Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Subspace identification of MIMO LPV systems: the PBSID approach

Jan-Willem van Wingerden and Michel Verhaegen

Abstract— **In this paper we present a novel algorithm to identify LPV systems with affine parameter dependence operating under open and closed-loop conditions. A factorization is introduced which makes it possible to form predictors which are based on past inputs, outputs, and scheduling data. The predictors contain the LPV equivalent of the Markov parameters. Using the predictors, ideas from Predictor Based Subspace IDentification (PBSID) are developed to estimate the state sequence from which the LPV system matrices can be constructed. A numerically efficient implementation is presented.**

Index Terms— **Subspace identification, Linear Parameter-Varying systems, System identification**

I. Introduction

Linear Parameter Varying (LPV) systems attracted considerable attention in the past years [1], [2], [3], [4]. Some of the applications with a considerable LPV behavior are unstable by nature and have to operate in closed loop before they can be identified, *e.g.* aero space applications [5] and wind turbines [6]. For these systems it is common practice to develop a low-level controller to stabilize the system and identify a number of local linear models in different operation points. Interpolation is performed between the different local models to obtain an LPV representation [7], [8]. In [9] it is shown that the interpolation between these local models can lead to unstable representations of the LPV structure while the original system is stable. In this paper we present a novel subspace identification algorithm to identify LPV systems operating under open and closed-loop conditions which does not require interpolation or identification of local models.

An overview of literature in the area of LPV identification is given in [10]. In the input-output setting, work can be found in [11], [12], [13]. However, to deal with multiple input and output systems and to exploit the numerical properties of subspace techniques (these techniques are solely depending on well-established techniques from linear algebra) the focus of this paper is on subspace-based LPV identification. Recently, a number of papers appeared where the structure of the scheduling sequence is exploited. It turns out that if the scheduling is periodic [14], piecewise constant [15], [16], or white noise [17], well-established LTI subspace techniques can be used to identify LPV or bilinear systems. The identification of LPV systems with arbitrarily varying scheduling sequences has proven to be challenging from a numerical point of view [18]. The data matrices involved in this algorithm grow exponentially with the size of the prediction window. With the introduction of the kernel method to this framework the curse of dimensionality was partially solved, however, a bias was introduced [19]. For optimization based algorithms [20] these models appeared to be a good starting point.

The literature on the identification of LPV systems so far is dedicated to the open-loop setting. Recently, a paper appeared [21] where they extend the $\text{PBSID}_{\text{opt}}$ algorithm [22] to LPV systems (we will refer to this algorithm as LPV-$\text{PBSID}_{\text{opt}}$). In this paper we will follow a similar approach, however, instead of using the $\text{PBSID}_{\text{opt}}$ algorithm we use the PBSID algorithm [23] (we will refer to this algorithm as LPV-PBSID). The big difference between these two algorithms is the number of terms you take into account in the product between the observability and controllability matrix. In the LPV-PBSID algorithm the matrices have a similar size as in [24] and are significantly larger than the matrices in [21]. However, it is of interest to see how these algorithms relate and furthermore their kernel representation is completely different (see [25] for a more elaborate discussion). With respect to [21] we also contribute by presenting a computationally efficient scheme.

The outline of this paper is as follows; we start in Section II with the problem formulation and assumptions. In Section III we present a factorization that separates the unknown system matrices from the known input, output, and scheduling data. In Section IV the basic idea behind the LPV-PBSID identification scheme is presented and the curse of dimensionality will appear. In Section V the kernel method is presented, where compact formulations of the kernels are presented. In Section VI a simulation example is presented where we compare LPV-$\text{PBSID}_{\text{opt}}$ with LPV-PBSID. We end this paper with our conclusions.

II. Problem formulation and assumptions

For the derivation of the algorithm we consider the following LPV system:

$$x_{k+1} = \sum_{i=1}^{m} \mu_k^{(i)} \left(A^{(i)} x_k + B^{(i)} u_k + K^{(i)} e_k \right), \quad (1)$$

$$y_k = C x_k + e_k, \quad (2)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^r$, $y_k \in \mathbb{R}^\ell$, are the state, input and output vectors. $e_k \in \mathbb{R}^\ell$ denotes the zero mean white innovation process. The matrices $A^{(i)} \in \mathbb{R}^{n \times n}$, $B^{(i)} \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{\ell \times n}$, $K^{(i)} \in \mathbb{R}^{n \times \ell}$ are the local system, input, output, direct feed through, and the observer matrices; and $\mu_k^{(i)} \in \mathbb{R}$ the local weights. The index m is referred to as the number of local models or scheduling parameters. Note that the system,

