Joint order and dependency reduction for LPV state-space models

Muhammad Mohsin Siraj, Roland Tóth and Siep Weiland

Abstract—In the reduction of Linear Parameter-Varying (LPV) models, decreasing model complexity is not only limited to model order reduction (state reduction), but also to the simplification of the dependency of the model on the scheduling variable. This is due to the fact that the concept of minimality is strongly connected to both types of complexities. While order reduction of LPV models has been deeply studied in the literature resulting in the extension of various reduction approaches of the LTI system theory, reduction of the scheduling dependency still remains to be a largely open problem. In this paper, a model reduction method for LPV state-space models is proposed which achieves both state-order and scheduling dependency reduction. The introduced approach is based on an LPV Ho-Kalman algorithm via imposing a sparsity expectation on the extended Hankel matrix of the model to be reduced. This sparsity is realized by an $L_1$-norm based minimization of a priori selected set of dependencies associated sub-Markov parameters. The performance of the proposed method is demonstrated via a representative simulation example.

Index Terms—Linear parameter-varying systems; model reduction; Ho-Kalman algorithm; $L_1$ relaxation; realization.

I. INTRODUCTION

The principle question that arises in modeling of most physical, chemical and engineering systems is how to decrease the complexity of first-principle models or models inferred from data while preserving a sufficient level of predictive power i.e., accuracy in the model. Reduction of Linear Parameter-Varying (LPV) models has been studied in the literature since the middle 1990s to support low-order controller synthesis, see, e.g., [1]–[3]. An interesting feature in the LPV setting is that decreasing the complexity corresponds not only the reduction of the complexity with respect to the order of the system model (state reduction), but also to the dependency on the scheduling variable.

Due to the linearity of LPV models, the problem of state-order reduction is similar to the model reduction in the LTI case, hence various methods available for LTI state reduction, such as co-prime factor based, optimal Hankel norm and balanced truncation methods have been extended to the LPV case with some moderate modifications [1], [2], [4]–[7]. Some of these approaches are also implemented in the Enhanced LFR toolbox [3], [6]. Most of the existing LPV reduction techniques are either based on Linear-Fractional Representation (LFR) or State-Space (SS) representations with affine dependency. In most of these approaches, the prime emphasis is on reducing the state dimension of the LPV model [1]. However, the problem of reduction of the dimension of the scheduling variable is of equal interest and generally disregarded in the literature. This paper aims to remedy this.

One of the LPV model reduction methods is the LPV Ho-Kalman algorithm studied in [4]. Unlike other LPV model reduction techniques, this approach, in addition to its simplicity, does not necessitate quadratic stabilizability or detectability of the full order model and can be employed for both stable and unstable systems without imposing any modifications. Furthermore, it can also be used for a “minimal” state-space realization of a large variety of model structures used in LPV identification. However, this algorithm also focuses only on state-order reduction, hence it does not offer reduction with respect to the scheduling dependency.

In this paper, we aim to explore the question of reduction of LPV system models from the view point of both state-order and scheduling-dependency reduction. For this purpose, we apply a modified version of the LPV Ho-Kalman algorithm via imposing a sparsity expectation on the extended Hankel matrix of the model to be reduced. LPV state-space representations with affine functional dependencies consist of state-space system matrices $A(p), B(p), C(p)$ and $D(p)$ that affinely depend on the scheduling parameter $p: \mathbb{Z} \rightarrow \mathbb{P} \subseteq \mathbb{R}^n$ in the sense that, like for $A(p)$

$$A(p) = A_0 + \sum_{i=1}^s A_i \Psi_i(p),$$

where $A_i \in \mathbb{R}^{\times \times}$ are the constant sub-parameters and $\Psi_i(\cdot): \mathbb{P} \rightarrow \mathbb{R}$ are linearly independent functional dependencies. Reduction of these sub-parameters of the system model to their minimal set results in the reduction of the functional dependencies. In this paper, combined with the Ho-Kalman based state-reduction, this is proposed to be achieved through the $L_1$-norm based convex relaxation of the following minimization problem:

$$\min_{x \in \mathbb{R}^s} \|x\|_0,$$  \hspace{1cm} (2)

where $x$ is the column vector of all sub-parameters and $\|x\|_0$ is the $L_0$ pseudo-norm that returns the number of non-zero components in its arguments [8]. This corresponds to the investigation of which functional dependencies have a significant role in an accurate LPV state-space representation of the system dynamics. It will also be shown that both the reduction of functional dependencies (through $L_1$ minimization) and state-order reduction (via the LPV Ho-Kalman algorithm) can be carried out independently in case of affine dependency.

