



HAL
open science

Trajectory Tracking Control Design for Large-Scale Linear Dynamical Systems With Applications to Soft Robotics

Maxime Thieffry, Alexandre Kruszewski, Thierry-Marie Guerra, Christian Duriez

► **To cite this version:**

Maxime Thieffry, Alexandre Kruszewski, Thierry-Marie Guerra, Christian Duriez. Trajectory Tracking Control Design for Large-Scale Linear Dynamical Systems With Applications to Soft Robotics. *IEEE Transactions on Control Systems Technology*, 2021, 29 (2), pp.556-566. 10.1109/TCST.2019.2953624 . hal-02404003

HAL Id: hal-02404003

<https://hal.science/hal-02404003v1>

Submitted on 11 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Trajectory Tracking Control Design for Large-Scale Linear Dynamical Systems With Applications to Soft Robotics

Maxime Thieffry, Alexandre Kruszewski, Thierry Marie Guerra, Christian Duriez

► **To cite this version:**

Maxime Thieffry, Alexandre Kruszewski, Thierry Marie Guerra, Christian Duriez. Trajectory Tracking Control Design for Large-Scale Linear Dynamical Systems With Applications to Soft Robotics. IEEE Transactions on Control Systems Technology, Institute of Electrical and Electronics Engineers, In press, pp.1-11. 10.1109/TCST.2019.2953624 . hal-02404003

HAL Id: hal-02404003

<https://hal.archives-ouvertes.fr/hal-02404003>

Submitted on 11 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Trajectory Tracking Control Design for Large-Scale Linear Dynamical Systems With Applications to Soft Robotics

Maxime Thieffry , Alexandre Kruszewski, Thierry-Marie Guerra, and Christian Duriez, *Senior Member, IEEE*

Abstract—This article presents new results to control process modeled through linear large-scale systems. Numerical methods are widely used to model physical systems, and the finite-element method is one of the most common methods. However, for this method to be precise, it requires a precise spatial mesh of the process. Large-scale dynamical systems arise from this spatial discretization. We propose a methodology to design an observer-based output feedback controller. First, a model reduction step is used to get a system of acceptable dimension. Based on this low-order system, two linear matrix inequality problems provide us, respectively, with the observer and controller gains. In both the cases, model and reduction errors are taken into account in the computations. This provides robustness with respect to the reduction step and guarantees the stability of the original large-scale system. Finally, the proposed method is applied to a physical setup—a soft robotics platform—to show its feasibility.

Index Terms—Large-scale systems, model-order reduction, robust control, soft robotics.

I. INTRODUCTION

THE use of simulation shortens the development time of engineering systems. Different designs, actuation or sensing methods, and control approaches can be tested out quickly. For computer-based simulations, the demand for a highly accurate description of realistic phenomena leads to work with systems of very large dimensions. There are many

Manuscript received March 12, 2019; revised September 25, 2019; accepted November 4, 2019. Manuscript received in final form November 12, 2019. This work was supported in part by Agence Nationale de la Recherche (ANR) under Project ANR-17-ERC2-0029, in part by the European Union through the European Regional Development Fund (ERDF), in part by the French Ministry of Higher Education and Research, in part by the National Center for Scientific Research (CNRS), and in part by the Hauts-de-France Region. Recommended by Associate Editor S. Gumusoy. (*Corresponding author: Maxime Thieffry.*)

M. Thieffry is with the Polytechnic University Hauts-de-France, CNRS, UMR 8201, Laboratory of Industrial and Human Automation, Mechanics and Computer Science (LAMIH), F-59313 Valenciennes, France, and also with the DEFormable RObotic SofTware (DEFROST) Team, Inria, University of Lille, Centrale Lille, Centre de Recherche en Informatique Signal et Automatique de Lille (CRISAL), UMR 9189, 59655 Villeneuve d’Ascq, France (e-mail: maxime.thieffry@isir.upmc.fr).

A. Kruszewski and C. Duriez are with the DEFormable RObotic SofTware (DEFROST) Team, Inria, University of Lille, Centrale Lille, Centre de Recherche en Informatique Signal et Automatique de Lille (CRISAL), UMR 9189, 59655 Villeneuve d’Ascq, France (e-mail: alexandre.kruszewski@centralelille.fr).

T.-M. Guerra is with the Polytechnic University Hauts-de-France, CNRS, UMR 8201, Laboratory of Industrial and Human Automation, Mechanics and Computer Science (LAMIH), F-59313 Valenciennes, France (e-mail: thierry.guerra@uphf.fr).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2019.2953624

examples of large-scale systems that present challenges, such as power networks, urban traffic networks, digital communication networks, or robotics. When these systems are part of a control loop, one is interested in their reactions with respect to these inputs. This article is about designing a controller for such large systems.

Solutions have been proposed to control large-scale systems: for nonlinear systems, Tong *et al.* [1] and Hsiao *et al.* [2] present different methodologies to design controllers for nonlinear large-scale systems based on decentralized fuzzy control. Simulation results show the interest and the effectiveness of the approach, but examples are restricted to interconnected systems of small dimensions compared with examples available in the numerical simulations’ literature.

A second track investigated to study the systems of large dimensions is model-order reduction. It provides a low-order model that represents the behavior of the large-scale system and it is more suitable to design a controller for this reduced-order model. However, due to the approximation errors, controllers designed for this low-order model can lead to unstable closed-loop systems when applied to the large-scale plants. To handle this issue, a method to build a low-dimension controller by optimizing the H_∞ norm of the low-order model is presented in [3]. This method ensures the stability of the large-scale closed-loop model. A similar idea is presented in [4], where a low-order controller is computed while guaranteeing the large-scale closed-loop stability through linear matrix inequality (LMI) constraints.

The theoretical results of this article aim at being generic regarding the range of applications. In this article, soft robotics examples illustrate the proposed method. Compliant bodies offer high adaptability to environment and inherent safe interactions with humans. In addition, soft robotics is an active field of research, especially for closed-loop dynamic control [5], [6] that makes this new type of robots a good experimental platform. When dealing with soft robots control, significant results exist for kinematics control, but a few methods exist for dynamic control [6]. Many works are focused on open-loop control of soft robots, such as [7] or [8]. The computational cost of accurate models is a major drawback of model-based controllers, both because of the complexity of the design and for the practical implementation on the hardware. To avoid this constraint, Thuruthel *et al.* [9] proposed a learning-based open-loop dynamic controller to perform dynamic motions. However, open-loop strategies suffer from well-known bottlenecks, such as the robustness of the controller with respect

to modeling uncertainties and the response of the system with respect to external perturbations. This is why closed-loop algorithms are presented hereafter. A dynamic controller based on constant curvature assumptions is presented in [10], where the model is used to generate a feed-forward action coupled with a PID controller. Also based on the piecewise constant curvature model, Della Santina *et al.* [11] presented a dynamic controller that enables dynamic trajectory tracking for a continuous soft robot while handling interactions with the environment. Our approach aims at being generic regarding the geometry of the robot, and thus, we propose to design the control law based on a mechanical model of the structure.

A simulation framework to design, model, and control soft robots is presented in [12]. It uses the finite-element method (FEM) to simulate soft bodies in real time and it has been used to propose different methods to control soft robots. Coevoet *et al.* [13] presented an open-loop optimization-based method to control soft robots interacting with its environment. Then, [14] proposes a closed-loop inverse kinematics controller based on a quasi-static FEM model. To overcome this quasi-static assumption, [15] presents a model-based dynamic controller for soft robots. This article proposes to use FEM to build a large-scale dynamic model of the system and proposes a methodology to perform dynamic trajectory tracking.

A. Notations and Definitions

The notations used in this article are standard. We denote that \mathbb{R} is the set of real numbers, \mathbb{R}^n is the set of real vectors of dimension n , and $\mathbb{R}^{n \times m}$ is the set of matrices with n rows and m columns. For any vector or matrices x or M in \mathbb{R}^n , $\mathbb{R}^{n \times m}$, x^T , and M^T are their transpose. For any matrix $M \in \mathbb{R}^{n \times n}$, $M = M^T > 0$ defines a symmetric positive definite matrix. For a vector $x \in \mathbb{R}^n$, the notation **diag**(x) refers to a square diagonal matrix, where the elements of the diagonal are the elements of x and the entries outside the diagonal are all zeros.

The time dependence of velocity, displacement, state, and input vectors $v(t)$, $q(t)$, $x(t)$, and $u(t)$ is omitted. This article is concerned with discrete-time systems and the value of any variable x at time t_k is simply written as $x = x(t_k)$ and the value of this variable at time t_{k+1} , i.e., at the next time step, is written as $x_+ = x(t_{k+1})$.