The authors are with the Delft Center for Systems and Control (DCSC), Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands J.W.vanWingerden@tudelft.nl m.verhaegen@moesp.org

input, and the observer matrices depend linearly on the time-varying scheduling vector. The time-varying system matrix is now given by:

$$A_k = \sum_{i=1}^{m} \mu_k^{(i)} A^{(i)},$$

and a similar thing can be done for the other system matrices. We assume that we have an affine dependence and the scheduling is given by:

$$\mu_k = \begin{bmatrix} 1, & \mu_k^{(2)}, & \cdots, & \mu_k^{(m)} \end{bmatrix}^T.$$

We can rewrite (1)-(2) in the predictor form as:

$$x_{k+1} = \sum_{i=1}^{m} \mu_k^{(i)} \left(\tilde{A}^{(i)} x_k + B^{(i)} u_k + K^{(i)} y_k \right), \quad (3)$$

$$y_k = C x_k + e_k, \quad (4)$$

with

$$\tilde{A}^{(i)} = A^{(i)} - K^{(i)} C.$$

It is well-known that an invertible linear transformation of the state does not change the input-output behavior of a state-space system. Therefore, we can only determine the system matrices up to a similarity transformation $T \in \mathbb{R}^{n \times n}$: $T^{-1}A^{(i)}T$, $T^{-1}B^{(i)}$, $T^{-1}K^{(i)}$, and CT.

The identification problem can now be formulated as: given the input sequence u_k, the output sequence y_k, and the scheduling sequence μ_k over a time $k = \{0, \ldots, N-1\}$; find, if they exist, the LPV system matrices $A^{(i)}$, $B^{(i)}$, $K^{(i)}$, and C up to a global similarity transformation.

A. Assumptions and notation

First we define the transition matrix for discrete-time time-varying systems [26] and this is given by:

$$\phi_{j,k} = \tilde{A}_{k+j-1} \cdots \tilde{A}_{k+1} \tilde{A}_k. \quad (5)$$

To make the notation more transparent we define: $z_k = \begin{bmatrix} u_k^T, & y_k^T \end{bmatrix}^T$, $\breve{B}_k = \begin{bmatrix} B_k, & K_k \end{bmatrix}$, and $\overline{B}^{(i)} = \begin{bmatrix} B^{(i)}, & K^{(i)} \end{bmatrix}$. Similar as in [23] we define a past and a future window denoted by p and f, respectively. The past window is used to define the following stacked vector:

$$\bar{z}_k^p = \begin{bmatrix} z_k^T, & z_{k+1}^T, & \cdots, & z_{k+p-1}^T \end{bmatrix}^T.$$

We assume that the state sequence:

$$X = \begin{bmatrix} x_{p+1}, & \cdots, & x_{N-f+1} \end{bmatrix},$$

has full row rank and the matrix:

$$\Gamma^f = \begin{bmatrix} C^T, & (C\tilde{A}^{(1)})^T, & \ldots, & \left(C \left(\tilde{A}^{(1)} \right)^{f-1} \right)^T \end{bmatrix}^T, \quad (6)$$

has full column rank. This last matrix can be interpreted as the extended observability matrix of the first local model. For persistency of excitation it is also required that the scheduling sequence satisfies the following relation:

$$\operatorname{rank}\left(\begin{bmatrix} \mu_0, & \mu_1, & \cdots, & \mu_{N-p-f} \end{bmatrix} \right) = m,$$

and $N - p - f + 1 > m$. The problem formulation so far does not require any assumptions on the correlation between the input and noise sequence which opens the possibility to apply the algorithm in closed-loop.

These definitions and assumptions are used in Section IV but first we define a factorization to extend the framework described in [22] to LPV systems.

III. Factorizations

In this section we define a fundamental factorization in which we separate the unknown system matrices from the known weighting sequence. The same factorization is introduced in [21] but to make the derivation of the algorithm complete we include this crucial factorization.

We will factorize the time-varying extended controllability matrix which is defined here.

