Mechanical Systems and Signal Processing 23 (2009) 2395–2417

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/jnlabr/ymssp

Interpolating model identification for SISO linear parameter-varying systems

Jan De Caigny[a], Juan F. Camino[b,*], Jan Swevers[a]

[a] Department of Mechanical Engineering, Katholieke Universiteit Leuven, 300B, B-3001 Heverlee, Belgium
[b] School of Mechanical Engineering, University of Campinas - UNICAMP, P.O. Box 6122, 13083-970 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 25 February 2008
Received in revised form
15 April 2009
Accepted 29 April 2009
Available online 9 May 2009

Keywords:
Linear parameter-varying systems
System identification
State-space model interpolation

ABSTRACT

This paper presents a new method to estimate linear parameter-varying (LPV) state-space models for single-input single-output systems whose dynamics depend on one or more time-varying parameters, called scheduling parameters. The method is based on the interpolation of linear time-invariant models that are identified for fixed operating conditions of the system, that is, for constant values of the scheduling parameters. The proposed method can account for multiple scheduling parameters and yields either a polynomial or an affine LPV model that is numerically well-conditioned and therefore suitable for LPV control synthesis. The underlying interpolation technique is formulated as a nonlinear least-squares optimization problem that can be solved efficiently by standard solvers. The new interpolation method is applied to an electromechanical system that depends on two scheduling parameters. The numerical results are compared to existing techniques in the literature, demonstrating the potential and advantages of the proposed method.

1. Introduction

For more than a decade, linear parameter-varying (LPV) synthesis procedures have received a lot of attention from the control community (for instance, see [1–4]). In the LPV control framework, the scheduling parameters that govern the variation of the dynamics of the system are assumed to be unknown, but measurable in real-time [5]. There is a continuing effort to design LPV controllers that achieve higher performance while still guaranteeing stability for all possible parameter variations [6–17]. Most of these control design methods rely on the availability of an LPV model of the system that accurately describes the variation of the dynamics over the workspace. Although identification techniques and algorithms for linear time-invariant (LTI) systems based on measured input–output data are well-known and widely spread, estimation of LPV models remains a difficult problem that is still in a state of development.

Two main LPV model identification approaches exist: the *global* approaches and the *local* approaches. The global approaches are based on the assumption that it is possible to perform one global identification experiment by exciting the system while the scheduling parameters are persistently changing the system dynamics. The local approaches interpolate a set of local LTI models that are estimated based on a set of local measurements, obtained by exciting the system for different fixed operating conditions, that is, for constant values of the scheduling parameters.

Several examples of the global LPV identification approach are found in the literature. In [18], a recursive least-squares algorithm is presented to estimate a single-input single-output (SISO) LPV model with a linear fractional transform (LFT)

[*] Corresponding author. Tel.: +55 19 3521 2908; fax: +55 19 3289 3722.
E-mail address: camino@fem.unicamp.br (J.F. Camino).

0888-3270/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ymssp.2009.04.007

representation. The method is restricted to systems with one scheduling parameter and is based on the assumption that all states of the system can be measured. This, however, is not possible for most practical engineering applications, since the number of available sensors is usually limited and smaller than the order of the system, implying that full state measurement is not possible. To identify MIMO LPV models, a nonlinear programming approach is proposed in [19]. However, the resulting nonconvex optimization problem is highly sensitive to the initial starting values. For MIMO LPV systems with an affine parameter-dependency, a subspace identification method is presented in [20]. In this approach, the matrices involved in the calculations grow exponentially with the order of the system. An attempt to overcome this curse of dimensionality is proposed in [21]. A new subspace identification approach was recently suggested in [22]. This approach uses a periodic scheduling sequence for the varying parameter, but the performance of the algorithm is affected by the choice of the scheduling sequence. In [23], an identification approach is proposed that uses a discrete-time input–output representation whose coefficients have a polynomial dependency on the scheduling parameter. The paper presents least-squares techniques to obtain both time-varying and recursive autoregressive with exogenous input (ARX) models. However, compared to a state-space representation, the input–output representation is not suitable for most existing LPV control design techniques. Although it is shown in [24] that it is possible to transform the input–output representation to an equivalent state-space representation, it is not straightforward to apply this transformation and the resulting state-space representation can have a dynamic dependency on the scheduling parameter. For discrete-time systems, this means a dependency on the scheduling parameter at different time instants and for continuous-time systems a dependency on the scheduling parameter and its derivative(s). It is also shown in [24] that even if the system dynamics changes smoothly with the time-varying parameter, the coefficients of the input–output representation may change in a complicated way.

As previously stated, the basic assumption in the global approaches is the possibility of performing a global identification experiment by exciting the system while the scheduling parameters are persistently changing the system dynamics. This type of experiment, however, is not possible for some applications, as for example the flight applications in [25] and the vibroacoustic system in [26]. In case a global identification experiment is impossible, it is appropriate to use a local LPV identification approach, based on the interpolation of local LTI models that are obtained from different local experiments performed for fixed operating conditions of the system, that is, for constant values of the scheduling parameters. Local approaches have the important practical advantage that well-known LTI identification algorithms can be used to estimate the local LTI models. Subsequently, an appropriate interpolation method is applied to construct an LPV model from the local LTI models. As the estimated local LTI models can be chosen to be either continuous- or discrete-time, both continuous- and discrete-time LPV models can be obtained, which is another advantage of the local approaches.

Several papers have proposed identification schemes based on the local LPV identification approach. To properly interpolate the local LTI models, all local approaches require that the local LTI state-space models are expressed in a consistent form, that is, all local LTI models need to be defined with respect to the same state-space basis. For instance, the method proposed in [27] interpolates the system matrices of local SISO LTI state-space models represented in the controllable form. However, as indicated in [10], the well-known numerical ill-conditioning of the controllable form reduces the practical use of the method for high-order systems. To avoid this numerical issue, a frequency-domain subspace identification technique is used in [28], that exploits properties of internally balanced realizations [29,30] to provide the local LTI models whose system matrices can be appropriately interpolated. Since the local LTI models are identified using subspace techniques, this method can be used to interpolate both SISO as well as MIMO local LTI models. Although in some cases, the non-uniqueness of the balancing transformation can be a problem for the interpolation. Moreover, since the internally balanced realization is based on the controllability and observability Gramians, all local LTI models need to be controllable and observable.

In [31], a different approach was proposed to obtain well-conditioned state-space LPV models. First, the poles and zeros of the local LTI models are interpolated by polynomials and afterwards, the interpolating LPV model is obtained as the series connection of low-order LPV submodels, constructed based on these interpolating polynomials. This method has been successfully applied to several industrial applications, but has some significant drawbacks. First, the method cannot handle systems with multiple scheduling parameters. Second, the method cannot cope with the transition of a complex conjugate pole (or zero) pair to a pair of real poles (or zeros) from one local LTI model to the next.

Although useful in practice, the local approaches have disadvantages as well. The main drawback is the fact that no information about the time variation and propagation of the scheduling parameters is used in the local approaches since they are based on the interpolation of local LTI models that are estimated for fixed values of the scheduling parameters. Consequently, it is not guaranteed that the interpolating LPV model will accurately describe the system for time-varying scheduling parameters. Therefore, local approaches are more suitable for systems whose scheduling parameters vary slowly and whose dynamics exhibit a smooth variation as a function of the scheduling parameters. These are common guidelines in interpolating gain-scheduling control practice [32]. A second drawback is that in order to obtain the local LTI models, the scheduling parameters must be constant during the identification experiment. It is therefore assumed that the system can operate in different fixed operating conditions. This assumption is satisfied in many applications, as for instance, mechatronic motion systems with position-dependent dynamics [10,27,31] and vibroacoustic systems [26]. Nevertheless, in some applications, the scheduling parameter might be constantly changing, such that no local LTI models can be obtained and a global approach should be considered.

The aim of this paper is to present a new local LPV identification method that extends the work presented in [31] by alleviating its restrictions. The most important extension of the proposed method is the fact that it can handle systems that depend on multiple scheduling parameters. Furthermore, in the method presented here, the idea of interpolating the poles and zeros of the local LTI models is discarded and a different interpolation strategy is proposed. Throughout the paper, it will be shown that this new interpolation strategy has two clear advantages with respect to the method proposed in [31]: (i) it can handle the transition from a complex conjugate pair of poles (or zeros) to a pair of real poles (or zeros) from one local LTI model to the next and (ii) the nonlinear least-squares optimization problem solved to estimate the coefficients of the interpolating LPV model is less complex and can be solved efficiently using standard solvers.

The paper is organized as follows. A brief outline and flowchart of the interpolation method is presented in Section 2. Section 3 focuses on the division of the local LTI models into submodels. For the single scheduling parameter case, Section 4 formulates the interpolation of local LTI models as an optimization problem and discusses algorithms to solve them. The extension to multiple scheduling parameters is the topic of Section 5. Section 6 discusses the application of the proposed interpolation method on an electromechanical system that depends on two scheduling parameters and presents a comparison with existing local techniques from the literature. Section 7 presents the conclusions.

1.1. Notation

The set of real numbers is denoted by $\mathbb{R}$ and the set of natural numbers by $\mathbb{N}$. Matrices, vectors and scalars associated with the LPV model are denoted using standard font, e.g., A_k, while matrices, vectors and scalars associated with the local LTI models are denoted using San Serif font, e.g., A_ℓ. Throughout the paper, the following short notation:

$$H := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

is used to indicate the state-space model

$$H := \begin{cases} \delta[x] & = Ax + Bu, \\ y & = Cx + Du, \end{cases}$$

where the operator $\delta[\cdot]$ denotes the time derivative for a continuous-time model and the forward time shift for a discrete-time model. The symbol $\prod$ should be interpreted as the series connection of SISO state-space models, obtained by using the output of the first model as input of the next one, and so on. For example, the series connection of two SISO state-space models is given by

$$\prod_{i=1}^{2} \left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & D_i \end{array}\right] = \left[\begin{array}{cc|c} A_1 & 0 & B_1 \\ B_2C_1 & A_2 & B_2D_1 \\ \hline D_2C_1 & C_2 & D_2D_1 \end{array}\right].$$

2. Outline of the interpolation method

This section presents an overview of the proposed interpolation method. To simplify the discussion, only one scheduling parameter is considered in this outline and in Sections 3 and 4. Later, in Section 5, the extension of the method for systems with multiple scheduling parameters is presented. For a single scheduling parameter, the interpolating LPV model is parameterized using the following state-space representation:

$$H(\theta(t)) := \left[\begin{array}{c|c} A(\theta(t)) & B(\theta(t)) \\ \hline C(\theta(t)) & D(\theta(t)) \end{array}\right] \tag{1}$$

where $\theta(t) \in \mathbb{R}$ is the scheduling parameter. The parameter-dependent system matrices in (1) have compatible dimensions and are chosen to have the following polynomial dependency on the scheduling parameter $\theta(t)$:

$$\begin{aligned} A(\theta(t)) &= \sum_{k=0}^{N} A_k \rho_k \theta(t)^k, \quad B(\theta(t)) = \sum_{k=0}^{N} B_k \rho_k \theta(t)^k, \\ C(\theta(t)) &= \sum_{k=0}^{N} C_k \rho_k \theta(t)^k, \quad D(\theta(t)) = \sum_{k=0}^{N} D_k \rho_k \theta(t)^k, \end{aligned} \tag{2}$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times 1}$, $C_k \in \mathbb{R}^{1\times n}$, $D_k \in \mathbb{R}^{1\times 1}$ and $\rho_k \in \mathbb{R}$, for $k = 0, \ldots, N$. The polynomial degree of the LPV model (1) with the system matrices (2) is denoted by $N \in \mathbb{N}$. An immediate consequence of this polynomial parameterization is that the obtained LPV model can be used in the linear fractional transformation (LFT) framework, which has proven to be useful for LPV control design (see, for example, [1]). As will be shown in Section 4.1, two interesting subclasses of the general expression (1) can be derived: a polynomial LPV model in $\theta(t)$ and an affine LPV model in $\rho(\theta(t)) = \sum_{k=1}^{N} \rho_k \theta(t)^k$.

The purpose of the LPV model (1) is to interpolate m local LTI models

$$\mathsf{H}_\ell := \left[\begin{array}{c|c} \mathsf{A}_\ell & \mathsf{B}_\ell \\ \hline \mathsf{C}_\ell & \mathsf{D}_\ell \end{array}\right], \quad \ell = 1,\dots,m, \tag{3}$$

identified for m different values θ_ℓ of the scheduling parameter. All local LTI models are single-input single-output and are assumed to have the same number of states n, that is, $\mathsf{A}_\ell \in \mathbb{R}^{n\times n}$, $\mathsf{B}_\ell \in \mathbb{R}^{n\times 1}$, $\mathsf{C}_\ell \in \mathbb{R}^{1\times n}$ and $\mathsf{D}_\ell \in \mathbb{R}^{1\times 1}$, for all ℓ.

To properly interpolate the local LTI models (3), it is required that they are consistently defined with respect to the same state-space basis. In this paper, this is ensured by representing the local LTI models (3) as a gain multiplied with the series connection of first and second order submodels (see Section 3 for details). These local LTI submodels are then interpolated by LPV submodels by minimizing an appropriate least-squares cost function. Subsequently, the unknown matrices A_k, B_k, C_k, D_k and scalars ρ_k in (2) are computed, for $k = 0,\dots,N$, from the series connection of the obtained LPV submodels.

Before the technical details are given, a brief outline of the proposed method is presented (see the flowchart in Fig. 1). Throughout the paper, the subscript ℓ indicates the index of the local LTI model and the superscript τ indicates the index of the submodel.

Assume that m local LTI models H_ℓ, for $\ell = 1,\dots,m$, obtained for m different values θ_ℓ of the scheduling parameter are available. Then:

- *Step* 1: For each local LTI model H_ℓ, calculate the set of poles p_ℓ and the set of zeros z_ℓ. Sort the poles and zeros in each set p_ℓ and z_ℓ using a consistent ordering for $\ell = 1,\dots,m$ (Section 3.1).
- *Step* 2: Divide each local LTI model into a gain K_ℓ multiplied by the series connection of τ_1 first and τ_2 second order submodels H_ℓ^τ (Section 3.2).
- *Step* 3: For each local LTI model, calculate the gain K_ℓ associated with the zero-pole-gain representation (Appendix A).
- *Step* 4: Select the subclass of interpolating LPV model: either polynomial in $\theta(t)$ or affine in $\rho(\theta(t))$ (Section 4.1).
- *Step* 5: Formulate and solve the appropriate optimization problem to obtain the LPV submodels $H^\tau(\theta(t))$ and the parameter-dependent gain $K(\theta(t))$ that interpolate the local LTI submodels and gains (Section 4.2).
- *Step* 6: Construct the interpolating LPV model $H(\theta(t))$ as the gain $K(\theta(t))$ multiplied by the series connection of the LPV submodels (Section 4.3).

