

Combining Analytic and Experimental Information for Linear Parameter-Varying Model Identification: Application to a Flexible Robotic Manipulator

Dániel Vizer^{1,2,*} / Guillaume Mercère² / Olivier Prot³ / Edouard Laroche⁴

RESEARCH ARTICLE

RECEIVED 28 APRIL 2014; ACCEPTED AFTER REVISION 22 DECEMBER 2014

Abstract

Identifying a linear parameter-varying (LPV) model of a non-linear system from local experimental data is still a problem which deserves attention. Many difficulties related to the determination of the local models with respect to coherent bases have been recently pointed out and must be solved in order to ensure a good behavior of the interpolated LPV model. Rather than building a model either from the law of physics or from experimental data independently, the combination of an analytic and an experimental approach is used in this paper to identify an LPV model of a flexible robotic manipulator. This technique focuses on the interpolation step by combining local re-structured linear time-invariant (LTI) state-space models satisfying a state-space parameterization deduced from the non-linear equations governing the dynamic behavior of the system. A dedicated H_∞ -norm technique is introduced to solve the underlying re-structuring problem. This contribution shows that prior information can be really helpful when the problem of coherent basis selection arises. As a sample, the case of the identification of a 2-DoF non-linear flexible manipulator is addressed.

Keywords

linear parameter-varying model · analytic modeling · robotic manipulator · non-smooth cost function · numerical optimization

1 Introduction

When models of robotic systems are required, a standard solution consists in resorting to a white-box model obtained by combining the laws of physics governing the behavior of the system. Their descriptions are usually based on the Euler-Lagrange equations and the virtual work principle [9]. Interesting from a theoretical point of view, the exclusive use of the standard laws of physics makes the final model quite complex and requires an accurate knowledge of the manipulator as well as high-level skills in robotics especially when different robot structures are handled. This is all the more true when the user wants to have access to physical parameters of the system which are imperfectly known. In order to circumvent these difficulties, efforts dedicated to robot identification are more and more made in industry [17]. However, a direct identification of a non-linear black-box model is often complicated because

- strong non-linearities may vary with the working conditions can appear,
- the development of a global non-linear model structure can rely on strong assumptions such as a uniform density of the manipulator segments or the nature of the deformations if any.

In robotics, linear time-invariant (LTI) models are not sufficient when the system is used over a large workspace. Rather than using a specific non-linear model, a more generic linear parameter-varying (LPV) model can be considered (see, e.g., [37]). The development of LPV model identification for the experimental modeling of robots is advocated for two main reasons. First, from an identification point of view, the introduction of such a structure allows the use of standard tools dedicated to LTI models for the estimation of models with a structural flexibility able to picture time-varying as well as non-linear dynamics. Second, from a control viewpoint, the construction of a reliable LPV model can be seen as a standard but essential initial step for many new control law determination techniques developed in robotics [7].

As far as the LPV model identification is concerned, two families of approaches can be found in the literature. On the one hand,

¹ Department of Control Engineering and Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, H-1117 Budapest, Magyar Tudósok krt. 1-3., Hungary

² Laboratoire d'Informatique et d'Automatique pour les Systèmes, University of Poitiers, 2 rue P. Brousse, bâtiment B25, TSA 41105, 86073 Poitiers Cedex 9, France

³ Institut de Recherche XLIM, University of Limoges, 123 avenue A. Thomas, 87060 Limoges Cedex, France

⁴ University of Strasbourg, Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie, 300 bd Sébastien Brant, CS 10413, 67412 Illkirch Cedex, France

* Corresponding author, e-mail: vizer@iit.bme.hu

some developments focused on a global procedure and assumed that one global experiment can be performed in which the control inputs as well as the scheduling variables can be both excited (see, e.g., [19]). By construction, these techniques are restricted to specific systems where the components of the scheduling variables $\boldsymbol{\kappa} \in \mathbb{K} \subset \mathbb{R}^{n_\kappa}$ are totally controllable and excitable (and not only measurable). On the other hand, other methods are based on a multi-step procedure (see, e.g., [37, 12]) where

1. a finite set of scheduling variable values $\{\boldsymbol{\kappa}_i\}$, $i = \{1, \dots, N_{op}\}$, is handled,
2. local experiments (corresponding to almost constant scheduling variable values) are carried out for each $\boldsymbol{\kappa}_i$, $i = \{1, \dots, N_{op}\}$
3. local LTI models are estimated from the sets of local I/O measurements, for each $\boldsymbol{\kappa}_i$,
4. a global $\boldsymbol{\kappa}$ -dependent model is built from the interpolation of the local LTI models.

This latter viewpoint is often considered for robots or mechanical systems identification (see, e.g., [37]). In the following, such a local approach is considered.

As shown in [35, 34], the description of the local models in coherent bases before the interpolation step is the most difficult step in the local identification procedure. That is the reason why the main theoretical developments suggested in this paper will focus on Step (4) of the afore-described procedure.

Inspired by the discussion in [6], the idea used in the current paper consists in resorting to the knowledge available from the study of the non-linear equations governing the system behavior in order to fix the structure of the global LPV model, then using the available experimental data sets in order to estimate the unknown parameters and to refine the analytic model composed of unknown values. Taking advantages of an initial study of the non-linear equations governing the behavior of the process is suggested in this paper as an interesting solution to circumvent the challenging problem of realizing all the local models with respect to the same state variables.

The paper is organized as follows. Section 2 focuses on the new local data-based identification procedure developed in this paper. A specific attention is paid to the step dedicated to the transformation of the locally-estimated fully-parameterized state-space representations into structured ones via a new H_∞ -based optimization algorithm. Section 3 addresses the validation of the local LPV model identification method on data generated by a simulator of a non-linear flexible robotic manipulator. Concluding remarks are finally gathered into Section 4.

2 Description of the identification procedure

2.1 Problem formulation

In many applicative fields such as robotics, energetics, aeronautics, it is not rare that the modeling of the system is performed by using either an analytic approach or an experimental

one without a strong interaction between both. It is obvious that both families of techniques suffer from limitations and drawbacks which could be relaxed by combining results coming from the study of both approaches. As highlighted in [6], this is the case in the LPV modeling and identification framework where, according to these authors, the synergy of these two research areas (the analytic approach and the experimental one) are not exploited sufficiently well. In this paper, it is suggested these two complementary approaches in order to circumvent a standard problem encountered in the local LPV model identification framework: the description of the local models with respect to a coherent state basis.

As mentioned in Section 1, the identification procedure developed in this paper involves several steps which are presented hereafter.

1. The first one tackles the problem of the determination of a reliable LPV model structure by converting the available non-linear physical representation of the system into an LPV form. This step is called analytic in this paper. Hereafter, it is assumed that this analytic study leads to a (continuous-time) gray-box LPV state-space representation of the system defined as

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\kappa}(t), \boldsymbol{\vartheta})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\kappa}(t), \boldsymbol{\vartheta})\mathbf{u}(t) \quad (1a)$$

$$\mathbf{y}(t) = \mathbf{C}(\boldsymbol{\kappa}(t), \boldsymbol{\vartheta})\mathbf{x}(t) \quad (1b)$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are the input signals, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ are the output signals, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector and $\boldsymbol{\vartheta}$ is the vector of the unknown parameters and $t \in \mathbb{R}$. In this paper, by gray-box, it is meant that the analytic study allows us to fix the structure of the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, i.e., to fix some matrix entries equal to 0 or 1 while the parameters found in the $\boldsymbol{\vartheta}$ vector (to estimate) are affine, rational even non-linear functions of the real physical ones. Contrary to the standard LTI state-space forms, the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ are functions of measurable time-varying signals, gathered into the vector $\boldsymbol{\kappa}(t) \in \mathbb{R}^{n_\kappa}$ and called the scheduling variables, \mathbb{K} being the so-called “scheduling space” [36]. It is furthermore assumed that the system matrices satisfy a static dependence on $\boldsymbol{\kappa}(t)$ [34], i.e., they do not depend on the time-derivatives $(\dot{\boldsymbol{\kappa}}(t), \ddot{\boldsymbol{\kappa}}(t), \dots)$ of the scheduling variables. Notice that because dynamic dependency is not identifiable from local experiments, only static dependence is needed to be tackled by the approach applied in this paper.

2. The second one, more experimental, consists of
 - the selection of a set of constant scheduling variables values for $N_{op} \in \mathbb{N}^*$ local working points so that all the working range of the system is covered and the “distance” between two constant working points is small enough, (see [16, 39] for recent effective solutions for the determination of these local working points)

- the generation, for each working point, of local experimental data sets for which the inputs are correctly excited so that local conditions of excitation are satisfied (see [21, 38] for details according to the techniques used for the identification of the local models as well as discussions concerning the consistence of excitation conditions).
3. The third one, standard when a local approach is undertaken, aims at identifying consistent local black-box LTI models by using dedicated toolboxes and well-known techniques such as the ones available in [21, 38] for DT models or in [10] for CT models.
 4. The fourth one, which is the major contribution of this paper, addresses the conversion of the local (fully-parameterized) black-box models into re-parameterized gray-box state-space forms derived from the frozen structure of the LPV representation obtained in Step (1) and calculated for the considered working points.
 5. Finally, by having access to re-parameterized local models, the structure of which depends on the initial analytic study, the last step of the procedure consists in interpolating the entries of the locally-estimated state-space matrices so that the following global LPV model can provide a good approximation of the behavior of the system over the considered range of operating points.

The main goal of the identification procedure developed in this paper is to yield an accurate gray-box LPV model able to mimic the global I/O dynamic behavior of the system to identify by bypassing the difficult problem of local coherent bases. The solution suggested herein to circumvent this problem resorts to the reliable estimation of local LTI gray-box model parameter vectors \mathcal{G}_i , $i \in \{1, \dots, N_{op}\}$. That is the main reason why in the following of this Section, a particular attention is paid to Step (4). This re-parameterization step is indeed an interesting novelty of this paper and a significant improvement compared to the results available in the literature.

2.2 From a black-box state-space form to a structured one

The main theoretical development introduced in this paper consists in solving the re-structuring problem highlighted in Step (4) of the afore-mentioned identification procedure. In order to reach this goal, the following assumptions must be stated for each working point $i \in \{1, \dots, N_{op}\}$.

- A local frozen LPV structure¹ [36], obtained from Eq. (1) and satisfying a gray-box LTI state-space representation

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\kappa}_i, \mathcal{G}_i) \mathbf{x}(t) + \mathbf{B}(\boldsymbol{\kappa}_i, \mathcal{G}_i) \mathbf{u}(t) \quad (2a)$$

$$\mathbf{y}(t) = \mathbf{C}(\boldsymbol{\kappa}_i, \mathcal{G}_i) \mathbf{x}(t) \quad (2b)$$

can be extracted from the analytic study of the system to identify where the involved matrices are smooth functions of relatively few unknown parameters gathered into a vector \mathcal{G}_i while, $\boldsymbol{\kappa}_i$ stands for the measured frozen scheduling parameter vector.

