Automatica 45 (2009) 372–381

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Subspace identification of Bilinear and LPV systems for open- and closed-loop data☆

Jan-Willem van Wingerden *, Michel Verhaegen

Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628CD Delft, The Netherlands

ARTICLE INFO

Article history:
Received 31 January 2008
Received in revised form
30 May 2008
Accepted 29 August 2008
Available online 17 December 2008

Keywords:
Identification
Subspace methods
Closed-loop identification
LPV systems
Time-varying systems

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 a predictor that predicts the output, which is based on past inputs, outputs, and scheduling data. The predictor contains the LPV equivalent of the Markov parameters. Using this predictor, ideas from closed-loop LTI identification are developed to estimate the state sequence from which the LPV system matrices can be constructed. A numerically efficient implementation is presented using the kernel method. It turns out that if structure is present in the scheduling sequence the computational complexity reduces even more.

1. Introduction

From a system theoretic point of view the identification and control of Linear Parameter-Varying (LPV) systems have attracted considerable attention in the past years, see Bamieh and Giarre (2002), Felici, van Wingerden, and Verhaegen (2007), Lee and Poolla (1999), Liu (1997), Verdult (2002), and Apkarian and Adams (1998), Scherer (2001), Shamma and Athans (1991), Zhou, Doyle, and Glover (1996) respectively. Recently, a number of applications of such systems were published: compressors (Giarre, Bauso, Falugi, & Bamieh, 2006), wind turbines (Bianchi, Mantz, & Christiansen, 2005; van Wingerden, Houtzager, Felici, & Verhaegen, 2009), aerospace applications (Barker & Balas, 2000), biomedical applications (Takahashi & Massaquoi, 2007), and motion platforms (Groot Wasink, Van De Wal, Scherer & Bosgra, 2005) are systems depending on a known scheduling vector, which shows the potential of LPV system theory for industrial applications.

Some of the applications are unstable by nature and have to operate in closed loop before they can be identified, *e.g.* aerospace applications (Barker & Balas, 2000) and wind turbines (Bianchi et al., 2005; van Wingerden et al., 2009). 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 (Groot Wassink et al., 2005; Lovera & Mercere, 2007). In Tóth, Felici, Heuberger, and Van den Hof (2007a) 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 based identification algorithm to identify LPV systems under closed-loop conditions which does not require interpolation or identification of local models.

The identification of LPV systems with arbitrarily varying scheduling sequences has proven to be challenging from a numerical point of view (Verdult & Verhaegen, 2002). 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 (Verdult & Verhaegen, 2005). For optimization based algorithms (Lee & Poolla, 1999) these models appeared to be a good starting point. Recently, a number of papers appeared where the structure of the scheduling sequence is exploited. It turns out that if the scheduling is periodic (Felici et al., 2007; van Wingerden et al., 2009), piecewise constant (van Wingerden & Verhaegen, 2007; van Wingerden, Felici, & Verhaegen, 2007), or white noise (Favoreel, De Moor, & Van Overschee, 1999; Santos, Ramos, & Carvalho, 2005), well-established LTI subspace techniques can be used to identify LPV or bilinear systems.

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Wolfgang Scherrer under the direction of Editor Torsten Söderström.

* Corresponding author. Tel.: +31 0 15 27 81720; fax: +31 15 27 86679.
E-mail addresses: j.w.vanwingerden@tudelft.nl (J.W. van Wingerden), m.verhaegen@moesp.org (M. Verhaegen).

0005-1098/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2008.08.015

The literature on the identification of LPV systems so far is dedicated to the open-loop setting. An overview of literature in the area of open-loop LPV identification is given in Verdult (2002). More recent work can be found in Borges, Verdult, Verhaegen, and Botto (2004) where they use separable least squares to identify an LPV model in the state space setting. In the input–output setting, work can be found in Bamieh and Giarre (2002), Previdi and Lovera (2004) and Tóth, Heuberger, and Van den Hof (2007b). 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.

The first main contribution of this paper is that we present a novel set of algorithms to identify LPV systems for the open- and closed-loop setting. We present a factorization which makes it possible to formulate a predictor that contains the LPV equivalent of the Markov parameters. This contribution is an extension of the work in van Wingerden and Verhaegen (2008b). The second main contribution is that we present an approach that stays close to the formulations given in Chiuso (2007). The computational complexity of the algorithm is significantly smaller than the algorithms in Verdult and Verhaegen (2002), but still the dimensions grow exponentially. Similar to what is done in Verdult and Verhaegen (2005) we present the kernel method to reduce the computational complexity. However, we derive computationally efficient formulations of the kernels, which is the third main contribution of this paper. We also show that if the scheduling sequence is periodic, piecewise constant, or structured in some sense, the identification procedure significantly simplifies even more from a computational point of view.

