

# Model order reduction of LPV systems based on parameter varying modal decomposition

István Gőzse, Tamás Luspay, Tamás Péni, Zoltán Szabó and Bálint Vanek

**Abstract**—The model reduction problem of high dimensional Linear Parameter Varying (LPV) systems is addressed in the paper. Modal representation of local systems is computed first for fixed values of the scheduling parameter. Modes are then matched and a smooth quasi-modal form is obtained over the entire parameter domain. Classification of system modes is applied to explore dynamic coherence of the model. Structured parameter-varying Gramians are constructed and used for balancing the model and eliminating negligible components. Numerical example illustrates the effectiveness of the methodology.

## I. INTRODUCTION

The problem of model reduction for Linear Parameter Varying (LPV) systems has been first discussed in [20], [21]. The goal is to reduce the state dimension (and accordingly the complexity of the underlying mathematical model) for parameter-varying systems, with minimal change in the overall input-output behaviour. The approach in [20] and [21] was based on the extension of internally balanced realization for Linear Time Invariant (LTI) systems [12]. The corresponding observability and controllability Gramians are equal and have a diagonal structure, with the singular values of the LPV system in the diagonal. These singular values characterize the contribution of each state-space element in the input-output map of the system, consequently, less significant states can be removed by truncation on the basis of the balanced realization.

In the LPV model reduction there are basically two main challenges to face with. First, the computation of the generalized Gramians for LPV systems leads to an optimization problem with Linear Matrix Inequality (LMI) constraints. Consequently, the method suffers from numerical limitations. An attractive idea to overcome this problem, is to consider the LPV system as a set of LTI models, obtained at different frozen values of the parameter [14], [1], [13], [18]. Numerically well established, linear techniques (such as balanced truncation [1], [18] or Krylov-based projection [14]) can be then applied individually for these local systems. The main drawback of such an approach is the lack of guarantee for state-space consistency of the reduced order models. It may happen that different modal content is preserved for different regions of the scheduling parameter. The offered methods for the connection and interpolation of local reduced-order models mostly neglect this issue.

The second challenge lies in the similarity transformation for LPV systems. In general, it is parameter-varying, hence the resulting transformed system depends not only on the scheduling parameter but also on its derivative. Two remedies are known for this problem: using parameter-independent transformation or neglecting the derivative dependent terms. Both solutions represent conservativeness in the methodology, in addition, dropping the rate-dependent terms represents an unknown uncertainty in the system. Recognizing these issues, a parameter-varying oblique projection based

model-reduction is proposed recently in [17]. The algorithm produces a reduced order model with consistent state-space such that the model does not depend on the derivative of the scheduling parameter. Finally, another aspect, namely, the problem of reducing the dimension of the scheduling variables for LPV systems has been discussed in [7], [15].

The paper proposes a novel reduction technique for LPV system, presented as follows. The forthcoming section II formally states the LPV model-reduction problem. Section III summarizes the methods and concepts, necessary for the construction of the model reduction algorithm. The algorithm combines local techniques with parameter-varying methodologies, where the emphasis is put on preserving the modal consistency throughout the steps. The methodology is discussed in details under Section IV. Numerical results are reported in Section V, which is followed by conclusions and future research directions.

An extended version of the method, which solves some of the raised problems, is readily available and currently under review [10].

## II. PROBLEM FORMULATION

Linear Parameter Varying (LPV) systems are considered in the paper with state space dynamics

$$G(\rho) : \begin{cases} \dot{x} = A(\rho)x + B(\rho)u \\ y = C(\rho)x + D(\rho)u, \end{cases} \quad (1)$$

where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^{n_u}$  and  $y \in \mathbb{R}^{n_y}$ , are the state, input and output vectors, respectively. The matrix functions  $A : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}^{n \times n}$ ,  $B : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}^{n \times n_u}$ ,  $C : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}^{n_y \times n}$ ,  $D : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}^{n_y \times n_u}$  are assumed to be continuous functions of the scheduling parameter vector  $\rho \in \mathbb{R}^{n_\rho}$ .

In this paper we assume that the dimension of  $\rho$  is 1 and both  $\rho(t)$  and  $\dot{\rho}(t)$  are bounded:  $\rho(t) \in [\rho_{min}, \rho_{max}]$  and  $|\dot{\rho}(t)| \leq \delta$  for all  $t$ . Though the concept of the proposed method can be extended to systems having more than one scheduling parameters, the numerical details of the algorithm have to be elaborated.

The LPV system (1) is given by the pair  $(\mathcal{G}, \Gamma)$ , where  $\mathcal{G}$  is a set of LTI models obtained by evaluating (1) at the scheduling parameter values  $\rho_1 = \rho_{min} < \rho_2 < \dots < \rho_K = \rho_{max}$  and  $\Gamma = \{\rho_1, \dots, \rho_K\}$  (grid-based LPV description). Formally,

$$\mathcal{G} = \left\{ G_k \mid G_k = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix}, \begin{matrix} A_k = A(\rho_k), B_k = B(\rho_k), \\ C_k = C(\rho_k), D_k = D(\rho_k) \end{matrix} \right\}.$$

Between the grid points linear interpolation is assumed. (This definition of LPV systems is typical if the model is generated by trimming a nonlinear system at different operating points [11].)

The aim of the LPV model reduction is to find  $G_{red}(\rho, \hat{\rho})$  of order  $n_{red} < n$ , such that exact bound can be derived to the induced  $\mathcal{L}_2$  norm of the LPV difference system  $G(\rho) - G_{red}(\rho, \hat{\rho})$  and  $G_{red}(\rho, \hat{\rho})$  minimizes this bound.