Definition 1: Given the transition matrix in (5) the time-varying extended controllability matrix is given by:

$$\overline{\mathscr{K}}_k^p = \begin{bmatrix} \phi_{p-1,k+1}\breve{B}_k, & \cdots, & \phi_{1,k+p-1}\breve{B}_{k+p-2}, & \breve{B}_{k+p-1} \end{bmatrix}.$$

The time-varying extended controllability matrix can be factorized in a matrix containing only the scheduling terms and a constant matrix which depends only on the system matrices $\tilde{A}^{(i)}$, and $B^{(i)}$. Before we formulate this factorization in a lemma we have to introduce a number of definitions. We start with the following definition:

Definition 2: We define the matrix:

$$\mathscr{L}_j = \begin{bmatrix} \tilde{A}^{(1)}\mathscr{L}_{j-1}, & \cdots, & \tilde{A}^{(m)}\mathscr{L}_{j-1} \end{bmatrix},$$

with

$$\mathscr{L}_1 = \begin{bmatrix} B^{(1)}, & \cdots, & B^{(m)} \end{bmatrix}.$$

To illustrate this definition see the following example:

Example 1: For $m = 2$ one obtains:

$$\mathscr{L}_1 = \begin{bmatrix} B^{(1)}, & B^{(2)} \end{bmatrix},$$

$$\mathscr{L}_2 = \begin{bmatrix} \tilde{A}^{(1)}B^{(1)}, & \tilde{A}^{(1)}B^{(2)}, & \tilde{A}^{(2)}B^{(1)}, & \tilde{A}^{(2)}B^{(2)} \end{bmatrix}.$$

The number of block-columns grows exponentially as m^j. Using this definition we define the matrix $\mathscr{K}^p$ which we refer to as LPV extended controllability matrix.

Definition 3: The operator $\mathscr{L}_j$ is used to define the LPV extended controllability matrix:

$$\mathscr{K}^p = \begin{bmatrix} \mathscr{L}_p, & \mathscr{L}_{p-1}, & \cdots, & \mathscr{L}_1 \end{bmatrix}. \quad (7)$$

To present the factorized expression of the time-varying extended controllability matrix in Lemma 1, we still need the following two definitions:

Definition 4: We define the matrix:

$$P_{p|k} = \mu_{k+p-1} \otimes \cdots \otimes \mu_k \otimes I_{r+\ell}$$

where $\otimes$ is the Kronecker product defined in [27].

Now we define:

Definition 5: With Definition 4 we can define:

$$N_k^p = \begin{bmatrix} P_{p|k} & & & 0 \\ & P_{p-1|k+1} & & \\ & & \ddots & \\ 0 & & & P_{1|k+p-1} \end{bmatrix}, \quad (8)$$

with $N_k^p \in \mathbb{R}^{\tilde{q} \times p(r+\ell)}$.

Now we can state the following lemma:

Lemma 1: Given the model structure in (3)-(4) we use Definition 3, and 5 to obtain:

$$\overline{\mathscr{K}}_k^p = \mathscr{K}^p N_k^p,$$

where $\overline{\mathscr{K}}_k^p$ is the time-varying extended controllability matrix which equals Definition 1, N_k^p depends on the known scheduling sequence (8), and $\mathscr{K}^p$ is an unknown matrix defined in (7). Note that the number of columns of $\mathscr{K}^p$ (rows of N_k^p), denoted by $\tilde{q}$, increases exponentially with p according to the relation $\tilde{q} = (r+\ell)\sum_{j=1}^{p} m^j$.

Proof: Proof follows through straightforward computations. ■

IV. Closed-loop LPV identification

With the factorization defined in the previous section we now come to the core of this paper and present the LPV identification algorithm.

A. Regression problem

The first objective of the algorithm is to reconstruct the state sequence up to a similarity transformation. The state x_{k+p} is given by:

$$x_{k+p} = \phi_{p,k} x_k + \mathscr{K}^p N_k^p \overline{z}_k^p, \tag{9}$$

where $\phi_{p,k}$ is the transition matrix given in (5), $\mathscr{K}^p$ is the time-invariant LPV controllability matrix and the matrix N_k^p is a matrix solely depending on the scheduling sequence. The key approximation in this algorithm is that we assume that $\phi_{j,k} \approx 0$ for all $j \geq p$. Similar as in the LTI case it can be shown that if the system in (3)-(4) is uniformly exponentially stable the approximation error can be made arbitrarily small by making p large [18]. With this assumption the state x_{k+p} is approximately given by:

$$x_{k+p} \approx \mathscr{K}^p N_k^p \overline{z}_k^p. \tag{10}$$

In a number of LTI subspace methods it is well known to make this step [23], [28]. The input-output behavior is now approximately given by:

$$\begin{cases} y_{k+p} & \approx & C\mathscr{K}^p N_k^p \overline{z}_k^p + e_k, \\ y_{k+p+1} & \approx & C\mathscr{K}^{p+1} N_k^{p+1} \overline{z}_k^{p+1} + e_{k+1}, \\ \vdots & & \\ y_{k+p+f-1} & \approx & C\mathscr{K}^{p+f-1} N_k^{p+f-1} \overline{z}_k^{p+f-1} + e_{k+f-1}. \end{cases}$$