The outline of this paper is as follows; we start in Section 2 with the problem formulation and assumptions. In Section 3 we present a factorization that separates the unknown system matrices from the known input, output, and scheduling data. In Section 4 the basic idea behind the closed-loop identification scheme is presented and the curse of dimensionality will appear. In Section 5 the kernel method is presented, where compact formulations of the kernels are presented. In Section 6 we show that dedicated scheduling sequences significantly reduce the computational complexity. In Section 7 three simulation examples are presented that show the potential of the proposed algorithm. We end this paper with our conclusions.

2. Problem formulation and assumptions

In this section we present the problem formulation and assumptions.

2.1. Problem formulation

In this paper we consider LPV systems with parameter-independent output equation. This is done because many practical LPV systems have a parameter-independent output equation and to keep the notation and derivation of the algorithm simple. However, using a similar derivation as presented in this paper, similar results can be derived for the model structure with LPV output equation (van Wingerden, 2008). In this paper we focus on the following model:

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

$$y_k = C x_k + D u_k + e_k, \tag{2}$$

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 $A^{(i)} \in \mathbb{R}^{n\times n}$, $B^{(i)} \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{\ell\times n}$, $D \in \mathbb{R}^{\ell\times r}$, $K^{(i)} \in \mathbb{R}^{n\times \ell}$ 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, 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:

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

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

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

$$y_k = C x_k + D u_k + e_k, \tag{4}$$

with

$$\tilde{A}^{(i)} = A^{(i)} - K^{(i)} C, \qquad \tilde{B}^{(i)} = B^{(i)} - K^{(i)} D.$$

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 $T \in \mathbb{R}^{n\times n}$: $T^{-1}A^{(i)}T$, $T^{-1}B^{(i)}$, $T^{-1}K^{(i)}$, CT, and D.

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 $k = \{1, \ldots, N\}$; find, if they exist, the LPV system matrices $A^{(i)}$, $B^{(i)}$, $K^{(i)}$, C, and D for all $i \in \{1, 2, \ldots, m\}$ up to a global similarity transformation.

2.2. Assumptions and notation

First we define the transition matrix for discrete-time time-varying systems (Rugh, 1996) and this is given by:

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

To make the notation more transparent we define: $z_k = \begin{bmatrix} u_k^{\mathrm{T}}, & y_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $\breve{B}_k = \begin{bmatrix} \tilde{B}_k, & K_k \end{bmatrix}$, and $\overline{B}^{(i)} = \begin{bmatrix} \tilde{B}^{(i)}, & K^{(i)} \end{bmatrix}$. Similar as in Jansson (2005) and Chiuso (2007) we define a past window denoted by p. This window is used to define the following stacked vector:

$$\overline{z}_k^p = \begin{bmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{k+p-1} \end{bmatrix}.$$

We assume that the state sequence:

$$X = \begin{bmatrix} x_{p+1}, & \ldots, & x_N \end{bmatrix},$$

has full row rank and the matrix:

$$\Gamma^p = \begin{bmatrix} C \\ C\tilde{A}^{(1)} \\ \vdots \\ C\left(\tilde{A}^{(1)}\right)^{p-1} \end{bmatrix}, \tag{6}$$

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:

$$\text{rank}\left(\begin{bmatrix}\mu_1, & \mu_2, & \dots, & \mu_{N-p+1}\end{bmatrix}\right) = m,$$

and $N - p + 1 > m$. 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 4 but first we define a factorization to extend the framework described in Chiuso (2007) to LPV systems.

3. Factorizations

In this section we define a fundamental factorization in which we separate the unknown system matrices from the known scheduling sequence.

We will factorize the time-varying extended controllability matrix which is defined by:

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

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

with $\overline{\mathcal{K}}_k^p \in \mathbb{R}^{n\times(r+\ell)p}$.

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 $\overline{B}^{(i)}$. 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:

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

with

$$\mathcal{L}_1 = \begin{bmatrix}\overline{B}^{(1)}, & \dots, & \overline{B}^{(m)}\end{bmatrix},$$

and $\mathcal{L}_j \in \mathbb{R}^{n\times(r+\ell)m^j}$.

To illustrate this definition see the following example:

Example 3. For $m = 2$ one obtains:

$$\mathcal{L}_1 = \begin{bmatrix}\overline{B}^{(1)}, & \overline{B}^{(2)}\end{bmatrix},$$
$$\mathcal{L}_2 = \begin{bmatrix}\tilde{A}^{(1)}\overline{B}^{(1)}, & \tilde{A}^{(1)}\overline{B}^{(2)}, & \tilde{A}^{(2)}\overline{B}^{(1)}, & \tilde{A}^{(2)}\overline{B}^{(2)}\end{bmatrix}.$$

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

Definition 4. The operator $\mathcal{L}_j$ is used to define the LPV extended controllability matrix:

$$\mathcal{K}^p = \begin{bmatrix}\mathcal{L}_p, & \mathcal{L}_{p-1}, & \dots, & \mathcal{L}_1\end{bmatrix} \in \mathbb{R}^{n\times\tilde{q}},$$

with $\tilde{q} = (r+\ell)\sum_{j=1}^{p} m^j$.