The authors are with the Systems and Control Lab of the Institute for Computer Science and Control, Budapest, 1111 Kende u. 14-17, Hungary. Corresponding author: Tamás Péni, e-mail: peni.tamas@szttaki.mta.hu

### III. METHODS USED TO CONSTRUCT THE PROPOSED ALGORITHM

This section collects all of the methods that are necessary to construct the proposed model reduction algorithm.

#### A. LPV balanced model reduction

Balanced model reduction is one possible method to solve the model reduction problem addressed above. The method is based on balanced realization, which reflects the controllability and observability properties of the given LPV system [20],[21]. To obtain the parameter varying transformation  $T(\rho)$ , which transforms an LPV system into balanced form the observability and controllability gramians, denoted by  $X_o(\rho)$  and  $X_c(\rho)$ , have to be determined first. If the LPV system is given in state-space form, the Gramians can be obtained as a result of the following optimization problem:

$$\min_{X_o(\rho), X_c(\rho)} \int_{\rho_{min}}^{\rho_{max}} \text{trace } X_o(\rho)X_c(\rho) d\rho \quad (2)$$

$$\begin{aligned} \dot{X}_o(\rho) + A(\rho)^T X_o(\rho) + X_o(\rho)A(\rho) + C(\rho)^T C(\rho) &< 0 \\ -\dot{X}_c(\rho) + A(\rho)X_c(\rho) + X_c(\rho)A(\rho)^T + B(\rho)B(\rho)^T &< 0 \\ X_o(\rho) &> 0, X_c(\rho) > 0. \end{aligned}$$

The next step is computing the unique Cholesky factors of  $X_o(\rho)$  and  $X_c(\rho)$  ([20]) as

$$\begin{aligned} X_o(\rho) &= R_o^T(\rho)R_o(\rho), & R_o(\rho) & \text{upper triangular} \\ X_c(\rho) &= R_c(\rho)R_c^T(\rho), & R_c(\rho) & \text{lower triangular} \end{aligned}$$

Then, performing a singular value decomposition on the product  $R_o(\rho)R_c(\rho)$  results in

$$U(\rho)S(\rho)V^T(\rho) = R_o(\rho)R_c(\rho)$$

where  $U(\rho)$  and  $V(\rho)$  are unique up to the sign of the corresponding columns and  $S(\rho)$  is the diagonal matrix of the parameter-dependent singular values  $\sigma_i(\rho)$ , which play the same role as the generalized singular values ([4]) in LTI case. The transformation  $T(\rho)$  is then computed as

$$T(\rho) = R_c(\rho)V(\rho)S^{-\frac{1}{2}}(\rho).$$

Having applied the balancing transformation on (1), the states corresponding to small singular values can be removed. Since  $T(\rho)$  is parameter-dependent, the reduced system explicitly depends on  $\rho$  as well [20]. As for the approximation error, the following bound can be derived:

$$\|G(\rho) - G_{red}(\rho, \dot{\rho})\|_{\infty} \leq 2 \sum_{i=n_{red}+1}^n \sigma_i(\rho)$$

for all  $\rho \in [\rho_{min}, \rho_{max}]$  and  $\dot{\rho} = 0$ , i.e. the approximation error is bounded (at least for frozen scheduling parameter values) by the sum of the singular values corresponding to the neglected states. (If  $\dot{\rho} \neq 0$  the derivation of an upper bound is a more complex task but it is still possible in the possession of the singular value functions  $\sigma_i(\cdot)$  [20].)

Regarding the computational issues the following remarks can be made:

- the infinite number of matrix inequalities in (2) can be relaxed if they are evaluated over only the finite parameter grid  $\Gamma$  and the cost function is replaced by  $\sum_{k=1}^K \text{trace } X_o(\rho_k)X_c(\rho_k)$ ;
- if the parameter dependence of  $X_o(\rho)$  and  $X_c(\rho)$  is a-priori fixed, (e.g. if  $X_o(\rho)$  is sought in the form of  $X_{o,0} + \sum_{i=1}^{n_b} f_i(\rho)X_{o,i}$ , where  $f_i: \mathbb{R} \rightarrow \mathbb{R}$  are a-priori fixed basis functions), then (2) can be converted to an

iterative convex optimization problem. It is iterative, because the cost function is linear in the decision variables only if either  $X_o$  or  $X_c$  is fixed. If one of the two is constant, the problem reduces to a linear optimization problem with Linear Matrix Inequality (LMI) constraints. By alternately fixing  $X_o$  and  $X_c$  a numerically tractable iterative algorithm is obtained;

- the Cholesky factors  $R_o(\rho)$  and  $R_c(\rho)$  can be computed by using the algorithm given in the proof of Theorem 7.8.1. of [20];
- since  $U(\rho)$  and  $V(\rho)$  are unique apart from the sign, therefore by performing the SVD decomposition pointwise, on the matrices  $R_o(\rho_k)R_c(\rho_k)$  then after possible sign-corrections the pointwise  $V(\rho_k)$  matrices can be interpolated.

Regarding the algorithm above further technical issues (e.g. the crossing of  $\sigma_i(\rho)$  trajectories and the continuity of  $T(\rho)$ ) may arise. These issues are discussed in detail in [20].

Although the parameter-varying balanced reduction provides a reliable algorithm to reduce LPV models, it is computationally extremely expensive. If the dimension of the system is higher than 30-40, the LMI optimization problem becomes intractable by the off-the-shelf semidefinite solvers. Therefore, this approach cannot be applied in such applications (e.g. aeroelastic aircraft modeling [13], [3]), where the number of states can be very large (100-200 or even larger).