- The local LTI gray-box model structure defined by Eq. (2) is identifiable (at \mathcal{G}_i^*) at least locally [24, Chapter 2], [21]. This assumption must indeed be satisfied because, for each working, we need to estimate a unique parameter vector \mathcal{G}_i .
- A consistent local fully-parameterized minimal state-space realization $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$ has been estimated from the available local data set by using, *e.g.*, a subspace-based identification technique² [38] associated with a discrete-to-continuous time conversion procedure, or, according to the user's affinities, a continuous-time identification algorithm [10]. Notice that, at this step, the local models does not have to have the same number of states. On top of that, this fact is one of the main advantages of the proposed technique.

The solutions developed in the literature [40,30,31,32] mainly aim at determining uniquely, for $i \in \{1, \dots, N_{op}\}$, the N_{op} similarity transformations \mathbf{T}_i as well as the N_{op} vectors \mathcal{G}_i satisfying³

$$\mathbf{A}_i \mathbf{T}_i = \mathbf{T}_i \mathbf{A}(\mathcal{G}_i) \quad \mathbf{B}_i = \mathbf{T}_i \mathbf{B}(\mathcal{G}_i) \quad \mathbf{C}_i \mathbf{T}_i = \mathbf{T}_i \mathbf{C}(\mathcal{G}_i). \quad (3)$$

Unfortunately, these solutions still resort to a set of bilinear equations, feature which leads to non-convex optimizations. As claimed by L. Xie and L. Ljung in [40], “the fewer variables, the better”. In order to reduce the number of involved unknown parameters or matrices, it is suggested hereafter considering the equality between the transfer function forms as

$$\underbrace{\mathbf{C}_i(\mathcal{G}_i) (s\mathbf{I}_{n_x \times n_x} - \mathbf{A}_i(\mathcal{G}_i))^{-1} \mathbf{B}_i(\mathcal{G}_i)}_{\mathcal{G}_i(s, \mathcal{G}_i)} = \underbrace{\mathbf{C}_i (s\mathbf{I}_{n_x \times n_x} - \mathbf{A}_i)^{-1} \mathbf{B}_i}_{\mathcal{G}_i(s)}, \quad i \in \{1, \dots, N_{op}\}, \quad (4)$$

where s is the Laplace transform variable and for which the similarity transformation matrix disappears. Such a simplification leads us to determine the structured matrices, more precisely the parameter vector \mathcal{G} which optimizes

$$\|\mathcal{G}_i(s, \mathcal{G}) - \mathcal{G}_i(s)\|_0^2, \quad i \in \{1, \dots, N_{op}\}. \quad (5)$$

Besides discarding the similarity transformation, the use of local transfer functions does not constrain the user to identify, for each working point, local fully-parameterized models with

² The subspace-based identification techniques [21, 38] are really good candidates to solve this problem. These algorithms can indeed yield consistent estimates in many different noisy cases.

³ In order to lighten the notations, the subscript i will be dropped when the context is clear.

¹ By fixing $\boldsymbol{\kappa}(t)$ in the i th position, to have $\boldsymbol{\kappa}_i$.

equal orders. This observation is an interesting practical feature of the identification technique developed in this paper.

Hereafter, a specific attention is paid to the H_∞ -norm. So, the cost function is reformulated as follows,

$$F(\boldsymbol{\vartheta}) = \left\| \mathbf{G}_i(s, \boldsymbol{\vartheta}) - \mathfrak{G}_i(s) \right\|_\infty^2, \quad i \in \{1, \dots, N_{op}\}. \quad (6)$$

The use of the H_∞ -norm can first be justified by noticing that, by definition, the H_∞ -norm is the maximal singular value of the complex gain matrix over all the frequencies. Thus, thanks to the norm property, if the cost function in Eq. (6) is small enough (*i.e.*, a lot smaller than 1), then the distance between the involved black-box model and the structured gray-box one is small as well.

Before introducing an efficient algorithm dedicated to the optimization of the cost function $F(\boldsymbol{\vartheta})$, it is interesting to draw the links between the H_∞ -norm-based technique proposed herein and the methods developed in the literature until now. Such a discussion should indeed enhance the impact of the following developments.

Remark 1. *The use of the H_∞ -norm as well as the cost function $F(\boldsymbol{\vartheta})$ may echo back to the worst-case and the set-membership identification methods [13, 23, 26] for which the main goal consists in yielding a guaranteed uncertainty set usable by the standard robust control design tools. However, it is important to point out that the technique proposed herein does not aim at quantifying any model quality or uncertainty set through the H_∞ -norm. Besides the reasons highlighted beforehand, the H_∞ -norm is used herein because of the availability of efficient numerical algorithms able to minimize the criterion $F(\boldsymbol{\vartheta})$ efficiently.*

2.3 Comments and discussion

As mentioned above, the first studies focused on the bilinear equations defined by Eq. (3). More precisely, in [40, 30], then later in [31], the following cost function has been suggested in order to compute the value of the parameter vector $\boldsymbol{\vartheta}$ and the similarity matrix \mathbf{T} (which is assumed to be invertible) involved in Eq. (3)

$$F(\boldsymbol{\vartheta}, \mathbf{T}) = \left\| \mathfrak{A}\mathbf{T} - \mathbf{T}\mathbf{A}(\boldsymbol{\vartheta}) \right\|_F^2 + \left\| \mathfrak{B} - \mathbf{T}\mathbf{B}(\boldsymbol{\vartheta}) \right\|_F^2 + \left\| \mathfrak{C}\mathbf{T} - \mathbf{C}(\boldsymbol{\vartheta}) \right\|_F^2, \quad (7)$$

where $\|\cdot\|_F$ is the Forbenius norm [15]. The use of such a cost function is quite straightforward in our context when the objective is to solve the set of matrix equations (3). On the contrary, in this paper, for $\omega \in [0, \infty[$, we concentrate on the difference

$$\Sigma(j\omega, \boldsymbol{\vartheta}) = \mathbf{C}(\boldsymbol{\vartheta})(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}_i(\boldsymbol{\vartheta}))^{-1} \mathbf{B}(\boldsymbol{\vartheta}) - \mathfrak{C}(j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} \mathfrak{B}. \quad (8)$$

The goal of this subsection is to highlight the links between $F(\boldsymbol{\vartheta}, \mathbf{T})$ and $\|\Sigma(j\omega, \boldsymbol{\vartheta})\|_F$ in order to point out why dealing with $\Sigma(j\omega, \boldsymbol{\vartheta})$ should be favored. In order to reach this goal,

first let us determine a bound of $\|\Sigma(j\omega, \boldsymbol{\vartheta})\|_F$, then analyze this bound. A straightforward calculation shows that

$$\begin{aligned} \Sigma(j\omega, \boldsymbol{\vartheta}) &= \mathfrak{C}(j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} (\mathbf{T}\mathbf{B}(\boldsymbol{\vartheta}) - \mathfrak{B}) \\ &\quad + (\mathbf{C}(\boldsymbol{\vartheta}) - \mathfrak{C}\mathbf{T})(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} \mathbf{B}(\boldsymbol{\vartheta}) \\ &\quad + \mathfrak{C} \left(\mathbf{T}(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} - (j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} \mathbf{T} \right) \mathbf{B}(\boldsymbol{\vartheta}). \end{aligned} \quad (9)$$

Now, by using the identity $(\mathbf{A}^{-1} - \mathbf{B}^{-1})^{-1} = \mathbf{A}(\mathbf{B} - \mathbf{A})^{-1}\mathbf{B}$ [15], that is, $\mathbf{A}^{-1} - \mathbf{B}^{-1} = \mathbf{B}^{-1}(\mathbf{B} - \mathbf{A})\mathbf{A}^{-1}$, the following relation can be deduced

$$\begin{aligned} &\left(\mathbf{T}(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} - (j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} \mathbf{T} \right)^{-1} = \\ &\left(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}) \right) (\mathbf{T}\mathbf{A}(\boldsymbol{\vartheta}) - \mathfrak{A}\mathbf{T})^{-1} (j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A}). \end{aligned} \quad (10)$$

Thus,

$$\begin{aligned} \Sigma(j\omega, \boldsymbol{\vartheta}) &= \mathfrak{C}(j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} (\mathbf{T}\mathbf{B}(\boldsymbol{\vartheta}) - \mathfrak{B}) \\ &\quad + (\mathbf{C}(\boldsymbol{\vartheta}) - \mathfrak{C}\mathbf{T})(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} \mathbf{B}(\boldsymbol{\vartheta}) \\ &\quad + \mathfrak{C}(j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} (\mathbf{T}\mathbf{A}(\boldsymbol{\vartheta}) - \mathfrak{A}\mathbf{T})(j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} \mathbf{B}(\boldsymbol{\vartheta}). \end{aligned} \quad (11)$$

Finally, with

$$\begin{aligned} \Sigma_B(j\omega, \boldsymbol{\vartheta}) &= \left\| (j\omega\mathbf{I}_{n_x \times n_x} - \mathbf{A}(\boldsymbol{\vartheta}))^{-1} \mathbf{B}(\boldsymbol{\vartheta}) \right\|_F, \\ \Sigma_C(j\omega, \boldsymbol{\vartheta}) &= \left\| \mathfrak{C}(j\omega\mathbf{I}_{n_x \times n_x} - \mathfrak{A})^{-1} \right\|_F \end{aligned} \quad (12)$$

and because $\|\Sigma(j\omega, \boldsymbol{\vartheta})\|_2 \leq \|\Sigma(j\omega, \boldsymbol{\vartheta})\|_F$ [15], the following inequality can be deduced

$$\begin{aligned} \sigma_1(\Sigma(j\omega, \boldsymbol{\vartheta})) &\leq \left\| \Sigma(j\omega, \boldsymbol{\vartheta}) \right\|_F \\ &\leq \sqrt{\varepsilon} \left[\Sigma_B(j\omega, \boldsymbol{\vartheta}) + \Sigma_C(j\omega, \boldsymbol{\vartheta}) \right], \end{aligned} \quad (13)$$

where $\sigma_1(\cdot)$ is the maximum singular value function [15] and where ε is the minimal value of the cost function $F(\boldsymbol{\vartheta}, \mathbf{T})$. This inequality shows that the distance between the transfer functions of both system representations, *i.e.*, $(\mathfrak{A}_i, \mathfrak{B}_i, \mathfrak{C}_i)$ and $(\mathbf{A}(\boldsymbol{\vartheta}), \mathbf{B}(\boldsymbol{\vartheta}), \mathbf{C}(\boldsymbol{\vartheta}))$, decreases with ε and is smaller for high values of ω . Indeed, even if the value of ε is small, the value of $\sigma_1(\Sigma(j\omega, \boldsymbol{\vartheta}))$ may be quite large at some frequencies. This is the case, for instance, if Σ_B or Σ_C have large resonance peaks. For all of these reasons, in the Authors' opinion, the minimization of the objective $F(\boldsymbol{\vartheta})$ should be favored to identify the parameter vector $\boldsymbol{\vartheta}$ rather than minimizing the cost function $F(\boldsymbol{\vartheta}, \mathbf{T})$ because of

- the direct interpretation of the optimal cost in terms of input/output energy,
- the non-explicit involvement of the similarity matrix into the cost function Eq. (6).