To present the factorized expression of the time-varying extended controllability matrix in Lemma 7, we still need the following two definitions:

Definition 5. We define the matrix:

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

with $P_{p|k} \in \mathbb{R}^{m^p(r+\ell)\times(r+\ell)}$ and $\otimes$ represents the Kronecker product (Brewer, 1978).

Now we define:

Definition 6. With Definition 5 we can define:

$$N_k^p = \begin{bmatrix}P_{p|k} & & & 0\\ & P_{p-1|k+1} & & \\ & & \ddots & \\ 0 & & & P_{1|k+p-1}\end{bmatrix}, \tag{7}$$

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

Now we can state the following lemma:

Lemma 7. *Given the model structure in (3)–(4) we use Definitions 4 and 6 to obtain:*

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

where $\overline{\mathcal{K}}_k^p$ is the time-varying extended controllability matrix, which equals Definition 1, N_k^p depends on the known scheduling sequence (7), and $\mathcal{K}^p$ is an unknown matrix defined in (4). Note that the number of columns of $\mathcal{K}^p$ (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. □

4. 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.

4.1. 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:

$$x_{k+p} = \phi_{p,k}x_k + \mathcal{K}^p N_k^p \bar{z}_k^p,$$

where $\phi_{p,k}$ is the transition matrix given in (5), $\mathcal{K}^p$ 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 $j \geq p$. For finite p this approximation might result in biased estimates. This approximation is often used in the LTI literature (*e.g.* N4SID (Van Overshee & De Moor, 1996), SSARX (Jansson, 2005), PBSID (Chiuso, 2007)) and it can be shown that, if the system in (3)–(4) is uniformly exponentially stable, the approximation error can be made arbitrarily small (Knudsen, 2001; Verdult & Verhaegen, 2002). With this approximation the state x_{k+p} is approximately given by:

$$x_{k+p} \approx \mathcal{K}^p N_k^p \bar{z}_k^p. \tag{8}$$

In a number of LTI subspace methods it is well known to make this step (Chiuso, 2007; Jansson, 2005). The input–output behavior is now approximately given by:

$$y_{k+p} \approx C\mathcal{K}^p N_k^p \bar{z}_k^p + Du_{k+p} + e_{k+p} := y_{k+p}^{(p)}. \tag{9}$$

Now we define the stacked matrices U, Y, and Z:

$$U = \begin{bmatrix} u_{p+1}, & \dots, & u_N \end{bmatrix}, \tag{10}$$

$$Y = \begin{bmatrix} y_{p+1}, & \dots, & y_N \end{bmatrix}, \tag{11}$$

$$Z = \begin{bmatrix} N_1^p \bar{z}_1^p, & \dots, & N_{N-p+1}^p \bar{z}_{N-p+1}^p \end{bmatrix}. \tag{12}$$

If the matrix $\begin{bmatrix} Z^{\mathrm{T}}, & U^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ has full row rank, the matrices $C\mathcal{K}^p$ and D can be estimated by solving the following linear regression problem:

$$\min_{C\mathcal{K}^p, D} \|Y - C\mathcal{K}^p Z - DU\|_F^2, \tag{13}$$

where $\|\cdots\|_F$ represents the Frobenius norm (Golub & van Loan, 1996). For finite p this linear problem will be biased due to the approximation made in (8). 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 (Chiuso, 2007; Chiuso & Picci, 2005; Knudsen, 2001). In Section 7 we demonstrate that a rather large approximation does not directly imply a large bias in the estimate of the system matrices.

4.2. Observability matrix times controllability matrix

The algorithm that we develop in this paper can be seen as the LPV counterpart of the PBSID$_{\text{opt}}$ algorithm (Chiuso, 2007; Chiuso & Picci, 2005). In the PBSID$_{\text{opt}}$ algorithm the LTI equivalent of $C\mathcal{K}^p$ is estimated to construct, approximately, the extended observability matrix times the extended controllability matrix (see (4.11) in Chiuso (2007)). For the LPV situation a similar approach can be followed. However, in this case we construct, approximately, the product between the extended observability matrix of the first local model, given in (6), and the extended LPV controllability matrix, given in Definition 4. This matrix can be constructed with Definition 2 to equal the following upper-block triangular matrix:

$$\Gamma^p \mathcal{K}^p \approx \begin{bmatrix} C\mathcal{L}_p & C\mathcal{L}_{p-1} & \cdots & C\mathcal{L}_1 \\ 0 & C\tilde{A}^{(1)}\mathcal{L}_{p-1} & \cdots & C\tilde{A}^{(1)}\mathcal{L}_1 \\ & & \ddots & \\ 0 & & & C\left(\tilde{A}^{(1)}\right)^{p-1}\mathcal{L}_1 \end{bmatrix}. \tag{14}$$