#### B. Modal form of LTI systems

Assume that the  $A$  matrix of the LTI dynamics  $\dot{x} = Ax + Bu$ ,  $y = Cx + Du$  is diagonalizable. Then there exists a state transformation  $\tilde{T}$  such that the matrix  $\tilde{T}^{-1}A\tilde{T}$  is block diagonal, i.e. the transformed system can be written in the form

$$\dot{x} = \begin{bmatrix} A_1 & 0 & & \\ 0 & A_2 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & A_m \end{bmatrix} x + \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_m \end{bmatrix} u, \quad (3)$$

$$y = [C_1 \ C_2 \ \dots \ C_m] x + Du, \quad (4)$$

where each block of the structured matrix  $\bar{A}$  corresponds to one mode of the system, characterized by the eigenvalues:

$$A_i = \begin{cases} \begin{bmatrix} \Re\{\lambda_i\} & \Im\{\lambda_i\} \\ -\Im\{\lambda_i\} & \Re\{\lambda_i\} \end{bmatrix} & \text{if } \lambda_i \in \mathbb{C} \\ [\lambda_i] & \text{if } \lambda_i \in \mathbb{R} \end{cases} \quad (5)$$

where  $\Re(\lambda_i)$  and  $\Im(\lambda_i)$  denote the real and imaginary part of the complex eigenvalue  $\lambda_i$ . The similarity transformation  $\tilde{T}$  can be constructed from the eigenspace of  $A$  as follows:

$$\tilde{T} = [ \Re\{v_1\} \ \Im\{v_1\} \ \dots \ \Re\{v_m\} \ \Im\{v_m\} ], \quad (6)$$

where  $v_i$  is the eigenvector associated with the eigenvalue  $\lambda_i$ .

#### C. Perfect matching in complete bipartite graphs

Let  $G(\mathcal{A}, \mathcal{B})$  be a complete bipartite graph with vertex sets  $\mathcal{A} = \{a_1, \dots, a_n\}$  and  $\mathcal{B} = \{b_1, \dots, b_n\}$ . Assume the edges are directed and go from  $\mathcal{A}$  to  $\mathcal{B}$ , that is, the edge set is defined as  $\mathcal{E} = \{(a_i, b_j) \mid \forall (i, j) \text{ pairs}\}$ . Let  $c_{ij} \in \mathbb{R}^+$  be a cost assigned to edge  $(a_i, b_j)$ . The perfect match in  $G$  is a subset of  $\mathcal{E}$  defined such that every  $a_i$  has exactly one pair in  $\mathcal{B}$  and every  $b_j$  is a pair of exactly one vertex in  $\mathcal{A}$ . Formally, if  $P$  denotes a permutation of the integer numbers  $1, \dots, n$ , then the perfect match is  $\bar{\mathcal{E}} = \{(a_i, b_{P(i)}) \mid \forall i\}$ , where  $P(i)$  denotes the  $i$ -th element of  $P$ . The cost of a

perfect match is the sum of the costs of the edges in  $\bar{\mathcal{E}}$ . If  $G$  is given, the perfect match with minimum cost can be found in polynomial time by using the Hungarian method, that is also known as Kuhn-Munkres algorithm [9].

#### D. Hierarchical clustering

Hierarchical clustering is a well-known method to group data by creating a hierarchy of clusters, also known as cluster tree [6]. In agglomerative clustering each object starts from its own cluster and pairs of clusters are merged. In order to do so, first, an appropriate metric, denoted by  $d(a, b)$ , is necessary, which measures the distance between two objects  $a$  and  $b$  in the data set. The metric is chosen to measure the similarity between the objects, in the given nature of interest. Secondly, the  $L(\mathcal{A}, \mathcal{B}, d(\cdot, \cdot))$  linkage function is introduced, which characterizes the distance between two sets of objects ( $\mathcal{A}$  and  $\mathcal{B}$ ) as a function of the pairwise distances between the objects (i.e. based on  $d(a, b)$ ;  $a \in \mathcal{A}$ ,  $b \in \mathcal{B}$ ). Finally, partitioning of the data takes place, where the obtained cluster tree is cut into groups. The number of clusters generated actually is either automatically determined by the clustering algorithm or it is specified a-priori by the user.

#### E. Hyperbolic distance

The hyperbolic geometry [5], [2] using the Poincaré disk model is a useful tool for the analysis and identification of dynamical systems [16], [19]. In this model the plane is the unit disk and points are Euclidean ones, which connects this mathematical structure to discrete time systems in control theory. Imaginary axis of the  $s$ -domain is mapped to the unit circle of the  $z$ -domain and left half-plane poles and zeros transformed to the interior of the circle, preserving stability. Therefore, stable, discrete-time systems can be naturally investigated in a hyperbolic setting [16]. Application of the hyperbolic distance for unstable poles is also possible by appropriately mapping them inside the unit circle. Details can be found in [10].

In the Poincaré model the hyperbolic distance between two points  $\gamma_1 \in \mathbb{C}$  and  $\gamma_2 \in \mathbb{C}$  is defined as:

$$h(\gamma_1, \gamma_2) = 2 \operatorname{arctanh}^{-1} \frac{|\gamma_1 - \gamma_2|}{|1 - \gamma_2 \bar{\gamma}_1|}. \quad (7)$$

The above defined distance is in fact a useful metric to compare two complex points in terms of the dynamical behaviour they determine as poles [2]. For stable systems, the transient response is determined by the location, as well as, the real and imaginary parts of the system poles. This property is reflected by the hyperbolic distance within the unit disk; small changes towards the unit circle correspond to large variation in the hyperbolic distance.