Given a vector $x \in \mathbb{R}^n$ and function $V(x) : \mathbb{R}^n \mapsto \mathbb{R}$, we denote $\Delta V(x) = V(x_+) - V(x)$. A star $*$ indicates a transposed quantity in an expression or in a symmetric matrix. For example, $*PA - P < 0$ stands for $A^T P A - P < 0$ and $(\begin{smallmatrix} P & * \\ A & P \end{smallmatrix}) < 0$ stands for $(\begin{smallmatrix} P & A^T \\ A & P \end{smallmatrix}) < 0$.

II. PROBLEM STATEMENT

A. Introduction

Let us start from a linear large-scale system given in the following equation:

$$M : \begin{cases} x_+ = Ax + Bu \\ y = Cx \end{cases} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, m being the number of actuators, $C \in \mathbb{R}^{p \times n}$ is a matrix defining the system's outputs, and p is the number of outputs.

System matrices A , B , and C come from numerical methods, such as the FEM. For this model to be precise, the number of variables in the state vector x has to be significantly high and system (1) is thus a large-scale system with $n \gg 1$.

The large dimension of system (1) makes difficult the use of standard tools of automatic control (such as pole placement or LMI constraints problem) to design a controller for this system. To tackle this problem, we propose to use model-order reduction methods to obtain a low-order system that represents faithfully the full-order system. Based on this low-order state, the designs of both the controller and the observer are tractable.

B. Presentation of Reduction Algorithms

In this article, we construct the low-order system using projection-based model reduction. Two main categories of the algorithm are found in the literature: singular value decomposition (SVD)-based methods and Krylov (moment-matching)-based methods. Balanced truncation and proper orthogonal decomposition (POD) are two SVD-based methods. The first one requires the computation of the system's gramian, which can sometimes be computationally expensive, but has the major advantage to offer an *a priori* error bound. The second method, POD, also offers such a bound and is, in addition, directly usable for nonlinear large-scale systems. For moment-matching reduction algorithms, one can find the iterative rational Krylov algorithm [16] and its multi-input–multioutput (MIMO) version [17]. These methods are computationally tractable even for very large-scale systems. In this article, we can use any of the aforementioned reduction methods.

The large-scale state x can be decomposed into a low-order state $x_r \in \mathbb{R}^r$ and a neglected state $x_{\bar{r}} \in \mathbb{R}^{n-r}$ such that

$$x = V_r x_r + V_{\bar{r}} x_{\bar{r}} \quad \text{with} \quad \begin{cases} x_r = W_r^T x \\ x_{\bar{r}} = W_{\bar{r}}^T x \end{cases} \quad (2)$$

The projectors W and V are orthogonal, and it holds $W_r^T V_r = I$ and $W_r^T V_{\bar{r}} = 0$. It is now possible to design a controller based on the low-order state x_r . From (1) and (2), the dynamics of the low-order state writes

$$x_{r,+} = \underbrace{W_r^T A V_r}_{A_r} x_r + \underbrace{W_r^T B}_{B_r} u + W_r^T A V_{\bar{r}} x_{\bar{r}}. \quad (3)$$

The dynamics of the low-order state x_r still depends on the neglected state $x_{\bar{r}}$ and the matrix $W_r^T A V_{\bar{r}} \in \mathbb{R}^{r \times (n-r)}$ is of large dimensions. Of course, the controller designed should ensure stability of the exact low-order model (3), and the states $x_{\bar{r}}$ have to be taken into account. Considering that the term based on the unknown state $x_{\bar{r}}$ should be modest compared with x_r (due to the model reduction), a way to tackle this issue is to consider the error reduction as an unknown disturbance d_u associated with an unavoidable modeling error ε . Thus, an adequate controller has to cope with these uncertainties to make them vanish via the robustness property of the feedback control law.

C. Contributions

This article is a very important extension to our previous work [15] that presented an FEM-based dynamic controller for soft robots. Especially, it introduces dynamic trajectory tracking by modifying the structure of the control law, and it presents new synthesis via LMI constraints' problems to set the gains of both control law and observer. Moreover, a stability proof of the observer-based control is provided, including a guaranteed region of convergence via input to state stability (ISS) properties.

In the rest of this article, we present a trajectory tracking controller based on a low-order model with uncertainties [see model M_r , (8)]. The control strategy is presented in Sections IV and V, and it includes both feed-forward and observer-based output feedback elements. Under the assumption made, the proposed method ensures the stability of the full-order model M [see (1)] through LMI problems. The algorithm, computed thanks to the low-order model, is then applied to the full-order model and real-world experiments are gathered in Section VI.

III. LOW-ORDER MODEL

As said earlier, (3) presents the exact dynamics of the low-order state x_r , but it still depends on the neglected state $x_{\bar{r}}$ that is of large dimensions. To ensure the stability of the exact reduced-order model (3), we decompose the reduction error $W_r^T AV_{\bar{r}}x_{\bar{r}}$ into two orthogonal parts: an input disturbance d_u plus an unavoidable modeling error ε

$$W_r^T AV_{\bar{r}}x_{\bar{r}} = B_r d_u + B_r^\perp \varepsilon = \begin{pmatrix} B_r & B_r^\perp \end{pmatrix} \begin{pmatrix} d_u \\ \varepsilon \end{pmatrix} \quad (4)$$

where B_r^\perp is an orthogonal complement of B_r , such as $B_r^{\perp T} B_r = 0$ and $B_r^{\perp T} B_r^\perp = I$ and the square matrix $\begin{pmatrix} B_r & B_r^\perp \end{pmatrix}$ is full rank.

Lemma 1: Assuming that $x_{\bar{r}}$ is bounded in a domain around the trajectory, then a bound of ε is given by the following equation:

$$\|\varepsilon\| \leq \eta \quad (5)$$

and a overestimated upper bound η of the modeling error ε is

$$\begin{aligned} W_r^T AV_{\bar{r}}x_{\bar{r}} &= B_r d_u + B_r^\perp \varepsilon \\ B_r^{\perp T} W_r^T AV_{\bar{r}}x_{\bar{r}} &= B_r^{\perp T} B_r d_u + B_r^{\perp T} B_r^\perp \varepsilon \\ B_r^{\perp T} W_r^T AV_{\bar{r}}x_{\bar{r}} &= \varepsilon \\ \Rightarrow \|\varepsilon\| &\leq \|B_r^{\perp T} W_r^T AV_{\bar{r}}x_{\bar{r}}\| = \eta. \end{aligned} \quad (6)$$

The study of the reduction error $W_r^T AV_{\bar{r}}x_{\bar{r}}$ gives a value of this bound. Not all reduction methods listed earlier provide the computation of the large matrix $V_{\bar{r}}$, as it is computationally expensive, but one can study this bound using

$$\begin{aligned} x_+ &= Ax + Bu \\ W_r^T x_+ &= W_r^T Ax + W_r^T Bu \\ W_r^T x_+ &= W_r^T A(V_r x_r + V_{\bar{r}} x_{\bar{r}}) + W_r^T Bu \\ W_r^T x_+ &= W_r^T AV_r W_r^T x + W_r^T AV_{\bar{r}} x_{\bar{r}} + W_r^T Bu \\ \Rightarrow W_r^T AV_{\bar{r}} x_{\bar{r}} &= W_r^T x_+ - A_r W_r^T x - B_r u. \end{aligned} \quad (7)$$

It is always possible to get an estimation of the upper bound of ε . For a simple trajectory, running simulations around the trajectory is sufficient. For a global bound, simulations can be provided covering the entire workspace of the effector and considering the worst case. Experimental values of this bound are provided in Section VI with the experimental validation of the method.

Finally, with notations presented above and from (3) and (4), the low-order dynamics writes

$$M_r : \begin{cases} x_{r+} = A_r x_r + B_r u + B_r d_u + B_r^\perp \varepsilon \\ y_r = C_r x_r. \end{cases} \quad (8)$$

IV. CONTROL DESIGN

A. Reference Model

The trajectory is defined through a linear reference model M^* sharing the same dimensions as the low-order model M_r , and s^* is the reference signal

$$M^* : \begin{cases} x_{r+}^* = A_r^* x_r^* + B_r^* s^* \\ y_r^* = C_r^* x_r^*. \end{cases} \quad (9)$$

The trajectory error e_t is defined in the reduced-order state space as

$$e_t = x_r - x_r^*. \quad (10)$$

The objective is to design a controller such that it minimizes the tracking error e_t . To achieve this objective, an observer-based output feedback controller is designed that uses both feed-forward and feedback elements.