Steps 1–3 are discussed in detail in the next section.

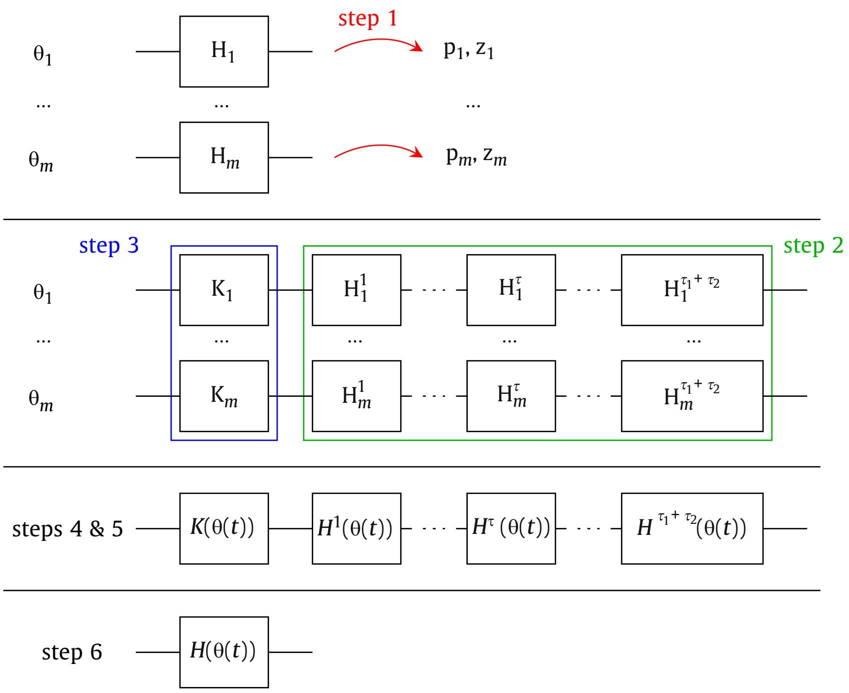

Fig. 1. Flowchart of the interpolation method.

3. Division of models in submodels

This section explains how to divide the local LTI models in a gain multiplied by the series connection of first and second order LTI submodels. In Step 1, the set of poles p_ℓ and set of zeros z_ℓ are calculated and sorted for each local LTI model H_ℓ. In Step 2, the sorted poles and zeros are used to define the local LTI submodels and in Step 3 the local gains K_ℓ are calculated.

3.1. Calculating and sorting the poles and zeros (Step 1)

In Step 1, the aim is to calculate and sort the poles and zeros of the m local LTI models H_ℓ. The set of poles p_ℓ of each local LTI model is readily obtained as the set of n eigenvalues of $\mathsf{A}_\ell \in \mathbb{R}^{n\times n}$. The set of zeros z_ℓ is computed by solving a generalized eigenvalue problem that involves A_ℓ, B_ℓ, C_ℓ and D_ℓ (see [33] for details). Since all local LTI models have the same order n, they all have the same number n of poles $\mathsf{p}_{\ell,i} \in \mathsf{p}_\ell$, for $i = 1, \ldots, n$. Furthermore, it is assumed that all local LTI models H_ℓ have the same number n_z of zeros $\mathsf{z}_{\ell,i} \in \mathsf{z}_\ell$, for $i = 1, \ldots, n_z$. Since a common guideline in gain-scheduling applications [32] is that the variation of the system dynamics should be smooth, the restriction posed by the assumption of an equal number n_z of zeros for all local LTI models is not too severe.

Once the poles and zeros have been computed for all local LTI models, they can be sorted. The sorting procedure is necessary to assure that for all local LTI models H_ℓ, the set of poles p_ℓ and the set of zeros z_ℓ are arranged in the same order. This consistent ordering is necessary in the proposed interpolation method to consistently define the LTI submodels in Step 2. Unfortunately, no general procedure exists that guarantees to yield the right sorting of the poles and zeros of the local LTI models for all applications. However, for most applications, the sorting can be easily performed based on a plot of the poles p_ℓ and the zeros z_ℓ in the complex plane as a function of the scheduling parameters. In Section 6, this plot will be shown for an electromechanical system. After the poles and zeros are sorted, the local LTI submodels can be constructed.

3.2. Constructing the local LTI submodels (Steps 2 and 3)

In this section, the original local LTI models H_ℓ are represented by a gain K_ℓ multiplied by the series connection of τ_1 first order and τ_2 second order submodels, that is, H_ℓ is represented as

$$\mathsf{H}_\ell := \mathsf{K}_\ell \prod_{\tau=1}^{\tau_1+\tau_2} \mathsf{H}_\ell^\tau, \tag{4}$$

with

$$\mathsf{H}_\ell^\tau := \left[\begin{array}{c|c} \mathsf{A}_\ell^\tau & \mathsf{B}_\ell^\tau \\ \hline \mathsf{C}_\ell^\tau & \mathsf{D}_\ell^\tau \end{array}\right], \tag{5}$$

for $\tau = 1, \ldots, \tau_1 + \tau_2$. The scalar K_ℓ is the gain, the integer τ_1 is the number of first order submodels associated with a real pole and the integer τ_2 is the number of second order submodels associated with a pair of complex conjugate poles or with a pair of two real poles. All local LTI models H_ℓ are forced to have the same number τ_1 of first order submodels and the same number τ_2 of second order submodels. Fig. 2 shows this representation for all local LTI models H_ℓ. The blue dashed rectangle indicates the m local gains K_ℓ while each of the $\tau_1 + \tau_2$ green solid rectangles indicates a set of m consistent local LTI submodels H_ℓ^τ, for $\ell = 1, \ldots, m$.

Since a first order submodel has one pole and can have either no zeros or one zero, two types of first order submodels are therefore defined. Likewise, a second order submodel has two poles and can have no, one or two zeros, and consequently three types of second order submodels are defined. Table 1 shows the five different types T_τ of submodels.

Different ways exist to divide an LTI model in first and second order submodels. Here, the number τ_1 of first order and the number τ_2 of second order submodels are determined as follows. If the model order n is even, the number of second order submodels $\tau_2 = n/2$ and there are no first order submodels ($\tau_1 = 0$). If the model order n is odd, $\tau_2 = (n-1)/2$ and $\tau_1 = 1$.

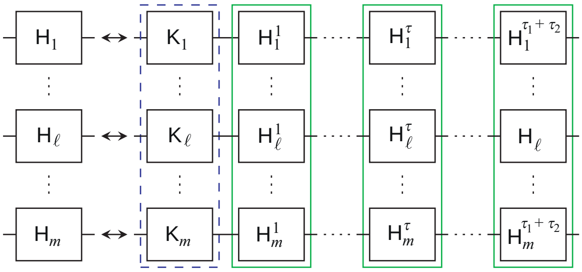

Fig. 2. Division of the local SISO models in consistent submodels.

Table 1
Types T_τ of submodels depending on the number of poles and of zeros.

Order	First		Second		
Type T_τ	1	2	3	4	5
number of poles	1	1	2	2	2
number of zeros	0	1	0	1	2

The local LTI submodels are constructed by consistently assigning the sorted poles p_ℓ and zeros z_ℓ to the $\tau_1+\tau_2$ local LTI submodels for all local LTI models H_ℓ. A pole $\mathsf{p}_{\ell,i}$ from the set of poles p_ℓ assigned to submodel τ of local LTI model H_ℓ is denoted by $\mathsf{p}^\tau_{\ell,i}$ (idem for the zeros). To assure that the local LTI submodels are consistently defined, they are all represented in the observable form. For example, a type 5 submodel which has two poles $\mathsf{p}^\tau_{\ell,1}$, $\mathsf{p}^\tau_{\ell,2}$ and two zeros $\mathsf{z}^\tau_{\ell,1}$, $\mathsf{z}^\tau_{\ell,2}$ is represented by

$$\mathsf{H}^\tau_\ell := \left[\begin{array}{cc|c} 0 & \mathsf{a}^\tau_{\ell,1} & \mathsf{b}^\tau_{\ell,1} \\ 1 & \mathsf{a}^\tau_{\ell,2} & \mathsf{b}^\tau_{\ell,2} \\ \hline 0 & 1 & 1 \end{array}\right], \tag{6}$$

where the entries of the system matrices of this observable form can be readily computed as

$$\begin{aligned} \mathsf{a}^\tau_{\ell,1} &= -\mathsf{p}^\tau_{\ell,1}\mathsf{p}^\tau_{\ell,2}, \quad & \mathsf{b}^\tau_{\ell,1} &= \mathsf{z}^\tau_{\ell,1}\mathsf{z}^\tau_{\ell,2} - \mathsf{p}^\tau_{\ell,1}\mathsf{p}^\tau_{\ell,2}, \\ \mathsf{a}^\tau_{\ell,2} &= \mathsf{p}^\tau_{\ell,1} + \mathsf{p}^\tau_{\ell,2}, \quad & \mathsf{b}^\tau_{\ell,2} &= -(\mathsf{z}^\tau_{\ell,1} + \mathsf{z}^\tau_{\ell,2}) + (\mathsf{p}^\tau_{\ell,1} + \mathsf{p}^\tau_{\ell,2}). \end{aligned}$$

For the other types of local LTI submodels, the observable form is analogously derived. In this observable form, the entries of the system matrices C^τ_ℓ and D^τ_ℓ are independent of the poles p^τ_ℓ and zeros z^τ_ℓ of the submodel, as (6) clearly shows for the type 5 submodel. Consequently, only the entries of the matrices A^τ_ℓ and B^τ_ℓ need to be interpolated. Representing the local LTI submodels in the observable form does not lead to ill-conditioning since the submodels are of first and second order only. Recall from (4) that the gain of the local LTI models is provided by K_ℓ, such that all local LTI submodels (5) have unit gain.

Remark 1. Throughout the paper, the term "gain" denotes the gain associated with the zero-pole-gain factorization of the model and not the DC-gain. Appendix A describes how the gain K_ℓ can be obtained without using the numerically ill-conditioned transfer function representation.

3.3. *Parameterization of the LPV submodels*

Once the division of the local LTI models into submodels is performed, these local LTI submodels can be interpolated using appropriate LPV submodels. Therefore, the desired LPV model is parameterized using a similar decomposition in a parameter-dependent gain $K(\theta(t))$ multiplied by the series connection of $\tau_1+\tau_2$ LPV submodels, that is, the interpolating LPV model is parameterized as

$$H(\theta(t)) := K(\theta(t)) \prod_{\tau=1}^{\tau_1+\tau_2} H^\tau(\theta(t)), \tag{7}$$

with

$$H^\tau(\theta(t)) := \left[\begin{array}{c|c} A^\tau(\theta(t)) & B^\tau(\theta(t)) \\ \hline C^\tau & D^\tau \end{array}\right] = \left[\begin{array}{c|c} \sum_{k=0}^{N} A^\tau_k \rho_k \theta(t)^k & \sum_{k=0}^{N} B^\tau_k \rho_k \theta(t)^k \\ \hline C^\tau & D^\tau \end{array}\right], \tag{8}$$

for $\tau = 1, \ldots, \tau_1+\tau_2$. Recall that only the entries of the system matrices A^τ_ℓ and B^τ_ℓ of the local LTI submodels need to be interpolated and therefore matrices C^τ and D^τ of the LPV submodels are independent of $\theta(t)$. Similar to the local LTI submodels (Table 1), five types of LPV submodels can be defined. Each LPV submodel is parameterized such that it resembles the form of the local LTI submodels it needs to interpolate. For example, an LPV submodel of type 5 has the following form:

$$H^{\tau}(\theta(t)) := \left[\begin{array}{cc|c} 0 & \sum_{k=0}^{N} a_{1,k}^{\tau}\rho_k\theta(t)^k & \sum_{k=0}^{N} b_{1,k}^{\tau}\rho_k\theta(t)^k \\ 1 & \sum_{k=0}^{N} a_{2,k}^{\tau}\rho_k\theta(t)^k & \sum_{k=0}^{N} b_{2,k}^{\tau}\rho_k\theta(t)^k \\ \hline 0 & 1 & 1 \end{array}\right], \tag{9}$$

where $a_{i,k}^{\tau}$ (resp. $b_{i,k}^{\tau}$) denotes the coefficient i of matrix A_k^{τ} (resp. matrix B_k^{τ}) of the LPV submodel $H^{\tau}(\theta(t))$. Note that the above parameterization of the LPV submodels (8) is still in the general form, since the subclass (polynomial in $\theta(t)$ or affine in $\rho(\theta(t))$) of interpolating LPV model has not yet been defined. This is the subject of the next section.

Remark 2. The fact that the output equation in (8) is independent of the scheduling parameter has an important advantage. It ensures that the LPV model obtained as the series connection of the LPV submodels has the same polynomial order N as the LPV submodels. In addition, once the series connection is obtained, the output equation can be multiplied by the parameter-dependent gain $K(\theta(t))$ without changing the polynomial order N. For instance, an LPV model given by a gain multiplied by the series connection of two LPV submodels is represented as

$$H(\theta(t)) = K(\theta(t)) \prod_{\tau=1}^{2} H^{\tau}(\theta(t)) = \left[\begin{array}{cc|c} A^1(\theta(t)) & 0 & B^1(\theta) \\ B^2(\theta(t))C^1 & A^2(\theta(t)) & B^2(\theta(t))D^1 \\ \hline K(\theta(t))D^2C^1 & K(\theta(t))C^2 & K(\theta(t))D^2D^1 \end{array}\right]. \tag{10}$$

4. Interpolation and optimization

This section presents the interpolation scheme and the related optimization problem that needs to be solved to find the unknown parameters A_k, B_k, C_k, D_k and ρ_k, for $k = 0,\dots,N$, of the LPV model (1). The goal is to interpolate the local LTI submodels using LPV submodels of the corresponding type, that is, each set of m local LTI submodels H_{ℓ}^{τ} for $\ell = 1,\dots,m$ (indicated by the green solid rectangles in Fig. 2), of a given type T_{τ}, will be interpolated by an LPV submodel $H^{\tau}(\theta(t))$ of type T_{τ}. To formulate the optimization problem, an appropriate cost function needs to be defined. Depending on the choice of the subclass of the interpolating LPV model, this cost function will lead to different optimization problems.