A simple example is given here to illustrate the aforementioned properties of the hyperbolic approach. Consider a second order, strictly proper SISO discrete transfer function  $G_0(z)$  with complex poles of  $p_0 = 0.9e^{\pm i\frac{\pi}{8}}$  inside the unit disk and static gain equal to 1. Perturbing  $|p_0|$  by  $\pm 10\%$  yields  $G_1(z)$  and  $G_2(z)$  with poles  $p_1 = 0.99e^{\pm i\frac{\pi}{8}}$  and  $p_2 = 0.81e^{\pm i\frac{\pi}{8}}$ , respectively. The Euclidean distance of the perturbed poles from the nominal ones are equally 0.09, hence they cannot be distinguished. However, the impulse responses of the three systems differ greatly, as depicted in Fig. 1. At the same time, the hyperbolic distance between  $p_0$  and  $p_1$  is 2.3489, compared to the value of 0.6904 between  $p_0$  and  $p_2$ . This indicates more dynamical similarity between  $G_0$  and  $G_2$ , which is expected by observing the settling times and overshoots in Figure 1.

The hyperbolic distance of two stable, continuous time eigenvalues (poles) can be defined by transforming them into

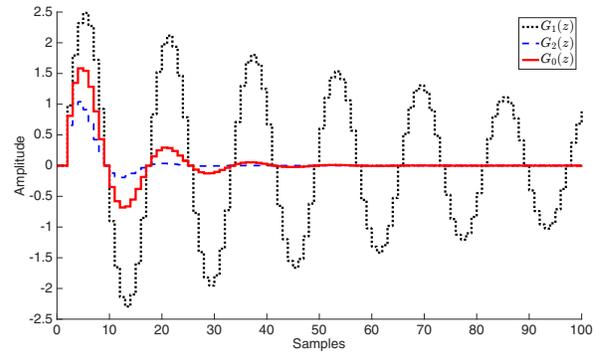


Fig. 1. Impulse responses of  $G_1(z)$ ,  $G_2(z)$  and  $G_N(z)$  systems

the unit disk and applying (7). The transformation can be the mapping  $\exp(\lambda T_s)$ , where  $\lambda$  lies on the open left half plane and  $T_s$  is a suitable small value. This transformation corresponds to computing the discrete time counterpart of  $\lambda$  at sampling time  $T_s$ . For notational convenience we denote the continuous-time version of (7) as follows:

$$h_c(\lambda_1, \lambda_2, T_s) = h(\exp(\lambda_1 T_s), \exp(\lambda_2 T_s)) \quad (8)$$

#### IV. ALGORITHM FOR MODEL ORDER REDUCTION OF LPV SYSTEMS

This section comprises the main contribution of the paper as an algorithm for model reduction of LPV systems. The main steps are detailed below.

**Input data.** The algorithm starts from the pair  $(\mathcal{G}, \Gamma)$  representing the LPV system to be reduced. To simplify the discussion, we make the following assumptions: each  $A_k$  is stable, diagonalizable and its every eigenvalue has multiplicity 1. (These restrictions are rather technical and most of them can be relaxed. The details can be found in the discussion of the the steps of the algorithm.)

**Step 1.** The first step is to compute the eigenvalue decomposition of each  $A_k$ . So let  $V_k^{-1} A_k V_k = \Lambda_k$ , where  $\Lambda_k = \text{blockdiag}(\lambda_{k,1}, \dots, \lambda_{k,n})$  collects the eigenvalues and  $V_k$  stores the eigenvectors of  $A_k$ . In order that the  $\Lambda_k$  matrices be consistent, the ordering of the eigenvalues have to be modified in each grid point such that for any  $1 \leq i \leq n$ , the sequence  $\lambda_{i,1}, \dots, \lambda_{i,K}$  corresponds to  $\lambda_k(\rho_1), \dots, \lambda_k(\rho_K)$ , where  $\lambda_k(\rho)$  is the  $k$ -th parameter dependent eigenvalue of  $A(\rho)$ . To find the right ordering of the eigenvalues we formulate the problem as a perfect matching problem in complete bipartite graphs. For this, let for all  $k < K$ , the vertex sets  $\mathcal{A}$  and  $\mathcal{B}$  be constructed from the eigenvalues as follows:  $\mathcal{A} := \{\lambda_{k,1}, \dots, \lambda_{k,n}\}$  and  $\mathcal{B} := \{\lambda_{k+1,1}, \dots, \lambda_{k+1,n}\}$ . Define now directed edges from every  $\lambda_{k,i}$  to every  $\lambda_{k+1,j}$  with cost  $h_c(\lambda_{k,i}, \lambda_{k+1,j}, T_s)$ . By using this bipartite graph, the eigenvalue matching problem can be formulated as seeking a perfect matching with minimal cost. Based on subsection III-C, this problem can be efficiently solved by the Hungarian method. Of course, if at some  $k$  the entries of  $\Lambda_k$  are reordered, the eigenvectors in  $V_k$  have to be consistently reordered as well.

*Remark 1:* The eigenvalue matching problem can be extended for unstable systems by consistently mapping the eigenvalues inside the unit circle. The technical details are omitted here, but can be found in an extended version of the paper in [10].

**Step 2.** Our aim is to use the spectral decomposition of the  $A_k$  matrices to construct a parameter-dependent transformation that decouples (at least approximately) the LPV dynamics into modal form. For this, a set of local modal transformations  $\tilde{T}_k$  will be constructed from the eigenvectors  $V_k$  and they will be interpolated to give a parameter-dependent state transformation. In order to make the smooth interpolation possible, the eigenvectors  $V_k$  are corrected in this step. The correction aims to transform  $v_{k+1,j}$  (which is the  $j$ -th column of  $V_{k+1}$ ) close to  $v_{k,j}$  such that the eigenvector property of  $v_{k+1,j}$  is preserved. If the multiplicity of each eigenvalue is 1 the correction can be performed easily by multiplying  $v_{k+1,j}$  by the (complex) number

$$\alpha_{k,j}^* := \frac{v_{k,j}^T v_{k+1,j}}{\|v_{k,j}\| \|v_{k+1,j}\|}.$$

It is easy to check that  $\alpha_{k,j}^*$  is the minimizer of the difference  $\|v_{k,j} - \alpha_{k,j}^* v_{k+1,j}\|$ . It is important to note, if there is an eigenvalue with multiplicity  $m > 1$ , but the diagonalisability of the  $A_k$  matrices holds true, the correction leads to a complex-valued Procrustes problem between the corresponding  $m$ -dimensional eigenspaces (This extension is also discussed in details in [10]).