The paper is organized as follows: In the preceding section, the LPV Ho-Kalman algorithm and the concept of
the extended Hankel matrix for LPV systems are briefly discussed. This is followed by introducing the $L_1$-norm based recovery problem in Section III. In Section IV, the reduction of the scheduling dependency is formulated as an $L_1$-norm based recovery problem and it is discussed how the recovered solution can be integrated with the LPV Ho-Kalman algorithm to achieve joint reduction of the dependency and the model order. In Section V, a simulation example is given to demonstrate the performance of the proposed method and finally the conclusions of the presented results are given in Section VI.

II. THE LPV HO-KALMAN ALGORITHM

In this section, a brief overview of the LPV extension of the Ho-Kalman algorithm is given. This approach, introduced in [4], aims at the model reduction of the state-space representation of an LPV system with an affine dependence on the scheduling variable.

State-space representation of an LPV system in discrete-time is commonly defined as

\[
qx = A(p)x + B(p)u,
\]
\[
y = C(p)x + D(p)u,
\]

where $u : \mathbb{Z} \to \mathbb{R}^{n_u}$, $y : \mathbb{Z} \to \mathbb{R}^{n_y}$ and $x : \mathbb{Z} \to \mathbb{R}^{n_x}$ are the input, output and state signals of the system respectively, $q$ is the forward time-shift operator, i.e., $q(x(k)) = x(k+1)$ and the system matrices $A, B, C, D$ are functions of the scheduling signal $p : \mathbb{Z} \to \mathcal{P}$. Affine dependence of the system matrices on $p$ is formulated as (1) and

\[
B(p) = B_0 + \sum_{i=1}^{s} B_i \Psi_i(p),
\]
\[
C(p) = C_0 + \sum_{i=1}^{s} C_i \Psi_i(p),
\]

where $\Psi_i(\cdot) : \mathcal{P} \to \mathbb{R}$ are bounded, linearly independent functions on $\mathcal{P}$ and $\{A_i, B_i, C_i\}_{i=0}^{s}$ are constant matrices with appropriate dimensions. Without loss of generality of the approach, from now on, we assume that $D(p) = 0$. Note that for the sake of simplicity of the upcoming derivations, we consider here all matrices $A, B$ and $C$ to depend on the same set of functions $\Psi_i$, this assumption will be relaxed later on.

For general definition of state-space representations of an LPV system and the application of the Ho-Kalman algorithm see [4] and [9].

The LPV system represented by (3a-b) also has an Infinite Impulse Response (IIR) representation in the form of

\[
y(k) = \sum_{i=0}^{\infty} g_i(p,k)u(k-i),
\]

where $g_i$ are the Markov parameters (functions) of the LPV system computed as $g_i(p,k) = C(p(k)) \prod_{\ell=1}^{i-1} A(p(k-i+\ell))B(p(k-i))$.

In order to introduce the LPV Ho-Kalman algorithm, an extended Hankel matrix for LPV systems is defined next and then the steps involved in the algorithm are discussed.

1) The extended Hankel matrix in the LPV case: Recursively define, for any $j$,

\[
M_1 = [B_0 \cdots B_s],
\]
\[
M_j = [A_0 M_{j-1} \cdots A_s M_{j-1}],
\]

and introduce

\[
R_k = [M_1 \cdots M_k],
\]

where $R_k \in \mathbb{R}^{n_x \times (n_u \sum_{i=1}^{s} (1+s)^i)}$. The matrix $R_k$ is called the $k$-step extended reachability matrix, which characterizes the structural state-reachability of the state-space representation of an LPV system [9]. As a next step, recursively define, for any $j$.