B. Observer design

The design is based on a so-called unknown input PI-observer. It assumes that the dynamics of d_u can be captured via a cascade of integrators, i.e., the p th variation of d_u is zero. Under this assumption, denoting $\tilde{\mathbf{B}}_r = (B_r \ 0 \ \dots \ 0)$ and defining an extended state $(\hat{x}_r^T \ \hat{D}_u^T)^T$, the observer writes

$$\begin{pmatrix} \hat{x}_{r+} \\ \hat{D}_{u+} \end{pmatrix} = \begin{pmatrix} A_r & \tilde{\mathbf{B}}_r \\ 0 & J \end{pmatrix} \begin{pmatrix} \hat{x}_r \\ \hat{D}_u \end{pmatrix} - \mathbf{K}_o C_r (x_r - \hat{x}_r) + \begin{pmatrix} B_r \\ 0 \end{pmatrix} u \quad (11)$$

with

$$\underbrace{\begin{pmatrix} \hat{d}_{u+} \\ \hat{d}_{u+}^{(1)} \\ \vdots \\ \hat{d}_{u+}^{(p-1)} \\ \hat{d}_{u+}^{(p)} \end{pmatrix}}_{\hat{D}_{u+}} = \underbrace{\begin{pmatrix} I & I & 0 & \dots & 0 \\ 0 & I & I & \ddots & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & I & I \\ 0 & 0 & \dots & 0 & I \end{pmatrix}}_J \underbrace{\begin{pmatrix} \hat{d}_u \\ \hat{d}_u^{(1)} \\ \vdots \\ \hat{d}_u^{(p-1)} \\ \hat{d}_u^{(p)} \end{pmatrix}}_{\hat{D}_u}. \quad (12)$$

To keep the observability property, the following constraint must be satisfied:

$$\text{rank}(B_r) \leq p \quad (13)$$

where p is the number of outputs. Let us define the observation error e_o

$$\begin{aligned} e_o &= \begin{pmatrix} x_r - \hat{x}_r \\ D_u - \hat{D}_u \end{pmatrix} = \begin{pmatrix} e_{o_x} \\ e_{o_d} \end{pmatrix} \\ \Rightarrow e_{o_+} &= \underbrace{\begin{bmatrix} A_r & \tilde{\mathbf{B}}_r \\ 0 & J \end{bmatrix} - \mathbf{K}_o(C_r \ O)}_{\mathbf{A}_o - \mathbf{K}_o \mathbf{C}_o} e_o + \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \varepsilon \end{aligned} \quad (14)$$

The computation of the matrix \mathbf{K}_o is detailed in Section V.

C. Observer-Based Output Feedback

The control law used to perform dynamic trajectory tracking is a PI-like control (L_i , integral part) based on a reference model state x_r^* (L^*) and the estimated states \hat{x}_r , \hat{d}_u (L and L_u)

$$u = -L^* x_r^* - (L \quad L_u \quad L_i) \begin{pmatrix} \hat{x}_r \\ \hat{d}_u \\ x_i \end{pmatrix}. \quad (15)$$

The dynamics of the trajectory error e_t writes, from (9) and (8)

$$\begin{aligned} e_{t_+} &= x_{r_+} - x_{r_+}^* \\ &= A_r x_r + B_r u + B_r d_u + \varepsilon - A_r^* x_r^* - B_r^* s^*. \end{aligned} \quad (16)$$

The proposed control law leads to the following trajectory error dynamics:

$$\begin{aligned} e_{t_+} &= A_r x_r - B_r (L \hat{x}_r + L^* x_r^* + L_u \hat{d}_u + L_i x_i) + B_r d_u + B_r^\perp \varepsilon \\ &\quad - A_r^* x_r^* - B_r^* s^* \\ e_{t_+} &= A_r (e_t + x_r^*) - B_r (L \hat{x}_r + L^* x_r^* + L_u \hat{d}_u + L_i x_i) \\ &\quad + B_r d_u + B_r^\perp \varepsilon - A_r^* x_r^* - B_r^* s^*. \end{aligned} \quad (17)$$

As $\hat{x}_r = x_r - e_o = e_t + x_r^* - e_o$, it holds

$$\begin{aligned} e_{t_+} &= A_r (e_t + x_r^*) \\ &\quad - B_r (L (e_t + x_r^* - e_{o_x}) + L^* x_r^* + L_u \hat{d}_u + L_i x_i) \\ &\quad + B_r d_u + B_r^\perp \varepsilon - A_r^* x_r^* - B_r^* s^* \\ e_{t_+} &= (A_r - B_r L) e_t + (A_r - A_r^* - B_r (L + L^*)) x_r^* \\ &\quad + B_r L e_{o_x} + B_r (d_u - L_u \hat{d}_u) - B_r L_i x_i - B_r^* s^* + B_r^\perp \varepsilon. \end{aligned} \quad (18)$$

In (18), an adequate choice to recover $d_u - \hat{d}_u$ is to set $L_u = I$ to get

$$\begin{aligned} e_{t_+} &= (A_r - B_r L) e_t + (A_r - A_r^* - B_r (L + L^*)) x_r^* \\ &\quad + (B_r L \quad B_r) \begin{pmatrix} e_{o_x} \\ d_u - \hat{d}_u \end{pmatrix} - B_r L_i x_i - B_r^* s^* + B_r^\perp \varepsilon \\ e_{t_+} &= (A_r - B_r L) e_t + (A_r - A_r^* - B_r (L + L^*)) x_r^* \\ &\quad + (B_r L \quad \tilde{\mathbf{B}}_r) e_o - B_r L_i x_i - B_r^* s^* + B_r^\perp \varepsilon. \end{aligned} \quad (19)$$

The integral term x_i is defined as

$$\begin{aligned} x_{i_+} &= x_i + (y_r - y_r^*) \\ x_{i_+} &= x_i + C_r e_t + (C_r - C_r^*) x_r^*. \end{aligned} \quad (20)$$

$$\begin{aligned} \text{Defining } \mathbf{z} &= \begin{pmatrix} e_t \\ x_r^* \\ x_i \end{pmatrix} \text{ and } \mathbf{A}_c = \begin{pmatrix} A_r & A_r - A_r^* & 0 \\ 0 & A_r^* & 0 \\ C_r & C_r - C_r^* & I \end{pmatrix}, \text{ it follows:} \\ \mathbf{z}_+ &= \begin{bmatrix} \mathbf{A}_c - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} (L \quad (L + L^*) \quad L_i) \\ \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{bmatrix} \mathbf{z} \\ &\quad + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \begin{pmatrix} B_r^\perp \\ 0 \\ 0 \end{pmatrix} \varepsilon. \end{aligned} \quad (21)$$

Now, we can write the closed-loop model, including the observer and the controller as

$$\begin{pmatrix} \mathbf{z}_+ \\ e_{o_+} \end{pmatrix} = \mathbf{G} \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \tilde{\mathbf{B}}_r^* s^* + \tilde{\Phi} \varepsilon \quad (22)$$

with

$$\mathbf{G} = \begin{pmatrix} \mathbf{A}_c - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} (L \quad (L + L^*) \quad L_i) & \begin{pmatrix} B_r L & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \\ 0 & \mathbf{A}_o - \mathbf{K}_o \mathbf{C}_o \end{pmatrix} \quad (23)$$

and

$$\tilde{\mathbf{B}}_r^* = \begin{pmatrix} \begin{bmatrix} -B_r^* \\ B_r^* \\ 0 \\ 0 \end{bmatrix} \end{pmatrix}; \quad \tilde{\Phi} = \begin{pmatrix} \begin{bmatrix} B_r^\perp \\ 0 \\ 0 \\ B_r^\perp \\ 0 \end{bmatrix} \end{pmatrix}. \quad (24)$$

Section V present how to compute the observer and controller gains that, under the assumption mentioned in Lemma 1, stabilize the closed-loop system and minimize the trajectory error. In the general case, finding all-in-one the Lyapunov function, the observer, and controller gains result is a non-convex problem. The structure of (21) associated with the matrix \mathbf{G} of (22) allows to combine a quasi-separation principle (solving the observer design apart from the controller design) and ISS properties to solve the problem. Global stability of the closed loop together with the region of convergence is provided. The control algorithm can thus be summarized as follows.

- Step 1:* Computation of the observer gain.
- Step 2:* Computation of the controller gains.
- Step 3:* Proof of stability and study of convergence region for the closed-loop algorithm [see system (22)].

V. COMPUTATION OF CONTROLLER AND OBSERVER GAINS

Let us recall the definition of ISS.