The consistent eigenvectors can be used to construct in each grid point  $k$  the modal transformations  $\tilde{T}_k$  by using (6). Due to the corrections above the entries of  $\tilde{T}_k$  can be smoothly interpolated along the parameter grid. By performing this interpolation, the parameter dependent transformation  $\tilde{T}(\rho)$  is obtained.

*Remark 2:* It can be proved if  $A(\rho)$  depends continuously on  $\rho$  and all of its eigenvalues have multiplicity 1, then the eigenvectors of  $A(\rho)$  are also continuous in  $\rho$  [8]. This result guarantees that after the correction above the eigenvectors and thus the local  $\tilde{T}_k$  transformations can be interpolated smoothly. In case of multiple eigenvalues, the smoothness requires further technical conditions, that have to be checked individually in the actual problem [8].

Having constructed  $\tilde{T}(\rho)$ , a new state vector  $\xi$  can be defined such that  $\tilde{T}(\rho)\xi = x$ , the original LPV system (1) transforms into

$$\begin{aligned} \dot{\xi} &= \left( \tilde{T}^{-1}(\rho)A(\rho)\tilde{T}(\rho) + E(\rho)\dot{\rho} \right) \xi + \tilde{T}^{-1}(\rho)B(\rho)u, \\ y &= C(\rho)\tilde{T}(\rho)\xi + D(\rho)u. \end{aligned} \quad (9)$$

where  $E(\rho) = -\tilde{T}^{-1}(\rho)\frac{\partial\tilde{T}(\rho)}{\partial\rho}$ . By the construction of  $\tilde{T}(\rho)$  the transformed system matrices (apart from the  $\dot{\rho}$ -dependent  $E(\rho)\dot{\rho}x$  term) define a decoupled, modal LPV system, that can be given by the pair  $(\tilde{\mathcal{G}}, \Gamma)$ , where

$$\begin{aligned} \tilde{\mathcal{G}} &= \left\{ \tilde{G}_k \mid \tilde{G}_k = \begin{bmatrix} \tilde{A}_k & \tilde{B}_k \\ C_k & D_k \end{bmatrix} \right\} \\ \tilde{A}_k &= \tilde{T}^{-1}(\rho_k)A(\rho_k)\tilde{T}(\rho_k) = \tilde{T}_k^{-1}A_k\tilde{T}_k \\ \tilde{B}_k &= \tilde{T}^{-1}(\rho_k)B(\rho_k) = \tilde{T}_k^{-1}B_k \\ \tilde{C}_k &= C(\rho_k)\tilde{T}(\rho_k) = C_k\tilde{T}_k \end{aligned}$$

Completed with the  $\dot{\rho}$ -dependent term, the LPV system can be formally defined by the triplet  $(\mathcal{E}, \tilde{\mathcal{G}}, \Gamma)$ , where  $\mathcal{E} = \{E_k \mid E_k = E(\rho_k), \rho_k \in \Gamma\}$ . Three cases are distinguished now, depending on how  $E(\rho)\dot{\rho}$  influences the dynamical behavior of the modal system:

- 1) If  $E(\rho)\dot{\rho}$  generates significant off-diagonal terms the modal decomposition does not provide any advantage.
- 2) If  $E(\rho)\dot{\rho}$  is less significant, but still not negligible the modal transformation  $\tilde{T}(\rho)$  does not decouple the LPV

dynamics, but makes it block-diagonally dominant<sup>1</sup>. This property can be exploited later in balanced reduction.

- 3) There are practical cases, when the effect of  $E(\rho)\dot{\rho}$  is so small, that it can be neglected without the input/output behavior significantly changing. In this case, only the fully decoupled, modal system  $(\tilde{\mathcal{G}}, \Gamma)$  has to be preserved. The decoupled, modal structure enables us to form subsystems from the modes and reduce them independently.

In order to proceed we assume that item 2 or 3 holds, i.e. the modal transformation transforms the LPV system at least in block-diagonally dominant form.

*Remark 3:* Note that, similar  $\dot{\rho}$ -dependent terms appear at other model reduction methods as well [18]. The most LPV model reduction algorithms working on local LTI models start with local model reduction, i.e. the local LTI models  $G_k \in \mathcal{G}$  are independently reduced by using some LTI reduction method, e.g. balanced truncation. The reduction procedure involves at each grid point a different state (e.g. balancing) transformation. Note that, on the LPV level this corresponds to an incorrectly applied parameter varying transformation, since the  $\dot{\rho}$ -dependent term is neglected.