\[
N_1 = [C_0^T \cdots C_s^T]^T,
\]
\[
N_j = [(N_{j-1} A_0)^T \cdots (N_{j-1} A_s)^T]^T.
\]

Similarly to the reachability case, introduce the $k$-step extended observability matrix, which characterizes the structural state-observability of the state-space representation of an LPV system [9], as:

\[
O_k = [N_1^T \cdots N_k^T]^T,
\]

where $O_k \in \mathbb{R}^{(n_y \sum_{i=1}^{s} (1+s)^i) \times n_x}$. Define the extended Hankel matrix of the system as,

H_{i,j} = O_i R_j \in \mathbb{R}^{(n_y \sum_{i=1}^{s} (1+s)^i) \times (n_u \times n_y \sum_{i=1}^{s} (1+s)^i)}.

In case of $x(k-j) = 0_{n_x}$ and $u(l) = 0$ for $l \geq k$,

\[
Y_{k,i} = N_{k,i} H_{i,j} M_{k,j},
\]

where $i,j \geq 1$, $Y_{k,i} = [y(k) \cdots y(k+i-1)]^T$ and $U_{k,j} = [u(k-1) \cdots u(k-j+1)]^T$ together with the combination of the changes of the scheduling dependencies expressed by

\[
W_k = \begin{bmatrix}
1 & \Psi_1(p(k)) & \cdots & \Psi_s(p(k))
\end{bmatrix}^T,
\]

\[
L_{k+i} = I_{n_y} \otimes (W_k \otimes \cdots \otimes W_{k-i})^T,
\]

\[
N_{k,i} = \text{diag}(L_{k+i}, \cdots, L_{k+i-1}),
\]

\[
K_{k,j} = W_k \otimes \cdots \otimes W_{k-j} \otimes I_{n_u},
\]

\[
M_{k,j} = \text{diag}(K_{k,j-1} \cdots K_{k-1,j-1}),
\]

where $I_n$ is the $n \times n$ identity matrix and $\otimes$ is the Kronecker product. For $s = 2$, $H_{i,j}$ is

\[
\begin{bmatrix}
C_0 B_0 & C_0 B_1 & C_0 A_0 B_0 & C_0 A_0 B_1 & \cdots \\
C_1 B_0 & C_1 B_1 & C_1 A_0 B_0 & C_1 A_0 B_1 & \cdots \\
C_0 A_0 B_0 & C_0 A_0 B_1 & C_0 A_0 B_0 & C_0 A_0 B_1 & \cdots \\
C_1 A_0 B_0 & C_1 A_0 B_1 & C_1 A_0 B_0 & C_1 A_0 B_1 & \cdots \\
C_0 A_1 B_0 & C_0 A_1 B_1 & C_0 A_1 B_0 & C_0 A_1 B_1 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{bmatrix}
\]

where $C_0 B_0, C_0 B_1, C_0 A_0 B_0, \ldots$ are called the sub-Markov parameters of the state-space representation of an LPV system. These parameters are constants multiplied by the functional dependencies as shown below for the $N$th Markov parameter $g_N$ in case of polynomial dependence:

\[
g_N = \left( \sum_{i=0}^{s} C_i P_i^N \right) \left( \prod_{i_1=1}^{N-1} \sum_{i_2=0}^{s} A_{i_1} P_i^{i_2} \right) \left( \sum_{i=0}^{s} B_i P_i^0 \right)
\]
where $P_N$ is the scalar value of the scheduling variable $p$ at the time instant $N$, i.e., $P_N = p(N)$.

2) The LPV Ho-Kalman algorithm: The following steps are involved for computing the system matrices $A(p(k)), B(p(k)), C(p(k))$ with a minimal or reduced state order $n$ from a given Hankel matrix $H_{i,j}$.

- Compute the SVD $H_{i,j} = U_n \Sigma_n V_n$ to determine $n$ (the number of non-zero singular values in $\Sigma_n$). In case of reduction, the significant non-zero singular values will determine the number of state variables $n$.
- Construct $H_1$ and $H_2$ according to:

$$H_1 = U_n \Sigma_n^2 \hat{\mathcal{O}}_1,$$  \hspace{1cm} (13a)
$$H_2 = \Sigma_n^2 V_n^\top \hat{\mathcal{C}}_j,$$  \hspace{1cm} (13b)

where $\text{rank}(H_1) = \text{rank}(H_2) = n$. $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{C}}_j$ are the extended observability and controllability matrices respectively corresponding to the resulting 'balanced' state basis.