[18, Th. 5]: A system with state x and input u is ISS if and only if it admits a smooth Lyapunov function $V(x)$ such that its variation satisfies

$$\Delta V(x, u) \leq -\alpha(\|x\|) + \gamma(\|u\|) \quad (25)$$

where $\alpha, \gamma \in \mathcal{K}_\infty$. \square

In the rest of this article, we will use quadratic \mathcal{K}_∞ functions defined as

$$\alpha(\|x\|) = \alpha x^T M x, \quad \alpha > 0, \quad M > 0. \quad (26)$$

The ISS property will come at hand to study the stability property of the closed loop according to both the modeling error ϵ and the input disturbance d_u .

A. Computation of Observer Gain

The first objective is to compute \mathbf{K}_o in (14) while minimizing the impact of the unknown modeling error ϵ on the observer error e_o that writes

Find \mathbf{K}_o and $V_o(e_o) = e_o^T P_o e_o$ for the model:

$$e_{o+} = (\mathbf{A}_o - \mathbf{K}_o \mathbf{C}_o) e_o + \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \epsilon$$

such that:

$$\Delta V_o < -\alpha_o e_o^T P_o e_o + \gamma_o \epsilon^T \epsilon$$

with: $\alpha_o > 0$, $\gamma_o > 0$, $P_o > 0$.

(27)

(27) is equivalent to

$$\begin{aligned} & [*] P_o \left[(\mathbf{A}_o - \mathbf{K}_o \mathbf{C}_o) e_o + \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \epsilon \right] \\ & - e_o P_o e_o < -\alpha_o e_o^T P_o e_o + \gamma_o \epsilon^T \epsilon \\ \Leftrightarrow & \begin{pmatrix} e_o^T & \epsilon^T \end{pmatrix} [*] P_o \begin{bmatrix} \mathbf{A}_o - \mathbf{K}_o \mathbf{C}_o & \begin{pmatrix} B_r^\perp \\ 0 \end{pmatrix} \end{bmatrix} \begin{pmatrix} e_o \\ \epsilon \end{pmatrix} \\ & + \begin{pmatrix} e_o^T & \epsilon^T \end{pmatrix} \begin{pmatrix} -P_o + \alpha_o P_o & 0 \\ 0 & -\gamma_o I \end{pmatrix} \begin{pmatrix} e_o \\ \epsilon \end{pmatrix} < 0. \end{aligned} \quad (28)$$

Using Schur's complement, this is equivalent to the following matrix inequality condition:

$$(27) \Leftrightarrow \begin{pmatrix} -P_o + \alpha_o P_o & 0 & * \\ 0 & -\gamma_o I & * \\ P_o \mathbf{A}_o - \tilde{\mathbf{K}}_o \mathbf{C}_o & P_o \begin{bmatrix} B_r^\perp \\ 0 \end{bmatrix} & -P_o \end{pmatrix} < 0 \quad (29)$$

with the classical change of variables $\tilde{\mathbf{K}}_o = P_o \mathbf{K}_o$. It ensures that for large enough time, the trajectory error converges to a hyperball, whose radius depends on γ_o and α_o .

B. Computation of Controller Gain

The second objective is to compute the controller gains L , L^* , and L_i in (15), while minimizing the impact of the modeling and reduction errors on the output tracking error $(y - y^*)$

Find L , L^* , L_i and $V_t(\mathbf{z}) = \mathbf{z}^T P_t \mathbf{z}$ for the model:

$$\begin{aligned} \mathbf{z}_+ = & \begin{bmatrix} \mathbf{A}_c - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} (L \quad (L + L^*) \quad L_i) \\ B_r L \quad \tilde{\mathbf{B}}_r \\ 0 \quad 0 \end{bmatrix} \mathbf{z} \\ & + \begin{pmatrix} B_r L \quad \tilde{\mathbf{B}}_r \\ 0 \quad 0 \end{pmatrix} e_o + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \begin{pmatrix} B_r^\perp \\ 0 \\ 0 \end{pmatrix} \epsilon \end{aligned} \quad (30)$$

such that:

$$\Delta V_t < -\alpha_t (y - y^*)^T (y - y^*) + \beta_t e_o^T \begin{pmatrix} P_t & 0 \\ 0 & I \end{pmatrix} e_o$$

$$+ \gamma_t \epsilon^T \epsilon + \omega_t s^{*T} s^*$$

with: $(\alpha_t, \beta_t, \gamma_t, \omega_t) > 0$ and $P_t > 0$.

It holds

$$\begin{aligned} (y - y^*) &= (C_r \quad C_r - C_r^* \quad 0) \begin{pmatrix} e_t \\ x_r^* \\ x_i \end{pmatrix} = \tilde{\mathbf{C}}_r \mathbf{z} \\ \Rightarrow (y - y^*)^T (y - y^*) &= \mathbf{z}^T \tilde{\mathbf{C}}_r^T \tilde{\mathbf{C}}_r \mathbf{z}. \end{aligned} \quad (31)$$

Thus, problem (31) is equivalent to

$$\begin{aligned} [*] P_t & \left[\begin{pmatrix} \mathbf{A}_c - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} (L \quad (L + L^*) \quad L_i) \\ B_r L \quad \tilde{\mathbf{B}}_r \\ 0 \quad 0 \end{pmatrix} \mathbf{z} \right. \\ & \left. + \begin{pmatrix} B_r L \quad \tilde{\mathbf{B}}_r \\ 0 \quad 0 \end{pmatrix} e_o + \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} s^* + \begin{pmatrix} B_r^\perp \\ 0 \\ 0 \end{pmatrix} \epsilon \right] - \mathbf{z}^T P_t \mathbf{z} \\ & < -\alpha_t \mathbf{z}^T \tilde{\mathbf{C}}_r^T \tilde{\mathbf{C}}_r \mathbf{z} + \beta_t e_o^T \begin{pmatrix} P_t & 0 \\ 0 & I \end{pmatrix} e_o + \gamma_t \epsilon^T \epsilon + \omega_t s^{*T} s^*. \end{aligned} \quad (32)$$

Denoting $X_t = P_t^{-1}$ and using Schur's complement and congruence property with $\mathbf{diag}(X_t, [X_t \quad I], I, I, I)$, condition (33) shown in bottom of the next page, defines an LMI problem corresponding to the ISS condition (31). LMI problems are solved using YALMIP for MATLAB [19].

C. Proof of Stability for Complete Low-Order System

The two previous LMI problems compute the controller and observer gains. This section optimizes the convergence region, where the proof of stability is given. Once the observer and controllers gains are obtained, the matrix \mathbf{G} defined in (23) is fully defined. The stability of the complete closed loop (i.e., the controller and observer) is guaranteed if the following problem is solved.

System (22) is ISS if it exists a Lyapunov function $V(\mathbf{z}, e_o) = (\mathbf{z}^T \quad e_o^T) P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}$ such that

$$\Delta V(\mathbf{z}, e_o) < -\alpha (\|\mathbf{z}\| \quad \|e_o\|) + \gamma (\|\epsilon\|) + \omega (\|s^*\|). \quad (34)$$

We define $\alpha(\|\mathbf{z}\|) = \alpha x^T P x$, $\omega(\|s^*\|) = \omega s^{*T} s^*$, and $\gamma(\|\epsilon\|) = \gamma \epsilon^T \epsilon$ with $P > 0$, $\alpha > 0$, $\omega > 0$, and $\gamma > 0$. Then, the previous ISS condition is satisfied if

$$\begin{aligned} (34) \Leftrightarrow & [*] P \left[\mathbf{G} \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \tilde{\mathbf{B}}_r^* s^* + \tilde{\Phi} \epsilon \right] - \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} \\ & < -\alpha \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} + \gamma \epsilon^T \epsilon + \omega s^{*T} s^* \\ \Leftrightarrow & [*] P [G \quad \tilde{\mathbf{B}}_r^* \quad \tilde{\Phi}] - \mathbf{diag}((1 - \alpha)P, \omega I, \gamma I) < 0. \end{aligned} \quad (35)$$

This defines a generalized eigenvalue problem in P , α , γ , and ω and ensures the stability of system (34) with the observer gain \mathbf{K}_o and controller gains L , L^* , and L_i . Moreover, it ensures that $V(\mathbf{z}, e_o)$ is decreasing for large enough vectors (\mathbf{z}, e_o) . The system finally converges to an invariant manifold defined as

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} < \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\alpha}. \quad (36)$$

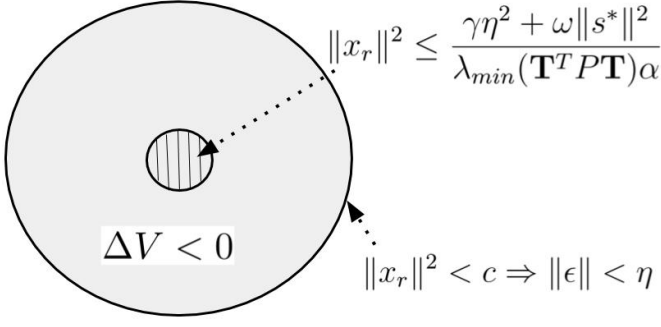


Fig. 1. Illustration of the convergence region using the ISS property. The hatching circle corresponds to invariant manifold defined in (41).