**Step 3.** The main idea in this step is to group the dynamical modes (real eigenvalues and complex eigenvalue pairs) into clusters, based on their dynamical behaviour. If item 3 of Step 2 holds, then the LPV subsystems defined by the clusters can be independently reduced by LPV balanced reduction (subsection III-A). If item 2 holds the clusters still carry information on the structure of the system, which can be exploited by seeking structured Gramians when (2) is solved. To construct the clusters the hierarchical clustering algorithm described in subsection III-D is performed. The considered individual objects are the (locally obtained and matched) eigenvalue sequences  $o_i := (\lambda_{i,1}, \dots, \lambda_{i,K})$ , ( $i = 1, \dots, n$ ). The distance metric  $d(\cdot, \cdot)$  between two objects are defined as follows:

$$d(o_i, o_j) = \min \left( \max_k h_c(\lambda_{i,k}, \lambda_{j,k}, T_s), \max_k h_c(\lambda_{i,k}, \bar{\lambda}_{j,k}, T_s) \right). \quad (10)$$

$d(o_i, o_j)$  is thus simply the maximum of the point-wise hyperbolic distances computed between the eigenvalue pairs corresponding to the same scheduling parameter value. The additional inclusion of the complex conjugate of  $o_j$  ensures the grouping of complex conjugate pairs into the same cluster. A maximum (also known as complete) linkage function has been selected to measure distance between two sets of objects, i.e.:

$$L(\mathcal{A}, \mathcal{B}, d(\cdot, \cdot)) = \{\max(d(o_i, o_j)), o_i \in \mathcal{A}, o_j \in \mathcal{B}\}. \quad (11)$$

The defined metric and linkage function renders a cluster tree to the system. Based on this information,  $M$  groups are formed from modes with similar dynamical behaviour.

*Remark 4:* In this step, we used only the hyperbolic metric to compare the modes, i.e. the modes were clustered based only on the location of the eigenvalues. It is also possible to extend the distance metric (10) with residuum (direction) information provided by the  $\tilde{B}_k$  and  $\tilde{C}_k$  matrices [18]. The improvement of the distance metric is part of the

<sup>1</sup>Here block-diagonal dominance is not a rigorous mathematical notion, it expresses only the fact that the blocks on the main diagonal are much more significant compared to the other entries of the matrix.

future work.

**Step 4.** The last step of model reduction is performing the LPV balanced truncation (subsection III-A) on the structured model. If the LPV system could be decoupled (item 3 in Step 2), then the balanced reduction can be performed on each cluster independently. If the system is only approximately decoupled, but it is block-diagonally dominant (item 2 in Step 2), then the balanced reduction algorithm has to be run on the full order model, but the block-diagonal dominant structure makes it possible to choose structured Gramians in (2). This highly reduces the number of free variables and thus results in smaller computational times. Formally, let  $\Pi$  be the permutation matrix that reorders the states of  $(\mathcal{E}, \hat{\mathcal{G}}, \Gamma)$  according to the clusters found in the previous step. Then we can define the clustered LPV model by the triplet  $(\hat{\mathcal{E}}, \hat{\mathcal{G}}, \Gamma)$ , where

$$\begin{aligned}\hat{\mathcal{E}} &= \{\hat{E}_k \mid \hat{E}_k = \Pi^T E_k \Pi\}, \\ \hat{\mathcal{G}} &= \left\{ \hat{G}_k \mid \hat{G}_k = \begin{bmatrix} \hat{A}_k & \hat{B}_k \\ \hat{C}_k & D_k \end{bmatrix} \right\}, \\ \hat{A}_k &= \Pi^T \tilde{A}_k \Pi = \text{block-diag}(\hat{A}_{k,1}, \hat{A}_{k,2}, \dots, \hat{A}_{k,M}), \\ \hat{B}_k &= \Pi^T \tilde{B}_k = [\hat{B}_{k,1}^T \dots \hat{B}_{k,M}^T]^T, \\ \hat{C}_k &= \tilde{C}_k \Pi = [\hat{C}_{k,1} \dots \hat{C}_{k,M}].\end{aligned}$$

The dimension of each block equals to the size of the corresponding cluster. It is reasonable to choose  $X_o(\rho)$ ,  $X_c(\rho)$  in the same structure as the model has, i.e.

$$\begin{aligned}X_o(\rho) &:= \text{block-diag}(X_{o,1}(\rho), \dots, X_{o,M}(\rho)), \\ X_c(\rho) &:= \text{block-diag}(X_{c,1}(\rho), \dots, X_{c,M}(\rho))\end{aligned}$$

and to perform the optimization (2) with these structured variables. The optimization problem that is actually solved can be given as follows:

$$\begin{aligned}\min_{X_{o,i}, X_{c,i}} \sum_{k=1}^K \text{trace } X_o(\rho_k) X_c(\rho_k) \\ \frac{dX_o(\rho)}{d\rho} s + (\hat{A}_k + \hat{E}_k s)^T X_o(\rho_k) + \dots \\ X_o(\rho_k) (\hat{A}_k + \hat{E}_k s) + \hat{C}_k^T \hat{C}_k < 0 \\ -\frac{dX_c(\rho)}{d\rho} s + (\hat{A}_k + \hat{E}_k s) X_c(\rho) + \dots \\ X_c(\rho) (\hat{A}_k + \hat{E}_k s)^T + \hat{B}_k \hat{B}_k^T < 0 \\ X_o(\rho_k) \succ 0, X_c(\rho_k) \succ 0, \forall \rho_k \in \Gamma \text{ and } s \in \{-\delta, \delta\}\end{aligned} \quad (12)$$

where  $X_o(\rho) = X_{o,0} + \sum_{i=1}^{n_{bo}} f_i(\rho) X_{o,i}$  and  $X_c(\rho) = X_{c,0} + \sum_{i=1}^{n_{bc}} g_i(\rho) X_{c,i}$  with a-priori selected basis functions  $f_i$  and  $g_i$ . Note that the conservativeness of the solution of (12) is influenced also by the terms  $\hat{C}_k^T \hat{C}_k$  and  $\hat{B}_k \hat{B}_k^T$ , because they can destroy the block diagonally dominant structure of the LMIs. But as  $\hat{A}_k$  are block diagonal and the effect of  $\hat{E}_k$  is relatively small, it makes sense anyway to expect a less conservative, reasonable result.