4.1. Subclasses of interpolating models (Step 4)

As briefly stated in the introduction, the general LPV model (1) is polynomial in the scheduling parameter $\theta(t)$ and can be particularized into the following two subclasses:

(1) By fixing $\rho_k = 1$ for $k = 0,\dots,N$, the LPV model $H(\theta(t))$ becomes polynomial in $\theta(t)$:

$$H(\theta(t)) := \left[\begin{array}{c|c} \sum_{k=0}^{N} A_k\theta(t)^k & \sum_{k=0}^{N} B_k\theta(t)^k \\ \hline \sum_{k=0}^{N} C_k\theta(t)^k & \sum_{k=0}^{N} D_k\theta(t)^k \end{array}\right]. \tag{11}$$

(2) By fixing $\rho_0 = 1$ and $A_k = A_1$, $B_k = B_1$, $C_k = C_1$ and $D_k = D_1$ for $k = 2,\dots,N$, the LPV model $H(\theta(t))$ becomes affine in $\rho(\theta(t))$:

$$H(\theta(t)) := \left[\begin{array}{c|c} A_0 + A_1\rho(\theta(t)) & B_0 + B_1\rho(\theta(t)) \\ \hline C_0 + C_1\rho(\theta(t)) & D_0 + D_1\rho(\theta(t)) \end{array}\right], \quad \text{with } \rho(\theta(t)) = \sum_{k=1}^{N} \rho_k\theta(t)^k. \tag{12}$$

Which subclass of LPV model to choose depends on the application at hand. From an interpolation point of view, the first subclass (11) offers more freedom and consequently allows better interpolation of the local LTI models. On the other hand, the second subclass (12) with affine dependency on $\rho(\theta(t))$ is appealing for LPV control synthesis since many control design strategies exist for affine LPV models. Moreover, when the scheduling parameter is bounded, that is, $\underline{\theta} \leq \theta(t) \leq \overline{\theta}$, the affine LPV model (12) can be easily converted in a polytopic LPV model $G(\alpha(\theta(t)))$ depending on the scheduling parameter $\alpha(\theta(t)) = [\alpha_1(\theta(t))\ \ \alpha_2(\theta(t))]'$:

$$G(\alpha(\theta(t))) := \left[\begin{array}{c|c} (A_0 + A_1\overline{\rho})\alpha_1(\theta(t)) + (A_0 + A_1\underline{\rho})\alpha_2(\theta(t)) & (B_0 + B_1\overline{\rho})\alpha_1(\theta(t)) + (B_0 + B_1\underline{\rho})\alpha_2(\theta(t)) \\ \hline (C_0 + C_1\overline{\rho})\alpha_1(\theta(t)) + (C_0 + C_1\underline{\rho})\alpha_2(\theta(t)) & (D_0 + D_1\overline{\rho})\alpha_1(\theta(t)) + (D_0 + D_1\underline{\rho})\alpha_2(\theta(t)) \end{array}\right]$$

where

$$\alpha_1(\theta(t)) = \frac{\rho(\theta(t)) - \underline{\rho}}{\overline{\rho} - \underline{\rho}} \quad \text{and} \quad \alpha_2(\theta(t)) = 1 - \alpha_1(\theta(t)),$$

with

$$\overline{\rho} = \max_{\underline{\theta} \le \theta(t) \le \overline{\theta}} \rho(\theta(t)) \quad \text{and} \quad \underline{\rho} = \min_{\underline{\theta} \le \theta(t) \le \overline{\theta}} \rho(\theta(t)).$$

By construction, it is clear that $0 \le \alpha_i(\theta(t)) \le 1$, for $i = 1, 2$. From an LPV control synthesis point of view, polytopic LPV models are of particular interest as can be seen from the vast amount of publications in the area (see [15,16] and the references therein).

As shown in the next section, the optimization problems arising from the polynomial and affine subclass are significantly different.

4.2. Optimization approach (Step 5)

This section presents the details of the optimization problem and the algorithms used to solve them. First, an appropriate cost function needs to be formulated. Naturally, this cost function depends on the number and the types of local LTI submodels that need to be interpolated. For instance, the cost function to interpolate a set of m local LTI submodels of type 5, given by (6), by an LPV submodel of type 5, given by (9), is chosen to be

$$E_\tau(a^\tau, b^\tau, \rho) = \sum_{\ell=1}^{m} \sum_{i=1}^{2} \left| \mathsf{a}_{\ell,i}^{\tau} - \sum_{k=0}^{N} a_{i,k}^{\tau} \rho_k \theta_\ell^k \right|^2 + \left| \mathsf{b}_{\ell,i}^{\tau} - \sum_{k=0}^{N} b_{i,k}^{\tau} \rho_k \theta_\ell^k \right|^2,$$

where $a^\tau = \{a_{i,k}^\tau\}$, $b^\tau = \{b_{i,k}^\tau\}$ for $i = 1, 2$, and $k = 0, \ldots, N$, and $\rho = \{\rho_k\}$ for $k = 0, \ldots, N$. In an analogous way, the cost functions related to the other types of submodels can be constructed. The cost function to fit the interpolating gain, parameterized by

$$K(\theta(t)) = \sum_{k=0}^{N} g_k \rho_k \theta(t)^k,$$

to the local gains K_ℓ is defined as

$$E_K(g, \rho) = \sum_{\ell=1}^{m} \left| \mathsf{K}_\ell - \sum_{k=0}^{N} g_k \rho_k \theta_\ell^k \right|^2,$$

with $g = \{g_k\}$ for $k = 0, \ldots, N$. The total cost function can now be formulated as

$$E(a, b, g, \rho) = E_K(g, \rho) + \sum_{\tau=1}^{\tau_1+\tau_2} E_\tau(a^\tau, b^\tau, \rho),$$

with $a = \{a^\tau\}$ and $b = \{b^\tau\}$ for $\tau = 1, \ldots, \tau_1 + \tau_2$. This cost function can be conveniently rewritten as

$$E(a, b, g, \rho) = \|F(q)\|_2^2 = F(q)'F(q),$$

with $F(q)$ the vector-valued function

$$F(q) = \left[\begin{array}{c} \mathsf{K}_1 - \left(g_0\rho_0 + g_1\rho_1\theta_1 + \cdots + g_N\rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{K}_m - \left(g_0\rho_0 + g_1\rho_1\theta_m + \cdots + g_N\rho_N\theta_m^N\right) \\ \hline \mathsf{a}_{1,1}^1 - \left(a_{1,0}^1\ \rho_0 + a_{1,1}^1\ \rho_1\theta_1 + \cdots + a_{1,N}^1\ \rho_N\theta_1^N\right) \\ \mathsf{b}_{1,1}^1 - \left(b_{1,0}^1\ \rho_0 + b_{1,1}^1\ \rho_1\theta_1 + \cdots + b_{1,N}^1\ \rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{a}_{\ell,i}^\tau - \left(a_{i,0}^\tau\ \rho_0 + a_{i,1}^\tau\ \rho_1\theta_\ell + \cdots + a_{i,N}^\tau\ \rho_N\theta_\ell^N\right) \\ \mathsf{b}_{\ell,i}^\tau - \left(b_{i,0}^\tau\ \rho_0 + b_{i,1}^\tau\ \rho_1\theta_\ell + \cdots + b_{i,N}^\tau\ \rho_N\theta_\ell^N\right) \\ \vdots \\ \mathsf{a}_{m,2}^{\tau_1+\tau_2} - \left(a_{2,0}^{\tau_1+\tau_2}\ \rho_0 + a_{2,1}^{\tau_1+\tau_2}\ \rho_1\theta_m + \cdots + a_{2,N}^{\tau_1+\tau_2}\ \rho_N\theta_m^N\right) \\ \mathsf{b}_{m,2}^{\tau_1+\tau_2} - \left(b_{2,0}^{\tau_1+\tau_2}\ \rho_0 + b_{2,1}^{\tau_1+\tau_2}\ \rho_1\theta_m + \cdots + b_{2,N}^{\tau_1+\tau_2}\ \rho_N\theta_m^N\right) \end{array} \right], \tag{13}$$

in the vector of variables

$$q = \left[\begin{array}{ccc|ccc|ccccccc} \rho_0 & \cdots & \rho_N & g_0 & \cdots & g_N & a^1_{1,0} & b^1_{1,0} & \cdots & a^\tau_{i,k} & b^\tau_{i,k} & \cdots & a^{\tau_1+\tau_2}_{2,N} & b^{\tau_1+\tau_2}_{2,N} \end{array}\right]',$$

for $\tau = 1, \ldots, \tau_1 + \tau_2, k = 0, \ldots, N$, and $i = 1, 2$.

Depending on the choice of the subclass (polynomial in $\theta(t)$ or affine in $\rho(\theta(t))$) of the interpolating LPV model presented in Section 4.1, the cost function $E(a, b, g, \rho)$ can be particularized, leading to either a linear or a nonlinear least-squares optimization problem. These two optimization problems and the algorithms to solve them are presented next.

4.2.1. Polynomial dependency on $\theta(t)$

In this case, the parameters $\rho_k = 1$ are fixed for $k = 0, \ldots, N$, and the vector-valued function (13) simplifies to

$$F_P(q) = \left[\begin{array}{c} \mathsf{K}_1 - \left(g_0 + g_1\theta_1 + \cdots + g_N\theta_1^N\right) \\ \vdots \\ \mathsf{K}_m - \left(g_0 + g_1\theta_m + \cdots + g_N\theta_m^N\right) \\ \hline \mathsf{a}^1_{1,1} - \left(a^1_{1,0} + a^1_{1,1}\theta_1 + \cdots + a^1_{1,N}\theta_1^N\right) \\ \mathsf{b}^1_{1,1} - \left(b^1_{1,0} + b^1_{1,1}\theta_1 + \cdots + b^1_{1,N}\theta_1^N\right) \\ \vdots \\ \mathsf{a}^\tau_{\ell,i} - \left(a^\tau_{i,0} + a^\tau_{i,1}\theta_\ell + \cdots + a^\tau_{i,N}\theta_\ell^N\right) \\ \mathsf{b}^\tau_{\ell,i} - \left(b^\tau_{i,0} + b^\tau_{i,1}\theta_\ell + \cdots + b^\tau_{i,N}\theta_\ell^N\right) \\ \vdots \\ \mathsf{a}^{\tau_1+\tau_2}_{m,2} - \left(a^{\tau_1+\tau_2}_{2,0} + a^{\tau_1+\tau_2}_{2,1}\theta_m + \cdots + a^{\tau_1+\tau_2}_{2,N}\theta_m^N\right) \\ \mathsf{b}^{\tau_1+\tau_2}_{m,2} - \left(b^{\tau_1+\tau_2}_{2,0} + b^{\tau_1+\tau_2}_{2,1}\theta_m + \cdots + b^{\tau_1+\tau_2}_{2,N}\theta_m^N\right) \end{array}\right],$$

which is clearly linear in the reduced vector of variables

$$q = [g_0 \ \cdots \ g_N \ | \ a^1_{1,0} \ b^1_{1,0} \ \cdots \ a^\tau_{i,k} \ b^\tau_{i,k} \ \cdots \ a^{\tau_1+\tau_2}_{2,N} \ b^{\tau_1+\tau_2}_{2,N}]'.$$

This function can be written as

$$F_P(q) = \mathbf{F}q - \mathbf{b}, \tag{14}$$

with the matrix $\mathbf{F}$ and the vector $\mathbf{b}$ promptly determined from $F_P(q)$. With this cost function, the optimization problem

$$\min_{a,b,g} E(a, b, g) = \min_q \|\mathbf{F}q - \mathbf{b}\|_2^2$$

is a standard linear least-squares problem that has the well-known minimum-length solution $q = \mathbf{F}^\dagger\mathbf{b}$, where $\mathbf{F}^\dagger$ is the pseudo-inverse of $\mathbf{F}$.

4.2.2. Affine dependency on $\rho(\theta(t))$

In the case of affine dependency on $\rho(\theta(t))$, ρ_0 is fixed to be $\rho_0 = 1$ and the parameters $a^\tau_{i,k} = a^\tau_{i,1}$ and $b^\tau_{i,k} = b^\tau_{i,1}$ are fixed for $k = 2, \ldots, N$. Consequently, the vector-valued function (13) simplifies to

$$F_A(q) = \left[\begin{array}{c} \mathsf{K}_1 - g_0 - g_1\left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{K}_m - g_0 - g_1\left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \\ \hline \mathsf{a}^1_{1,1} - a^1_{1,0} - a^1_{1,1}\left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \mathsf{b}^1_{1,1} - b^1_{1,0} - b^1_{1,1}\left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{a}^\tau_{\ell,i} - a^\tau_{i,0} - a^\tau_{i,1}\left(\rho_1\theta_\ell + \rho_2\theta_\ell^2 + \ldots + \rho_N\theta_\ell^N\right) \\ \mathsf{b}^\tau_{\ell,i} - b^\tau_{i,0} - b^\tau_{i,1}\left(\rho_1\theta_\ell + \rho_2\theta_\ell^2 + \ldots + \rho_N\theta_\ell^N\right) \\ \vdots \\ \mathsf{a}^{\tau_1+\tau_2}_{m,2} - a^{\tau_1+\tau_2}_{2,0} - a^{\tau_1+\tau_2}_{2,1}\left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \\ \mathsf{b}^{\tau_1+\tau_2}_{m,2} - b^{\tau_1+\tau_2}_{2,0} - b^{\tau_1+\tau_2}_{2,1}\left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \end{array}\right], \tag{15}$$

which is now nonlinear in the reduced vector of variables

$$q = \left[\begin{array}{ccc|cc|ccccc} \rho_1 & \ldots & \rho_N & g_0 & g_1 & a^{\tau}_{i,0} & b^{\tau}_{i,0} & \ldots & a^{\tau}_{i,1} & b^{\tau}_{i,1} \end{array}\right]'.$$

The vector-valued function (15) is nonlinear due to the multiplication of the variables ρ_k with the variables g_1, $a^{\tau}_{i,1}$ and $b^{\tau}_{i,1}$. With this cost function, the optimization problem

$$\min_{a,b,g,\rho} E(a,b,g,\rho) = \min_{q} \|F_A(q)\|_2^2 \tag{16}$$

is a nonlinear least-squares problem, which can be solved, for instance, using the standard implementation of the Levenberg–Marquardt algorithm found in the optimization toolbox from Matlab [34]. This algorithm requires the Jacobian matrix $J(q) = \partial F_A(q)/\partial q$, which is promptly derived from (15).