The zeros appear in this equation based on the approximation that $\phi_{j,k} \approx 0$ for all $j \geq p$ (similarly as in the LTI counterpart PBSID$_{\text{opt}}$ (Chiuso, 2007)).[1] Recall that based on Definitions 2 and 4 the following two relations hold:

$$C\mathcal{K}^p = \begin{bmatrix} C\mathcal{L}_p, & C\mathcal{L}_{p-1}, & \dots & C\mathcal{L}_1 \end{bmatrix},$$

$$C\mathcal{L}_p = \begin{bmatrix} C\tilde{A}^{(1)}\mathcal{L}_{p-1}, & \dots, & C\tilde{A}^{(m)}\mathcal{L}_{p-1} \end{bmatrix}.$$

With these two relations and the estimate of $C\mathcal{K}^p$, obtained from the linear problem in (13), we can construct the matrix in (14).

4.3. Estimation of the state sequence

Now we can compute $\Gamma^p \mathcal{K}^p Z$, which equals by definition the extended observability matrix times the state sequence, $\Gamma^p X$. Under the assumptions stated in Section 2.2 that X and Γ^p both have full rank and that $p\ell > n$, we can estimate the state sequence

[1] Without this approximation the algorithm becomes more complex and computationally intensive (see the LPV-PBSID algorithm in van Wingerden and Verhaegen (2008a), van Wingerden (2008)).

Table 1
Total number of rows in the matrix Z for $m = 4$ and $r = \ell$.

	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$
$p = 2$	40	80	120	160	200
$p = 3$	168	336	504	672	840
$p = 4$	680	1360	2040	2 720	3 400
$p = 5$	2728	5456	8184	10 912	13 640

and the order of the system based on a rank revealing Singular Value Decomposition (SVD). We will use the following SVD:

$$\widehat{\Gamma^p \mathcal{K}^p} Z = \begin{bmatrix} \mathcal{U} & \mathcal{U}_{\sigma\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma \end{bmatrix} \begin{bmatrix} V \\ V_\perp \end{bmatrix}, \tag{15}$$

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 largest singular values by detecting a gap between the singular values (Verhaegen & Verdult, 2007). 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 system matrices can be estimated (Lovera, 1997; Nemani, Ravikanth, & Bamieh, 1995; Verdult & Verhaegen, 2002). First we use (2) which is now a linear relation in C and D and where e_k represents white noise. From this equation an estimate can be found of the C and D matrix while also the noise sequence can be estimated. The estimated noise sequence is used to transform (1) into a linear expression depending on $A^{(i)}$, $B^{(i)}$, and $K^{(i)}$ and consequently all the system matrices can be estimated.

4.4. Curse of dimensionality

Like in Verdult and Verhaegen (2002) the method suffers from the curse of dimensionality. The number of rows of Z grows exponentially with the size of the past window. The number of rows is given by:

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

In Table 1 the curse of dimensionality is illustrated. Observe that the growth of the dimensions is considerably smaller compared to the work of Verdult and Verhaegen (2002) (see Table 1 in Verdult and Verhaegen (2002)). However, the dimensions still grow rapidly with increasing past window. To overcome this drawback the kernel method will be introduced in the next section, but first we summarize the algorithm.

4.5. Summary of the algorithm

Algorithm 1 (*LPV-PBSID*$_{\text{opt}}$). The algorithm can be summarized as follows:

(1) Create the matrices U, Y, and Z using (10)–(12),
(2) Solve the linear problem given in (13),
(3) Construct $\Gamma^p \mathcal{K}^p Z$ using (12) and (14),
(4) Estimate the state sequence using (15) and (16),
(5) With the estimated state, use the linear relations (1)–(2) to obtain the system matrices.

5. Kernel Method

The LPV identification method presented in the previous paragraph suffers from the curse of dimensionality. However, like in Verdult and Verhaegen (2005) we can use the kernel method to overcome this drawback. In Section 5.1 we present the kernel method for the proposed LPV identification scheme. In Section 5.2 a computationally efficient formula is presented for the proposed model structure. The kernel method is normally ill-conditioned, but in Section 5.3 regularization is proposed to overcome this drawback. In the last subsection a summary of the algorithm with kernels is given.

5.1. Kernel Method

The LPV identification approach presented in the previous section resulted in a linear problem formulated in (13), from now on also referred to as the primal problem. This equation can be solved by using traditional Least Squares (LS). However, the data matrices grow exponentially with the past window, p. In Verdult and Verhaegen (2005) it was shown that the solution of this primal problem is equal to the solution of the dual problem if the solution with the minimum-norm (Golub & van Loan, 1996) is considered. In this subsection we show how the kernel method can be exploited for the presented LPV identification scheme.