- $[\hat{\mathcal{C}}_1^\top \cdots \hat{\mathcal{C}}_s^\top]^\top$ are extracted by taking the first $n_g(1+s)$ rows of (13a) while $[\hat{B}_0 \cdots \hat{B}_s]$ is obtained by taking the first $n_u(1+s)$ columns of (13b).

- $[\hat{A}_0 \cdots \hat{A}_s]$ is isolated from $H_{i,j}$, which is obtained by shifting the Hankel matrix one block column, i.e., $n_u(1+s)$ columns, to the left. $H_{i,j}$ can be written as

$$H_{i,j} = \hat{\mathcal{O}}_1 [A_0 \cdots A_s] (I_{1+s} \otimes R_{j-1}),$$  \hspace{1cm} (14a)
$$= H_1 [A_0 \cdots A_s] (I_{1+s} \otimes H_2),$$  \hspace{1cm} (14b)

where $\otimes$ is introduced as a block-wise Kronecker product defined as follows:

$$I_{1+s} \otimes R_{j-1} = [I_{1+s} \otimes M_1 \cdots I_{1+s} \otimes M_{j-1}].$$  \hspace{1cm} (15)

$H_2$ is generated from $H_2$ by leaving out the last $n_u(1+s)$ columns. As a consequence, it holds that

$$H_1^\top [H_1 \otimes H_2] = [A_0 \cdots A_s],$$  \hspace{1cm} (16)

where $H_1^\top$ and $H_2^\top$ are referred to as the left pseudo-inverse and right pseudo-inverse of $H_1$ and $H_2$ respectively.

It is important to highlight that $s$ and the set $\{\Psi_i\}_{i=1}^s$ has an implicit role in the formulation of the Hankel matrix and the whole realization/reduction procedure. Therefore, this method does not offer reduction with respect to the functional dependencies. Next an $L_1$ recovery concept is introduced which later will be applied for the reduction of the sub-Markov parameters and hence the scheduling dependency.

III. $L_1$ Recovery

To propose a solution for the reduction of scheduling dependency, first we briefly study the concept of $L_1$ recovery. Suppose that $\Phi \in \mathbb{R}^{N \times m}, b \in \mathbb{R}^N$ and there exists an $x \in \mathbb{R}^m$, s.t. $\Phi x = b$. If $b$ contains the samples of a signal, then $x$ is called a representation of $b$ with respect to the $\Phi$ matrix [8]. Furthermore, $x$ is called $s$-sparse when $\|x\|_0 \leq s$.

The basic objective of “recovery” is to represent the signal $b$ by computing a $x$ with maximal sparsity. This corresponds to minimizing the $L_0$ norm of $x$ under the constraint that $\Phi x = b$. As this problem is non-convex and NP hard, a fruitful alternative is a convex relaxation based on the $L_1$ norm [8]:

$$\min \|x\|_1,$$  \hspace{1cm} (17a)
$$\text{subject to } b = \Phi x.$$  \hspace{1cm} (17b)

Such a relaxation is known to have a close approximation of the $L_0$ norm [8]. Note that (17) is a classical linear programming problem which is efficiently solvable, even in the case where $m \gg N$. These problems arise in many areas, including compressive sensing, statistics, signal processing, machine learning and approximation theory [8].

In case of the approximate representation of $b$ by $x$ with respect to the matrix $\Phi$, (17) modifies to

$$\min \|x\|_1,$$  \hspace{1cm} (18a)
$$\text{subject to } \|b - \Phi x\|_2 < \epsilon.$$  \hspace{1cm} (18b)

where $\epsilon > 0$ represents the approximation error which can be a priori chosen. (18a-b) is again a convex problem.