2.4 Computing the H_∞ -norm

Based on Eq. (6), the following cost function can be defined

$$\min_{\mathfrak{g}} F(\mathfrak{g}) \quad (14)$$

Then, in order to solve the term defined by Eq. (14), *i.e.*, in order to minimize the H_∞ -norm of $\Sigma(s, \mathfrak{g})$, the first step consists of the computation of the value of this function efficiently. Clearly, by definition the Laplace variable with $j\omega$,

$$\|\Sigma(j\omega, \mathfrak{g})\|_\infty^2 = \max_{\omega \in [0, \infty]} \lambda_1 \left(\Sigma^H(j\omega, \mathfrak{g}) \Sigma(j\omega, \mathfrak{g}) \right), \quad (15)$$

where $\lambda_1(\cdot)$ is the maximum eigenvalue function. Thus, solving the problem given by Eq. (14) is equivalent to the following optimization problem

$$\min_{\mathfrak{g}} \max_{\omega \in [0, \infty]} \lambda_1 \left(\Sigma^H(j\omega, \mathfrak{g}) \Sigma(j\omega, \mathfrak{g}) \right). \quad (16)$$

The maximum eigenvalue function is a convex but a non-smooth function [2]. As a consequence, from Eq. (16), the computation of the objective function seems to be a hard task because, by construction, the global optimum of a non-smooth function over $[0, \infty]$ must be estimated. Fortunately, when Σ has a state-space representation, this step can be performed efficiently by using a bisection algorithm based on a hamiltonian calculus [3]. Thus, under the assumption that Σ can be realized by a state-space representation, the map $\omega \rightarrow \Sigma(j\omega, \mathfrak{g})$ is a rational matrix function which is smooth for all $\omega \in [0, \infty]$. The bisection algorithm used to compute the value of the cost function involved in problem (16) moreover permits the computation of the set of active frequencies $\Omega(\mathfrak{g})$ defined as follows

$$\Omega(\mathfrak{g}) = \left\{ \omega \in [0, \infty] : \lambda_1 \left(\Sigma^H(j\omega, \mathfrak{g}) \Sigma(j\omega, \mathfrak{g}) \right) = \|\Sigma(\cdot, \mathfrak{g})\|_\infty^2 \right\}. \quad (17)$$

This set is of prime importance for the following non-smooth minimization algorithm. It is indeed used to define as well as to compute the sub-gradients of the underlying cost function. As shown in [3], the number of frequencies in $\Omega(\mathfrak{g})$ is either finite, or $\Omega(\mathfrak{g}) = [0, \infty]$. In our context, because the structured models ($A(\mathfrak{g}), B(\mathfrak{g}), C(\mathfrak{g})$) as well as the black-box ones ($\mathfrak{A}, \mathfrak{B}, \mathfrak{C}$) do not have a direct transmission from \mathbf{u} to \mathbf{y} , the case $\Omega(\mathfrak{g}) = [0, \infty]$ can only be observed when the cost function has been identified without any error, *i.e.*, when the parameter vector \mathfrak{g} has been identified.

2.5 Minimizing the H_∞ -norm

In the last years, many algorithms have been developed for the minimization of non-convex and non-smooth functions [1,4,14]. These algorithms have been successfully applied in the control framework, for example, in order to synthesize stabilizing controllers while this kind of problems are known to be NP-hard [28]. In this paper, it has been decided to use a proximity control algorithm, described in [29,3], to perform

the minimization of the non-smooth function $F(\mathfrak{g})$. Originally, this algorithm is an improvement of the method described in [1], with better convergences properties. The main idea of the algorithm consists in using a local convex model φ of the cost function as a ‘‘cutting plane generator’’ from which a polyhedral model of the objective is built and used to generate trial points in order to minimize the cost function value. Because of the inherent instability of the cutting plane method [5], a proximity control term is introduced into the model in order to stabilize the iterations. This algorithm is a first order method and is thus quite slow to converge.

In order to use this method for our identification problem, a local convex model for the objective function is required as well as specific cutting planes. This is the purpose of the **next Subsection**.

2.6 Local model and cutting planes computation

In order to compute the aforementioned local planes, the lines of [3] must be followed. More precisely the following local model for the objective function (6) at point \mathfrak{g}_k can be computed

$$\begin{aligned} \phi(\mathfrak{g}_{k+1}, \mathfrak{g}_k) &= \max_{\omega \in [0, \infty]} \max_{Z \in \mathcal{H}} Z \bullet \left[\Sigma^H(j\omega, \mathfrak{g}_k) \Sigma(j\omega, \mathfrak{g}_k) \right. \\ &+ \left(d \Sigma(j\omega, \mathfrak{g}_k) \cdot (\mathfrak{g}_{k+1} - \mathfrak{g}_k) \right)^H \Sigma(j\omega, \mathfrak{g}_k) \\ &\left. + \Sigma^H(j\omega, \mathfrak{g}_k) \left(d \Sigma(j\omega, \mathfrak{g}_k) \cdot (\mathfrak{g}_{k+1} - \mathfrak{g}_k) \right) \right] \end{aligned} \quad (18)$$

where \mathcal{H} is the set of hermitian semi-definite positive matrices with a trace equal to one, \bullet is the Frobenius scalar product and $d\Sigma(j\omega, \mathfrak{g}_k)$ denotes the differential of Σ with respect to the variable \mathfrak{g}_k . A cutting planes of $\varphi(\cdot, \mathfrak{g}_k)$ at point \mathfrak{g}_{k+1} reads

$$y \mapsto \alpha + g^T (y - x), \quad (19)$$

where

$$\begin{aligned} \alpha &= \mathbf{Z}_\omega \bullet \left(\Sigma^H(j\omega, \mathfrak{g}_k) \Sigma(j\omega, \mathfrak{g}_k) \right), \\ g &= \mathbf{Z}_\omega \bullet \left(\left(d \Sigma(j\omega, \mathfrak{g}_k) \cdot (\mathfrak{g}_2 - \mathfrak{g}_k) \right)^H \Sigma(j\omega, \mathfrak{g}_k) \right. \\ &\left. + \Sigma^H(j\omega, \mathfrak{g}_k) \left(d \Sigma(j\omega, \mathfrak{g}_k) \cdot (\mathfrak{g}_2 - \mathfrak{g}_k) \right) \right). \end{aligned} \quad (20)$$

In Eq. (20), $\omega \in \Omega(\mathfrak{g}_k)$ and \mathbf{Z}_ω is such that

$$\mathbf{Z}_\omega \bullet \left(\Sigma(j\omega, \mathfrak{g}_k)^H \Sigma(j\omega, \mathfrak{g}_k) \right) = \lambda_1 \left(\Sigma^H(j\omega, \mathfrak{g}_k) \Sigma(j\omega, \mathfrak{g}_k) \right). \quad (21)$$

There is no unique choice for the matrix \mathbf{Z}_ω which satisfies Eq. (21), but the algorithm just needs to compute one of them. This matrix can be easily derived from linear algebra, for example by using the eig function in Matlab.

In order to use the proximity control algorithm, the differential of the smooth function $\mathfrak{g} \rightarrow \Sigma(j\omega, \mathfrak{g})$ must be computed. Now, by assuming that Σ has the following state-space representation

$$\dot{\mathbf{x}} = \mathcal{A}(\boldsymbol{\vartheta})\mathbf{x} + \mathcal{B}(\boldsymbol{\vartheta})\mathbf{u}, \quad \mathbf{y} = \mathcal{C}(\boldsymbol{\vartheta})\mathbf{x}, \quad (22)$$

that is, $\Sigma(j\omega, \boldsymbol{\vartheta}) = \mathcal{C}(\boldsymbol{\vartheta})(j\omega\mathbf{I} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}\mathcal{B}(\boldsymbol{\vartheta})$, then, by defining

$$\mathbf{a}(\boldsymbol{\vartheta}) = \text{vec}(\mathcal{A}(\boldsymbol{\vartheta})), \quad \mathbf{b}(\boldsymbol{\vartheta}) = \text{vec}(\mathcal{B}(\boldsymbol{\vartheta})), \quad \mathbf{c}(\boldsymbol{\vartheta}) = \text{vec}(\mathcal{C}(\boldsymbol{\vartheta})), \quad (23)$$

where $\text{vec}(\cdot)$ is the vectorization operator [15], the differential of Σ with respect to variable $\boldsymbol{\vartheta}$ satisfies

$$\begin{aligned} d\Sigma(j\omega, \boldsymbol{\vartheta}) \cdot h &= (d\mathcal{C}(\boldsymbol{\vartheta}) \cdot h)(j\omega\mathbf{I} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}\mathcal{B}(\boldsymbol{\vartheta}) \\ &+ \mathcal{C}(\boldsymbol{\vartheta})(j\omega\mathbf{I}_{n_x \times n_x} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}(d\mathcal{A}(\boldsymbol{\vartheta}) \cdot h)(j\omega\mathbf{I}_{n_x \times n_x} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}\mathcal{B}(\boldsymbol{\vartheta}) \\ &+ \mathcal{C}(\boldsymbol{\vartheta})(j\omega\mathbf{I}_{n_x \times n_x} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}(d\mathcal{B}(\boldsymbol{\vartheta}) \cdot h). \end{aligned} \quad (24)$$

Then, by using again the vectorization operator, we get

$$\begin{aligned} \text{vec}(d\Sigma(j\omega, \boldsymbol{\vartheta}) \cdot h) &= \\ &\left[(\mathbf{M}_B(\boldsymbol{\vartheta})^\top \otimes \mathbf{I}_{n_x \times n_x}) J_c(\boldsymbol{\vartheta}) + (\mathbf{M}_B(\boldsymbol{\vartheta})^\top \otimes \mathbf{M}_C(\boldsymbol{\vartheta})) J_a(\boldsymbol{\vartheta}) \right. \\ &\left. + (\mathbf{I}_{n_x \times n_x} \otimes \mathbf{M}_C(\boldsymbol{\vartheta})) J_b(\boldsymbol{\vartheta}) \right] \cdot h \end{aligned} \quad (25)$$

where

$$\begin{aligned} \mathbf{M}_B(\boldsymbol{\vartheta}) &= (j\omega\mathbf{I}_{n_x \times n_x} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1}\mathcal{B}(\boldsymbol{\vartheta}) \\ \mathbf{M}_C(\boldsymbol{\vartheta}) &= \mathcal{C}(\boldsymbol{\vartheta})(j\omega\mathbf{I}_{n_x \times n_x} - \mathcal{A}(\boldsymbol{\vartheta}))^{-1} \end{aligned}$$

and J_a , J_b and J_c are the Jacobian of \mathbf{a} , \mathbf{b} and \mathbf{c} respectively. From Eq. (25), the expression of the Jacobian matrices of $\boldsymbol{\vartheta} \rightarrow \text{vec}(d\Sigma(j\omega, \boldsymbol{\vartheta}))$ can be obtained because the vectorization operator is linear. Then, we are finally able to compute the value of g in Eq. (20) by applying the inverse vectorization. All the ingredients for the use of the proximity control algorithm are available: a local model and a way to compute a cutting plane (α , g) at some points.