The linear problem in (13) has a unique solution if the matrix $\begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ has full row rank and is given by:

$$\begin{bmatrix} \widehat{C\mathcal{K}^p} & \widehat{D} \end{bmatrix} = Y \begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix} \left(\begin{bmatrix} Z \\ U \end{bmatrix} \begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix} \right)^{-1}.$$

When the matrix $\begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ has missing row rank the solution is not unique. This will occur when the past window is large. However, the solution with the smallest norm, $\min \| \begin{bmatrix} C\mathcal{K}^p, & D \end{bmatrix} \|_F^2$, can still be computed by using the SVD of the matrix:

$$\begin{bmatrix} Z \\ U \end{bmatrix} = \begin{bmatrix} \mathcal{U} & \mathcal{U}_\perp \end{bmatrix} \begin{bmatrix} \Sigma_m & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V^{\mathrm{T}} \\ V_\perp^{\mathrm{T}} \end{bmatrix}, \tag{17}$$

where Σ_m is the diagonal matrix containing the non-zero singular values and V^{T} and $\mathcal{U}$ are the corresponding row and column space, respectively. The solution with the minimum-norm is now given by:

$$\begin{bmatrix} \widehat{C\mathcal{K}^p} & \widehat{D} \end{bmatrix} = YV\Sigma_m^{-1}\mathcal{U}^{\mathrm{T}}. \tag{18}$$

The computations take place in a large dimensional space spanned by the columns of Z and U. If we consider the minimum-norm solution of (13) the dual problem (Suykens, van Gestel, DeBrabanter, DeMoor, & Vandewalle, 2002) avoids computations in this large dimensional space. The dual problem results in:

$$\min_{\alpha} \|\alpha\|_F^2 \quad \text{with } Y - \alpha \begin{bmatrix} Z^{\mathrm{T}}Z + U^{\mathrm{T}}U \end{bmatrix} = 0, \tag{19}$$

where α are the Lagrange Multipliers and $\begin{bmatrix} Z^{\mathrm{T}}Z + U^{\mathrm{T}}U \end{bmatrix}$is referred to as the kernel matrix. If the matrix $\begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$has full column rank the solution to this dual problem is given by:

$$\widehat{\alpha} = Y \left(\begin{bmatrix} Z^{\mathrm{T}}Z + U^{\mathrm{T}}U \end{bmatrix} \right)^{-1}, \tag{20}$$

$$= YV\Sigma_m^{-2}V^{\mathrm{T}}. \tag{21}$$

The estimate of $\begin{bmatrix} C\mathcal{K}^p & D \end{bmatrix}$ is now given by:

$$\begin{bmatrix} \widehat{C\mathcal{K}^p} & \widehat{D} \end{bmatrix} = \widehat{\alpha} \begin{bmatrix} Z^{\mathrm{T}} & U^{\mathrm{T}} \end{bmatrix}, \\ = YV\Sigma_m^{-1}\mathcal{U}^{\mathrm{T}}.$$

The construction of the matrix $\widehat{C\mathcal{K}^p}$ from the dual problem requires the matrix Z, explicitly. However, due to the curse of dimensionality this can lead to dimension problems. For the construction of $C\mathcal{K}^pZ$ we do not need the matrix Z explicitly, we only have to construct $Z^{\mathrm{T}}Z$ and $U^{\mathrm{T}}U$ for the computation of α and an estimate of $C\mathcal{K}^pZ$ is given by:

$$\widehat{C\mathcal{K}^p}Z = \hat{\alpha}Z^{\mathrm{T}}Z. \tag{22}$$

With this above we cannot reconstruct the extended controllability matrix times the state sequence directly. However, in the following lemma we show that with similar matrices as $Z^{\mathrm{T}}Z$ we can do this.

Lemma 8. *Given the model structure in (1)–(2), the assumptions in Section 2, and Definition 5, we can define the following matrices:*

$$Z^{i,j} = \begin{bmatrix} P_{p-j+1|j-i+1}z_{j-i+1}, & \ldots, & P_{p-j+1|\overline{N}+j-i}z_{\overline{N}+j-i} \end{bmatrix}, \tag{23}$$

with $Z^{i,j} \in \mathbb{R}^{m^{p-j+1}(r+\ell)\times\overline{N}}$ and $\overline{N} = N - p + 1$. Now we have:

$$C\left(\tilde{A}^{(1)}\right)^{i-1} \mathcal{L}_{p-j+1} = \alpha \left(Z^{i,j}\right)^{\mathrm{T}}, \tag{24}$$

and then we can construct the matrix $\Gamma^p\mathcal{K}^pZ$ as follows:

$$\Gamma^p\mathcal{K}^pZ = \begin{bmatrix} \alpha \sum_{j=1}^{p} \left(Z^{1,j}\right)^{\mathrm{T}} Z^{1,j} \\ \alpha \sum_{j=2}^{p} \left(Z^{2,j}\right)^{\mathrm{T}} Z^{1,j} \\ \vdots \\ \alpha \sum_{j=p}^{p} \left(Z^{p,j}\right)^{\mathrm{T}} Z^{1,j} \end{bmatrix}. \tag{25}$$

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 $Z^{i,j}$ but we only need $\left(Z^{i,j}\right)^{\mathrm{T}} Z^{i,j}$. This observation makes it possible to derive a computationally more efficient implementation.