Now we define the stacked matrices Y_i and Z_i for all $i \in \{0, \cdots, f-1\}$.

$$Y_i = \begin{bmatrix} y_{p+1+i}, & \cdots, & y_{N-f+1-i} \end{bmatrix}, \tag{11}$$

$$Z_i = \begin{bmatrix} N_1^{p+i} \overline{z}_1^{p+i}, & \cdots, & N_{N-p-f+1}^{p+i} \overline{z}_{N-p-f+1}^{p+i} \end{bmatrix}. \tag{12}$$

If the matrix Z_i for all $i \in \{0, \cdots, f-1\}$ has full row rank the matrix $C\mathscr{K}^{p+i}$ can be estimated by solving the following linear problems:

$$\min_{C\mathscr{K}^{p+i}} ||Y_i - C\mathscr{K}^{p+i} Z_i||_F^2, \qquad \forall i \in \{0, \cdots, f-1\}. \tag{13}$$

for all $i \in \{0, \cdots, f-1\}$ and where $||\cdots||_F$ represents the Frobenius norm [29]. For finite p this linear problem will be biased due the approximation made in (10). In the LTI literature a number of papers appeared that studied the effect of the window size and although they proved the asymptotic properties of the algorithms (if $p \to \infty$ the bias disappears) it is hard to quantify the effect for finite p [22], [23], [30]. The product $\mathscr{K}^p Z_0$, which by definition represents the state sequence X, can not directly be estimated. In the LTI literature it is common practice to use the estimates of similar matrices as $C\mathscr{K}^{p+i}$ to construct the extended observability matrix times the controllability matrix. For the LPV situation a similar approach can be followed. However, in this case we look at the product $\Gamma^f \mathscr{K}^p$ with Γ^f defined in (6). This full block matrix can be constructed with Definition 2 to equal the following matrix

$$\Gamma^f \mathscr{K}^p = \begin{bmatrix} C\mathscr{K}^p \\ C\left(\tilde{A}^{(1)}\right)\mathscr{K}^p \\ \vdots \\ C\left(\tilde{A}^{(1)}\right)^{f-1}\mathscr{K}^p \end{bmatrix}. \tag{14}$$

Observe that from the linear problems formulated in (13) and the factorizations presented in Definition 2 we can construct the $(i+1)^{th}$ block row of $\Gamma^f \mathscr{K}^p$ from the estimate of $C\mathscr{K}^{p+i}$. From which we can construct $\Gamma^f \mathscr{K}^p Z_0$ which equals by definition the extended observability matrix times the state sequence, $\Gamma^f X$. The construction of $\Gamma^f \mathscr{K}^p$ is a rather cumbersome task. However, when we introduce the kernel method the construction of this matrix significantly simplifies.

In this particular step the big difference with LPV-PBSID$_{\text{opt}}$ appears [21]. In LPV-PBSID$_{\text{opt}}$ the assumption that $\phi_{k,j} = 0$ if $j \geq p$ is also used in the construction of $\Gamma^f \mathscr{K}^p$. This assumption results in an upper block triangular matrix representation of $\Gamma^f \mathscr{K}^p$ and this representation only contains elements from $C\mathscr{K}^p$. Consequently, we only have to solve the linear problem in (13) for $i = 0$ which significantly simplifies the identification algorithm.