This defines a region of convergence for the vector (\mathbf{z}, e_o) . In order to study the convergence of the reduced-order state x_r , let us write

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} = \begin{pmatrix} e_t \\ x_r^* \\ x_i \\ e_o \end{pmatrix} = \begin{pmatrix} I & -I & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}. \quad (37)$$

Equation (36) is also equivalent to

$$\begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix}^T P \begin{pmatrix} \mathbf{z} \\ e_o \end{pmatrix} = \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} < \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\alpha}. \quad (38)$$

In addition, we have

$$\begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \geq \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \lambda_{\min}(\mathbf{T}^T P \mathbf{T}) \quad (39)$$

where $\lambda_{\min}(A)$ is the smallest eigenvalue of A . It yields

$$\begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix}^T \mathbf{T}^T P \mathbf{T} \begin{pmatrix} x_r \\ x_r^* \\ x_i \\ e_o \end{pmatrix} \geq x_r^T x_r \lambda_{\min}(\mathbf{T}^T P \mathbf{T}). \quad (40)$$

Finally, the invariant manifold with respect to the reduced-order state writes

$$\|x_r\|^2 \leq \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\lambda_{\min}(\mathbf{T}^T P \mathbf{T}) \alpha}. \quad (41)$$

A graphical scheme of the convergence region defined by (41) is shown in Fig. 1. Section VI will give an example of

the estimation of this bound as well as the experimental results from the studied robot.

D. Summary of Closed-Loop Algorithm

The study of the open-loop system says that for a given workspace in which the reduced-order state x_r is bounded by a given constant c , the modeling error ϵ is bounded by η

$$\|x_r\|^2 < c \Rightarrow \|\epsilon\| < \eta.$$

Then, three LMI problems guarantee that if the modeling error ϵ is bounded by η , then the reduced-order state converges to an invariant manifold defined in (36) and (41). If this region of convergence is smaller than the workspace in which $\|x_r\|^2 < c$ is satisfied, then the closed-loop system is stable.

1) Control Design Algorithm:

Study of open-loop system

$$\|\epsilon\| < \eta \Rightarrow \|x_r\|^2 \leq \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\lambda_{\min}(\mathbf{T}^T P \mathbf{T}) \alpha}.$$

Solve three LMI problems.

$$\text{If } \frac{\gamma \eta^2 + \omega \|s^*\|^2}{\lambda_{\min}(\mathbf{T}^T P \mathbf{T}) \alpha} < c$$

then the closed-loop is stable.

VI. EXPERIMENTAL VALIDATION

To illustrate the interest and the effectiveness of the methodology proposed, it is tested on soft robotics application. The modeling step is done using the SOFA framework [12], a unified software to model soft structures thanks to the FEM. For this model to be precise, the number of nodes of the mesh has to be significantly high and the model studied is also a large-scale system.

A. Experimental Setup

Experiments are conducted on a trunk-like robot presented in Fig. 2 with a schematic view in Fig. 3. It is entirely made of silicone, the structure is 18 cm long, and the thicknesses at its base and its tip are, respectively, 2.5 and 1 cm. The structure is driven by four cables—actuated by four servomotors—that permit to work in the three dimensions of space. The output of the system is the position of the tip (red point in Fig. 2) that is measured using a magnetic microsensors, whose frequency can reach 240 Hz. In the following, the sampling time for the real-time experiments is set to 25 Hz.

$$\left(\begin{array}{cccccccc} & -X_t & & & * & * & * & * \\ & 0 & & -\beta_t \begin{pmatrix} X_t & 0 \\ 0 & I \end{pmatrix} & * & * & * & * \\ & 0 & & 0 & -\omega_t I & * & * & * \\ & 0 & & 0 & 0 & -\gamma_t I & * & * \\ \mathbf{A}_c - \begin{pmatrix} B_r \\ 0 \\ 0 \end{pmatrix} (L \quad (L + L^*) \quad L_i) X_t & \begin{pmatrix} B_r L X_t & \tilde{\mathbf{B}}_r \\ 0 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} -B_r^* \\ B_r^* \\ 0 \end{pmatrix} & \begin{pmatrix} B_r^\perp \\ 0 \\ 0 \end{pmatrix} & -X_t & * & & \\ & \tilde{\mathbf{C}}_r X_t & & 0 & 0 & 0 & 0 & -\alpha_t I \end{array} \right) < 0 \quad (33)$$

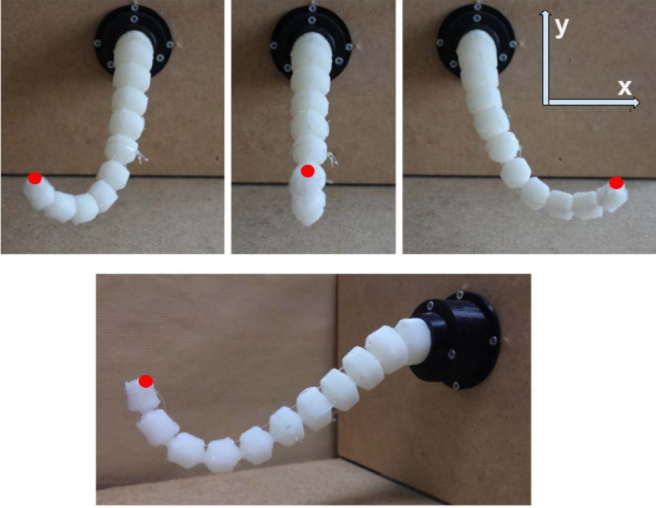


Fig. 2. Soft robot used for experimental validation. It is entirely made of silicone. Top: front view of the robot. Bottom: side view. Red = end-effector, location of the sensor.

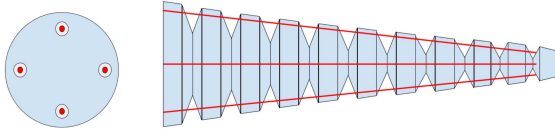


Fig. 3. Design of the robot. Slice view (left) and side view (right). The robot is actuated with four cables in red.

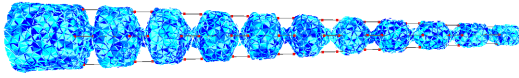


Fig. 4. FEM mesh of the trunk-like robot presented in Fig. 2. The mesh is made of 1557 nodes and 2972 tetrahedrons.

B. Modeling

The finite-element mesh of this robot is made of 1557 nodes, see Fig. 4. The dimensions of the state vector in system (55) is also $1557 \times 3 \times 2 = 9342$ state variables (three directions of space for displacement and velocity). Then, model-order reduction provides us with a low-order system of dimension 6, as explained in Section VI-D.

1) *Nonlinear Large-Scale Model*: Let us start with the formulation given by the second law of Newton that models the dynamic behavior of a body as

$$\mathbb{M}(q)\dot{v} = \mathbb{P}(q) - \mathbb{F}(q, v) + \mathbb{H}^T(q)\lambda \quad (42)$$

where $q \in \mathbb{R}^n$ is the vector of generalized degrees of freedom, $\mathbb{M}(q)$ is the inertia matrix, and $v = \dot{q} \in \mathbb{R}^n$ is the vector of velocity. $\mathbb{F}(q, v)$ represents internal forces applied to the simulated object depending on the current state and $\mathbb{P}(q)$ gathers known external forces. The matrices \mathbb{M} , \mathbb{F} , and \mathbb{P} are square matrices of dimensions $n \times n$. $\mathbb{H}^T(q)$ is the matrix containing the constraint directions, while $\lambda \in \mathbb{R}^m$ is the vector of actuators forces. The procedure to get these different terms is given shortly, and the reader can refer to [12] for details. We integrate (42) using a time-stepping implicit scheme (backward Euler) to have unconditional stability. Let us consider the time interval $[t_k, t_{k+1}]$ whose length is $h = t_{k+1} - t_k$.