5.2. Computation of the kernel matrices

In the previous subsection it was already stressed that we do not have to compute $Z^{i,j}$ but we only need $\left(Z^{i,j}\right)^{\mathrm{T}} Z^{i,j}$. In this section an analytical expression is given which does not require the calculation of $Z^{i,j}$. First we define the matrix $\tilde{N} = \begin{bmatrix} 1, & 2, & \ldots, & \overline{N} \end{bmatrix}$, and the following Lemma.

Lemma 9. *Given the vectors $\lambda_1, \lambda_2, \ldots, \lambda_\nu \in \mathbb{R}^{\kappa\times 1}$ and $\theta_1, \theta_2, \ldots, \theta_\nu \in \mathbb{R}^{\kappa\times 1}$ the product*

$$\left(\lambda_1 \otimes \lambda_2 \otimes \cdots \otimes \lambda_\nu\right)^{\mathrm{T}} \left(\theta_1 \otimes \theta_2 \otimes \cdots \otimes \theta_\nu\right), \tag{26}$$

is given by:

$$\prod_{j=1}^{\nu} \lambda_j^{\mathrm{T}}\theta_j. \tag{27}$$

Proof. With the properties of the Kronecker product ($\otimes$) defined in Brewer (1978), $(A \otimes B)(C \otimes D) = AC \otimes BD$, we can rewrite (26) as: $\lambda_1^{\mathrm{T}}\theta_1 \otimes \cdots \otimes \lambda_\nu^{\mathrm{T}}\theta_\nu$ and observing that all the elements between the Kronecker products are scalers results in (27). □

With Lemma 9 we can define the kernels for the model structure given in (1)–(2).

Theorem 10 (*Kernels LPV*). *Given Lemma 9 and the model structure given in (1)–(2) we have for* $j \geq i$:[2]

$$\left(Z^{i,j}\right)^{\mathrm{T}} Z^{1,j} = \left(\prod_{v=0}^{p-j} \mu_{\tilde{N}+v+j-i}^{\mathrm{T}} \mu_{\tilde{N}+v+j-1}\right) \left(z_{\tilde{N}+j-i}^{\mathrm{T}} z_{\tilde{N}+j-1}\right)$$

and

$$Z^{\mathrm{T}} Z = \sum_{j=1}^{p} \left(Z^{1,j}\right)^{\mathrm{T}} Z^{1,j}.$$

We can solve (19) and construct (25).

Proof. Using Lemma 9 the proof follows by straightforward computations. □

For $N \gg p$ and $N \gg m$ the computational complexity of the indirect computation, so first constructing Z and then computing the product $\left(Z^{\mathrm{T}} Z\right)$, is of order $\mathcal{O}(N^2 \hat{q})$ with $\hat{q} = \sum_{j=1}^{p} m^j$ and direct construction of $\left(Z^{\mathrm{T}} Z\right)$ is of the order $\mathcal{O}(N^2)$ which illustrates the computational efficiency of the algorithm.

The kernels are valid for the model structure given in (1)–(2). The whole derivation of the kernel matrices can be repeated for different model structures. For this derivation we refer to van Wingerden (2008).

5.3. Regularization

The kernel $Z^{\mathrm{T}} Z$ 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 through regularization. There are a number of regularization techniques (for a detailed overview see Sima (2006)). In Verdult and Verhaegen (2005) 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.

5.4. Summary of the algorithm

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

Algorithm 2 (*LPV-PBSID*$_{\mathrm{opt}}$ *(kernel)*). The algorithm can be summarized as follows:

(1) Create the matrices $\left(Z^{i,j}\right)^{\mathrm{T}} Z^{i,j}$ and $Z^{\mathrm{T}} Z$ using Theorem 10 and compute $U^{\mathrm{T}} U$ using (10),
(2) Solve the linear problem given in (19). If desired regularized,
(3) Construct $\Gamma^p \mathcal{K}^p Z$ using (25),
(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.

6. Dedicated scheduling sequences

Recently, a number of papers appeared where the structure of the scheduling sequence is exploited to overcome the computational complexity of the general LPV identification approach. It turns out that if periodic scheduling (Felici et al., 2007; van Wingerden et al., 2009), piecewise constant scheduling (van Wingerden & Verhaegen, 2007; van Wingerden et al., 2007), or white-noise scheduling (Favoreel et al., 1999; Santos et al., 2005) is used well-established LTI subspace techniques can be used to identify LPV systems or bilinear systems. However, in Felici et al. (2007), van Wingerden et al. (2007) and van Wingerden and Verhaegen (2007) they have to solve a numerical sensitive intersection problem and in van Wingerden et al. (2007) and van Wingerden and Verhaegen (2007) the local models are interconnected by formulating a number of least squares problems which require accurate local models. In Santos et al. (2005) an iterative approach is used to obtain the system matrices. In this section we show how the structure in the scheduling further reduces the computational complexity of the algorithm presented in this article.