It is well-known that nonconvex optimization problems are usually sensitive to the starting value of the optimization variables. Unfortunately, there are no general rules for choosing the initial condition. For problem (16), one possible starting value is obtained by fixing the variables g_1, $a^{\tau}_{i,1}$ and $b^{\tau}_{i,1}$ equal to 1. This results in the linear vector-valued function

$$F_{SV}(q) = \left[\begin{array}{c} \mathsf{K}_1 - g_0 - \left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{K}_m - g_0 - \left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \\ \hline \mathsf{a}^1_{1,1} - a^1_{1,0} - \left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \mathsf{b}^1_{1,1} - b^1_{1,0} - \left(\rho_1\theta_1 + \rho_2\theta_1^2 + \ldots + \rho_N\theta_1^N\right) \\ \vdots \\ \mathsf{a}^{\tau}_{\ell,i} - a^{\tau}_{i,0} - \left(\rho_1\theta_\ell + \rho_2\theta_\ell^2 + \ldots + \rho_N\theta_\ell^N\right) \\ \mathsf{b}^{\tau}_{\ell,i} - b^{\tau}_{i,0} - \left(\rho_1\theta_\ell + \rho_2\theta_\ell^2 + \ldots + \rho_N\theta_\ell^N\right) \\ \vdots \\ \mathsf{a}^{\tau_1+\tau_2}_{m,2} - a^{\tau_1+\tau_2}_{2,0} - \left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \\ \mathsf{b}^{\tau_1+\tau_2}_{m,2} - b^{\tau_1+\tau_2}_{2,0} - \left(\rho_1\theta_m + \rho_2\theta_m^2 + \ldots + \rho_N\theta_m^N\right) \end{array}\right].$$

The solution q^* of this linear least-squares problem, $q^* = \arg\min\|F_{SV}(q)\|_2^2$, together with $g_1 = a^{\tau}_{i,1} = b^{\tau}_{i,1} = 1$, provides the starting value for the nonlinear optimization problem (16). Using this starting value, the nonlinear least-squares problem (16) is solved to obtain the polynomial $\rho(\theta(t))$ and the coefficients of the system matrices of the LPV submodels. Numerical experiments and comparisons with randomly generated starting values have shown that the above procedure usually yields good results. However, it is important to emphasize that most existing nonlinear optimization solvers cannot guarantee convergence to a global optimum.

4.3. Construction of the interpolating LPV model (Step 6)

Once the coefficients $a^{\tau}_{i,k}$, $b^{\tau}_{i,k}$, ρ_k and g_k are obtained from the above optimization problem, the LPV submodels are readily constructed. Subsequently, the final interpolating LPV model can be obtained by explicitly computing the series connection (7). For an LPV model with two LPV submodels, this series connection is shown in (10).

5. Multiparameter-dependency

In this section, the interpolation method is extended to systems whose dynamics depend on a vector of scheduling parameters denoted by $\phi(t) = [\phi_1(t) \ \ldots \ \phi_M(t)]' \in \mathbb{R}^M$, with M the number of parameters. In this case, the interpolating LPV model of polynomial degree N is parameterized by the following state-space representation:

$$H(\phi(t)) := \left[\begin{array}{c|c} \sum_{k\in\Gamma_M(N)} A_k\rho_k\phi(t)^k & \sum_{k\in\Gamma_M(N)} B_k\rho_k\phi(t)^k \\ \hline \sum_{k\in\Gamma_M(N)} C_k\rho_k\phi(t)^k & \sum_{k\in\Gamma_M(N)} D_k\rho_k\phi(t)^k \end{array}\right], \tag{17}$$

where $\Gamma_M(N)$ represents the set of all possible M-tuples of degree N or less, given by

$$\Gamma_M(N) = \left\{k = [k_1 \ \ldots \ k_M] \in \mathbb{N}^M : \sum_{i=1}^{M} k_i \le N\right\}, \tag{18}$$

and the monomial $\phi(t)^k$ is given by $\phi(t)^k{:=}\phi_1(t)^{k_1}\phi_2(t)^{k_2}\ldots\phi_M(t)^{k_M}$. Similarly to the single scheduling parameter case, the purpose of the above LPV model (17) is to interpolate m local LTI models identified for fixed operating conditions, that is, for constant values ϕ_ℓ, for $\ell = 1,\ldots,m$, of the vector of scheduling parameters $\phi(t)$.

Basically, all six steps of the interpolation method presented in the two previous sections can be adopted to the multiparameter case. The first three steps, calculating and sorting the poles and zeros of the local LTI models (Step 1), representing the local LTI models by a gain multiplied with the series connection of first and second order submodels (Step 2) and calculating the local gains for all local LTI models (Step 3) are performed in exactly the same way as described in Section 3. In order to interpolate the local LTI submodels and local gains defined in Steps 2 and 3, the parameterization of the LPV submodels and LPV gain is extended to incorporate the multiparameter-dependency using multivariable polynomials. For instance, the LPV submodel of type 5, defined in (9) for the single scheduling parameter case, is extended to the multiple parameter case as follows:

$$H^\tau(\phi(t)) := \left[\begin{array}{cc|c} 0 & \sum\limits_{k\in\Gamma_M(N)} a^\tau_{1,k}\rho_k\phi(t)^k & \sum\limits_{k\in\Gamma_M(N)} b^\tau_{1,k}\rho_k\phi(t)^k \\ 1 & \sum\limits_{k\in\Gamma_M(N)} a^\tau_{2,k}\rho_k\phi(t)^k & \sum\limits_{k\in\Gamma_M(N)} b^\tau_{2,k}\rho_k\phi(t)^k \\ \hline 0 & 1 & 1 \end{array}\right], \tag{19}$$

with $\Gamma_M(N)$ defined in (18).

The next step in the interpolation method (Step 4) is the choice of the subclass of interpolating LPV model. Similarly to the single parameter case, a polynomial LPV model in $\phi(t)$ and an affine model in $\rho(\phi(t))$ can be defined by fixing different combinations of the coefficients in (17).

(1) By fixing $\rho_k = 1$ for all k the LPV model $H(\phi(t))$ becomes polynomial in $\phi(t)$:

$$H(\phi(t)) := \left[\begin{array}{c|c} \sum_{k\in\Gamma_M(N)} A_k\phi(t)^k & \sum_{k\in\Gamma_M(N)} B_k\phi(t)^k \\ \hline \sum_{k\in\Gamma_M(N)} C_k\phi(t)^k & \sum_{k\in\Gamma_M(N)} D_k\phi(t)^k \end{array}\right]. \tag{20}$$

(2) By fixing $\rho_0 = 1$ and $A_k = A_1$, $B_k = B_1$, $C_k = C_1$ and $D_k = D_1$ for $k \neq 0$, the LPV model $H(\phi(t))$ becomes affine in $\rho(\phi(t))$:

$$H(\phi(t)) := \left[\begin{array}{c|c} A_0 + A_1\rho(\phi(t)) & B_0 + B_1\rho(\phi(t)) \\ \hline C_0 + C_1\rho(\phi(t)) & D_0 + D_1\rho(\phi(t)) \end{array}\right], \quad \text{with } \rho(\phi(t)) = \sum_{k\in\Gamma_M(N)/\{0\}} \rho_k\phi(t)^k \tag{21}$$

where $\Gamma_M(N)/\{0\}$ denotes the set of M-tuples of degree N or less, without the zero element.

Moreover, since the LPV model now depends on multiple scheduling parameters, more subclasses can be defined. For instance, it is straightforward to define an LPV model by

$$H(\phi(t)) := \left[\begin{array}{c|c} A_0 + A_1\rho_1(\phi_1(t)) + \ldots + A_M\rho_M(\phi_M(t)) & B_0 + B_1\rho_1(\phi_1(t)) + \ldots + B_M\rho_M(\phi_M(t)) \\ \hline C_0 + C_1\rho_1(\phi_1(t)) + \ldots + C_M\rho_M(\phi_M(t)) & D_0 + D_1\rho_1(\phi_1(t)) + \ldots + D_M\rho_M(\phi_M(t)) \end{array}\right], \tag{22}$$

that is affine in M polynomials $\rho_i(\phi_i(t)) = \sum_{j=1}^N \rho_{i,j}\phi_i(t)^j$. This subclass is interesting whenever it is a priori known that there is no physical interaction between the different scheduling parameters. Formulating and solving the optimization problems for these subclasses is analogous to the single parameter case presented in Section 4.2. For example, the cost function to interpolate a set of m local LTI submodels of type 5, given by (6), by an LPV submodel of type 5, given by (19), is chosen to be

$$E_\tau(a^\tau, b^\tau, \rho) = \sum_{\ell=1}^{m}\sum_{i=1}^{2} \left| \mathrm{a}^\tau_{\ell,i} - \sum_{k\in\Gamma_M(N)} a^\tau_{i,k}\rho_k\phi_\ell^k \right|^2 + \left| \mathrm{b}^\tau_{\ell,i} - \sum_{k\in\Gamma_M(N)} b^\tau_{i,k}\rho_k\phi_\ell^k \right|^2,$$

where $a^\tau = \{a^\tau_{i,k}\}$, $b^\tau = \{b^\tau_{i,k}\}$, for $i = 1, 2$ and $k \in \Gamma_M(N)$, and $\rho = \{\rho_k\}$ for $k \in \Gamma_M(N)$. The cost functions to interpolate the other types of submodels and the local gains are analogously defined. Thus, the total cost function is given by

$$E(a, b, g, \rho) = E_K(g, \rho) + \sum_{\tau=1}^{\tau_1+\tau_2} E_\tau(a^\tau, b^\tau, \rho),$$

with $a = \{a^\tau\}$ and $b = \{b^\tau\}$ for $\tau = 1, \ldots, \tau_1 + \tau_2$. Depending on the type of the interpolating LPV model chosen in Step 4, this cost function leads to either a linear or a nonlinear least-squares optimization problem that can be solved using the same techniques discussed in Section 4.2.

6. Numerical example

This section discusses the application of the interpolation method on an electromechanical system that depends on two scheduling parameters. First, in Section 6.1, an analytic LPV model of the electromechanical system is derived based on its equations of motion. This analytical LPV model will be used to validate the proposed interpolation technique. To apply the interpolation method, it is necessary to have local LTI models, which, in this case, are readily obtained by evaluating the analytic LPV model for different values of the two scheduling parameters. Subsequently, Section 6.2 shows how the

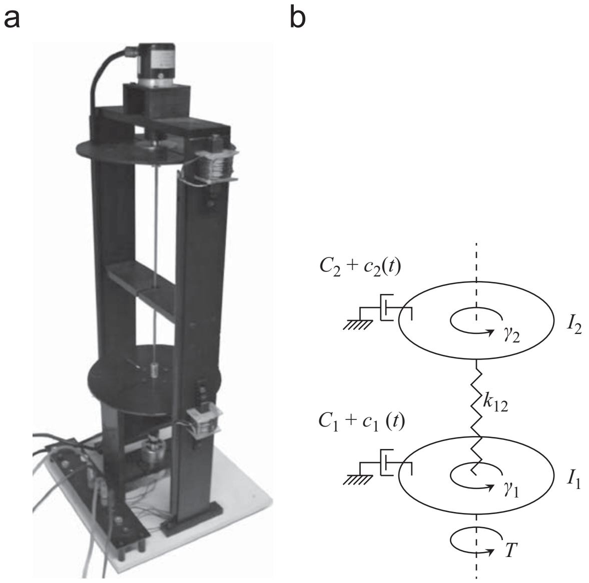

Fig. 3. The electromechanical system with two scheduling parameters $c_1(t)$ and $c_2(t)$. (a) Electromechanical system. (b) Lumped parameter model.

different steps of the interpolation method are applied to the local LTI models. Finally, Section 6.3 presents the numerical results and Section 6.4 compares the proposed method to others techniques available in the literature.

6.1. The electromechanical system

The electromechanical system (presented in Fig. 3a) consists of two rotating discs connected by a flexible torsional beam. Each disc has an electromagnetic brake with a time-varying damping coefficient. Fig. 3b shows a lumped parameter model for this system. The discs are represented by their moment of inertia ($I_1 = 5.4625 \times 10^{-5}$ and $I_2 = 5.4088 \times 10^{-5}\,\mathrm{kg\,m^2}$), the flexible beam by a torsional spring with stiffness $k_{12} = 0.0256\,\mathrm{N\,m/rad}$ and viscous friction between each disc and the fixed world by the damping coefficients $C_1 = 3.2223 \times 10^{-4}$ and $C_2 = 5.1921 \times 10^{-4}\,\mathrm{N\,m\,s/rad}$. The electromagnetic brakes are represented by two additional damping coefficients $c_1(t)$ and $c_2(t)$. The angular displacement of the two discs is denoted by $\gamma_1(t)$ and $\gamma_2(t)$. The input to the system is the torque $T(t)$ applied by a DC-motor to the first disc. Based on Fig. 3b, the equations of motion are readily obtained as

$$\begin{aligned}
&I_1\ddot{\gamma}_1(t) = -(C_1 + c_1(t))\dot{\gamma}_1(t) + k_{12}(\gamma_2(t) - \gamma_1(t)) + T(t)\\
&I_2\ddot{\gamma}_2(t) = -(C_2 + c_2(t))\dot{\gamma}_2(t) + k_{12}(\gamma_1(t) - \gamma_2(t)).
\end{aligned}$$

These equations can be conveniently represented in a state-space form, affine in the two scheduling parameters $c_1(t)$ and $c_2(t)$:

$$\begin{bmatrix} \dot{\gamma}_1(t) \\ \dot{\gamma}_2(t) \\ \ddot{\gamma}_1(t) \\ \ddot{\gamma}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_{12}/I_1 & k_{12}/I_1 & -(C_1 + c_1(t))/I_1 & 0 \\ k_{12}/I_2 & -k_{12}/I_2 & 0 & -(C_2 + c_2(t))/I_2 \end{bmatrix} \begin{bmatrix} \gamma_1(t) \\ \gamma_2(t) \\ \dot{\gamma}_1(t) \\ \dot{\gamma}_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/I_1 \\ 0 \end{bmatrix} u(t), \tag{23}$$

where the system input $u(t)$ is defined as the torque $T(t)$. The output of the system is the angular displacement of disc 1. Hence, the output equation is given by $y(t) = \gamma_1(t)$.