By computing a Singular Value Decomposition (SVD) of $\widehat{\Gamma^f \mathscr{K}^p} Z_0$ we can estimate the state sequence and the order of the system. We will use the following SVD:

$$\widehat{\Gamma^f \mathscr{K}^p} Z_0 = \begin{bmatrix} U & U_{\sigma\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V \\ V_\perp \end{bmatrix}, \tag{15}$$

where Σ_n is the diagonal matrix containing the n largest singular values and V is the corresponding row space. Note that we can find the smallest singular values by detecting a gap between the singular values [31]. The state is now estimated by:

$$\widehat{X} = \Sigma_n V. \tag{16}$$

It is well known that when the state, input, output, and scheduling sequence are known the LPV system matrices can be estimated [18]. First we use (2) which is now a linear relation in C and where e_k represents white noise. From this equation an estimate can be found of the C matrix while also the noise sequence can be estimated. The estimated noise sequence is used to transform (1) into a linear expression

TABLE I
TOTAL NUMBER OF ROWS IN THE MATRIX Z_i FOR $m = 4$ AND $r = l = 1$

	i=0	i=1	i=2	i=3	i=4
p=2	40	168	680	2728	10920
p=3	168	680	2728	10920	43688
p=4	680	2728	10920	43688	174760
p=5	2728	10920	43688	174760	699048

depending on $A^{(i)}$, $B^{(i)}$, and $K^{(i)}$ and consequently all the system matrices can be estimated.

B. Curse of dimensionality

Like in [18] the method suffers from the curse of dimensionality. The number of rows of Z_i grows exponentially with the past and future window. The number of rows is given by:

$$\rho_{Z_i} = (r+\ell)\sum_{j=1}^{p+i} m^j.$$

In Table I the curse of dimensionality is illustrated. Observe that the growth of the dimensions is almost similar to the work of [18] (see Table 1 in [18]) . In the LPV-PBSID$_{\text{opt}}$ version of the proposed algorithm only one least squares problem should be solved which coincides with a future window of 1. So we can conclude that the computational complexity of the proposed algorithm is significantly higher. In LPV-PBSID$_{\text{opt}}$ the dimensional grows rapidly and the kernel method was proposed. A similar thing can be done for LPV-PBSID, but first we summarize the algorithm.

C. Summary of the algorithm

Algorithm 1 (LPV-PBSID): The algorithm can be summarized as follows:

1) Create the matrices Y_i, and Z_i using (11) and (12),
2) Solve the linear problems given in (13),
3) Construct $\Gamma^f \mathscr{K}^p Z_0$ using (12) and (14),
4) Compute the state sequence using (15) and (16),
5) With the estimated state use the linear relations (1)-(2) to obtain the system matrices.

V. KERNEL METHOD

The LPV identification method presented in the previous paragraph suffers from the curse of dimensionality. However, like in [19] we can use the kernel method to overcome this drawback. In Section V-A we present the kernel method for the proposed LPV identification scheme. In Section V-B a computationally efficient formula is presented for the proposed model structure. The kernel method is normally ill-conditioned, but in Section V-C regularization is proposed to overcome this drawback. In the last subsection a summary of the algorithm with kernels is given.

A. Kernel Method

The LPV identification approach presented in the previous section resulted in a set of linear problems formulated in (13). This equation can be solved by using traditional Least Squares (LS). However, the data matrices grow exponentially with the past and future window, p and f. In [19] it was shown that the solution of this least squares problem is equal to the solution of the dual problem if the solution with the minimum two norm is considered. In this subsection we show how the kernel method can be exploited for the presented LPV identification scheme.

The least squares problem in (13) has a unique solution if the matrix Z_i has full row rank and is given by:

$$\widehat{C\mathscr{K}^{p+i}} = Y_i Z_i^T \left(Z_i Z_i^T\right)^{-1}.$$

When the matrix Z_i has no full rank the solution is not unique. This will occur when the past window is large. However, the solution with the smallest norm, $\min ||C\mathscr{K}^{p+i}||_F^2$, can still be computed by using the SVD of the matrix:

$$Z_i = \begin{bmatrix} U_i & U_{i,\perp} \end{bmatrix} \begin{bmatrix} \Sigma_{i,n} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_i^T \\ V_{i,\perp}^T \end{bmatrix}, \tag{17}$$

where $\Sigma_{i,n}$ is the diagonal matrix containing the largest singular values and V_i^T and U_i are the corresponding row and column space, respectively. The solution with the minimum-norm is now given by:

$$\widehat{C\mathscr{K}^{p+i}} = Y_i V_i \Sigma_{i,n}^{-1} U_i^T. \tag{18}$$

The computations take place in a large dimensional space spanned by the columns of Z_i. If we consider the minimum norm solution of (13) the dual problem [32] avoids computations in this large dimensional space. The dual problem results in:

$$\min_{\alpha_i} \|\alpha_i\|_F^2 \quad \text{with} \quad Y_i - \alpha_i Z_i^T Z_i = 0, \tag{19}$$

where α_i are the Lagrange Multipliers and $Z_i^T Z_i$ is referred to as the kernel matrix. If the matrix Z_i has full column rank the solution to this dual problem is given by:

$$\widehat{\alpha}_i = Y_i \left(Z_i^T Z_i\right)^{-1} \tag{20}$$

$$= Y_i V_i \Sigma_{i,n}^{-2} V_i^T. \tag{21}$$

The estimate of $C\mathscr{K}^{p+i}$ is now given by:

$$\widehat{C\mathscr{K}^{p+i}} = \widehat{\alpha} Z_i^T,$$
$$= Y_i V_i \Sigma_{i,n}^{-1} U_i^T.$$