6.1. Periodic scheduling

We will use the kernel algorithm to show that the computational complexity significantly reduces when the scheduling is periodic, piecewise constant or is structured in some sense.

The matrix $Z^{\mathrm{T}} Z$ is given by:

$$Q^{\mathrm{T}} \begin{bmatrix} \left(N_1^p\right)^{\mathrm{T}} N_1^p & \left(N_1^p\right)^{\mathrm{T}} N_2^p & \cdots & \left(N_1^p\right)^{\mathrm{T}} N_{\bar{N}}^p \\ \left(N_2^p\right)^{\mathrm{T}} N_1^p & \left(N_2^p\right)^{\mathrm{T}} N_2^p & \cdots & \left(N_2^p\right)^{\mathrm{T}} N_{\bar{N}}^p \\ \vdots & & \ddots & \\ \left(N_{\bar{N}}^p\right)^{\mathrm{T}} N_1^p & \left(N_{\bar{N}}^p\right)^{\mathrm{T}} N_2^p & \cdots & \left(N_{\bar{N}}^p\right)^{\mathrm{T}} N_{\bar{N}}^p \end{bmatrix} Q,$$

with $Q = \mathrm{diag}\left(\left[\bar{z}_1^p, \quad \bar{z}_2^p, \quad \ldots, \quad \bar{z}_{\bar{N}}^p\right]\right)$. If the system is periodic with period P we observe that $N_i^p = N_{i+P}^p$ and consequently the matrix $Z^{\mathrm{T}} Z$ will lose rank. In the kernel method we are only interested in the row space of $C\mathcal{K}^p Z$. In the previous section an estimate of $C\mathcal{K}^p Z$ was given by $\alpha Z^{\mathrm{T}} Z$: if the matrix $Z^{\mathrm{T}} Z$ has not full row rank we can select the rows that span the row space of this matrix without altering the row space of $C\mathcal{K}^p Z$. This means that we only have to select the rows of $Z^{\mathrm{T}} Z$ that span the row space of this matrix. For periodic scheduling it is sufficient to select the first $P \times (r + \ell)p$ rows if the matrix Q has full rank. For piecewise constant scheduling (van Wingerden et al., 2007) the same time-variation will be present a number of times and in this case the matrix $Z^{\mathrm{T}} Z$ will lose rank. For this situation it is harder to select the rows that span the row space of $Z^{\mathrm{T}} Z$. There are more scheduling sequences where the same or almost the same time-variation in the dynamics occurs, structured in some sense. In the next subsection we present a sketch of an algorithm how to select the right rows for structured scheduling sequences.

6.2. Kernel selection

To exploit structure, like periodic or piecewise constant scheduling, it is important to select the linear-independent rows of $Z^{\mathrm{T}} Z$. However, we can also select the rows that are independent up to a certain threshold, ϵ. In this way only the most dominant rows are selected. There are different methods to select the most dominant rows but we summarize the most straightforward approach in Algorithm 3.

Algorithm 3 (*Kernel Selection Algorithm*). The algorithm can be summarized as follows:[3]

[2] We define $\mu_{\tilde{N}+v+j-1} = \left[\mu_{1+v+j-1}, \quad \ldots, \quad \mu_{\bar{N}+v+j-1}\right]$.

[3] Matlab notation is used for convenience.

Init Give a tolerance level ϵ
define the matrix:
$\mathcal{Z} = \left((Z)^{\mathrm{T}}(Z)\right)(1,:) + \left((U)^{\mathrm{T}}(U)\right)(1,:)$
$\mathcal{D} = \{1\}$
For $j = 2 : \bar{N}$
Compute:
$\overline{\mathcal{Z}} = \left((Z)^{\mathrm{T}}(Z)\right)(j,:) + \left((U)^{\mathrm{T}}(U)\right)(j,:)$
If $\min_{\Xi} \|\mathcal{Z}\Xi - \overline{\mathcal{Z}}\|_F^2 > \epsilon$
$\mathcal{Z} = \begin{bmatrix} \mathcal{Z}^{\mathrm{T}} & \overline{\mathcal{Z}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$
Add j to the set $\mathcal{D}$
End
End

Solve $\min_{\tilde{\alpha}} \|Y - \tilde{\alpha}\mathcal{Z}\|_F^2$. The estimate of the extended observability matrix is now given by (25) where only the rows indicated by the set $\mathcal{D}$ of $\sum_{j=i}^{p} \left(Z^{i,j}\right)^{\mathrm{T}} Z^{1,j}$ are used. Similar as in the original algorithm an SVD can be computed to obtain an estimate of the state and consequently the system matrices can be estimated.