The analytic state-space LPV model (23) is now used to obtain m local continuous-time LTI models

$$\mathsf{H}_\ell := \left[\begin{array}{c|c} \mathsf{A}_\ell & \mathsf{B}_\ell \\ \hline \mathsf{C}_\ell & \mathsf{D}_\ell \end{array}\right], \quad \ell = 1, \ldots, m, \tag{24}$$

for m different fixed operating conditions. Note that there are many different possibilities to choose the operating conditions. Essentially, the choice depends on the dynamics of the system and the expertise of the user. However, one general approach to select the fixed operation conditions is to choose, for each scheduling parameter, an equidistant grid of values between its lower and upper bound and to take all possible combinations. The density of the grid for each scheduling parameter should be chosen as small as possible (to reduce the number of LTI identification experiments), but high enough such that the influence of the parameter on the system dynamics is captured. For the electromechanical

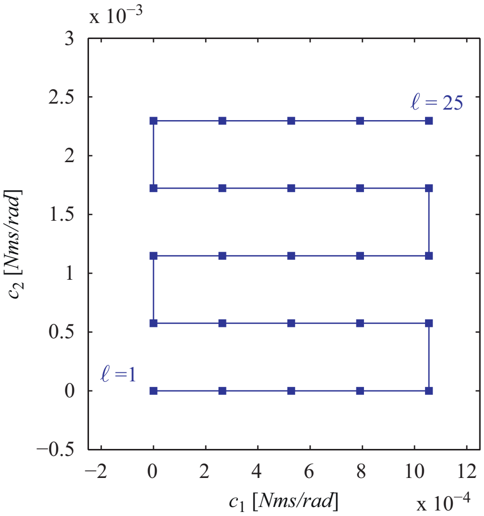

Fig. 4. Constant values for the pair $(c_{1,\ell}, c_{2,\ell})$ (blue, squares) of the scheduling parameters for $\ell = 1, \ldots, 25$.

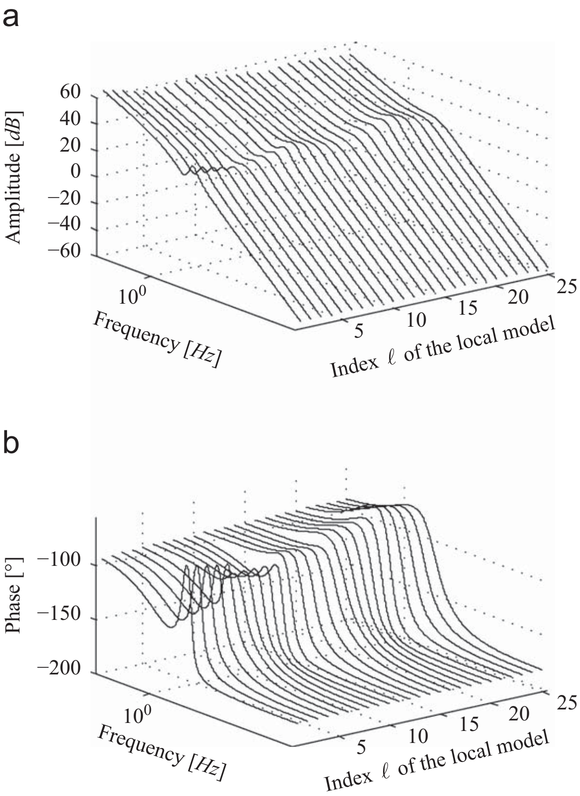

Fig. 5. Bode plot of the 25 local LTI models as a function of the index ℓ. (a) amplitude and (b) phase.

system, an equidistant grid of five constant values is chosen for each scheduling parameters

$$\begin{aligned} c_1 &\in \left\{0, \frac{\bar{c_1}}{4}, \frac{\bar{c_1}}{2}, \frac{3\bar{c_1}}{4}, \bar{c_1}\right\} \ (\mathrm{N\,m\,s/rad}), \\ c_2 &\in \left\{0, \frac{\bar{c_2}}{4}, \frac{\bar{c_2}}{2}, \frac{3\bar{c_2}}{4}, \bar{c_2}\right\} \ (\mathrm{N\,m\,s/rad}), \end{aligned} \quad (25)$$

where $\bar{c_1} = 1.0552 \times 10^{-3}$ and $\bar{c_2} = 2.2977 \times 10^{-3}\,\mathrm{N\,m\,s/rad}$ represent the maximum additional damping that can be added by the electromagnetic brakes. Fig. 4 presents all possible ($m = 25$) different values for the pair $(c_{1,\ell}, c_{2,\ell})$ for $\ell = 1, \ldots, 25$. The dynamics of the resulting m local LTI models provide a reasonable representation of the different possible operating conditions of the system. Fig. 5 shows the Bode plot of the 25 local LTI models as a function of the index ℓ of the local model.

With the 25 local LTI models (24) obtained for the values given in (25), the proposed interpolation method can be applied. As stated in Section 4.1, the method can be used to obtain both polynomial and affine LPV models. To analyze the differences and similarities between the subclasses of interpolating LPV model, both a polynomial and an affine interpolating LPV model are obtained in the next section.

6.2. *Application of the interpolation method*

Since the dynamics of the electromechanical system depend on two scheduling parameters $c_1(t)$ and $c_2(t)$, the vector of scheduling parameters $\phi(t)$ is given by

$$\phi(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \end{bmatrix} = \begin{bmatrix} c_1(t) \\ c_2(t) \end{bmatrix} \in \mathbb{R}^2.$$

The proposed interpolation method is now applied to the 25 local LTI models to obtain both a polynomial interpolating LPV model and an affine interpolating LPV model, respectively, denoted by $H_P(\phi(t))$ and by $H_A(\phi(t))$.

6.2.1. *Step 1. For each local LTI model, calculate the set of poles p_ℓ and the set of zeros z_ℓ*

For each of the 25 local LTI models, Fig. 6a shows the four poles in the complex plane as a function of the index ℓ of the local LTI model. There is a fixed pole $\mathrm{p}_{\ell,1}$ at 0 (blue, squares), a varying real pole $\mathrm{p}_{\ell,2}$ (black, crosses) and a varying complex conjugate pole pair $\mathrm{p}_{\ell,3}$ and $\mathrm{p}_{\ell,4}$ (red, diamonds). All local LTI models have two zeros $\mathrm{z}_{\ell,1}$ and $\mathrm{z}_{\ell,2}$. As shown in Fig. 6b (green, circles), these zeros change from a complex conjugate pair to a pair of real zeros between $\ell = 20$ and 21.

6.2.2. *Step 2. Divide each local LTI model into a series connection of first and second order submodels*

Given the sorted poles and zeros, the local LTI models can be represented by a gain multiplied by the series connection of first and second order submodels. Since the order n of the local LTI models is 4, the number of second order submodels

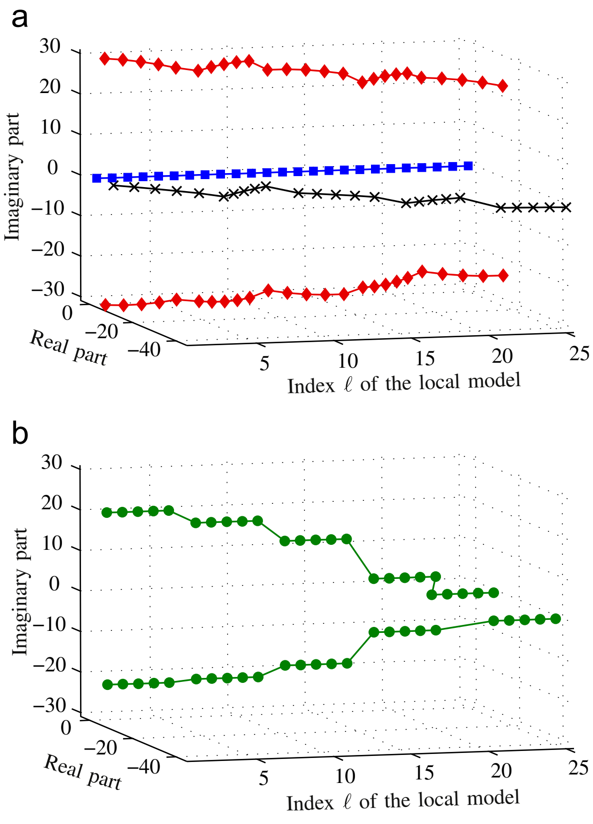

Fig. 6. Real and imaginary part of the poles and zeros of the 25 local LTI models as a function of the index ℓ. (a) poles, (b) zeros.

$\tau_2 = 2$ and there are no first order submodels ($\tau_1 = 0$). The first submodel H_ℓ^1 is chosen to contain the two real poles $\mathsf{p}_{\ell,1}$ and $\mathsf{p}_{\ell,2}$ and no zeros. Thus, this submodel is of type 3. The second submodel H_ℓ^2 is chosen to contain the complex conjugate pole pair and the two zeros, thus being of type 5. These submodels are represented by

$$\mathsf{H}_\ell^1 := \left[\begin{array}{cc|c} 0 & \mathsf{a}_{\ell,1}^1 & 1 \\ 1 & \mathsf{a}_{\ell,2}^1 & 0 \\ \hline 0 & 1 & 0 \end{array}\right] \quad \text{and} \quad \mathsf{H}_\ell^2 := \left[\begin{array}{cc|c} 0 & \mathsf{a}_{\ell,1}^2 & \mathsf{b}_{\ell,1}^2 \\ 1 & \mathsf{a}_{\ell,2}^2 & \mathsf{b}_{\ell,2}^2 \\ \hline 0 & 1 & 1 \end{array}\right],$$

where

$$\begin{aligned} &\mathsf{a}_{\ell,1}^1 = -\mathsf{p}_{\ell,1}\mathsf{p}_{\ell,2}, \quad \mathsf{a}_{\ell,2}^1 = \mathsf{p}_{\ell,1} + \mathsf{p}_{\ell,2}, \\ &\mathsf{a}_{\ell,1}^2 = -\mathsf{p}_{\ell,3}\mathsf{p}_{\ell,4}, \quad \mathsf{a}_{\ell,2}^2 = \mathsf{p}_{\ell,3} + \mathsf{p}_{\ell,4}, \\ &\mathsf{b}_{\ell,1}^2 = \mathsf{z}_{\ell,1}\mathsf{z}_{\ell,2} - \mathsf{p}_{\ell,3}\mathsf{p}_{\ell,4}, \quad \mathsf{b}_{\ell,2}^2 = -(\mathsf{z}_{\ell,1} + \mathsf{z}_{\ell,2}) + (\mathsf{p}_{\ell,3} + \mathsf{p}_{\ell,4}), \end{aligned} \tag{26}$$

for $\ell = 1, \ldots, 25$. Computing the series connection of these two submodels, the local LTI models H_ℓ can now be expressed as

$$\mathsf{H}_\ell := \mathsf{K}_\ell \prod_{\tau=1}^{2} \mathsf{H}_\ell^\tau = \left[\begin{array}{cccc|c} 0 & \mathsf{a}_{\ell,1}^1 & 0 & 0 & 1 \\ 1 & \mathsf{a}_{\ell,2}^1 & 0 & 0 & 0 \\ 0 & \mathsf{b}_{\ell,1}^2 & 0 & \mathsf{a}_{\ell,1}^2 & 0 \\ 0 & \mathsf{b}_{\ell,2}^2 & 1 & \mathsf{a}_{\ell,2}^2 & 0 \\ \hline 0 & \mathsf{K}_\ell & 0 & \mathsf{K}_\ell & 0 \end{array}\right]. \tag{27}$$

6.2.3. Step 3. Calculate the gain

The local gain K_ℓ in (27) is calculated using the procedure of Appendix A. For this example, the gain was found to be $\mathsf{K}_\ell = 1.8307 \times 10^4$ for all local LTI models.

6.2.4. Step 4. Select the subclass of LPV model

For this application the aim is to compute both a polynomial interpolating LPV model $H_P(\phi(t))$ and an affine interpolating LPV model $H_A(\phi(t))$. The polynomial degree of $H_P(\phi(t))$ and $H_A(\phi(t))$ is chosen to be $N = 2$. Since the system has two scheduling parameters, the LPV models $H_P(\phi(t))$ and $H_A(\phi(t))$ will have a polynomial dependency of degree 2 in two variables and therefore, using (18), the possible exponents of the monomials are given by $\Gamma_2(2) = \{00, 10, 01, 20, 11, 02\}$. For the polynomial LPV model $H_P(\phi(t))$, the two LPV submodels, denoted by $H_P^1(\phi(t))$ and $H_P^2(\phi(t))$, are parameterized as

$$H_P{}^1(\phi(t)) := \left[\begin{array}{cc|c} 0 & a_{P}{}_1^1(\phi(t)) & 1 \\ 1 & a_{P}{}_2^1(\phi(t)) & 0 \\ \hline 0 & 1 & 0 \end{array}\right] \quad \text{and} \quad H_P{}^2(\phi) := \left[\begin{array}{cc|c} 0 & a_{P}{}_1^2(\phi(t)) & b_{P}{}_1^2(\phi(t)) \\ 1 & a_{P}{}_2^2(\phi(t)) & b_{P}{}_2^2(\phi(t)) \\ \hline 0 & 1 & 1 \end{array}\right],$$

where

$$\begin{aligned} a_{P}{}_1^1(\phi(t)) &= a_{P}{}_{1,00}^1 + a_{P}{}_{1,10}^1\phi_1(t) + a_{P}{}_{1,01}^1\phi_2(t) + a_{P}{}_{1,20}^1\phi_1(t)^2 + a_{P}{}_{1,11}^1\phi_1(t)\phi_2(t) + a_{P}{}_{1,02}^1\phi_2(t)^2, \\ a_{P}{}_2^1(\phi(t)) &= a_{P}{}_{2,00}^1 + a_{P}{}_{2,10}^1\phi_1(t) + a_{P}{}_{2,01}^1\phi_2(t) + a_{P}{}_{2,20}^1\phi_1(t)^2 + a_{P}{}_{2,11}^1\phi_1(t)\phi_2(t) + a_{P}{}_{2,02}^1\phi_2(t)^2, \\ a_{P}{}_1^2(\phi(t)) &= a_{P}{}_{1,00}^2 + a_{P}{}_{1,10}^2\phi_1(t) + a_{P}{}_{1,01}^2\phi_2(t) + a_{P}{}_{1,20}^2\phi_1(t)^2 + a_{P}{}_{1,11}^2\phi_1(t)\phi_2(t) + a_{P}{}_{1,02}^2\phi_2(t)^2, \\ a_{P}{}_2^2(\phi(t)) &= a_{P}{}_{2,00}^2 + a_{P}{}_{2,10}^2\phi_1(t) + a_{P}{}_{2,01}^2\phi_2(t) + a_{P}{}_{2,20}^2\phi_1(t)^2 + a_{P}{}_{2,11}^2\phi_1(t)\phi_2(t) + a_{P}{}_{2,02}^2\phi_2(t)^2, \\ b_{P}{}_1^2(\phi(t)) &= b_{P}{}_{1,00}^2 + b_{P}{}_{1,10}^2\phi_1(t) + b_{P}{}_{1,01}^2\phi_2(t) + b_{P}{}_{1,20}^2\phi_1(t)^2 + b_{P}{}_{1,11}^2\phi_1(t)\phi_2(t) + b_{P}{}_{1,02}^2\phi_2(t)^2, \\ b_{P}{}_2^2(\phi(t)) &= b_{P}{}_{2,00}^2 + b_{P}{}_{2,10}^2\phi_1(t) + b_{P}{}_{2,01}^2\phi_2(t) + b_{P}{}_{2,20}^2\phi_1(t)^2 + b_{P}{}_{2,11}^2\phi_1(t)\phi_2(t) + b_{P}{}_{2,02}^2\phi_2(t)^2. \end{aligned} \tag{28}$$

The two polynomial LPV submodels are completely parameterized by the 36 unknown coefficients $a_{P}{}_{i,k}^{\tau}$ and $b_{P}{}_{i,k}^{2}$, for $i = 1, 2$, $\tau = 1, 2$ and $k \in \Gamma_2(2) = \{00, 10, 01, 20, 11, 02\}$.