In practice due to hardware limitation, the time step is equal to 0.04 s. Let us denote $\delta q = q_+ - q = hv_+$ and $\delta v = v_+ - v$; with these notations, (42) writes

$$\begin{aligned} \mathbb{M}(q_+)\delta v &= h(\mathbb{P}(q_+) - \mathbb{F}(q_+, v_+)) + h\mathbb{H}^T(q_+)\lambda \\ q_+ &= q + hv_+. \end{aligned} \quad (43)$$

The internal forces \mathbb{F} are a nonlinear function of the positions and the velocities. We then apply a Taylor series expansion to \mathbb{F} and make the following first-order approximation:

$$\begin{aligned} \mathbb{F}(q_+, v_+) &= \mathbb{F}(q + \delta q, v + \delta v) \\ &\approx \mathbb{F}(q, v) + \frac{\partial \mathbb{F}(q, v)}{\partial q} \delta q + \frac{\partial \mathbb{F}(q, v)}{\partial v} \delta v. \end{aligned} \quad (44)$$

Defining

$$\mathbf{S}(q, v) = \left(\mathbb{M}(q) + h \frac{\partial \mathbb{F}(q, v)}{\partial v} + h^2 \frac{\partial^2 \mathbb{F}(q, v)}{\partial q^2} \right). \quad (45)$$

As $\delta q = hv_+ = h\delta v + hv$, it follows:

$$\mathbf{S}(q, v)\delta v = h \left(\mathbb{P}(q_+) - \mathbb{F}(q, v) - h \frac{\partial \mathbb{F}(q, v)}{\partial q} v \right) + h\mathbb{H}^T(q)\lambda. \quad (46)$$

And finally

$$\begin{aligned} v_+ &= v + h\mathbf{S}(q, v)^{-1} \left(\mathbb{P}(q_+) - \mathbb{F}(q, v) - h \frac{\partial \mathbb{F}(q, v)}{\partial q} v \right) \\ &\quad + h\mathbf{S}(q, v)^{-1} \mathbb{H}^T(q)\lambda. \end{aligned} \quad (47)$$

Let us define the state vector x as

$$x = \begin{pmatrix} v \\ q \end{pmatrix}. \quad (48)$$

The nonlinear state-space expression is given by (49), shown in bottom of the next page.

C. Large-Scale Linear Discrete-Time State-Space Model

Let $q_0 \in \mathbb{R}^n$ be a stable equilibrium point that could be obtained using an inverse problem. It is induced by the gravity field $\mathbb{P}(q_0)$ that is considered constant, i.e., $\mathbb{P}(q_0) = \mathbb{P}$, and the actuation input λ_0 . The Taylor expansion is made around this configuration q_0 and it holds

$$\frac{\partial \mathbb{F}(q, v)}{\partial q} \Big|_{\substack{q=q_0 \\ v=0}} = \mathbb{K} \quad \text{and} \quad \frac{\partial \mathbb{F}}{\partial v} \Big|_{\substack{q=q_0 \\ v=0}} = \mathbb{D}. \quad (50)$$

Moreover, we consider constant inertia matrix $\mathbb{M}(q) = \mathbb{M}$. With these notations, we get

$$\mathbf{S} = (\mathbb{M} + h\mathbb{D} + h^2\mathbb{K}). \quad (51)$$

Defining the new input vector $u = \lambda - \lambda_0$ and $\mathbf{q} = q - q_0$, the linear state-space equation is

$$\begin{pmatrix} I & 0 \\ -h & I \end{pmatrix} x_+ = \begin{pmatrix} v - h\mathbf{S}^{-1}[\mathbb{F} + h\mathbb{K}v] \\ \mathbf{q} \end{pmatrix} + \begin{pmatrix} h\mathbf{S}^{-1}\mathbb{H}^T \\ 0 \end{pmatrix} u \quad (52)$$

From (49) to (52), the value of \mathbb{P} vanishes as it is included in the equilibrium point. \mathbb{F} are the internal forces of the previous

time step that can be written as $\mathbb{F} = \mathbb{K}q$. Time step h is a nonnull scalar, and thus, we have

$$\begin{pmatrix} I & 0 \\ -h & I \end{pmatrix}^{-1} = \begin{pmatrix} I & 0 \\ h & I \end{pmatrix}. \quad (53)$$

With these notations, state space (52) writes

$$\begin{aligned} x_+ &= \begin{pmatrix} I & 0 \\ h & I \end{pmatrix} \begin{pmatrix} v - h\mathbf{S}^{-1}[\mathbb{K}q + h\mathbb{K}v] \\ \mathbf{q} \end{pmatrix} \\ &\quad + \begin{pmatrix} I & 0 \\ h & I \end{pmatrix} \begin{pmatrix} h\mathbf{S}^{-1}\mathbb{H}^T \\ 0 \end{pmatrix} u. \end{aligned} \quad (54)$$

Finally, the state-space equation can be written in the standard form

$$M : \begin{cases} x_+ = \underbrace{\begin{pmatrix} I - h^2\mathbf{S}^{-1}\mathbb{K} & -h\mathbf{S}^{-1}\mathbb{K} \\ hI - h^3\mathbf{S}^{-1}\mathbb{K} & I - h^2\mathbf{S}^{-1}\mathbb{K} \end{pmatrix}}_A \begin{pmatrix} v \\ \mathbf{q} \end{pmatrix} \\ \quad + \underbrace{\begin{pmatrix} h\mathbf{S}^{-1}\mathbb{H}^T \\ h^2\mathbf{S}^{-1}\mathbb{H}^T \end{pmatrix}}_B u \\ y = Cx \end{cases} \quad (55)$$

where $A \in \mathbb{R}^{2n \times 2n}$, $B \in \mathbb{R}^{2n \times m}$, m being the number of actuators and $C \in \mathbb{R}^{p \times 2n}$ is a matrix defining the end-effector coordinates, and p is the number of outputs.

D. Low-Order Model

As the computation of \mathcal{H}_∞ is computationally more expensive, a common way to measure the accuracy of the reduced-order system is to measure the \mathcal{H}_2 norm error between the full- and reduced-order models

$$\|M - M_r\|_{\mathcal{H}_2}. \quad (56)$$

In this article, we use a moment-matching reduction method, implemented within the MORE toolbox [20], to perform the reduction. Table I gathers the comparison of the \mathcal{H}_2 errors for different sizes of reduced systems, provided that the norm of the full-order model $\|M\|_{\mathcal{H}_2} = 0.42$. The best result regarding the \mathcal{H}_2 norm error is achieved for reduced-order systems of dimension 26 and higher; they present an error of 24.67% compared with the full-order model. The \mathcal{H}_2 error between the full-order system and the reduced-order system of dimension 2 is 25.4%, which is close to the optimal solution. Indeed, the \mathcal{H}_2 error decreases from a system with dimension 2 to dimension 6 and then converges slowly to the optimal solution. A good compromise between the accuracy and the complexity of the low-order model is also to choose a reduced-order system of dimension 6.

TABLE I

COMPARISON OF THE \mathcal{H}_2 NORM ERROR BETWEEN THE FULL- AND REDUCED-ORDER SYSTEMS FOR DIFFERENT DIMENSIONS OF REDUCED SYSTEMS

Dimensions of M_r	2	4	6	10	26	30	100
\mathcal{H}_2 norm error $\ M - M_r\ _{\mathcal{H}_2}$ in %	25.40	24.70	24.69	24.69	24.67	24.67	24.67

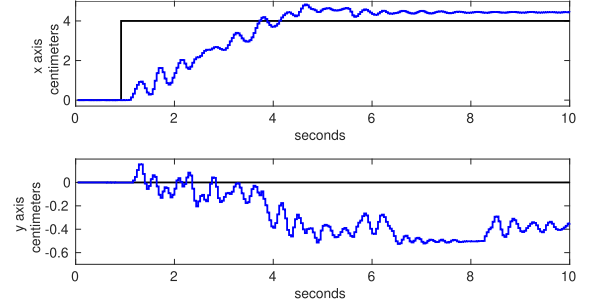


Fig. 5. Real-time open-loop behavior of the robot. Black: reference signal. Blue: end-effector's displacement. Top: measurements of displacement along the x -axis. Bottom: measurements of displacement along the y -axis.

E. Real Setup's Behavior in Open Loop

The open-loop behavior of the robot is shown in Fig. 5. It presents the displacements of the end-effector (i.e., the output of the system) in the open loop controlled by solving an inverse problem.

Remark: With an accurate calibration process, it is possible to obtain a zero static error in the open loop. However, the open-loop algorithms do not offer any guarantee about the performances of the control algorithm, especially in the presence of perturbations. Moreover, due to hardware limitations, the calibration has to be done regularly, as phenomena, such as fatigue, are difficult to model.