Next, by formulating an $L_1$ recovery problem of the Markov parameter sequence of the system (for a given trajectory of $p$) with respect to a priori assumed set of dependencies, it will be shown how the principle idea of maximal sparsity expectation can be used for the reduction of sub-Markov parameters and hence the scheduling dependency. Later, it will be demonstrated by a simulation example, that the modified version of the LPV Ho-Kalman algorithm based on this idea of ‘recovery’ is capable of recovering the minimal (both in order and dependency) state-space realization of a given LPV system.

IV. IMPROVED LPV HO-KALMAN ALGORITHM

So far, we have discussed both the LPV Ho-Kalman algorithm for state-order reduction and the concept of $L_1$ recovery. In this section, we will present and formulate the idea of utilizing these concepts to propose a joint reduction technique for state-order and scheduling dependency.

For the sake of simplicity, only Single-Input Single-Output (SISO) case is considered. It is assumed that the Markov parameters $g_0, g_1, \ldots, g_N$ of the LPV model with respect to different trajectories of the scheduling variable $p(k)$ are given. Note that the Markov parameters of an LPV system are functions of $p(k)$ and also depend on the shifted version of $p(k)$. On the other hand, the sub-Markov parameters are constants. Note that the system matrices $\{A, B, C\}$, associated with the underlying minimal representation and the scheduling dependency, are assumed to be unknown.

A. Problem formulation

The problem of scheduling dependency reduction is formulated as a linear problem, i.e., $\Phi x = b$. This minimal realization problem can be extended to the approximate model reduction problem as in (18) with a suitably chosen $\epsilon$ (see later the discussion on the selection of $\epsilon$). The $\Phi$ is a block diagonal matrix which contains the various combinations of the sequences $\{\Psi_i(p_k)\}_{k=1}^N$. The vector $b = [g_1(p, 1) \quad g_2(p, 2) \quad \ldots \quad g_N(p, N)]^\top$ contains the value of
the Markov parameters for a given trajectory of \( p(k) \) on \( k \in [1, N] \). This can be considered as the impulse response of the model along that scheduling trajectory. The sub-Markov parameters are collected in \( x \). In this way, \( x \) can be seen as the weights of the function components which compose each \( g_1, \ldots, g_N \) and hence their observed values along the scheduling variable \( p \) given in vector \( b \). To be consistent with our previous assumption, \( g_0 = 0 \) corresponding to \( D(p(k)) = 0 \). To illustrate this construction, consider the case when \( \Psi_i(p(k)) = p^i(k) \) with \( n_p = 1 \). Define

\[
L_i = \begin{bmatrix} 1 & p(i) & p^2(i) & \cdots & p^i(i) \end{bmatrix},
\]

for \( i \in \{0, \ldots, N\} \). Let

\[
K_i = L_i \otimes L_{i-1} \otimes \ldots \otimes L_0.
\]

Then, matrix \( \Phi \) is formed as follows:

\[
\Phi = \text{blkdiag}(K_1, K_2, \ldots, K_N),
\]

where 'blkdiag' is a block diagonal matrix with \( N \) rows and \( \sum_{i=1}^{N} (s + 1)^i \) columns, \( N \) is the number of Markov parameters and \( s \) is the expected (maximum allowed) order of the polynomial dependence on \( p \). The column vector \( b \) of the Markov parameters is defined as:

\[
b = \begin{bmatrix} g(p, 1) & g(p, 2) & \cdots & g(p, N) \end{bmatrix}^T.
\]

The vector of the sub-Markov parameters \( x \) is defined via

\[
W_j = [C_0 M_j^T \ C_1 M_j^T \ \cdots \ C_s M_j^T]^T
\]

giving

\[
x = [W_1 \ W_2 \ \cdots \ W_N]^T.
\]

Note that in MIMO case, the same procedure can be used to formulate a vectorized form of all sub-parameters.