For the affine LPV model $H_A(\phi(t))$, a parameterization of the form (22) is chosen. In this case, $H_A(\phi(t))$ is affine in two polynomials of degree $N = 2$: a polynomial that is only function of the first scheduling parameter $\phi_1(t)$, denoted by $\rho_1(\phi_1(t)) = \sum_{j=1}^{2} \rho_{1,j}\phi_1(t)^j$, and a polynomial that is only function of the second scheduling parameter $\phi_2(t)$, denoted by

$\rho_2(\phi_2(t)) = \sum_{j=1}^{2} \rho_{2,j}\phi_2(t)^j$. Consequently, the two LPV submodels, denoted by $H_A^1(\phi(t))$ and $H_A^2(\phi(t))$, are parameterized as

$$H_A{}^1(\phi(t)) := \left[\begin{array}{cc|c} 0 & {a_A}^1_1(\phi(t)) & 1 \\ 1 & {a_A}^1_2(\phi(t)) & 0 \\ \hline 0 & 1 & 0 \end{array}\right] \quad \text{and} \quad H_A{}^2(\phi(t)) := \left[\begin{array}{cc|c} 0 & {a_A}^2_1(\phi(t)) & {b_A}^2_1(\phi(t)) \\ 1 & {a_A}^2_2(\phi(t)) & {b_A}^2_2(\phi(t)) \\ \hline 0 & 1 & 1 \end{array}\right],$$

where

$$\begin{aligned}
{a_A}^1_1(\phi(t)) &= {a_A}^1_{1,0} + {a_A}^1_{1,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {a_A}^1_{1,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2),\\
{a_A}^1_2(\phi(t)) &= {a_A}^1_{2,0} + {a_A}^1_{2,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {a_A}^1_{2,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2),\\
{a_A}^2_1(\phi(t)) &= {a_A}^2_{1,0} + {a_A}^2_{1,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {a_A}^2_{1,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2),\\
{a_A}^2_2(\phi(t)) &= {a_A}^2_{2,0} + {a_A}^2_{2,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {a_A}^2_{2,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2),\\
{b_A}^2_1(\phi(t)) &= {b_A}^2_{1,0} + {b_A}^2_{1,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {b_A}^2_{1,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2),\\
{b_A}^2_2(\phi(t)) &= {b_A}^2_{2,0} + {b_A}^2_{1,1}(\rho_{1,1}\phi_1(t) + \rho_{1,2}\phi_1(t)^2) + {b_A}^2_{2,2}(\rho_{2,1}\phi_2(t) + \rho_{2,2}\phi_2(t)^2).
\end{aligned} \tag{29}$$

The two affine LPV submodels are completely parameterized by the four unknown coefficients $\rho_{i,j}$, for $i = 1,2$ and $j = 1,2$, and the eighteen unknown coefficients ${a_A}^i_{\tau,k}$ and ${b_A}^i_{2,k}$, for $i = 1,2$, $\tau = 1,2$ and $k = 0,1,2$.

6.2.5. Step 5. Formulate and solve the optimization problem

Using the entries of the system matrices of the constructed local LTI submodels (26) and the parameterization of the coefficients of the polynomial LPV submodels (28), the cost function for the linear least-squares problem is formulated as

$$E_P = \sum_{\ell=1}^{m}\sum_{\tau=1}^{2}\sum_{i=1}^{2} \left|\mathsf{a}^{\tau}_{\ell,i} - {a_P}^{\tau}_{i}(\phi_\ell)\right|^2 + \sum_{\ell=1}^{m}\sum_{i=1}^{2} \left|\mathsf{b}^{2}_{\ell,i} - {b_P}^{2}_{i}(\phi_\ell)\right|^2. \tag{30}$$

Analogously, using (26) and the parameterization of the coefficients of the affine LPV submodels (29), the cost function for the nonlinear least-squares problem is formulated as

$$E_A = \sum_{\ell=1}^{m}\sum_{\tau=1}^{2}\sum_{i=1}^{2} \left|\mathsf{a}^{\tau}_{\ell,i} - {a_A}^{\tau}_{i}(\phi_\ell)\right|^2 + \sum_{\ell=1}^{m}\sum_{i=1}^{2} \left|\mathsf{b}^{2}_{\ell,i} - {b_A}^{2}_{i}(\phi_\ell)\right|^2. \tag{31}$$

These cost functions are minimized using the techniques proposed in Section 4.2. For the linear problem (30), the optimal value is $E_P = 99.3341$ and for the nonlinear problem (31), the optimal value is $E_A = 240.6264$. The residue of the interpolation of the entries of the system matrices of the local LTI submodels is smaller for the polynomial LPV model, as expected, since the parameterization of this LPV model has more optimization variables.

It is unnecessary to interpolate the local gains since the local gain K_ℓ is the same for all local LTI models. Therefore, the gain of the polynomial interpolating LPV model, denoted by $K_P(\phi(t))$, and the gain of the affine interpolating LPV model, denoted by $K_A(\phi(t))$, are constant and given by $K_P(\phi(t)) = K_A(\phi(t)) = \mathsf{K}_\ell = 1.8307 \times 10^4$.

6.2.6. Step 6. Construct the interpolating LPV model

The interpolating LPV models are obtained as the series connection of the LPV submodels multiplied by the gain. For instance, using (10), the obtained affine interpolating LPV model is given by

$$H_A(\phi(t)) := \left[\begin{array}{cccc|c}
0 & {a_A}^1_{1,0} + {a_A}^1_{1,1}\rho_1(\phi_1(t)) + {a_A}^1_{1,2}\rho_2(\phi_2(t)) & 0 & 0 & 1\\
1 & {a_A}^1_{2,0} + {a_A}^1_{2,1}\rho_1(\phi_1(t)) + {a_A}^1_{2,2}\rho_2(\phi_2(t)) & 0 & 0 & 0\\
0 & {b_A}^2_{1,0} + {b_A}^2_{1,1}\rho_1(\phi_1(t)) + {b_A}^2_{1,2}\rho_2(\phi_2(t)) & 0 & {a_A}^2_{1,0} + {a_A}^2_{1,1}\rho_1(\phi_1(t)) + {a_A}^2_{1,2}\rho_2(\phi_2(t)) & 0\\
0 & {b_A}^2_{2,0} + {b_A}^2_{2,1}\rho_1(\phi_1(t)) + {b_A}^2_{2,2}\rho_2(\phi_2(t)) & 1 & {a_A}^2_{2,0} + {a_A}^2_{2,1}\rho_1(\phi_1(t)) + {a_A}^2_{2,2}\rho_2(\phi_2(t)) & 0\\
\hline
0 & K_A(\phi(t)) & 0 & K_A(\phi(t)) & 0
\end{array}\right].$$

6.3. Validation

The interpolating LPV models need to be validated to verify whether they accurately represent the dynamics of the electromechanical system, even for operating conditions that are not used during the LPV modeling. For this purpose, the analytical state-space LPV model (23) and the obtained interpolating LPV models are evaluated and compared for the following values of the scheduling parameters

$$c_{e1,i} = \frac{\bar{c_1}}{2} + \frac{\bar{c_1}}{2}\sin\left(\frac{\pi}{13}i\right)\ (\mathrm{N\,m\,s/rad}), \quad c_{e2,i} = \frac{\bar{c_2}}{2} + \frac{\bar{c_2}}{2}\sin\left(\frac{2\pi}{13}i\right)\ (\mathrm{N\,m\,s/rad}), \tag{32}$$

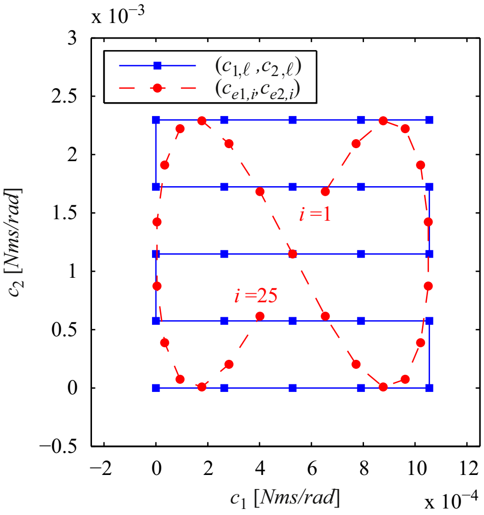

Fig. 7. Constant values for the pair $(c_{1,\ell}, c_{2,\ell})$ (blue, squares) for the 25 local LTI models and for the pair $(c_{e1,i}, c_{e2,i})$ (red, circles) for the 25 LTI models for the evaluation of the LPV models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

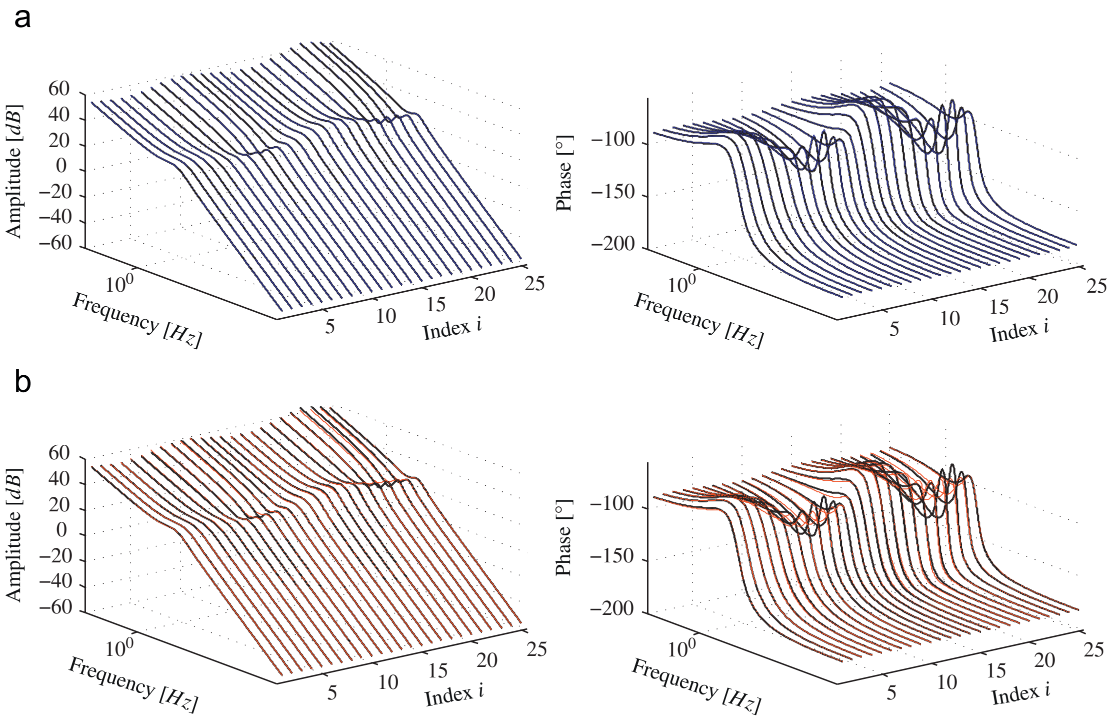

Fig. 8. Bode plot of the models in $\mathscr{L}_P$ (blue) and in $\mathscr{L}_A$ (red) compared to the Bode plot of the 25 LTI models obtained from the analytical LPV model (black). (a) Comparison for the models in $\mathscr{L}_P$. (b) Comparison for the models in $\mathscr{L}_A$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for $i = 1, \ldots, 25$. These values, presented in Fig. 7 with red circles, result in a smooth variation of the dynamics of the obtained LTI models as a function of the index i which facilitates visualization of the results.