F. Validation of Model in Simulations

1) *Validation of Model:* The experiments of Fig. 5 are repeated in simulation, and the results are shown in Fig. 6. Fig. 5 compared with Fig. 6 shows that the main behavior of the robot is captured by the model.

2) *Bound η of the Modeling Error:* By running multiple simulations, we cover a workspace of the robot that is considered as exhaustive (we cover the whole range of possible actuation) and we get the maximal values of the norms of the full- and reduced-order states

$$\|x\| = 2.39 \cdot 10^8 \text{ and } \|x_r\| = 2.37 \cdot 10^8. \quad (57)$$

In this workspace and from (7), the norm of the modeling error is

$$\eta = 4.19 \cdot 10^3. \quad (58)$$

$$\begin{pmatrix} I & 0 \\ -h & I \end{pmatrix} x_+ = \begin{pmatrix} v + h\mathbf{S}(q, v)^{-1}(\mathbb{P}(q_+) - \mathbb{F}(q, v) - h \frac{\partial \mathbb{F}(q, v)}{\partial q} v) \\ q \end{pmatrix} + \begin{pmatrix} h\mathbf{S}(q, v)^{-1}\mathbb{H}^T(q) \\ 0 \end{pmatrix} \lambda \quad (49)$$

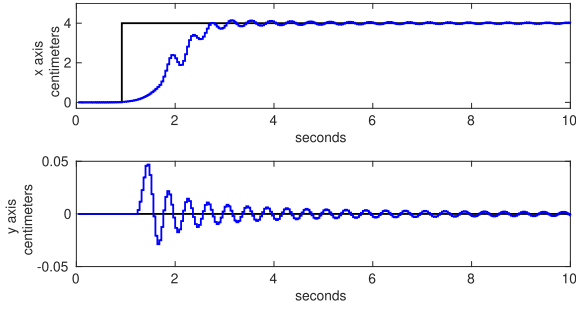


Fig. 6. Open-loop simulation of the robot controlled via an inverse problem. Black: reference signal. Blue: end-effector's displacement. Top: displacement along the x-axis. Bottom: displacement along the y-axis.

3) *Invariant Manifold for the System Studied*: The invariant manifold defined in (41) and depicted in Fig. 1 is defined with the following parameters:

$$\begin{aligned} \alpha &= 0.05 \\ \gamma &= 10^{-2} \\ \omega &= 2.10^{-3}. \end{aligned} \quad (59)$$

For the case where $s^* = 0$, i.e., the robot comes back to its rest shape, the invariant manifold writes

$$\|x_r\|^2 \leq \frac{\gamma \eta^2}{\lambda_{\min}(\mathbf{T}^T \mathbf{P} \mathbf{T}) \alpha} = 9.14 \cdot 10^6. \quad (60)$$

In this case, the invariant manifold corresponds to 3.87% of the robot workspace. Therefore, for an initial condition that satisfies (57), the system is stable and converges to the invariant manifold.

G. Experimental Validation

Experiments are conducted with the same sampling time as for the modeling process; point-to-point control and trajectory tracking experiments are conducted.

1) *Point-to-Point Control*: For the first experiment, the reference is set such that the robot starts from its rest position and converges to a desired position. Once the robot has reached the target, it is moved 1 cm away from the target. Fig. 7 shows that the control algorithm is robust enough to deal with this perturbation. For the second experiment, the robots start from its rest shape and converge to the first deformed position and then to the second deformed position. Results are presented in Fig. 8. It shows a diminution of oscillations, a faster time response, and a cancellation of the static error compared with the open loop.

2) *Sinusoidal Trajectory*: The second set of experiments consists of tracking a sinusoidal signal along the x-axis of the robot; the reference for y-axis is zero. Results are presented in Fig. 9. The behavior along the first axis accurately follows the trajectory, while the second output oscillates around the reference signal. Of course, this residual oscillation is due to the physical coupling between the actuators.

H. Discussion and Limitations

The limits of the algorithm are tested with the third experiment, where the goal is to reach a highly deformed position.

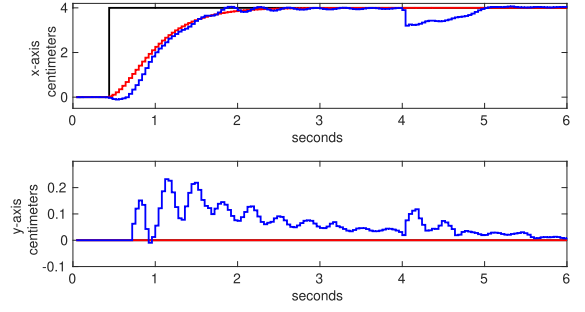


Fig. 7. Point-to-point closed-loop control. Top: output along the x-axis. Bottom: output along the y-axis. Black: reference signal. Red: output of reference model. Blue: Robot's end-effector's displacement.

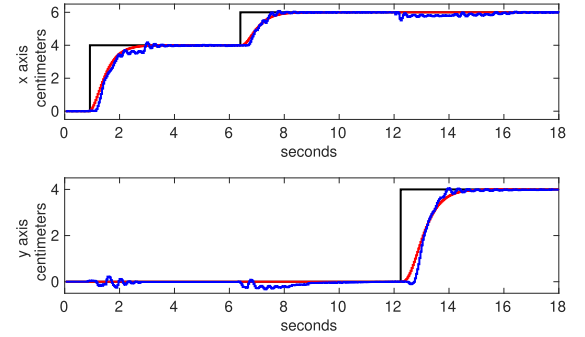


Fig. 8. Point-to-point closed-loop control. Desired position corresponds to Fig. 10(c). Top: output along the x-axis. Bottom: output along the y-axis. Black: reference signal. Red: output of the reference model. Blue: robot's end-effector's displacement.

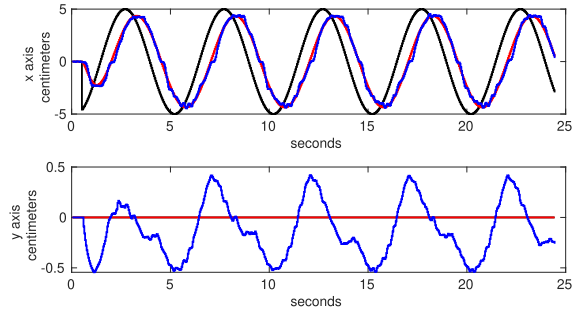


Fig. 9. Closed-loop trajectory tracking. Top: output along the x-axis. Bottom: output along the y-axis. Black: reference signal. Red: output of reference model. Blue: robot's end-effector's displacement.

Prior to the experiment, it has been checked via an inverse problem that the target is inside the robot workspace. Fig. 10 shows the different positions, where the algorithm is tested. Fig. 10(a) shows the rest position where the system is linearized. Fig. 10(b) shows a deformation of 5% where the linearization assumption is valid. Fig. 10(c) corresponds to the desired position of the successful attempt shown in Fig. 8. Fig. 10(d) shows an example of unsuccessful attempt to reach a highly deformed position. This limitation is shown in Fig. 11 and is probably due to the fact that the position is outside the validity range of the linearization.

Indeed, the proposed closed-loop controller is based on a linear model: we consider constant mass, stiffness, and damping. This linearization assumption, necessary to cope with the high-sized problem, naturally limits the workspace of the

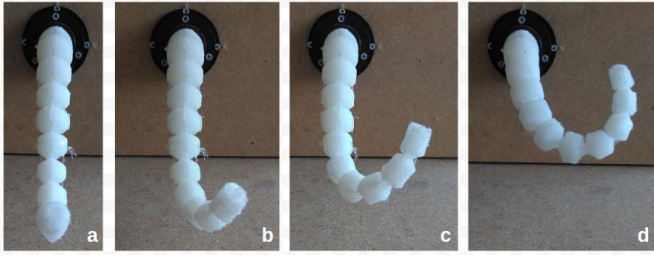


Fig. 10. (a) Rest position. (b) Limits of the linearization assumption (5% of deformation). (c) Desired position corresponding to Fig. 8. (d) Desired position corresponding to Fig. 11.