\section{The \( L_1 \) Optimization}

Now, in order to recover \( x \) with maximal sparsity, corresponding to the elimination of all non-significant sub-Markov parameters, the \( L_1 \) optimization in the form of (18a-b) is used with \( \epsilon > 0 \) chosen according to the expected approximation realizability error on the Markov parameter sequence. In case of a minimal realization problem, \( \epsilon \) is used to represent the machine round-off and hence it is typically chosen to be \( 0 < \epsilon \ll 1 \). In case of approximative realizability, \( \epsilon \) can be chosen as \( \epsilon = (1 - \delta)||b||_2 \) where \( 0 < \delta \ll 1 \) represents a user allowed percentage of error relative to the \( L_2 \) norm of the truncated impulse response of the model given by \( b \). Following the same derivation as in [4], this leads to a bound on the expected induced \( \mathcal{H}_\infty \) norm of the model approximation error. An alternative approach for the selection of \( \epsilon \) can be achieved by selection methodologies discussed in the sparse estimation literature [10]. The systematic choice of this threshold is regarded as a future research work.

A particularly important fact is that the reduction of the \( p \)-dependence of the Markov parameters can be carried out independently from the state-reduction due to the assumed linear parameterization of the matrices in \( \{\Psi_i\}_{i=1}^{N} \) and the linear independence of these functions on \( p \). This is due to the property of the Markov parameters, namely that they uniquely characterize the input-output map of the system (disregarding initial conditions). Therefore, reduction of their dependence in a structural form which is related to the aimed state-space description (\( 1^\text{st} \)-order parameter-varying difference equation with linear parameterization of the coefficients in a subset of \( \{\Psi_i\}_{i=1}^{N} \)), leaves the choice for the state-map and hence the state-variable selection independent. To ensure optimal selection of the subset of \( \{\Psi_i\}_{i=1}^{N} \) required for the reduction/minimal realization, the corresponding signal recovery problem must be solved with respect to all possible trajectories of \( p \), i.e., \( p \in \mathbb{R}^Z \). To formulate this in a computationally feasible way, a finite set of randomly generated trajectories are taken.

\section{Detection and reduction of the dependence}

As the extended Hankel matrix of the model to be reduced is obtained via imposing a sparsity expectation on it, the resulting matrix will contain some rows/columns of zeros specifying the reduced dependency structure. Denote by \( s_A \), \( s_B \) and \( s_C \) the order of dependencies of the \( p \)-dependent matrices in (1) and (4a-b) respectively which initially are equal to \( s \). In case the input matrix \( B_i \), \( i \in \{1, \ldots, s_B\} \), can be neglected in the \( L_1 \) recovery provided Hankel matrix, then the \( p \)-th column and every \( (s_B+1) \)-th column following it must be zero. The same holds for the output matrix \( C_s \) regarding the rows of the Hankel matrix. By detecting these columns and rows, the particular dependencies can be discarded from (1) and (4a-b). For the state matrix \( A \), the detection procedure is more complicated. Based on the construction scheme of the extended Hankel matrix (10), we can observe how the state matrices \( \{A_i\}_{i=0}^{s_A} \) appear in the structure. Then the following rules can be obtained to detect the absense of \( A_i \):

\begin{itemize}
  \item Check if there is any block of \((s_A + 1)^T\) zeros spaced \( i(s + 1)^T\) apart for \( i = 1, 2, \ldots \).
  \item Remove the columns associated with these blocks.
\end{itemize}

Due to symmetry of the extended Hankel matrix, a similar check can be made on the rows. The absence of these rows/columns specifies the absence of a particular \( A_i \).

\section{Improvements of the existing LPV Ho-Kalman algorithm}

As shown in Section II, \( s \) has an implicit role in the original Ho-Kalman realization/reduction procedure. Now the existing algorithm is improved to use the reduced set of dependencies for each system matrices, i.e., \( s_A \), \( s_B \) and \( s_C \) as determined in the previous sub-section. The improved LPV Ho-Kalman algorithm inherits the properties of the existing algorithm (see [4] for the details) and it can be extended to \textit{Multiple-input Multiple-Output} (MIMO) system models as well. Note that the \( L_1 \) recovery problem can be easily extended to the MIMO case via a block-wise formulation, but in terms of implementation, it requires a precise book keeping of the sub-Markov sequences.

\section{Example}

The purpose of the following example is to demonstrate the performance of the proposed model reduction scheme on a relevant simulation example. In this demonstration, we
will first focus on the exact reduction of a non-minimal state-space representation of an LPV system to a minimal form. This will be followed by a demonstration of approximate reduction via the proposed scheme.