3 Experimental modeling of a flexible arm

3.1 Robot manipulator description

In this study, the system is a horizontal flexible arm composed of two flexible segments ($n_\theta = 2$) as depicted in Fig. 1a. Such a structure can be found, e.g., when considering the two first rotoid joints of a SCARA manipulator. Under specific working conditions, this type of manipulator may have significant flexibilities. Indeed, even if these deformations only yield short displacements of the end-effector, this is sufficient to restrict the bandwidth of the control loop, as shown in [8]. Therefore, a model of these flexible modes is necessary in order to design a vision-based control loop with high bandwidth. For instance, these flexible characteristics are satisfied by a prototype designed by SINTERS and used in [11] (see Fig. 1b). This robot is lightweight because it was designed to attain fast dynamics in order to compensate the heart tissue motion for cardiac surgery. As a result, it is observed that the

bandwidth is restricted by flexible modes that can be attributed to small deformations of the segments. This flexible system, more precisely a simulator of this surgery robot, will be used in the following for the validation of the identification techniques introduced beforehand.

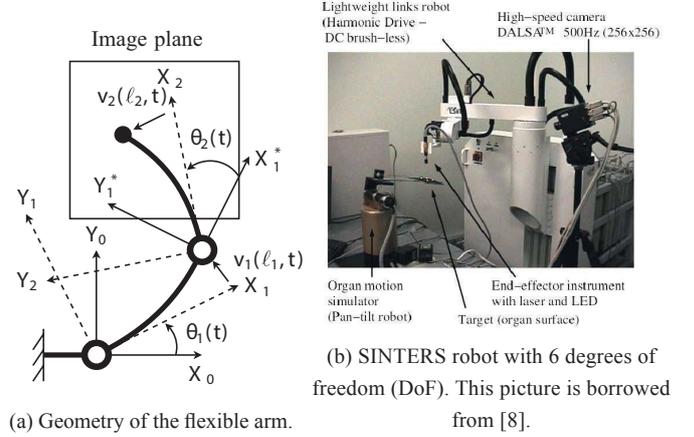


Fig. 1. Flexible robotic manipulator.

3.2 Non-linear and linearized dynamic models

Under the assumption of Euler-Bernoulli beam, the dynamic equations of a flexible arm can be derived by using the assumed-mode method where the deformation field is decomposed into a finite sum of elementary deformations [33]. In the current case, small deformations are considered and only one mode is chosen for the transverse deformation field. For segment $\#k$, $k = 1, \dots, n_\theta$, the deformation field writes $\delta_k(x, t) = x^2 v_k(t)$, where x represents the abscissa along the segment and $v_k(t)$ is the state of the deformation. Therefore, the resulting deformation at the end of the segment of length ℓ_k is $\delta_k(\ell_k, t) = \ell_k^2 v_k(t)$. The dynamic model is derived from the Virtual Work Principle using the DynaFlex toolbox developed on Maple (see [33]). By denoting $\boldsymbol{\theta} = [\theta_1 \ \theta_2]^\top \in \mathbb{R}^{n_\theta}$ and $\mathbf{v} = [v_1 \ v_2]^\top \in \mathbb{R}^{n_v}$, the resulting model relies on a generalized position vector $\mathbf{q} = [\boldsymbol{\theta}^\top \ \mathbf{v}^\top]^\top \in \mathbb{R}^{n_q}$, $n_q = n_\theta + n_v$, and writes (see Fig. 1a for the notations)

$$\mathcal{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) = \mathcal{F}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathcal{G}\mathbf{u}(t) \quad (26)$$

where $\mathcal{M}(\mathbf{q})$ is the inertia matrix, $\mathcal{F}(\mathbf{q}, \dot{\mathbf{q}})$ is a generalized force vector which includes the Coriolis and centrifugal effects (see [18] for details about the mathematical expressions of the matrix \mathcal{M} and the vector \mathcal{F}). The torque vector $\mathbf{u} = [u_1 \ u_2]^\top$ has only effects on the dynamics of the rigid positions θ_1 and θ_2 , corresponding to

$$\mathcal{G} = \begin{bmatrix} \mathbf{I}_{n_\theta \times n_\theta} \\ \mathbf{0}_{n_v \times n_\theta} \end{bmatrix} \quad (27)$$

The x and y positions of the end-effector can be written from the geometric model, resulting in the non-linear measurement equation $\mathbf{z} = \mathbf{g}(\mathbf{q})$, i.e.,

$$z_1 = \left(\ell_1 - \frac{2}{3} \ell_1^3 v_1^3 \right) \cos(\theta_1) - \ell_1^2 v_1 \sin(\theta_1) + \left(\ell_2 - \frac{2}{3} \ell_2^3 v_2^3 \right) \cos(\theta_{12}) - \ell_2^2 v_2 \sin(\theta_{12}) \quad (28a)$$

$$z_2 = \left(\ell_1 - \frac{2}{3} \ell_1^3 v_1^3 \right) \sin(\theta_1) + \ell_1^2 v_1 \cos(\theta_1) + \ell_2^2 v_2 \cos(\theta_{12}) + \left(\ell_2 - \frac{2}{3} \ell_2^3 v_2^3 \right) \sin(\theta_{12}) \quad (28b)$$

where $\theta_{12} = \theta_1 + \theta_2 + 2\ell_1 v_1$. Each arm is equipped with a local velocity controller with input

$$\mathbf{u}(t) = \Lambda(\dot{\theta}^*(t) - \dot{\theta}(t)) \quad (29)$$

(see [18] for a detailed description). In order to fix the structure of the global LPV model required by the technique introduced in this paper, a standard Jacobian linearization can be applied to the generalized second order model given in Eq. (26). This linearization step removes the Coriolis terms. Therefore, the resulting model is valid only when low velocities are demanded, which fits to the considered medical application where the movements have small amplitudes around the involved working points. In this special case, $\kappa = \cos(\theta_2)$ is selected to be the scheduling variable. The input of the system is the joint velocity reference vector denoted hereafter by $\dot{\theta}^*(t) = [\dot{\theta}_1^*(t) \dot{\theta}_2^*(t)]^T$.

Such a linear technique leads to the following minimal state-space representation of order 6,

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\kappa, \mathcal{G}) \mathbf{x}(t) + \mathbf{B}(\kappa, \mathcal{G}) \dot{\theta}^*(t), \quad (30a)$$

$$\mathbf{y}(t) = \mathbf{C} \mathbf{x}(t), \quad (30b)$$

$$\mathbf{x} = \left[\mathbf{v}^T \dot{\theta}^T \dot{\mathbf{v}}^T \right]^T, \quad (30c)$$

where⁴

$$\mathbf{A}(\kappa, \mathcal{G}) = \begin{bmatrix} \mathbf{0}_{n_v \times n_\theta} \\ \left(\mathbf{M}^{-1} \mathbf{K}(\kappa, \mathcal{G}) \right) (:, n_\theta + 1 : \text{end}) \\ \mathbf{0}_{n_v \times n_v} & \mathbf{I}_{n_v \times n_\theta} \\ -\mathbf{M}^{-1} \mathbf{G} \Lambda(\kappa, \mathcal{G}) & \mathbf{0}_{n_v \times n_v} \end{bmatrix} \quad (31a)$$

$$\mathbf{B}(\kappa, \mathcal{G}) = \begin{bmatrix} \mathbf{0}_{n_v \times n_\theta} \\ \mathbf{M}^{-1} \mathbf{G} \Lambda(\kappa, \mathcal{G}) \end{bmatrix} \quad \mathbf{C}(\kappa, \mathcal{G}) = \left[\mathbf{0}_{n_\theta \times n_v} \quad \mathbf{C}_1 \right], \quad (31b)$$

with

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & \ell_1 & 0 \\ 0 & 1 & \ell_1 & \ell_2 \end{bmatrix} \quad (31c)$$

and where $\mathbf{M}(\kappa, \mathcal{G})$, $\mathbf{K}(\kappa, \mathcal{G})$, $\mathbf{G}(\kappa, \mathcal{G})$, \mathbf{C}_1 are the linearized versions of the matrices and vectors defined by Eq. (26)-(28) and $\Lambda = \text{diag}(\lambda_1, \lambda_2)$ is the diagonal matrix of the control gains acting on the joints. The $\mathcal{G} = \{\mathcal{G}_j\}_{j=1}^{16}$ vector is composed of all the parameters of the system matrices except the ones of \mathbf{C} and the known parameters 0 and 1 found in the other matrices. More precisely, the matrices have the following fix structure for a frozen value of κ denoted hereafter by κ_i with $i = \{1, \dots, N_{op}\}$,

$$\mathbf{A}(\kappa_i, \mathcal{G}_i) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \mathcal{G}_{i1} & \mathcal{G}_{i5} & \mathcal{G}_{i9} & \mathcal{G}_{i12} & 0 & 0 \\ \mathcal{G}_{i2} & \mathcal{G}_{i6} & \mathcal{G}_{i10} & \mathcal{G}_{i13} & 0 & 0 \\ \mathcal{G}_{i3} & \mathcal{G}_{i7} & \mathcal{G}_{i11} & \mathcal{G}_{i14} & 0 & 0 \\ \mathcal{G}_{i4} & \mathcal{G}_{i8} & \mathcal{G}_{i12} & \mathcal{G}_{i15} & 0 & 0 \end{bmatrix}, \quad (32)$$

$$\mathbf{B}(\kappa_i, \mathcal{G}_i) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\mathcal{G}_{i9} & -\mathcal{G}_{i13} \\ -\mathcal{G}_{i10} & -\mathcal{G}_{i14} \\ -\mathcal{G}_{i11} & -\mathcal{G}_{i15} \\ -\mathcal{G}_{i12} & -\mathcal{G}_{i16} \end{bmatrix}.$$

This system of equations (see Eq. (30)-(31)) will be used in the following to fix the structure of the locally-estimated models and as a linearized analytic model, computable when its parameters are assumed to be known, for the validation of the local and the global models. These “known” (or computable) local models will be called analytic in the sequel.

3.3 Identification results

3.3.1 Local model estimation

The work performed in the previous Subsection gives access to a reliable LPV model structure. Now, the steps composing the local identification procedure described in Section 2 can be addressed. First, $N_{op} = 7$ local working points, then 7 local I/O data sets must be selected and generated. For the specific test-bed considered in this paper, 7 constant values of θ_2 in the set $\{i\pi/8 : i = 1, \dots, 7\}$ are selected, then 7 noise-free I/O data sets are acquired. This noise-free I/O data generation is made possible because, herein, a Simulink model of the flexible robotic manipulator was developed to carry out the simulations. The inputs of the system are angular velocity reference $\dot{\theta}^*(t) = [\dot{\theta}_1^*(t) \dot{\theta}_2^*(t)]^T$ and are chosen as two uncorrelated pseudo-random binary sequences built so that all the dynamics of the system are well-excited. The outputs of the model are \mathbf{y} , *i.e.*, the angular velocities of the fictitious rigid robot having the same geometry of the flexible robot. This noise-free I/O data is generated once for each value of $\theta_2 \in \{i\pi/8 : i = 1, \dots, 7\}$. Then, a Monte Carlo simulation of size 100 is used for the estimation of the local models. This Monte Carlo simulation is carried

⁴ For our system, $\mathbf{D} = \mathbf{0}_{n_v \times n_v}$.