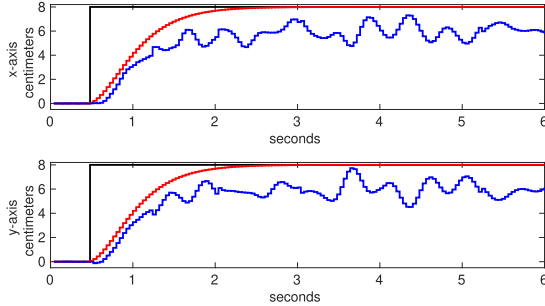


Fig. 11. Unsuccessful attempt to reach a more distant target [see Fig. 10(d)]. Top: output along the x -axis. Bottom: output along the y -axis. Black: reference signal. Red: output of reference model. Blue: robot's end-effector's displacement.

robot, and the model remains only valid in a neighborhood of the linearization. In practice, for the robot under consideration, the workspace is limited approximately to a circle of radius 6 cm around its rest position. If the reference trajectory is set outside of this workspace, the performances start to decrease. Even if it is recognized as a limiting factor, this is not a strong limitation as 6 cm represent 33.3% of the length of the structure. Further experimental tests with different robots should be performed to evaluate and characterize the full potential of the approach. For larger deformation, more elaborated models should come at hand (e.g., polytopic descriptions, nonlinear representation, and so on); this is an ongoing research topic.

In addition, the mechanical coupling between the actuators limits the performances of the control algorithm, as shown in Fig. 9. Moreover, the friction between the cables and the structure also limits the performances of the closed-loop algorithm, and this requires deepening the actuation system to remove this constraint.

VII. CONCLUSION

A large-scale model of a physical system is obtained using the numerical method, such as the FEM. The precision of the model is linked to the size of the model; as accuracy increases, so does the size of the state-space model.

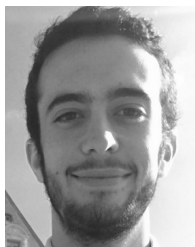
We deal with this dimension issue using projection-based model reduction. The design of a robust controller is done to take into account both the modeling errors and the modeling reduction. A so-called PI-like observer reconstructs the reduction part via an unknown vector. The conditions are written as three LMI constraints' problem. When a solution

to these problems is found, it is applied to the original full-order model.

To illustrate the interest of this article, theoretical results are applied to a soft robot. Dynamic control of soft robots is an open field of research and the results provided in this article show that the method is relevant to increase the accuracy, range of speed, and robustness of this kind of robots.

REFERENCES

- [1] S. Tong, L. Zhang, and Y. Li, "Observed-based adaptive fuzzy decentralized tracking control for switched uncertain nonlinear large-scale systems with dead zones," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 1, pp. 37–47, Jan. 2016.
- [2] F.-H. Hsiao, J.-D. Hwang, C.-W. Chen, and Z.-R. Tsai, "Robust stabilization of nonlinear multiple time-delay large-scale systems via decentralized fuzzy control," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 152–163, Feb. 2005.
- [3] P. Benner, T. Mitchell, and M. L. Overton, "Low-order control design using a reduced-order model with a stability constraint on the full-order model," 2018, *arXiv:1803.06549*. [Online]. Available: <https://arxiv.org/abs/1803.06549>
- [4] M. Thieffry, A. Kruszewski, T.-M. Guerra, and C. Duriez, "Reduced order control of soft robots with guaranteed stability," in *Proc. Eur. Control Conf. (ECC)*, Limassol, Cyprus, Jun. 2018, pp. 635–640.
- [5] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, p. 467, May 2015.
- [6] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robot.*, vol. 5, no. 2, pp. 149–163, 2018.
- [7] C. D. Onal and D. Rus, "Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot," *Bioinspiration Biomimetics*, vol. 8, no. 2, 2013, Art. no. 026003.
- [8] T. Umedachi, V. Vikas, and B. A. Trimmer, "Highly deformable 3-D printed soft robot generating inching and crawling locomotions with variable friction legs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4590–4595.
- [9] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, "Stable open loop control of soft robotic manipulators," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1292–1298, Apr. 2018.
- [10] V. Falkenhahn, A. Hildebrandt, R. Neumann, and O. Sawodny, "Model-based feedforward position control of constant curvature continuum robots using feedback linearization," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 762–767.
- [11] C. Della Santina, R. K. Katzschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *Proc. IEEE Int. Conf. Soft Robot. (RoboSoft)*, Apr. 2018, pp. 46–53.
- [12] E. Coevoet *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Adv. Robot.*, vol. 31, no. 22, pp. 1208–1224, 2017, doi: [10.1080/01691864.2017.1395362](https://doi.org/10.1080/01691864.2017.1395362).
- [13] E. Coevoet, A. Escande, and C. Duriez, "Optimization-based inverse model of soft robots with contact handling," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1413–1419, Jul. 2017.
- [14] T. M. Bieze, F. Largilliere, A. Kruszewski, Z. Zhang, R. Merzouki, and C. Duriez, "Finite element method-based kinematics and closed-loop control of soft, continuum manipulators," *Soft Robot.*, vol. 5, no. 3, pp. 348–364, 2018.
- [15] M. Thieffry, A. Kruszewski, C. Duriez, and T.-M. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 25–32, Jan. 2019.
- [16] S. Gugercin, A. C. Antoulas, and C. Beattie, " H_2 model reduction for large-scale linear dynamical systems," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 609–638, 2008.
- [17] C. A. Beattie and S. Gugercin, "A trust region method for optimal H_2 model reduction," in *Proc. 48th IEEE Conf. Decis. Control, Jointly, 28th Chin. Control Conf. (CDC/CCC)*, Dec. 2009, pp. 5370–5375.
- [18] E. D. Sontag, "Input to state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*. Berlin, Germany: Springer, 2008, pp. 163–220.
- [19] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, Taipei, Taiwan, 2004.
- [20] C. Poussot-Vassal and P. Vuillemin, "Introduction to MORE: A Model REduction toolbox," in *Proc. IEEE Int. Conf. Control Appl. (CCA)*, Dubrovnik, Croatia, Oct. 2012, pp. 776–781.



Maxime Thieffry is currently pursuing the Ph.D. degree with the Polytechnic University of Hauts-de-France (UPHF), Valenciennes, France.

He works with the DEFormable RObotic SoftWare (DEFROST) Team, Inria, University of Lille, Centrale Lille, CNRS, Villeneuve d'Ascq, France, dedicated to modeling, simulation, and control of soft robots. His research topics are control and observation of dynamical systems with applications to soft robotics.



Thierry-Marie Guerra received the Ph.D. degree in automatic control and the HDR degree from the Polytechnic University Hauts-de-France (UPHF), Valenciennes, France, in 1991 and 1999, respectively.

He is currently a Full Professor with the UPHF. He is the Head of the Laboratory of Industrial and Human Automation, Mechanics and Computer Science (LAMIH), CNRS, UMR 8201, Valenciennes (148 researchers and staff and 110 Ph.D. students and post-docs). His major research fields and topics of interest are wine, hard rock, stamps, nonlinear control, LPV, quasi-LPV (Takagi-Sugeno) models' control and observation, LMI constraints, nonlinear Lyapunov functions, and applications to mobility, soft robotics, and disabled persons.

Dr. Guerra is a member of the IFAC TC 7.1 "Automotive Control," the Chair of the Technical Committee 3.2 "Computational Intelligence in Control" for the International Federation of Automatic Control (IFAC), and an Area Editor of the international journals: *Fuzzy Sets and Systems* and the IEEE TRANSACTIONS ON FUZZY SYSTEMS.



Alexandre Kruszewski received the Ph.D. degree in automatic control from the University of Valenciennes et du Hainaut-Cambrésis, Valenciennes, France, in 2006, and the HDR degree from the University of Lille, Lille, France, in 2017.

He has been an Assistant Professor with Centrale Lille, Villeneuve-d'Ascq, France, since 2007. He is with the DEFormable RObotic SoftWare (DEFROST) Team, CRISTAL Laboratory, UMR 9189, Villeneuve d'Ascq. His main researches deal with the robust stabilization of polytopic systems with application to soft robots.

tems with application to soft robots.



Christian Duriez (S'04–M'13–SM'19) is the Research Director (DR2) of Inria and the Head of the DEFormable RObotic SoftWare (DEFROST) Team, Inria, University of Lille, Centrale Lille, CNRS, Villeneuve d'Ascq, France, dedicated to modeling, simulation, and control of soft robots. He has published more than 100 articles in international journals and conferences and he holds four patents. His research topics are soft robot models and control, fast finite-element methods, simulation of contact response and other complex mechanical interactions, and new algorithms for haptics.

Dr. Duriez is an Associate Editor of the journals the IEEE TRANSACTIONS ON HAPTICS and the IEEE ROBOTICS AND AUTOMATION LETTER. All his research results are developed in SOFA, which is a framework that he co-developed with other INRIA teams. He is also one of the founders of the start-up company InSimo which use his research results and employs more than 20 people.