Because we are not interested in $\widehat{C\mathscr{K}^{p+i}}$ but in $\widehat{C\mathscr{K}^{p+i}}Z_0$, we do not have to construct the matrices Z_i explicitly, we only have to construct $Z_i^T Z_i$ for the computation of α_i. The dual least squares problem can be solved and we can construct the matrix:

$$\widehat{C\mathscr{K}^{p+i}}Z_i = \hat{\alpha}_i (Z_i)^T Z_i, \tag{22}$$

with this above we can not reconstruct the extended controllability matrix times the state sequence directly. In the previous section we used this estimate to build the matrix in (14). We mentioned that this was a rather cumbersome task. However, in the next lemma things clear up.

Lemma 2: Given the model structure in (3)-(4) and the Definition 4. Now we have

$$C\left(\tilde{A}^{(1)}\right)^i \mathscr{K}^p = \alpha_i Z_0^T. \tag{23}$$

So with $\alpha = \begin{pmatrix} \alpha_0^T, & \alpha_1^T, & \cdots, & \alpha_{f-1}^T \end{pmatrix}^T$ we can construct the matrix $\Gamma^f \mathscr{K}^p Z_0$ as follows

$$\Gamma^f \mathscr{K}^p Z_0 = \alpha Z_0^T Z_0 \tag{24}$$

Proof: The proof follows from the derivation of the dual problem. ■

With this lemma we can go back to the original problem and compute an SVD of this estimate to find the state sequence. Again it is important to stress that we do not require Z_i but we only need $Z_i^T Z_i$. This observation makes it possible to derive a computationally more efficient implementation.

B. Computation of the kernel matrices

In the previous subsection it was already stressed that we do not have to compute Z_i but we only need $Z_i^T Z_i$. In this section an analytic expression is given which does not require the calculation of Z_i. First we define the matrix $\tilde{N} = \begin{bmatrix} 1 & 2 & \cdots & N-p-f+1 \end{bmatrix}$, and the following Lemma

Lemma 3: Given the vectors $\lambda_1, \lambda_2, \cdots, \lambda_\nu \in \mathbb{R}^{\kappa \times 1}$ and $\theta_1, \theta_2, \cdots, \theta_\nu \in \mathbb{R}^{\kappa \times 1}$ the product

$$(\lambda_1 \otimes \lambda_2 \otimes \cdots \otimes \lambda_\nu)^T (\theta_1 \otimes \theta_2 \otimes \cdots \otimes \theta_\nu), \tag{25}$$

is given by

$$\prod_{j=1}^{\nu} \lambda_j^T \theta_j \tag{26}$$

Proof: With the properties of the Kronecker product ($\otimes$) defined in [27], $(A \otimes B)(C \otimes D) = AC \otimes BD$, we can rewrite (25) as: $\lambda_1^T \theta_1 \otimes \cdots \otimes \lambda_\nu^T \theta_\nu$ and observing that all the elements between the Kronecker products are scalers results in (26). ■

With Lemma 3 we can define the kernels for the model structure given in (1)-(2)

Theorem 1 (Kernels LPV): Given Lemma 3 and the model structure given in (1)-(2) we have for $i \in \{0, \cdots, f-1\}$

$$Z_i^T Z_i = \sum_{q=0}^{p-1+i} \left(\left(\prod_{v=q}^{p-1+i} \mu_{\overline{N}+v}^T \mu_{\overline{N}+v} \right) \left(z_{\overline{N}+q}^T z_{\overline{N}+q} \right) \right) \tag{27}$$

We can solve (19) and construct (24) .

Proof: Using Lemma 3 the proof follows by straightforward computations ■

C. Regularization

The kernel $Z_i^T Z_i$ described in the previous paragraph is square and has the size of the number of data points available. This normally leads to an ill-conditioned set of equations. This conditioning problem can be circumvented by regularization. There are a number of regularization techniques. In [19] a simulation study is performed to select the optimal regularization technique and corresponding regularization parameter selection method. In this study they concluded that Tikhonov regularization with generalized cross validation regularization parameter selection gives the best result and this is what we use in the simulations.