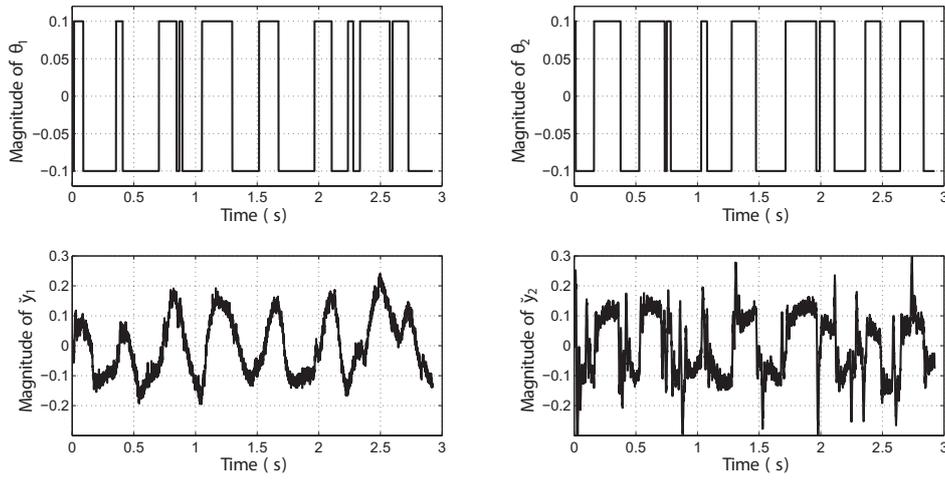


Fig. 2. I/O data sample.

out by adding up on the noise-free outputs 100 realizations of two uncorrelated zero-mean white Gaussian noises satisfying a signal-noise ratio⁵ equal to 30dB. The length of each local data set is equal to 30, 000 samples with a sampling period of 0.1 ms. For the identification, a down-sampling is performed with a rate equal to 10. A sample of noisy I/O data is given in Fig. 2.

The next step of the local identification procedure consists in estimating, for each run of the Monte Carlo simulation, reliable local models. To reach this goal,

- first the PI-MOESP algorithm [38] is applied to perform the local estimation from the local I/O noisy data sets. This subspace-based identification algorithm is indeed well-known to be efficient under the noisy conditions described beforehand,
- second the local fully-parametrized state-space representations are balanced (e.g., with the function **balreal** of MATLAB). This balancing step is performed essentially because the multi-step technique developed in this article will be compared hereafter with the technique suggested in [25] where local balanced realizations are required. However, we also use this balancing procedure in our technique in order to take advantage of the numerical reliability of the balanced realizations. While, for each working point, the local models can have different orders, the averaging step applied hereafter requires, for each value of $i \in \{1, \dots, 7\}$, local black-box models realized with respect to the same state basis and with the same order. The balanced state-space form satisfies such a constraint (up to the sign) [27] and may help us if model reduction is necessary.

This two-step procedure is third completed by a discrete-to-continuous-time transformation. This step is required because (i) the system under study is by construction continuous-time,

(ii) the PI-MOESP algorithm only leads to discrete-time models. This discrete-to-continuous-time domain conversion is performed by using the bilinear Tustin approximation to the derivative (e.g., the function **d2cm** available in MATLAB). Once 100 local models are estimated for each operating point, 7 local models can be computed by averaging these 100 local models. Notice that during this averaging step, it is beneficial that the local models are balanced. These 7 average local models are then locally validated by considering two complementary tools

- two I/O fit measurements (see Eq. (33)) evaluated on N_{op} new noise-free data sets generated for this task (cross-validation),
- a comparison of the frequency responses (magnitude Bode plots) of the estimated local model and an analytic local model calculated from a linearization of the non-linear equations governing the behavior of the system.

For $k \in [1, n_y]$, the following fit measurements⁶ are introduced in order to quantify the model quality on validation data (i.e., a data set different from the one used for the estimation)

$$\text{BFT}_k = 100 \times \max \left(1 - \frac{\|y_k - \hat{y}_k\|}{\|y_k - \text{mean}(y_k)\|}, 0 \right) \quad (33a)$$

$$\text{VAF}_k = 100 \times \max \left(1 - \frac{\text{var}(y_k - \hat{y}_k)}{\text{var}(y_k)}, 0 \right) \quad (33b)$$

Table 1 (see also the time responses in Fig. 3 for a qualitative validation for $\theta_2 = 5\pi / 8$) gathers these fit measurements for $\theta_2 \in \{i\pi/8 : i = 1, \dots, 7\}$. From these values, it can be concluded that, for each operating point, the estimated local LTI model describes the actual system quite well. The reader must keep in mind that the system behavior is highly non-linear.

⁵ The signal-to-noise ratio is defined as follows: $\text{SNR} = 10 \log \left(\frac{\text{cov}\{y_k\}}{\text{cov}\{v\}} \right)$, $k \in [1, n_y]$ where v stands for the noise acting on the noise-free output y_k .

⁶ y_k stands for the k^{th} system output and \hat{y}_k for its estimate. $\text{var}(\bullet)$ is the variance of \bullet .

Tab. 1. Performance metrics for the estimated LTI models on validation data.

θ_2	$\pi / 8$	$2\pi / 8$	$3\pi / 8$	$4\pi / 8$	$5\pi / 8$	$6\pi / 8$	$7\pi / 8$
BFT ₁ (%)	85.83	91.23	90.23	92.09	89.65	88.42	91.01
BFT ₂ (%)	67.42	98.18	85.07	97.83	88.97	89.82	97.23
VAF ₁ (%)	97.98	99.23	99.04	99.37	98.93	98.66	99.2
VAF ₂ (%)	89.39	99.96	97.77	99.9	98.78	98.96	99.9

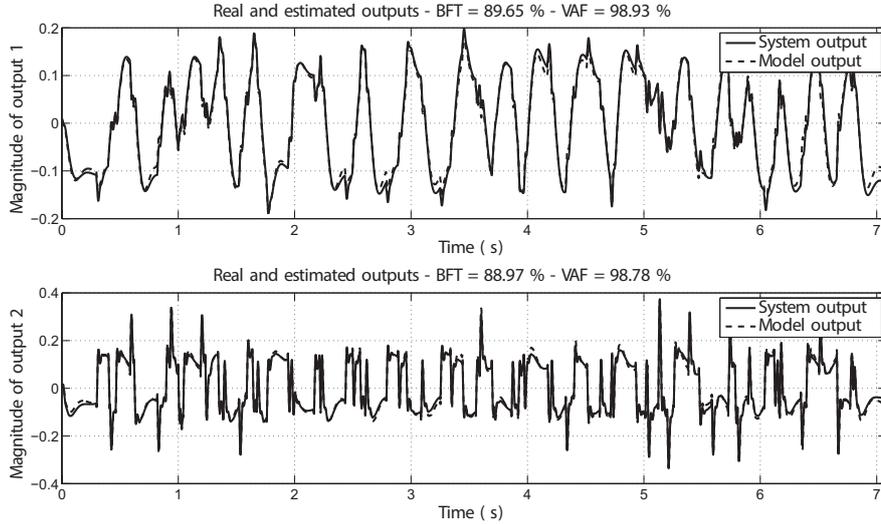


Fig. 3. Comparison of the time responses of the system (—) and the estimated local model (---) for $\theta_2 = 5\pi / 8$.

Having fit measurements in the range [67%, 98%] is, according to the Authors' experience, more than suitable. Notice also that the weakest values are obtained for $\theta_2 = \pi / 8$, *i.e.*, for a position close to singularity. This initial conclusion is enhanced by comparing the frequency responses of these estimated LTI models with the frequency responses of local analytic models. Figure 4 shows the magnitude Bode plots in the least favorable case, *i.e.*, $\theta_2 = \pi / 8$ of (i) the estimated local model (---), (ii) a local analytic model of the flexible system (—) obtained by linearizing the non-linear equations given in Subsection 3.2, (iii) a local analytic model of an equivalent rigid system (-o). These frequency plots show that the estimated local model is able to approximate the system behavior in a relatively large range of frequencies (up to $1e4 \text{ rad/s}$). Notice however that the estimated local model has difficulties to picture the gain of the system when the coupling transfers $y_1 \leftrightarrow u_2$ and $y_2 \leftrightarrow u_1$ are considered. The comparison with a rigid model (obtained from the non-linear equations given in Subsection 3.2) indicates that the estimated local models are a lot better at capturing the flexibilities of the system.

3.3.2 LPV model estimation

Once 7 average local black-box models are available, Step (4) of the procedure described in Section 2 can be carried out. More precisely, knowing the model structure ($\mathbf{A}(\boldsymbol{\theta})$, $\mathbf{B}(\boldsymbol{\theta})$, $\mathbf{C}(\boldsymbol{\theta})$)

given in Subsection 3.2 as well as a set of local fully-parameterized state-space forms ($\boldsymbol{\mathcal{A}}_i, \boldsymbol{\mathcal{B}}_i, \boldsymbol{\mathcal{C}}_i$) for $i \in \{1, \dots, 7\}$, local re-structured state-space representations ($\mathbf{A}(\boldsymbol{\theta})$, $\mathbf{B}(\boldsymbol{\theta})$, $\mathbf{C}(\boldsymbol{\theta})$) are extracted by following the steps of the H_∞ -norm-based approach described in Subsection 2.2. For each working point, a state-space form satisfying the gray-box structure defined by Eq. (32) is estimated. Because of the specific structure of \mathbf{C} (see Eq. (31) and the following discussion), this matrix is assumed to be known *a priori*. Notice that the redundancy of the parameters (up to the sign) obtained from the linearization procedure described in Subsection 3.1 (see Eq. (31)) is explicitly taken into account in the description of the matrices $\mathbf{A}(\boldsymbol{\theta})$ and $\mathbf{B}(\boldsymbol{\theta})$ in Eq. (32). This structural constraint can be easily tackled by the H_∞ -based algorithm developed in this paper. For each working point, 16 parameters are estimated with the help of $n_x(n_u + n_y) = 36$ equations. At this point, we may recall that, in order to find the desired parameter vector $\boldsymbol{\theta}_i$, $i \in \{1, \dots, 7\}$, (i) the global minimum of the involved objective function is sought, (ii) the parameters $\boldsymbol{\theta}_i$ must be unique for each value of $i \in \{1, \dots, 7\}$. The first problem, *i.e.*, the convergence towards to the global minimum of the cost function (6), can be bypassed by initializing the algorithm with a user-defined parameter vector in the vicinity of the global minimum. In this paper, it is suggested using the information available from the structure of the system in order to find such an initial point. More precisely,