Consider the example of a mass connected to a varying spring and damper depicted in Fig. 1. This problem is one of the typical phenomena occurring in the motion control of many mechatronic systems like in active suspension. Denote $x$ the position (in [m]) of the mass $m$ (in [kg]), $k_d > 0$ the varying stiffness of the spring and $c_d > 0$ the varying damping. These varying coefficients are assumed to be a function of a scheduling signal $p(t) : \mathbb{R} \rightarrow [0, 1]$. Introduce $F$ as the force (in [N]) acting on the mass $m$. In this example, we will treat $y(k) = \frac{dx}{dt}(k)$ as the sampled output and $u(k) = F(k)$ as the sampled input of the system for with a sampling period $T_s = 0.01 s$. The underlying system can be represented as a $2^{nd}$-order parameter-varying differential equation in a straight forward manner. However due to discretization followed by a state-space realization, the following DT representation of the system is obtained:

$$\begin{bmatrix}
A(p,k) & B(p,k) \\
C(p,k) & D(p,k)
\end{bmatrix}$$

$$\begin{bmatrix}
a_{11} & a_{12} & b_1 \\
a_{21} & a_{22} & b_2 \\
c_1 & c_2 & 0
\end{bmatrix}$$

(24)

where

$$a_{11}(p,k) = p_k^2(0.6p_k - 1.3 + p_k(0.5p_k - 2.15)) + 1.8p_k + 0.9$$

$$a_{12}(p,k) = -p_k^2(1.2p_k - 1 + 0.6 + p_k(0.4p_k - 1) + 4 - 3p_k + 0.25)$$

$$a_{21}(p,k) = 0.3p_k^2p_k - p_k(0.25p_k - 1 + 0.9p_k)$$

$$a_{22}(p,k) = -0.6p_k^2p_k - p_k(0.2p_k - 1 + 1.9) - 1.6p_k + 0.95$$

$$b_1(p,k) = 2.2p_k + 0.9, \quad c_1(p,k) = 1 - p_k,$$

$$b_2(p,k) = 1.1p_k - 0.1, \quad c_2(p,k) = 2p_k - 1.$$

3-D plot of the Frozen Frequency Response (FRF) of the given system model (frequency response of the system for constant $p(k)$, i.e., $p(k) = P$, $\forall k \in \mathbb{Z}$) is shown in Figure 2.

1) Exact reduction to minimal form: The Markov parameter sequences (for a finite set of randomly generated trajectories of the scheduling signal $p(t) : \mathbb{R} \rightarrow (0, 1)$) of the state-space representation of an LPV system given above in (24) are calculated. Then the $L_1$ recovery problem, as presented in (18a-b), is formulated. As this is the minimal realization case, $\epsilon$ is chosen to be very small, i.e., $0 < \epsilon << 1$, and it represents the machine round-off only. This recovery problem results in the elimination of all non-significant sub-Markov parameters corresponding to the negligible dependency structure. For the given system, dependency is determined to be only on $p_k$. The maximally sparse vector $x$ of the sub-Markov parameters is mapped to the extended Hankel matrix. The LPV H∞-Kalman algorithm is then invoked, the non-zero singular values are of magnitude $2.8501$ and $0.0142$ respectively, which determine the minimal state-order of the model. By using the steps in the Section II the system matrices are recovered as follows:

$$\begin{bmatrix}
A(p,k) & B(p,k) \\
C(p,k) & D(p,k)
\end{bmatrix} = \begin{bmatrix}
0.875 - 0.080p_k & -0.045 - 0.035p_k & -1.001 \\
0.039 + 0.043p_k & 0.974 - 0.019p_k & 0.045 \\
-1.001 & -0.061 & 0
\end{bmatrix}$$

(25)

The 3-D plot of the difference between the FRF of the given system model and its minimal realization is shown for all values of $P$ in Figure 3. The system model given in (24) can also be converted by a state-transformation to the minimal state-space representation as shown below:

$$\begin{bmatrix}
A(p,k) & B(p,k) \\
C(p,k) & D(p,k)
\end{bmatrix} = \begin{bmatrix}
0.9 - 0.1p_k & -(0.2 - 0.2p_k) & 1 \\
0 & 0.95 & 0.1 \\
1 & 0 & 0
\end{bmatrix}$$

(26)

This representation is indeed connected with a state-transformation to (24) via the state transformation matrix

$$T(p_k) = \begin{bmatrix}
2p_k + 1 & 2p_k - 1 \\
p_k & p_k - 1
\end{bmatrix}.$$  

(27)

This proves that that realization determined by our proposed method is a particular minimal realization of (24).