D. Summary of the algorithm

We end this chapter with the summary of the closed-loop kernel LPV identification algorithm.

Algorithm 2 (LPV-PBSID (kernel)): The algorithm can be summarized as follows:

1) Create the matrices $Z_i^T Z_i$ using Theorem 1.
2) Solve the linear problems given in (19). If desired regularized.
3) Construct $\Gamma^f \mathscr{K}^p Z_0$ using (24)
4) Compute the state sequence using (15) and (16)
5) With the estimated state use the linear relations (1)-(2) to obtain the system matrices.

VI. Simulation results

In this section we show some of the features of the new algorithm on a simulation example and we compare the performance with LPV-PBSID$_{\text{opt}}$.

We have tested the proposed LPV identification method on the benchmark model used in [18], [19], [21]. This is a fourth order MIMO open-loop LPV model with $m = 4$, $r = 2$, and $l = 3$. The collected data u_k, y_k, and μ_k are used for the identification algorithm. The algorithm described in Algorithm 2 with Tikhonov regularization is used to identify an LPV model.

The performance of the identified system is evaluated by looking at the value of the variance-accounted-for (VAF) on a data set different from the one used for identification. The VAF value is defined as:

$$VAF = \max\left\{1 - \frac{var(y_k - \hat{y}_k)}{var(y_k)}, 0\right\} * 100\%$$

where $\hat{y}_k$ denotes the output signal obtained by simulating the identified LPV system, y_k is the output signal of the true LPV system, and $var()$ denotes the variance of a quasi-stationary signal. To investigate the sensitivity of the identification algorithm with respect to noise, a Monte-Carlo simulation with 100 runs was carried out. For each of the 100 simulations a different realization of the input u_k and scheduling sequence μ_k is used.

In Table II the results of the different identification methods are summarized (where the VAF values are based on a validation data set). If we look at the identification results for the system with $N = 1000$ and no noise (SNR=∞) the results are significantly better than the results presented in [19]. However, the LPV-PBSID$_{\text{opt}}$ algorithm performs in general better. It is questionable if this is a fair comparison because we assumed that the past window equals the future window. In LPV-PBSID$_{\text{opt}}$ there is no future window and this indicates that the future window is an additional tuning parameter in the LPV-PBSID algorithm. In Table III where the past and future windows are decoupled this statement is confirmed.

VII. Conclusions

In this paper we presented a novel Linear Parameter Varying (LPV) subspace identification method which is an extension of LTI Prediction Based Subspace IDentification (PBSID). The methodology from LTI PBSID is used to

TABLE II

THE VAF ON A FRESH DATA SET FOR 100 MONTE CARLO SIMULATIONS. THE EXPERIMENTS ARE PERFORMED FOR DIFFERENT SETTINGS

$p = f = 3$	LPV-PBSID+reg.			LPV-PBSID$_{\text{opt}}$+reg.		
	Output 1	Output 2	Output 3	Output 1	Output 2	Output 3
N=500, SNR=40	94.1	94.2	94.2	96.8	96.7	96.8
N=500, SNR=∞	99.9	99.9	99.9	99.9	100.0	100.0
N=1000, SNR=40	97.3	97.3	97.3	98.4	98.4	98.5
N=1000, SNR=∞	100	100	100	99.9	99.9	99.9

TABLE III

THE VAF ON A FRESH DATA SET FOR 100 MONTE CARLO SIMULATIONS FOR A DECOUPLED SETTING OF F AND P

$p = 2,\ f = 3$	LPV-PBSID+reg.		
	Output 1	Output 2	Output 3
N=1000, SNR=40	98.4	98.3	98.4

formulate the input-output behavior of an LPV system. From this input-output behavior the LPV equivalent of the Markov parameters can be estimated. We showed that with this estimate the product between the observability and state sequence can be reconstructed and an SVD can be used to estimate the state sequence and consequently the system matrices. The curse of dimensionality in subspace LPV identification appeared and the kernel method was proposed. A computational efficient representation of the kernel is presented which makes the approach numerical attractive. We also showed the similarities with LPV-PBSID$_{\text{opt}}$ and on a benchmark problem we evaluated their performances.

VIII. ACKNOWLEDGMENTS

This research is supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Ministry of Economic Affairs.