Evaluating the polynomial interpolating LPV model $H_P(\phi(t))$ and the affine interpolating LPV model $H_A(\phi(t))$ for the 25 values of the two scheduling parameters (32), results in two sets of 25 models, which are denoted by

$$\mathscr{L}_P = \{H_P(\phi_{e,i}) | \phi_{e,i} = [c_{e1,i} \;\; c_{e2,i}]' \text{ for } i = 1, \ldots, 25\},$$

$$\mathscr{L}_A = \{H_A(\phi_{e,i}) | \phi_{e,i} = [c_{e1,i} \;\; c_{e2,i}]' \quad \text{for } i = 1, \ldots, 25\}.$$

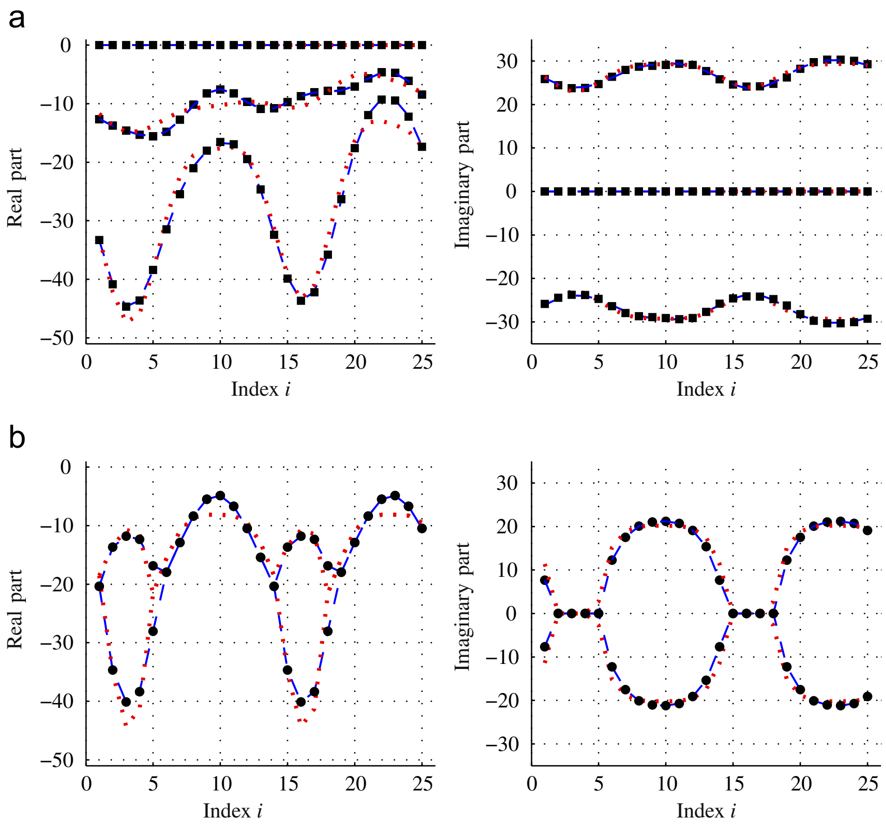

Fig. 9. Poles and zeros of the models in $\mathscr{L}_P$ (blue, dashed) and in $\mathscr{L}_A$ (red, dotted) and the poles and zeros of the 25 LTI models obtained from the analytical LPV model (black, squares and circles). (a) poles, (b) zeros. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 8 presents the Bode plots of the models in $\mathscr{L}_P$ (blue) and $\mathscr{L}_A$ (red) and the Bode plots of the 25 LTI models obtained by evaluating the analytical LPV model (23) for the operating conditions (32) (black). Clearly, the dynamics of the LTI models obtained from the analytical LPV model (23) are smoothly interpolated. The amplitude and phase plots show that the fit by the polynomial interpolating LPV model $H_P(\phi(t))$ is better than the fit by the affine interpolating LPV model $H_A(\phi(t))$. This can also be seen in Fig. 9, which shows the poles and zeros of the models in sets $\mathscr{L}_P$ (blue, dashed) and $\mathscr{L}_A$ (red, dotted) and the poles and zeros of the LTI models obtained from the analytical LPV model (23). The curves for the poles and zeros of the models in set $\mathscr{L}_P$ accurately fit the pole and zero curves of the local LTI models, while the fit by the curves for the poles and zeros of the models in set $\mathscr{L}_A$ is slightly worse. Recall that the optimal value E_P for the polynomial interpolating LPV model was significantly lower than the optimal value E_A for the affine interpolating LPV model (see Step 5 in Section 6.2). Therefore, it can be concluded that a better fit of the entries of the system matrices of the local LTI submodels results in a better fit of the dynamics of the local LTI models. Since the parameterization of the polynomial interpolating LPV model has more optimization variables, this model yields a better fit of the local LTI models.

6.4. *Comparison with existing techniques*

To show the advantages of the proposed interpolation method, the following three techniques from the literature are also applied to the 25 local LTI models:

- the interpolation of the poles and zeros, proposed in [31];
- the interpolation of the controllable form, proposed in [27];
- the interpolation of internally balanced realizations, proposed in [28].

The resulting LPV models from these three techniques are validated and compared to the polynomial interpolating LPV model $H_P(\phi(t))$ and the affine interpolating LPV model $H_A(\phi(t))$ obtained in Section 6.2.

6.4.1. *Interpolation of the poles and zeros*

For systems depending on a single scheduling parameter, the technique presented in [31] proposes a polynomial interpolation of the real and imaginary part of the poles and zeros of the local LTI models. Since this electromechanical system depends on two scheduling parameters and has a transition from a complex conjugate pair of zeros to a pair of real zeros (as shown in Fig. 6b), it is impossible to apply the technique from [31].

6.4.2. Interpolation of the controllable form

The technique presented in [27] represents the local LTI models in the controllable form and then interpolates the entries of the A and C system matrices. Although the technique presented in [27] only considers dependency on a single scheduling parameter, the technique can be extended, by fitting multivariable polynomials to the entries of the A and C system matrices, to provide a multiparameter-dependent interpolating LPV model. The resulting optimization problem can be posed as a linear least-squares optimization problem.

Figs. 10 and 11 show a validation of the obtained interpolating LPV model, denoted by $H_S(\phi(t))$. Fig. 10 shows the Bode plot of the 25 local LTI models and the Bode plot of the 25 models from the set

$$\mathscr{L}_S = \{H_S(\phi_{e,i})|\phi_{e,i} = [c_{e1,i}\ \ c_{e2,i}]' \quad \text{for } i = 1, \ldots, 25\},$$

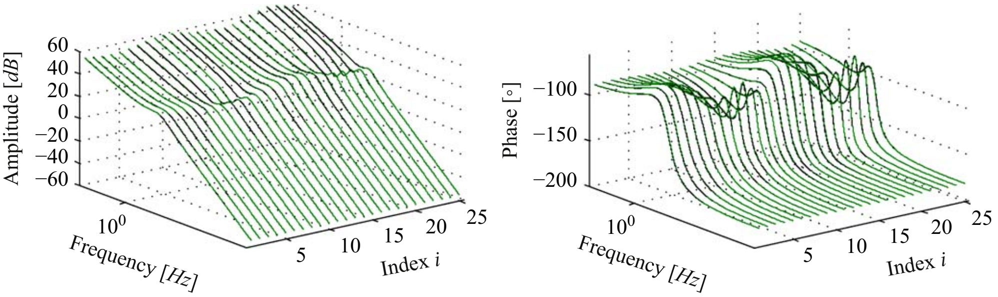

Fig. 10. Bode plot of the models in $\mathscr{L}_S$ (green) compared to the Bode plot of the 25 LTI models obtained from the analytical LPV model (black). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

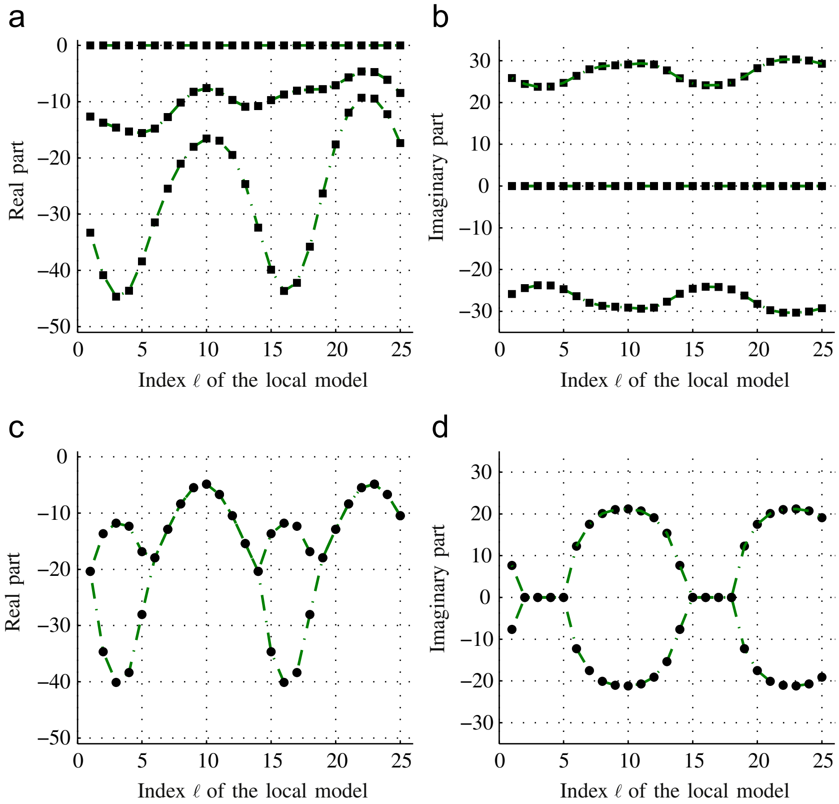

Fig. 11. Poles and zeros of the models in $\mathscr{L}_S$ (green, dash–dotted) and the poles and zeros of the 25 local LTI models (black, squares and circles). (a) Poles. (b) Zeros. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

obtained by evaluating $H_S(\phi(t))$ for the operating conditions (32). Clearly, the models in $\mathscr{L}_S$ smoothly interpolate the LTI dynamics obtained from the analytical LPV model (23). Figs. 11a and b show that the curves for the poles and zeros of the 25 models in the set $\mathscr{L}_S$ accurately fit the curves for the poles and zeros of the 25 LTI models obtained from the analytical LPV model. Therefore, it can be concluded that for this electromechanical system, the technique proposed in [27] results in an interpolating LPV model that smoothly interpolates the local LTI dynamics. Comparing Figs. 10 and 11 for $H_S(\phi(t))$ with Figs. 8 and 9 for $H_P(\phi(t))$, shows that both polynomial interpolating LPV models smoothly interpolate the local system dynamics. Consequently, for this electromechanical system, the proposed interpolation method and the technique of [27] yield similar results.

However, as indicated in [10,27], the technique from [27] suffers from the well-known numerical ill-conditioning of the controllable form [35], which arises for medium to high-order models with realistic data. For the application presented in [10], the local LTI models are restricted to be of order 8. It is concluded by the authors that to fully exploit the possibilities of state-of-the-art LPV control, parameter-dependent behavior that appears at higher frequencies should have been taken into account, which was not possible due to the poor numerical conditioning associated with the applied LPV modeling technique. On the other hand, the proposed interpolation method based on the division of the local LTI models in a series connection of first and second order submodels does not pose this problem. Thus, the size of the system is immaterial for the proposed technique, as the poles and zeros can be calculated very efficiently.

6.4.3. *Interpolation of internally balanced realizations*

The technique presented in [28] represents the local LTI models in an internally balanced state-space realization and then interpolates the entries of the system matrices. The internally balanced realization, implemented in Matlab as `balreal`, uses an eigenvalue decomposition of the product of the observability and controllability Gramians to construct the balancing transformation matrix that contains the corresponding eigenvectors (see [30] for more details). As stated in [28], the most interesting property of this balanced realization, with respect to the interpolation of local LTI models, is its uniqueness property (up to a sign change). In [28], three situations are defined.

- If the entries of the system matrices of the local LTI models show a smooth variation after the balancing transformations have been applied, then the interpolating LPV model can be obtained by interpolating these entries.
- If, on the other hand, the entries of the system matrices show abrupt sign changes, the eigensystem of the product of the Gramians needs to be checked.
  - If the eigenvalues are distinct, a so-called "mild" nonuniqueness is occurring, which can be corrected by changing the sign of the corresponding eigenvector.
  - If there are repeated eigenvalues, the local LTI model should be momentarily neglected and left out of the interpolation.

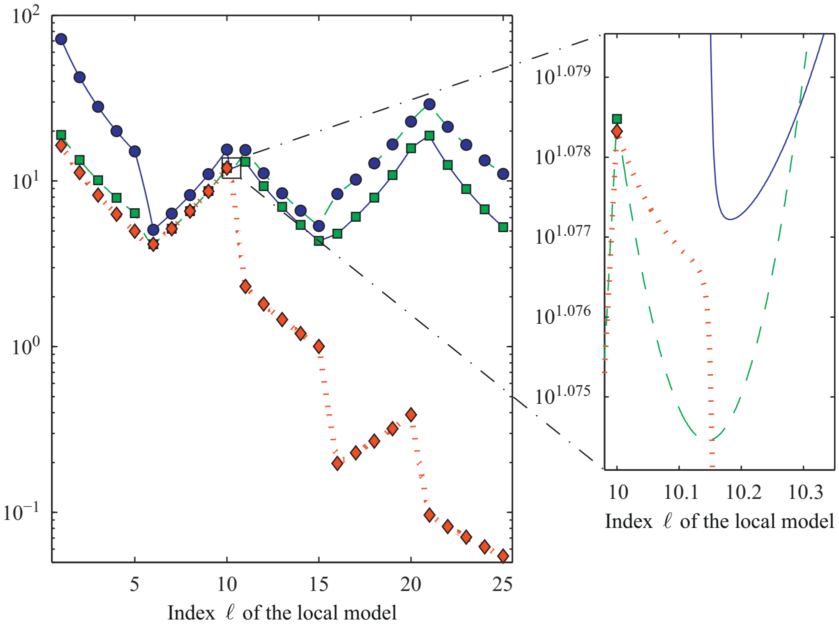

Fig. 12. The three eigenvalues of the product of the Gramians of the local LTI models (circles, squares and diamonds) and of the analytic LPV model (solid, dashed and dotted lines).

Using the Matlab command `balreal`, the balancing transformations for the 25 local LTI models of the electromechanical system are computed. Checking the variation of the entries of the system matrices, after these transformations have been applied, reveals that there are many abrupt changes. Therefore, the uniqueness of the eigenvalues of the product of the Gramians needs to be checked. Since all local LTI models are fourth order, the product of the Gramians of all local LTI models should have four eigenvalues. However, due to the pole at 0 in each local LTI model, `balreal` only computes three eigenvalues and sorts them in descending order. For the 25 local LTI models, these three eigenvalues are shown in Fig. 12 as a function of the index ℓ of the local model. In this figure, the largest eigenvalue is indicated with blue circles, the middle eigenvalue with green squares and the smallest with red diamonds. For each local LTI model, the eigenvalues are distinct. Therefore, according to [28], the sign changes can be fixed. Note, however, that for $5 \leq \ell \leq 10$, the green squares and red diamonds almost coincide.