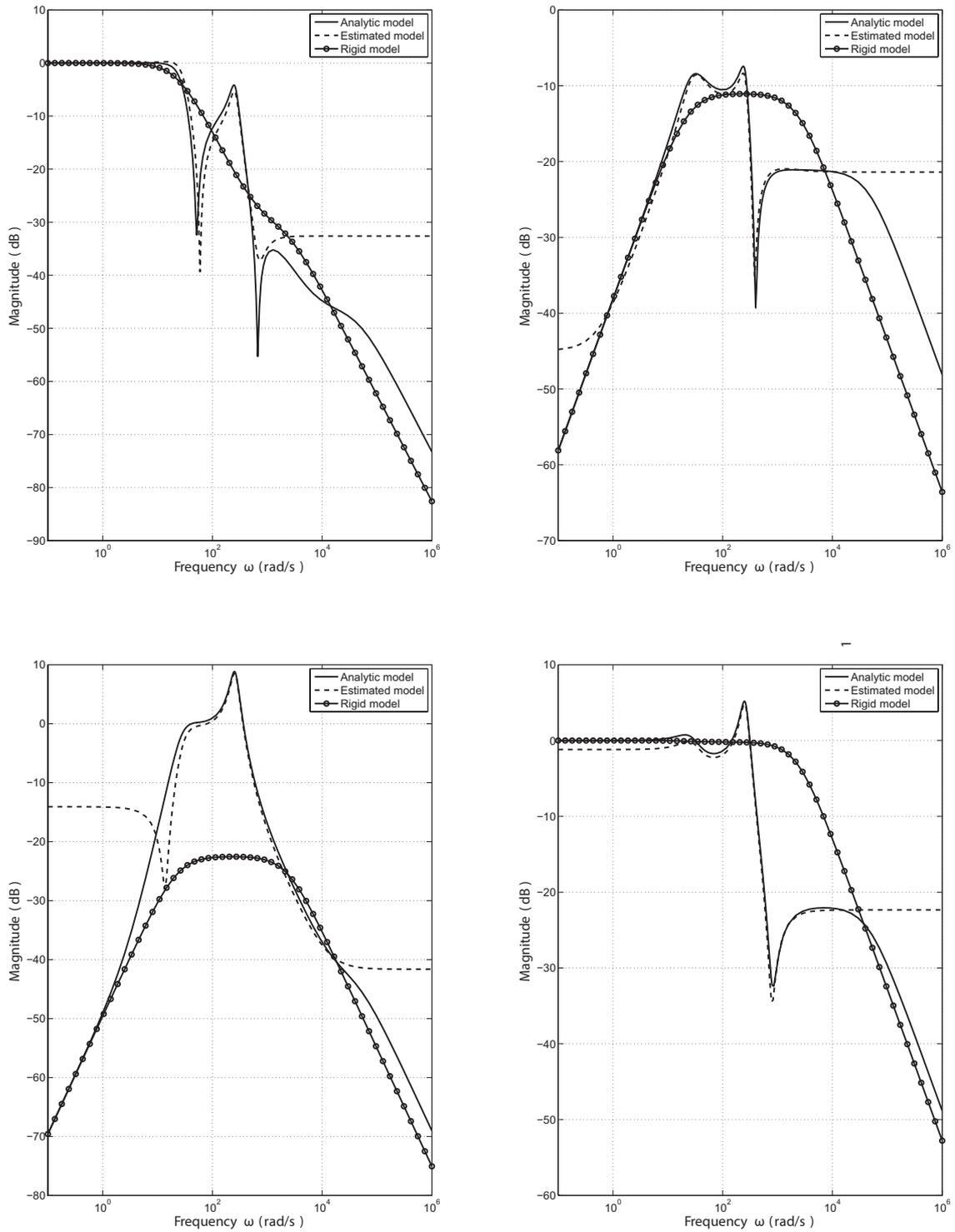


Fig. 4. Comparison of the frequency responses of the analytic model (—), a rigid model (—o) and the locally-estimated model (---) for $\theta_2 = \pi/8$.

if the system $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$ can be represented with the desired structure (32), then there is a similarity matrix \mathbf{T}_i such that

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{T}_i \mathbf{A}_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}_i, \quad (34a)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{T}_i \mathbf{B}_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad (34b)$$

$$\mathbf{C}_i = \mathbf{C} \mathbf{T}_i \quad (34c)$$

Clearly, Eq. (34a)-(34c) are linear in \mathbf{T}_i . So, this set of equations can be solved in the least squares sense, *i.e.*, by using a Moore-Penrose pseudo-inverse [15], in order to find the optimal similarity matrix \mathbf{T}_i . This set of equations must be solved in a least squares sense because, for this particular example, there are more equations than variables. Notice however that, in the general case, using a least squares minimization should be favored

Tab. 2. Performance metrics for the restructuring step.

θ_2	$\pi / 8$	$2\pi / 8$	$3\pi / 8$	$4\pi / 8$	$5\pi / 8$	$6\pi / 8$	$7\pi / 8$
$\frac{F(\mathbf{g}_{final})}{\ \mathbf{G}_{anal}(s, \theta_2)\ _{\infty}}$	4.03e-03	2.1e-03	1.48e-03	1.25e-03	1.03e-03	7.28e-04	4.79e-04

over, for instance, a standard LU decomposition, because of the measurement and numerical errors involved in the $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$ computations. Once the optimal \mathbf{T}_i matrix has been computed, and assuming that this matrix is invertible (which is the case in our study), an initial parameter vector \mathbf{g}_{i_0} can be extracted from the matrix $\mathbf{T}_i \mathbf{A}_i \mathbf{T}_i^{-1}$ and from the structure (32). The parameter vector \mathbf{g}_{i_0} can then be used to initialize the proximity control algorithm described above. As mentioned in Subsection 2.2, the second problem is related to the local identifiability of the model structure defined by Eq. (30). In this paper, the study of the identifiability of the involved model structure is carried out numerically. More precisely, starting from 10 small randomly generated perturbations of \mathbf{g}_{i_0} , the H_{∞} -norm-based optimization algorithm is run 10 times, then the final estimates are compared. Because the 10 optimizations return the same values for the parameters (up to the numerical precision), we can conclude that the model structure is identifiable, at least locally (30). Notice that the question of the sensitivity w.r.t noise is a challenging problem which will be addressed in a future study.

Remark 2. In order to increase the speed of convergence of the algorithm, a L-BFGS [20] method is applied before using the proximity control algorithm. For our numerical experiments, the open source software **libLBFGS** has been used. The computations have been carried out on a Linux Desktop computer, with an AMD FX 8-core processor running at 3.6 GHz. The total computation time (initialization + optimization for the seven working points) is lower than 90 seconds.

Table 2 gives the normed final values of the 7 H_{∞} -based cost functions used to re-structure the local state-space models, where $\mathbf{G}_{anal}(s, \theta_2)$ stands, more precisely, for the transfer function of the local analytical models. These figures prove the efficiency of the developed technique quantitatively. As far as the quality and reliability of the estimates are concerned, Fig. 5 shows the parameter evolution w.r.t. the scheduling parameter of the re-structured state-space forms. A comparison of the curves shown by Fig. 5, i.e., a comparison of the evolution of the parameters (w.r.t. $\kappa = \cos(\theta_2)$) of the estimated re-structured models and the analytic ones, illustrates the performance of the gray-box technique developed in this paper. These plots show indeed that the gray-box identification procedure suggested herein is able to give access to local models satisfying a parameter evolution (w.r.t. $\kappa = \cos(\theta_2)$) similar to the one verified by the analytic models obtained from the non-linear equations

given in Subsection 3.2. Notice that this dynamic evolution totally differs from the one satisfied by the local black-box balanced matrices $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$, $i \in \{1, \dots, 7\}$ (see Fig. 6) which was the case in [22] as well. This comparison shows that, at least locally, combining structural information with black-box estimates can lead to performance equivalent to the one reachable through a global analytic study which requires strong knowledge in robotics as well as the availability of all of the physical parameters governing the behavior of the system.

The last step of the identification procedure consists in interpolating the coefficients of the estimated local models in order to get a reliable interpolated LPV state-space model. Hereafter, the interpolation of the local re-structured models are compared with the analytic ones as well as the estimated local balanced ones as suggested in [22]. The curves in Fig. 5-6 lead us to choose a polynomial κ -dependent form, i.e.,

$$\mathbf{A}(\kappa) = \mathbf{A}_0 + \mathbf{A}_1 \kappa + \dots + \mathbf{A}_d \kappa^d \quad (35)$$

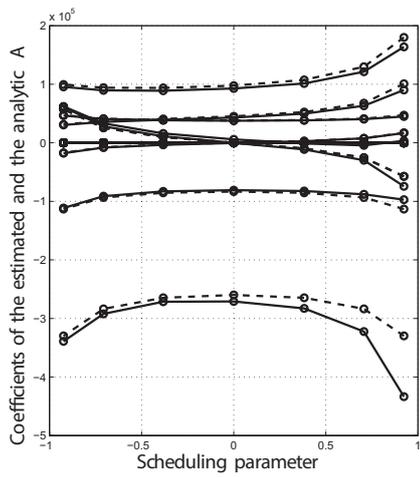
and similarly for $\mathbf{B}(\kappa)$ (and $\mathbf{C}(\kappa)$ and $\mathbf{D}(\kappa)$ for the balanced state-space forms). By using suitable regressors formed from the scheduling variable $\kappa = \cos(\theta_2)$, a least-squares algorithm can be used to estimate the matrices $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k, \mathbf{D}_k, k \in [0, d]$, and, by extension $\mathbf{A}(\kappa), \mathbf{B}(\kappa), \mathbf{C}(\kappa)$ and $\mathbf{D}(\kappa)$ for each interpolated LPV model. The plots in Fig. 5-6 show that the evolution is quite smooth and can be captured with a low-order polynomial. In order to validate the interpolation step, a second fit measurement is used

$$\text{FIT} = 100 \times \left(1 - \frac{\|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}\|}{\|\boldsymbol{\eta} - \text{mean}(\boldsymbol{\eta})\|} \right) \quad (36)$$

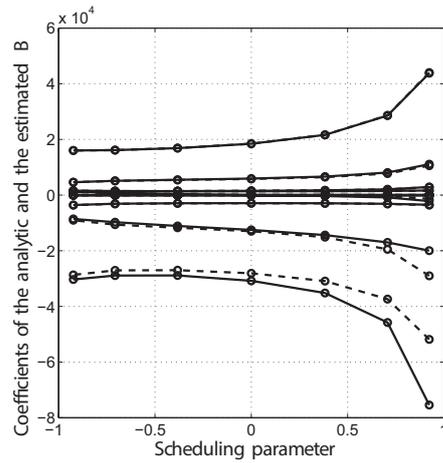
where $\boldsymbol{\eta}$ stands for a vector obtained after the vectorization of all the LTI model parameters estimated for $\theta_2 \in \{i\pi/8 : i = 1, \dots, 7\}$ and $\hat{\boldsymbol{\eta}}$ the corresponding simulated vector using Eq. (35). The fits for the estimated models with $d = 1$ and $d = 2$ are displayed in Table 3. These figures show that the proposed interpolation procedure is efficient and that a low-order polynomial LPV model, i.e., $d = 2$, is a good trade-off between complexity and efficiency.