REFERENCES

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

[2] K. Zhou, J.C. Doyle, and K. Glover. *Robust and optimal control*. Upper Saddle River, New Jersey, 1996.

[3] P. Apkarian and R. J. Adams. Advanced gain scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, 6(1):21–32, 1998.

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

[5] J. M. Barker and G. J. Balas. Comparing linear parameter-varying gain-scheduled control techniques for active flutter suppression. *Journal of Guidance, Control, and Dynamics*, 23(5):948–955, 2000.

[6] F.D. Bianchi, R.J. Mantz, and C.F. Christiansen. Gain scheduling control of variable-speed wind energy conversion systems using quasi-LPV models. *Control Engineering Practice*, 13(2):247–255, 2005.

[7] M. Groot Wassink, M. Van De Wal, C. W. Scherer, and O. Bosgra. LPV control for a wafer stage: beyond the theoretical solution. *Control Engineering Practice*, 13(2):231–245, 2005.

[8] M. Lovera and G. Mercere. Identification for gain-scheduling: balanced subspace approach. In *Proceedings of the American Control Conference (ACC)*, New York City, USA, 2007.

[9] R. Tóth, F. Felici, P. S. C. Heuberger, and P. M. J. Van den Hof. Discrete time LPV I/O and state-space representations, differences of behavior and pitfalls of interpolation. In *Proceedings of the European Control Conference (ECC)*, Kos, Greece, 2007.

[10] V. Verdult. *Nonlinear system identification; A state-space approach*. Ph.D. thesis, University of Twente, 2002.

[11] B. Bamieh and L. Giarre. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12(9):841–853, 2002.

[12] F. Previdi and M. Lovera. Identification of nonlinear parametrically varying models using seperable least squares. *International Journal of Control*, 77(16):1382–1392, 2004.

[13] R. Tóth, P. S. C. Heuberger, and P. M. J. Van Den Hof. LPV system identification with globally fixed orthonormal basis functions. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA, 2007.

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

[15] J. W. van Wingerden, F. Felici, and M. Verhaegen. Subspace identification of MIMO LPV systems using piecewise constant scheduling sequence with hard/soft switching. In *Proceedings of the European Control Conference (ECC)*, Kos, Greece, 2007.

[16] J. W. van Wingerden and M. Verhaegen. Subspace identification of bilinear systems using a dedicated input sequence. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, New Orleans, USA, 2007.

[17] P. L. dos Santos, J. A. Ramos, and J. L. M. de Carvalho. Identification of bilinear systems using an iterative deterministic-stochastic subspace approach. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC), and the European Control Conference (ECC)*, Seville, Spain, 2005.

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

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

[20] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using non-linear programming. *Journal of Dynamic Systems, Measurements and control*, 121(1):71–78, 1999.

[21] J. W. van Wingerden and M. Verhaegen. Subspace identification of multivariable LPV systems: a novel approach. In *Proceedings of the IEEE Multi-conference on Systems and Control (MSC)*, San Antonio, USA, 2008.

[22] A. Chiuso and G. Picci. Consistency analysis of certain closed-loop subspace identification methods. *Automatica*, 41(3):377–391, 2005.

[23] A. Chiuso. The role of vector auto regressive modeling in predictor based subspace identication. *Automatica*, 43(6):1034–1048, 2007.

[24] V. Verdult, L. Ljung, and M. Verhaegen. Identification of composite local linear state-space models using a projected gradient search. *International Journal of Control*, 75(16):1385–1398, 2002.

[25] J. W. van Wingerden. *Control of Wind Turbines with 'Smart' Rotors: Proof of Concept & LPV Subspace Identification*. PhD thesis, Delft University of Technology, 2008.

[26] W.J. Rugh. *Linear System Theory*. Upper Sadle River, 1996.

[27] J. W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on circuits and systems*, 25(9):772–781, 1978.

[28] P. Van Overschee and B De Moor. *Subspace Identification for linear systems*. Kluwer Academic Publishers, 1996.

[29] G. H. Golub and C. F. van Loan. *Matrix computations*. The John Hopkins University Press, 1996.

[30] T. Knudsen. Consistency analysis of subspace identification methods based on a linear regression approach. *Automatica*, 37(1):81–89, 2001.

[31] M. Verhaegen and V. Verdult. *Filtering and System Identification: An introduction*. 2007.

[32] J. A. K. Suykens, T. V. van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least squares support vector machines*. World Scientific, 2002.