Fig. 13 shows the entries of the A-matrices of the local LTI models after the balancing transformations have been applied *and* the sign changes have been fixed. The entries of the first row and column are zero (due to the pole at 0 in all local LTI models), but all other elements clearly show a nonsmooth variation with abrupt changes. The reason for this is visualized by the solid, dashed and dotted line in Fig. 12: these lines are calculated by evaluating the analytic LPV model (23) of the electromechanical system for 24 000 values of the scheduling parameters (1000 equidistantly values between each two consecutive pairs $(c_{1,\ell}, c_{2,\ell})$ and $(c_{1,\ell+1}, c_{2,\ell+1})$, for $\ell = 1, \ldots, 24$) and then calculating the eigenvalues of the product of the Gramians of these models. Using this fine grid, it is possible to analyze the variation of the eigenvalues of the product of the Gramians of the LPV model (23). Fig. 12 shows that the three lines cross each other between $\ell = 10$ and 11 (see magnification on the right in Fig. 12). As a result, the blue circles and green squares, indicating the two largest eigenvalues of the local LTI models are sorted in the wrong way for $\ell \geq 10$. At $\ell = 10$ Fig. 13 shows the abrupt changes in the elements of the A-matrices. Therefore, it can be concluded that a straightforward interpolation of the balanced realizations of the local LTI models is not possible without a correct sorting of the eigenvalues of the product of the Gramians. Based on

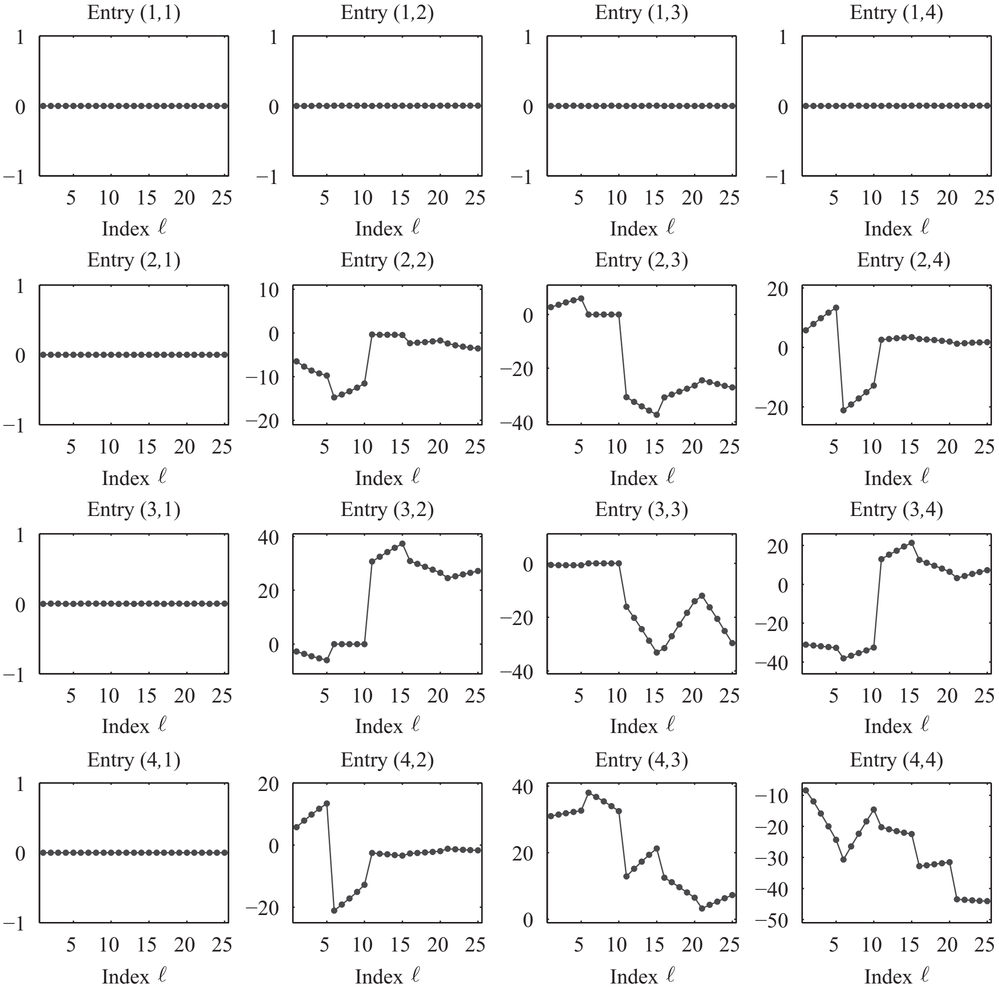

Fig. 13. Entries of the A-matrices after the balancing transformation.

the local LTI models, it is unclear how this sorting should be performed. Indeed, for the electromechanical system, the true variation is only clear after evaluating the analytical LPV model (23) for a very high number of 24 000 values of the scheduling parameters. In practice, it is usually impossible to collect this number of local LTI models from a system.

6.4.4. Implementation issues

The comparison of the proposed interpolation method with the three existing techniques is concluded with a final remark about the implementation of the different techniques.

- *The interpolation of the poles and zeros proposed in* [31].
  This technique requires the sorting of the poles and zeros of the local LTI models, for which an automatic procedure does not exist. However, when the local LTI dynamics show a smooth variation, their poles and zeros will also show a smooth variation and the sorting is straightforward. The other steps in the technique can be automated.
- *The interpolation of the controllable form proposed in* [27].
  This technique can be completely automated.
- *The interpolation of internally balanced realizations proposed in* [28].
  This technique requires the sorting of the eigenvalues of the product of the Gramians, for which an automatic procedure does not exist. Moreover, a smooth variation of the local LTI dynamics, does not necessarily result in a straightforward sorting of the eigenvalues, as shown for the electromechanical system. This is a drawback of this technique. An automatic implementation to correct the sign changes of the eigenvectors, to construct the balancing transformation matrices might be possible.
- *The proposed interpolation method.*
  The technique also requires the sorting of the poles and zeros, which cannot be automated. All other steps can be automated.

7. Conclusions

In this paper a new method is developed to obtain numerically well-conditioned SISO LPV models for systems depending on multiple scheduling parameters. The new method is based on the interpolation of a set of local LTI models that are obtained for fixed operating conditions of the system. Starting from a general parameterization, both polynomial and affine LPV models can be obtained. While the parameterization of the polynomial LPV model offers more degrees of freedom to yield a more accurate interpolation, the affine LPV model can be more interesting from a control point of view, since many LPV control design techniques are based on affine and polytopic LPV models. The application of the proposed interpolation method on an electromechanical system depending on two scheduling parameters and a comparison with three existing LPV identification techniques from the literature clearly shows the benefits of the proposed interpolation method. The extension to the interpolation of MIMO local LTI models is the topic of current research.

Acknowledgments

The authors are very grateful to Prof. Eurípedes G.O. Nóbrega for providing the numerical data of the electromechanical system. Juan F. Camino is supported through grants from CAPES and FAPESP. Jan De Caigny and Jan Swevers are supported through the following funding: Project G.0446.06 of the Research Foundation—Flanders (FWO—Vlaanderen), K.U.Leuven—BOF EF/05/006 Center-of-Excellence Optimization in Engineering and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its author(s).

Appendix A. Calculating the gain of an LTI model

Calculate the following series connection of the local LTI submodels, *without* addition of the gain K_ℓ,

$$\bar{\mathsf{H}}_\ell := \prod_{\tau=1}^{\tau_1+\tau_2} \mathsf{H}_\ell^\tau$$

and denote the system matrices of the obtained LTI model as

$$\bar{\mathsf{H}}_\ell := \left[\begin{array}{c|c} \bar{\mathsf{A}}_\ell & \bar{\mathsf{B}}_\ell \\ \hline \bar{\mathsf{C}}_\ell & \bar{\mathsf{D}}_\ell \end{array}\right].$$

Comparing $\bar{\mathsf{H}}_\ell$ with (4) yields

$$\mathsf{H}_\ell = \mathsf{K}_\ell \bar{\mathsf{H}}_\ell. \tag{33}$$

Since both H_ℓ and $\bar{\mathsf{H}}_\ell$ are SISO LTI models, it is clear that (33) implies the following relation between the transfer function of both LTI models

$$\mathsf{C}_\ell(qI-\mathsf{A}_\ell)^{-1}\mathsf{B}_\ell+\mathsf{D}_\ell=\mathsf{K}_\ell(\bar{\mathsf{C}}_\ell(qI-\bar{\mathsf{A}}_\ell)^{-1}\bar{\mathsf{B}}_\ell+\bar{\mathsf{D}}_\ell).$$

Since the matrices A_ℓ, B_ℓ, C_ℓ, D_ℓ, $\bar{\mathsf{A}}_\ell$, $\bar{\mathsf{B}}_\ell$, $\bar{\mathsf{C}}_\ell$ and $\bar{\mathsf{D}}_\ell$ are given, the gain K_ℓ can be calculated in a straightforward manner by evaluating this expression for some value of q that makes the determinants $|qI-\mathsf{A}_\ell|\neq 0$ and $|qI-\bar{\mathsf{A}}_\ell|\neq 0$.

References

[1] A. Packard, Gain scheduling via linear fractional transformations, Syst. Control Lett. 22 (2) (1994) 79–92.

[2] P. Apkarian, P. Gahinet, A convex characterization of gain-scheduled $\mathscr{H}_\infty$ controllers, IEEE Trans. Autom. Control 40 (5) (1995) 853–864.

[3] W.J. Rugh, J.S. Shamma, Research on gain scheduling, Automatica 36 (10) (2000) 1401–1425.

[4] D.J. Leith, W.E. Leithead, Survey of gain-scheduling analysis and design, Int. J. Control 73 (11) (2000) 1001–1025.

[5] J.S. Shamma, M. Athans, Guaranteed properties of gain scheduled control for linear parameter-varying plants, Automatica 27 (3) (1991) 559–564.

[6] C.W. Scherer, LPV control and full block multipliers, Automatica 37 (3) (2001) 361–375.

[7] J. Daafouz, J. Bernussou, Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties, Syst. Control Lett. 43 (5) (2001) 355–359.

[8] V.J.S. Leite, P.L.D. Peres, Robust control through piecewise Lyapunov functions for discrete time-varying uncertain systems, Int. J. Control 77 (3) (2004) 230–238.

[9] H. Karimi, P. Jabedar, B. Lohmann, B. Moshiri, $\mathscr{H}_\infty$ control of parameter-dependent state-delayed systems using polynomial parameter-dependent quadratic functions, Int. J. Control 78 (4) (2005) 254–263.

[10] M.G. Wassink, M.V. de Wal, C. Scherer, O. Bosgra, LPV control for a wafer stage: beyond the theoretical solution, Control Eng. Pract. 13 (2) (2005) 231–245.

[11] F. Amato, M. Mattei, A. Pironti, Gain scheduled control for discrete-time systems depending on bounded rate parameters, Int. J. Robust Nonlinear Control 15 (2005) 473–494.

[12] W. Xie, $\mathscr{H}_2$ gain scheduled state feedback for LPV system with new LMI formulation, IEE Proc.—Control Theory Appl. 152 (6) (2005) 693–697.

[13] C.E. de Souza, A. Trofino, Gain-scheduled $\mathscr{H}_2$ controller synthesis for linear parameter varying systems via parameter-dependent Lyapunov functions, Int. J. Robust Nonlinear Control 16 (5) (2006) 243–257.

[14] V.F. Montagner, R.C.L.F. Oliveira, P.L.D. Peres, P.-A. Bliman, Linear matrix inequality characterisation for $\mathscr{H}_\infty$ and $\mathscr{H}_2$ guaranteed cost gain-scheduling quadratic stabilisation of linear time-varying polytopic systems, IET Control Theory Appl. 1 (6) (2007) 1726–1735.

[15] J. De Caigny, J.F. Camino, R.C.L.F. Oliveira, P.L.D. Peres, J. Swevers, Gain-scheduled $\mathscr{H}_\infty$-control of discrete-time polytopic time-varying systems, in: Proceedings of the 47th IEEE Conference on Decision and Control, Cancun, Mexico, December 2008, pp. 3872–3877.

[16] J. De Caigny, J.F. Camino, R.C.L.F. Oliveira, P.L.D. Peres, J. Swevers, Gain-scheduled $\mathscr{H}_2$ and $\mathscr{H}_\infty$ control of discrete-time polytopic time-varying systems, IET Control Theory Appl. 2009, to appear.

[17] F. Wu, K. Dong, Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions, Automatica 42 (1) (2006) 39–50.

[18] M. Nemani, R. Ravikanth, B.A. Bamieh, Identification of linear parametrically varying systems, in: Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, LA, USA, December 1995, pp. 2990–2995.

[19] L.H. Lee, K. Poolla, Identification of linear parameter-varying systems using nonlinear programming, J. Dyn. Syst. Meas. Control 121 (1) (1999) 71–78.

[20] V. Verdult, M. Verhaegen, Subspace identification of multivariable linear parameter-varying systems, Automatica 38 (5) (2002) 805–814.

[21] V. Verdult, M. Verhaegen, Kernel methods for subspace identification of multivariable LPV and bilinear systems, Automatica 41 (9) (2005) 1557–1565.

[22] F. Felici, J.W. van Wingerden, M. Verhaegen, Subspace identification of MIMO LPV systems using a periodic scheduling sequence, Automatica 43 (10) (2007) 1684–1697.

[23] B. Bamieh, L. Giarre, Identification of linear parameter varying models, Int. J. Robust Nonlinear Control 12 (9) (2002) 841–853.

[24] R. Tóth, F. Felici, P.S.C. Heuberger, P.M.J.V. den Hof, Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation, in: Proceedings of the 2007 European Control Conference, Kos, Greece, July 2007, pp. 5418–5425.

[25] G.J. Balas, Linear, parameter-varying control and its application to a turbofan engine, Int. J. Robust Nonlinear Control 12 (2002) 763–796.

[26] J. De Caigny, J.F. Camino, B. Paijmans, J. Swevers, An application of interpolating gain-scheduling control, in: Proceedings of the Third IFAC Symposium on Systems, Structure and Control (SSSC07), Foz do Iguassu, Brazil, October 2007 (cdrom).

[27] M. Steinbuch, R. van de Molengraft, A. der Voort, Experimental modelling and LPV control of a motion system, in: Proceedings of the 2003 American Control Conference, Denver, CO, USA, 2003, pp. 1374–1379.

[28] M. Lovera, G. Mercere, Identification for gain-scheduling: a balanced subspace approach, in: Proceedings of the 2007 American Control Conference, New York, USA, July 2007, pp. 858–863.

[29] B.C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, IEEE Trans. Autom. Control 26 (1) (1981) 17–32.

[30] A. Laub, M. Heath, C. Paige, R. Ward, Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms, IEEE Trans. Autom. Control 32 (2) (1987) 115–122.

[31] B. Paijmans, W. Symens, H.V. Brussel, J. Swevers, Experimental identification of affine LPV models for mechatronic systems with one varying parameter, Eur. J. Control 14 (1) (2008) 16–29.

[32] J.S. Shamma, M. Athans, Gain scheduling: potential hazards and possible remedies, IEEE Control Syst. Mag. 12 (3) (1992) 101–107.

[33] A. Emami-Naeini, P. Van Dooren, Computation of zeros of linear multivariable systems, Automatica 18 (4) (1982) 415–430.

[34] J.J. Moré, The Levenberg–Marquardt algorithm: implementation and theory, in: G.A. Watson (Ed.), Numerical Analysis, Lecture Notes in Mathematics, Springer, Berlin, Heidelberg, vol. 630, 1978, pp. 105–116.

[35] C. Paige, Properties of numerical algorithms related to computing controllability, IEEE Trans. Autom. Control 26 (1) (1981) 130–138.