## V. NUMERICAL EXAMPLE

The proposed algorithm has been tested on a 20 dimensional benchmark example generated in the following way. First, a random SISO LTI system was created with 7 complex eigenvalue pairs and 6 real eigenvalues and was transformed to modal form. Then eigenvalue trajectories were constructed and assigned to 6 eigenvalues. These trajectories define how

the corresponding eigenvalue depends on the scheduling variable  $\rho$ . Finally a parameter-dependent transformation was applied on the system matrices. The final LPV model can be given in state space form as follows:

$$\begin{aligned}\dot{x} &= A(\rho)x + B(\rho)u \\ y &= C(\rho)x + Du\end{aligned}$$

where  $u, y \in \mathbb{R}$ ,  $x \in \mathbb{R}^{20}$  and  $\rho : \mathbb{R} \rightarrow \mathbb{R}$ ,  $\rho(t) \in [0, 1]$ ,  $|\dot{\rho}(t)| \leq 0.1$  for all  $t$ . From the LPV model, 50 LTI systems  $G_1, G_2, \dots, G_{50}$  were generated by evaluating the state matrices over the parameter grid  $\Gamma = \{0, \frac{1}{49}, \frac{2}{49}, \dots, 1\}$ . The eigenvalue trajectories of the systems are plotted in Fig. 2. The model reduction started from the LPV model represented by the pair  $(\mathcal{G}, \Gamma)$ , where  $\mathcal{G} = \{G_k, k = 1, \dots, 50\}$ .

Having performed the first step of the model reduction algorithm (eigenvalue decomposition and matching<sup>2</sup>) Fig. 2 was obtained, which shows that the algorithm managed to find the correct pair of each eigenvalue in the consecutive  $\rho$  values (see the zoomed part in the upper right corner). After rearranging the eigenvectors to be aligned with the matched eigenvalues the local modal transformations were constructed. From these, a parameter varying  $\tilde{T}(\rho)$  transformation was generated by computing the cubic spline interpolant of each entry. In the possession of  $\tilde{T}(\rho)$  the  $\rho$ -dependent term could also be computed as  $E(\rho)\dot{\rho} = -\tilde{T}^{-1}(\rho) \frac{d\tilde{T}(\rho)}{d\rho} \dot{\rho}$ . Before proceeding, the I/O behavior of the original and the modal LPV system was analyzed. Simulations verified that the effect of the  $\rho$ -dependent term is relatively small in this example, hence can be neglected.

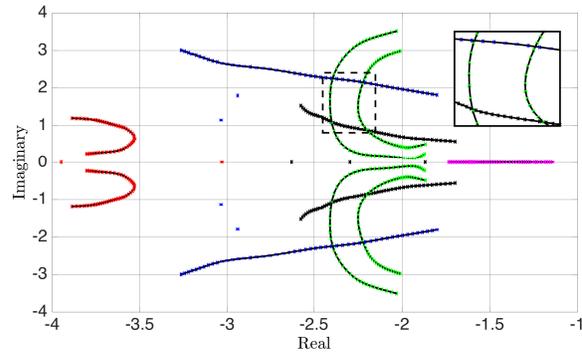


Fig. 2. Matching of eigenvalues in the consecutive scheduling parameter values

The next step of model reduction is the clustering of the modes. Performing the hierarchical clustering with predefined cluster number 5 the clusters depicted with different colors in Fig. 2. This means that the LPV model was divided into 5 smaller LPV systems of dimensions 4,5,4,6,1. Since the effect of  $E(\rho)\dot{\rho}$  is small, we could have decided at this point to neglect this component and reduce the 5 LPV models independently. Instead of this, the term  $E(\rho)\dot{\rho}$  was kept, the LPV balanced model reduction was performed on the whole 20 dimensional model, but the Gramians were sought in block diagonal form. The dimensions of the blocks were chosen to be aligned with the dimensions of the 5

<sup>2</sup>In the computation of the hyperbolic metric (eq. (8))  $T_s = 0.001$  was used.

subsystems. The Gramians were thus constructed as follows:

$$X_o(\rho) = \text{block-diag}(X_{o,0,1}, \dots, X_{o,0,5}) + \rho \cdot \text{block-diag}(X_{o,1,1}, \dots, X_{o,1,5})$$

$$X_c(\rho) = \text{block-diag}(X_{c,0,1}, \dots, X_{c,0,5}) + \rho \cdot \text{block-diag}(X_{c,1,1}, \dots, X_{c,1,5})$$

where  $X_{o,i,j}, X_{c,i,j} \in \mathbb{R}^{d(j) \times d(j)}$  with  $d = [4, 5, 4, 6, 1]$ . The LPV balanced reduction was also performed with full (non-structured)  $X_{o,0}, X_{o,1}, X_{c,0}, X_{c,1}$  variables in order to compare the results. Fig. 3 shows the singular values in both cases. It can be seen that running the balanced reduction without modal transformation proposes to neglect 12-14 states while the modal approach suggests removing 10-12. This result is quite promising, especially if the computation times are compared. With full  $X_{o,0}, X_{o,1}, X_{c,0}, X_{c,1}$  matrices the total number of free parameters were  $210 \times 4$  that resulted in 80-120 sec computation time per each iteration in (12). On the other hand, by using structured Gramians we had only  $57 \times 4$  free variables and thus each iteration could be solved less than 11 sec.

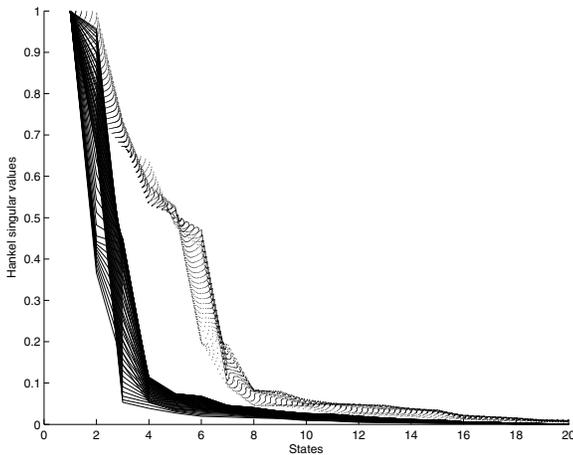


Fig. 3. Parameter-dependent singular values after balancing. Solid, left: case with (unstructured) Gramians. Dotted, right: with structured Gramians.