3.3.3 LPV model validation

The final LPV model must be validated. For that, a new set of I/O data is generated. The system is more precisely excited so that the whole range of θ_2 is visited (see Fig. 7). Notice that this evolution is relatively fast because the reachable range



(a) Evolution of $\vartheta_1, \dots, \vartheta_8$ of the analytic (---o) and the re-structured models (—o).



(b) Evolution of $-\vartheta_9, \dots, -\vartheta_{16}$ of the analytic (---o) and the re-structured models (—o).

Fig. 5. Parameter evolutions obtained from the local analytical models and by re-structuring the estimated fully-parameterized state-space forms $(\mathfrak{A}_i, \mathfrak{B}_i, \mathfrak{C}_i), i \in \{1, \dots, 7\}$ w.r.t. $\kappa = \cos(\theta_2)$.

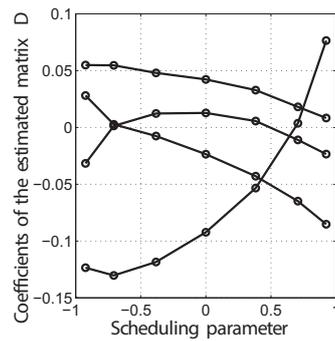
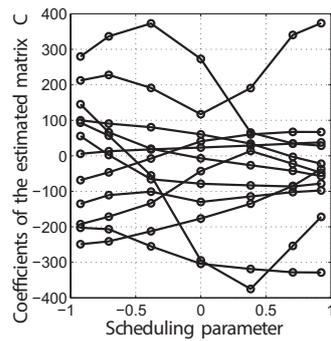
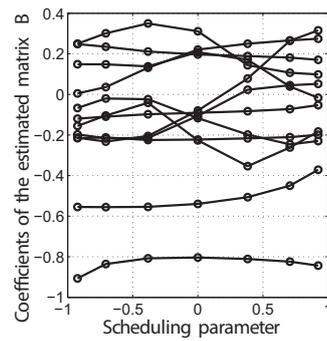
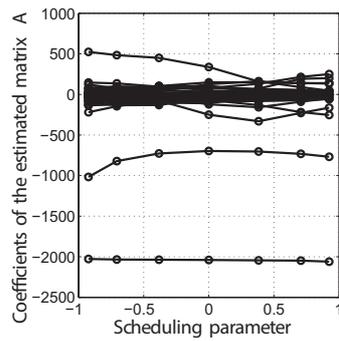


Fig. 6. Evolutions of the coefficients of the estimated local balanced LTI state-space matrices w.r.t. $\kappa = \cos(\theta_2)$.

Tab. 3. Parameter fit between the least-squares estimates and the local model parameters (re-structured (struc.), analytic (anal.), balanced (bal.)) for different values of the degree d .

	d	$A(\kappa)$	$B(\kappa)$	$C(\kappa)$	$D(\kappa)$
FIT (struc. %)	1	84.7	78.1	97.9	not det.
FIT (struc. %)	2	96.6	88.8	98.9	not det.
FIT (anal. %)	1	88.2	82.7	100	not det.
FIT (anal. %)	2	94.5	91.5	100	not det.
FIT (bal. %)	1	92	86.6	84.7	94.8
FIT (bal. %)	2	94.8	90.1	91.1	98.4

of values for θ_2 is visited in almost 3s. Once again, the time responses for each output of the different interpolated LPV models are compared with the outputs of the non-linear simulator (see Table 4 for a quantified comparison and Fig. 8-10 for specific samples). These fit measurements show that the LPV model designed from the technique suggested in this paper is able to capture the dynamic behavior of the non-linear system. Indeed, the performance of this interpolated LPV model is equivalent to the one of the LPV model obtained from the analytic study. Notice that these results are a lot better than the ones obtained from the locally-estimated balanced state-space representation, as shown by the drops in the fit measurements. This final validation is compulsory because, when we look at Fig. 6, the parameters of the local balanced models satisfy a smooth variation with respect to the scheduling variable which can be efficiently captured by a second order polynomial function (see the last two lines of Table 3). However, as shown in Fig. 10, the corresponding final interpolated model fails to

Tab. 4. Performance metrics (BFT and VAF) for the experimental LPV models (from the re-structured (struc.), analytic (anal.), balanced (bal.) local models) (with $d = 2$) on validation data.

	BFT (%)	VAF (%)
y_1 (struc.)	76.2	94.4
y_2 (struc.)	83	97.1
y_1 (anal.)	78.6	95.5
y_2 (anal.)	88.4	98.7
y_1 (bal.)	52.7	77.7
y_2 (bal.)	79.7	95.9

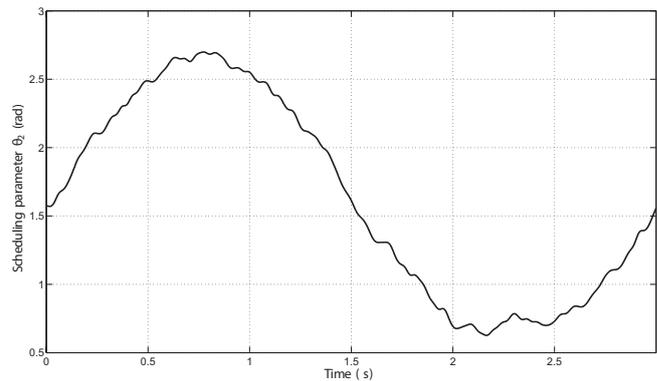


Fig. 7. Evolution of θ_2 during the LPV models validation.

picture the behavior of the non-linear model suitably. These results prove that adding up prior information (through the knowledge of the LPV model structure) and using locally re-structured state-space models is an efficient solution to result in an accurate LPV model when local data sets are available.

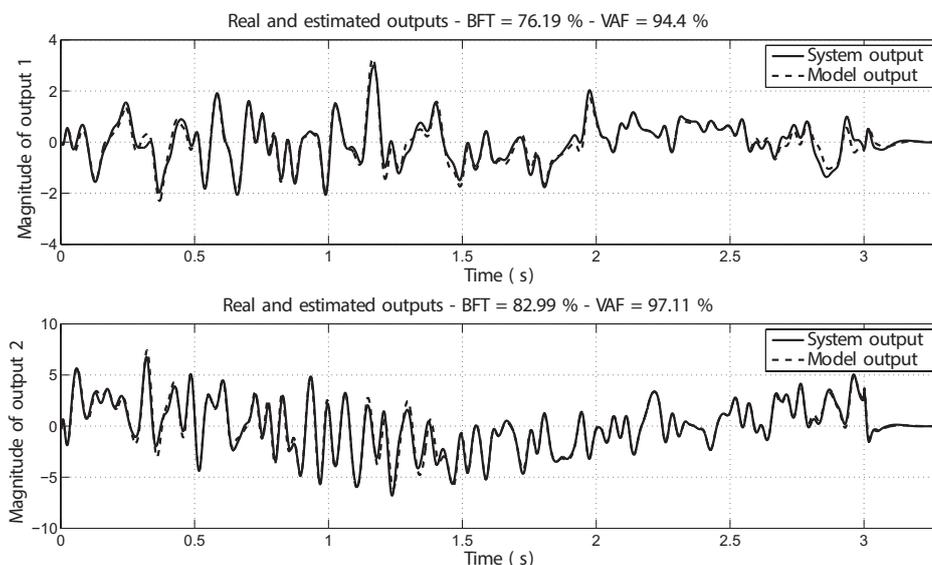


Fig. 8. Comparison of the time responses of the system (—) and the interpolated LPV model from the locally-estimated re-structured state-space forms (- -).

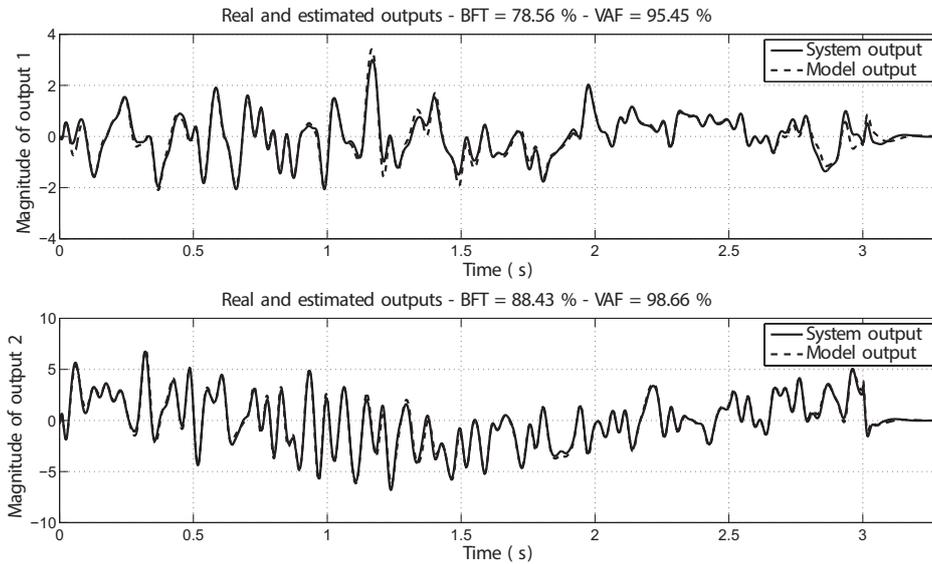


Fig. 9. Comparison of the time responses of the system (—) and the interpolated LPV model from the local analytic state-space forms (- -).

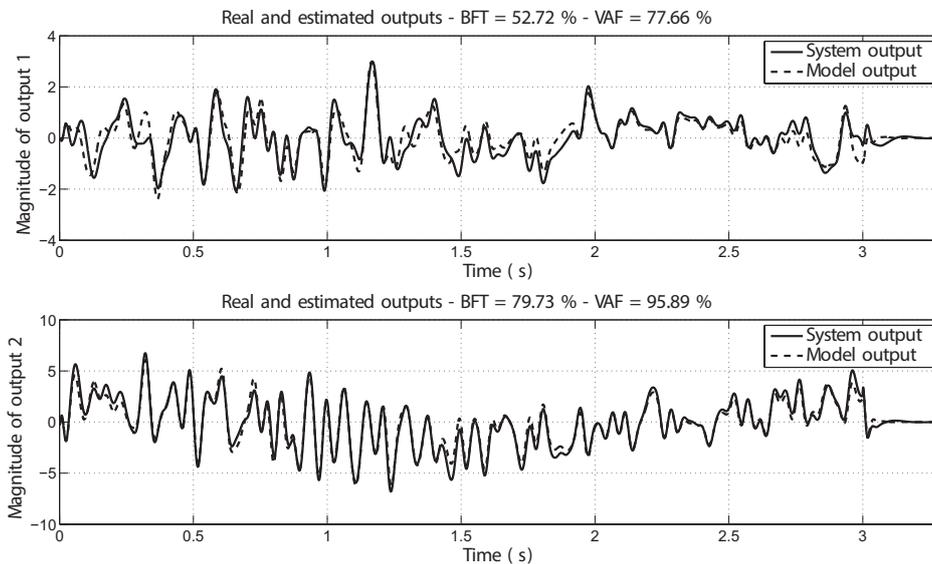


Fig. 10. Comparison of the time responses of the system (—) and the interpolated LPV model from the locally-estimated balanced state-space forms (- -).