The reduced model is also compared with the original system in terms of time domain responses with respect to a 1000 sample long randomly generated input sequence with $u(k) \in \mathcal{U}(0, 1)$ and $p(k) = 0.25 + 0.25w_1(k) + w_2(k)$ where $w_1$ is a normalized random phase multisine with frequencies $10, 100, 200$ Hz and $w_2(k) \in \mathcal{U}(0, 5)$. To quantify the time-domain approximation error, the Best Fit Rate (BFR)

$$\text{BFR} = 100\% \cdot \max \left( 1 - \frac{\|y(k) - \hat{y}(k)\|_2}{\|y(k) - \bar{y}\|_2}, 0 \right),$$

(28)
and the Variance Accounted For (VAF) percentage
\[
\text{VAF} = 100\% \max \left( 1 - \frac{\text{var}(y(k) - \hat{y}(k))}{\text{var}(y(k))} \right),
\]
are used where \(\hat{y}\) is the mean of \(y\) (the output of the given system) and \(\hat{y}\) is the simulated output of the algorithm provided model. In the minimum realization case, both the VAF and the BFR are 100%.

2) Reduced representation of the original system: In case of the reduced representation, the calculation of the Markov parameters and the \(L_1\) recovery problem is formulated in the similar way as shown in the minimal realization case, except that now, \(\epsilon\) is chosen according to the following selection rule to limit the approximation error:
\[
\epsilon = (1 + n_g \times \log(N)) \|b - \Phi^+b\|_2^2,
\]
where \(\Phi^+ = \Phi(\Phi^T\Phi)^{-1}\Phi^T\) is an orthogonal projection and \(n_g\) is the number of sub-Markov parameters in \(x\). This selection rule has a direct connection with the BIC based threshold selection for \(L_1\) sparse estimators [10]. Considering that in these scheduling dependencies the Markov parameters can be realized without representation error, therefore this bound \(\epsilon_{\text{opt}}\) and the resulting \(\epsilon\) are very small and they are in the magnitude of the machine round-off. The \(L_1\) recovery determines the significant dependency to be on \(p_k\) only. This is followed by applying the LPV Ho-Kalman approach. However now only the significant singular value, 2.850 is considered, due to the fact that it is much larger than the other singular value 0.0153. The corresponding first-order reduced system model is obtained. The 3-D plot of the FRF difference between the given system model and the reduced first order model for all values of \(p\) is shown in Figure 4. The reduced system matrices are given as:
\[
\begin{bmatrix}
A & B \\
C & 0
\end{bmatrix} = 
\begin{bmatrix}
0.8764 & -0.0814p_k & -1.0014 \\
-1.0014 & 0 & 0
\end{bmatrix}. 
\]

The resulting BFR is 39.59% due to a slight gain difference, while the VAF is 96.88% which shows that the output variations are perfectly explained by the reduced model.

VI. CONCLUSIONS
A new method has been proposed in this paper for the model reduction of LPV systems offering joint state-order and scheduling-dependency reduction. It has been shown that, in case of affine static dependency, reduction of the scheduling-dependency and order-reduction can be understood as two decoupled problems in terms of signal recovery and state-space realization. While the first problem can be solved via reducing the number of sub-Markov parameters, associated with a given affine dependency structure, by using \(L_1\) norm based convex optimization, the second problem can be handled by a previously introduced LPV Ho-Kalman algorithm. The resulting approach inherits the properties of the existing Ho-Kalman method in terms of simplicity and applicability on non quadratically-stabilizable systems. An example is presented to verify the proposed algorithm.

VII. ACKNOWLEDGEMENT
The authors would like to thank A.G. Dankers for fruitful discussions during the development of these results.

REFERENCES