Finally, the model reduction of the LPV system, based on the obtained singular values (as seen in Figure 3) had been carried out, following the steps described in Section III-A). The step-response of the full order (20 dimensional) system is compared with the reduced one (10 dimensional) in Figure 4. It can be depicted that the proposed reduction methodology preserved the IO behaviour of the system.

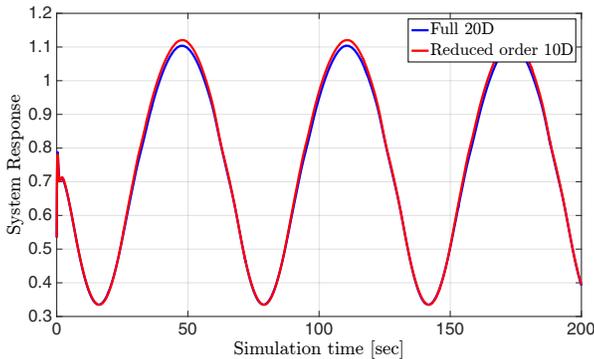


Fig. 4. Step response of the full order and the reduced order LPV system.

## VI. CONCLUSIONS AND FUTURE WORKS

A novel model order reduction algorithm has been proposed in the paper, for parameter-varying systems. The developed algorithm is based on the (approximate) modal decomposition of the LPV dynamics that makes it possible to perform the parameter-varying balanced reduction in a structured, more efficient way. Preserving the modal consistency of the system is an important, key-feature of the methodology. The efficiency of the algorithm has been demonstrated on a numerical example, encouraging for further development of the method.

The presented framework was restricted in its application; only stable LPV systems have been considered, with diagonalizable system matrices. Extension of the method for unstable or mixed stability systems, as well as, for systems with multiple eigenvalues have been already obtained and available [10]. However, the question of non-diagonalizable system matrices still remains an open issue for our future research.

## VII. ACKNOWLEDGMENTS

The research leading to these results is part of the FLEXOP project. This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 636307.

## REFERENCES

- [1] F. Adegas, I. Sonderby, M. Hansen, and J. Stoustrup. Reduced-order LPV model of flexible wind turbines from high fidelity aeroelastic codes. In *IEEE Conf. Control Applications*, pages 424–429, 2013.
- [2] A.F. Beardon and D. Minda. The hyperbolic metric and geometric function theory. In *Proceedings of the International Workshop on Quasiconformal Mappings and their Applications*, 2000.
- [3] S. Delannoy. Design and validation of A350 flexible structure control law. In *EURO-GNC*, 2015.
- [4] G. E. Dullerud and E. G. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Springer, New York, NY, USA, 2000.
- [5] William M. Goldman. *Complex Hyperbolic Geometry*. Oxford mathematical monographs. Clarendon Press, 1999.
- [6] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Data Mining, Inference, and Prediction. Springer, 2009.
- [7] A. Kwiatkowski and H. Werner. PCA-Based Parameter Set Mappings for LPV Models With Fewer Parameters and Less Overbounding. *IEEE Transactions on Control Systems Technology*, 16(4):781–788, 2008.
- [8] P. D. Lax. *Linear algebra*. Pure and applied mathematics. Wiley-Interscience, 1996.
- [9] L. Lovasz and M. D. Plummer. *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland Mathematics Studies, 1986.
- [10] T. Luspay, T. Péni, I. Gőzse, Z. Szabó, and B. Vanek. Model reduction for LPV systems based on approximate modal decomposition. *arXiv:1609.06948*, September 2016.
- [11] A. Marcos and G. Balas. Development of linear-parameter-varying models for aircraft. *Journal of Guidance, Control and Dynamics*, 27(2):218–228, 2004.
- [12] B. C. Moore. Principle Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, February 1981.
- [13] C. Moreno, P. Seiler, and G. Balas. Model Reduction for Aeroservoelastic Systems. *Journal of Aircraft*, 51(1):280–290, 2014.
- [14] C. Poussot-Vassal and C. Roos. Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models. *Control Engineering Practice*, 20:919–930, 2012.
- [15] S.Z. Rizvi, J. Mohammadpour, R. Tóth, and N. Meskin. A Kernel-Based PCA Approach to Model Reduction of Linear Parameter-Varying Systems. *IEEE Transactions on Control Systems Technology*, 24:1883–1891, 2016.
- [16] Zoltán Szabó and József Bokor. *Non-Euclidean Geometries in Modeling and Control*. Széchenyi University Press, 2015.
- [17] J. Theis, P. Seiler, and H. Werner. Model Order Reduction by Parameter-Varying Oblique Projection. In *American Control Conference 2016, to appear*, 2016.
- [18] J. Theis, B. Takarics, H. Pfifer, G. Balas, and H. Werner. Modal Matching for LPV Model Reduction of Aeroservoelastic Vehicles. In *AIAA Science and Technology Forum*, 2015.
- [19] Roland Toth. *Modeling and Identification of Linear Parameter-Varying Systems*. Springer-Verlag Berlin Heidelberg, 2010.
- [20] G. D. Wood. *Control of Parameter-Dependent Mechanical Systems*. PhD thesis, University of Cambridge, 1996.
- [21] G. D. Wood, P. J. Goddard, and K. Glover. Approximation of Linear Parameter-Varying Systems. In *Proceedings of the 35th Conference on Decision and Control*, pages 406–411, 1996.