4 Conclusions

In this paper, problems related to the local approach for LPV identification are illustrated and significant solutions are introduced. More specifically, when the variations of the scheduling variables are fast w.r.t. to the dynamics of the system, the local approach may lead to unreliable LPV models when no prior information of the system is available. For instance, despite its numerical robustness, the technique suggested in [22] may give access to interpolated LPV models unable to capture the dynamics of the system as illustrated in Section 3 of this paper. In order to bypass these difficulties and to use the local approach efficiently, the introduction of prior information, more precisely the structure of the LPV model, seems to be a

good solution. More particularly, the problem of recovering the numerical values of the parameters of a structured LPV representation from fully-parameterized state-space forms identified locally has been addressed. To reach this goal, a technique based on a specific H_∞ -model matching criterion has been developed. The benefits of this new multi-step identification technique have been illustrated through the identification of a non-linear flexible robot. As a conclusion, combining a technique able to re-structure local fully-parameterized state-space forms with the basic idea of the local approach can be seen as an efficient and promising solution to ensure the coherence of the coordinate bases used for each local model.

References

- 1 **Apkarian P., Noll D.** *Nonsmooth H_∞ synthesis*. IEEE Transactions on Automatic Control, 51. pp. 71–86. (2006)
- 2 **Apkarian P., Noll D., Prot O.** *A trust region spectral bundle method for nonconvex eigenvalue optimization*. SIAM Journal of Optimization. 19. pp. 281–306. (2008)
DOI: [10.1137/060665191](https://doi.org/10.1137/060665191)
- 3 **Apkarian P., Noll D., Prot O.** *A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions*. Journal of Convex Analysis. 16. pp. 641–666. (2009)
- 4 **Arzelier D., Deaconu G., Gumussoy S., Henrion D.** *H_2 for HIFOO*. In: Proceedings of the International Conference on Control and Optimization with Industrial Applications. Ankara, Turkey, August (2011)
- 5 **Bonnans J., Gilbert J., Lemaréchal C., Sagastizábal C.** *Numerical optimization*. Springer-Verlag. (2006)
DOI: [10.1007/978-3-540-35447-5](https://doi.org/10.1007/978-3-540-35447-5)
- 6 **Casella F., Lovera M.** *LPV/LFT modelling and identification: overview, synergies and a case study*. In: Proceedings of the IEEE Multi-Conference on Systems and Control. San Antonio, Texas, USA, September (2008)
DOI: [10.1109/cacsd.2008.4627358](https://doi.org/10.1109/cacsd.2008.4627358)
- 7 **Craig J.** *Introduction to robotics: mechanics and control*. Pearson/Prentice Hall. (2005)
DOI: [10.1016/0005-1098\(87\)90105-1](https://doi.org/10.1016/0005-1098(87)90105-1)
- 8 **Cuvillon L., Laroche E., Gangloff J., de Mathelin M.** *A multivariable methodology for fast visual servoing of flexible manipulators moving in a restricted workspace*. Advanced Robotics. 26. pp.1771–1797. (2012)
DOI: [10.1080/01691864.2012.685230](https://doi.org/10.1080/01691864.2012.685230)
- 9 **Dym C., Shames I.** *Solid mechanics: a variational Approach*. McGraw-Hill. (1973)
- 10 **Garnier H., Wang L.** *Identification of continuous-time models from sampled data*. London: Springer Verlag. (2008)
DOI: [10.1007/978-1-84800-161-9](https://doi.org/10.1007/978-1-84800-161-9)
- 11 **Ginhoux R., Gangloff J., de Mathelin M., Soler L., Arenas Sanchez M., Marescaux J.** *Active filtering of physiological motion in robotized surgery using predictive control*. IEEE Transactions on Robotics. 21. pp. 67–79. (2005)
DOI: [10.1109/tro.2004.833812](https://doi.org/10.1109/tro.2004.833812)
- 12 **Groot Wassink W., van de Wal M., Scherer C., Bosgra O.** *LPV control for a wafer stage: beyond the theoretical solution*. Control Engineering Practice. 13 (2). pp. 231–245. (2005)
DOI: [10.1016/j.conengprac.2004.03.008](https://doi.org/10.1016/j.conengprac.2004.03.008)
- 13 **Gu G., Khargonekar P.** *A class of algorithms for identification in H_∞* . Automatica. 28 (2). pp. 299–312. (1992)
DOI: [10.1016/0005-1098\(92\)90117-x](https://doi.org/10.1016/0005-1098(92)90117-x)
- 14 **Gumussoy S., Henrion D., Millstone M., Overton M.** *Multiobjective robust control with HIFOO 2.0*. In: Proceedings of the IFAC Symposium on Robust Control Design. Haifa, Israel, June 2009. DOI: [10.3182/20090616-3-il-2002.00025](https://doi.org/10.3182/20090616-3-il-2002.00025)
- 15 **Horn R., Johnson C.** *Matrix analysis*. Cambridge University Press. (1990)
- 16 **Khalate A., Bombois X., Tóth R., Babuska R.** *Optimal experimental design for LPV identification using a local approach*. In: Proceedings of the IFAC Symposium on System Identification, Saint-malo, France, July 2009. DOI: [10.3182/20090706-3-fr-2004.00027](https://doi.org/10.3182/20090706-3-fr-2004.00027)
- 17 **Kozłowski K.** *Modelling and identification in robotics*. Springer. (1998)
DOI: [10.1007/978-1-4471-0429-2](https://doi.org/10.1007/978-1-4471-0429-2)
- 18 **Laroche E., Cuvillon L.** *Dynamical model of a planar 2-DOF arm*. Technical report, Strasbourg University, Laboratoire des Sciences de l’Image de l’Informatique et de la Télé-detection. (2007)
- 19 **Lee L., Poolla K.** *Identification of linear parameter varying systems using non linear programming*. Journal of Dynamic Systems Measurements and Control. 121 (1). pp. 71–78. (1999)
DOI: [10.1115/1.2802444](https://doi.org/10.1115/1.2802444)
- 20 **Liu D., Nocedal J., Dong C.** *On the limited memory BFGS method for large scale optimization*. Mathematical Programming. 45 (1-3). pp. 503–528. (1989)
DOI: [10.1007/bf01589116](https://doi.org/10.1007/bf01589116)
- 21 **Ljung L.** *System identification. Theory for the user*. Prentice Hall, Upper Saddle River. (1999)
- 22 **Lovera M., Mercère G.** *Identification for gain scheduling: a balanced subspace approach*. In: Proceedings of the American Control Conference, New York, USA, (2007)
DOI: [10.1109/acc.2007.4282899](https://doi.org/10.1109/acc.2007.4282899)
- 23 **Mäkilä P., Partington J.** *Robustness in H_∞ identification*. Automatica. 36. pp. 1685–1691. (2000)
- 24 **McKelvey T.** *Identification of state space models from time and frequency data*. PhD thesis, Linköping University, Linköping, Sweden. (1995)
- 25 **Mercère G., Lovera M.** *Convergence analysis of instrumental variable recursive subspace identification algorithms*. Automatica. 43 (8). p. 1377–1386. (2007)
DOI: [10.1016/j.automatica.2007.01.014](https://doi.org/10.1016/j.automatica.2007.01.014)
- 26 **Milanese M., Taragna M.** *H_∞ set membership identification: a survey*. Automatica. 41 (12). p. 2019–2032. (2005)
DOI: [10.1016/j.automatica.2005.07.007](https://doi.org/10.1016/j.automatica.2005.07.007)
- 27 **Moore B.** *Principal component analysis in linear systems: controllability, observability and model reduction*. IEEE Transactions on Automatic Control. 26. p. 17–32. (1981)
DOI: [10.1109/tac.1981.1102568](https://doi.org/10.1109/tac.1981.1102568)
- 28 **Nemirovski, A.** *Several NP-hard problems arising in robust stability analysis*. Mathematics of Control, Signals and Systems. 6 (2). p. 99–105. (1993)
DOI: [10.1007/bf01211741](https://doi.org/10.1007/bf01211741)
- 29 **Noll D., Prot O., Rondepierre A.** *A proximity control algorithm to minimize nonsmooth and nonconvex functions*. Pacific Journal of Optimization. 4. p. 571–604. (2008)
- 30 **Parrilo P., Ljung L.** *Initialization of physical parameter estimates*. In: Proceedings of the IFAC Symposium on System Identification, Rotterdam, The Netherlands. (2003)
- 31 **Prot O., Mercère G.** *Initialization of gradient-based optimization algorithms for the identification of structured state-space models*. In: Proceedings of the IFAC World Congress, Milan, Italy. (2011)
DOI: [10.3182/20110828-6-it-1002.02879](https://doi.org/10.3182/20110828-6-it-1002.02879)
- 32 **Prot O., Mercère G., Ramos J.** *A null-space-based technique for the estimation of linear-time invariant structured state-space representations*. In: Proceedings of the IFAC Symposium on System Identification, Brussels, Belgium. (2012)
DOI: [10.3182/20120711-3-be-2027.00075](https://doi.org/10.3182/20120711-3-be-2027.00075)
- 33 **Shi P., McPhee J.** *DynaFlex Users’ Guide*. Systems Design Engineering, University of Waterloo. (2002)

- 34 **Tóth R.** *Identification and Modeling of Linear Parameter-Varying Systems*. Springer Verlag. Lecture Notes in Control and Information Sciences. 403. (2010)
- 35 **Tóth R., Felici F., Heuberger P., Van den Hof P.** *Discrete-time LPV I/O and state-space representations differences of behavior and pitfalls of interpolation*. In: Proceedings of the European Control Conference, Kos, Greece. (2007)
- 36 **Tóth R., Willems J. Heuberger P., Van den Hof P.** *The behavioral approach to linear parameter-varying systems*. IEEE Transactions on Automatic Control. 56 (11). pp. 2499–2514. (2011)
DOI: [10.1109/tac.2011.2109439](https://doi.org/10.1109/tac.2011.2109439)
- 37 **Helvoort J. van, Steinbuch M., Lambrechts P., van de Molengraft R.** *Analytical and experimental modelling for gain scheduling of a double scara robot*. In: Proceedings of the IFAC Symposium on Mechatronic Systems, Sydney, Australia. (2004)
- 38 **Verhaegen M., Verdult V.** *Filtering and system identification: a least squares approach*. Cambridge University Press. (2007)
DOI: [10.1017/cbo9780511618888](https://doi.org/10.1017/cbo9780511618888)
- 39 **Vizer D., Mercère G.** *An H_∞ -norm-based approach for operating point selection and LPV model identification from local experiments*. Periodica Polytechnica Electrical Engineering and Computer Science. 58 (3). (2014)
DOI: [10.3311/PPee.7354](https://doi.org/10.3311/PPee.7354)
- 40 **Xie L., Ljung L.** *Estimate physical parameters by black box modeling*. In: Proceedings of the Chinese Control Conference. Hangzhou, China. (2002)