With this algorithm the number of selected rows can still be large. More advanced kernel selection algorithms can be used to select the most dominant rows of $Z^{\mathrm{T}}Z$. In Suykens et al. (2002) and Espinoza, Suykens, and De Moor (2006) they use fixed-sized kernels. In machine learning literature more methods can be found on how to select a fixed number of kernels (see for instance Smola and Scholkopf (2000)).

The advantage of the method presented above is that we can deal with large data sets when the row space of $Z^{\mathrm{T}}Z$ is small. In the next section we present simulation examples that show the potential of the proposed LPV identification algorithm. We also show that we can deal with a large data set if we have periodic scheduling.

7. Simulation results

In this section we show some features of the novel algorithm on three different simulation examples.

7.1. Open-loop LPV identification example

We have tested the proposed LPV identification on the benchmark model used in Verdult and Verhaegen (2002) and Verdult and Verhaegen (2005). 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 and without 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:

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

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 output and process 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 2 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 $= \infty$) the results are significantly better than the results presented in Verdult and Verhaegen (2005). Remarkably, we observe that the VAF is 100% for a finite p. However, if we compute the VAF between y_k and $y_k^{(p)}$ defined in (9), which basically indicates the approximation we made, the VAF values are significantly smaller. This stresses the point that although the approximation error is rather large we cannot predict how large the bias term is and how it transfers to the final estimate of the system matrices. That is why from a theoretical point of view we can state that it is better to pick a large p because then the consistency can be proven (Verdult & Verhaegen, 2002). However, a large p, due to the curse of dimensionality, and a finite $\bar{N}$ implies a large number of variables to be estimated leading to minimum-norm solutions with a larger variance. From this perspective it is better to choose a small p. In this particular simulation example this trade-off already appears for a small p. Estimation with a finite p is an interesting research field for both LTI and LPV system identification. In Section 7.3 we clearly show the trade-off between bias and variance on a simple bilinear example.

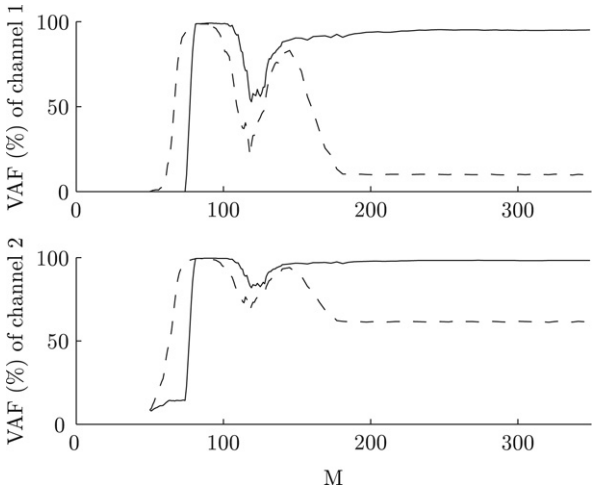

Fig. 1. The VAF for 100 Monte Carlo simulations as a function of the numbers of rows of $Z^{\mathrm{T}}Z$ that are taken into account. The solid line and the dashed line represent the regularized and the un-regularized version of the algorithm.

7.2. Closed-loop LPV identification example

In this section we show the operation of the algorithm with data collected in closed loop. We use the model described in Felici et al. (2007) which represents the flapping dynamics of a wind turbine blade (Eggleston & Stoddard, 1987). To apply state feedback we now assume that this model has an identity output matrix. We take a past window of 10 and limit ourselves to 2000 data points. To show the potential of the closed-loop setting a time-varying state feedback is used. The controller is synthesized using a discrete-time periodic Riccati equation (Hench & Laub, 1994; Varga, 2005). Process noise is added with a variance of 0.1.

An LPV model was identified using Algorithm 2 and we exploit the fact that the scheduling sequence contains structure, it is periodic. We take the first M rows of $Z^{\mathrm{T}}Z$ into account. As mentioned earlier there are smarter ways to select these rows however this will illustrate the effectiveness of the approach. In Fig. 1 the VAF value as a function of M is presented for the regularized and the un-regularized situation. We observe that we can select an optimum of 100 rows and we see that regularization is not required anymore.

In Fig. 2 the eigenvalues of the estimated models are compared with their true values. As expected the closed-loop algorithm gives

Table 2
The mean VAF on a fresh data set for 100 Monte Carlo simulations. The experiments are performed for different settings.

$p = 3$	Algorithm 2			Algorithm 2 + Tikhonov reg. + GCV		
	Output 1	Output 2	Output 3	Output 1	Output 2	Output 3
$N = 500$, SNR $= 40$	93.2	93.5	93.8	96.8	96.7	96.8
$N = 500$, SNR $= \infty$	100.0	100.0	100.0	99.9	100.0	100.0
$N = 1000$, SNR $= 40$	98.0	98.0	98.0	98.4	98.4	98.5
$N = 1000$, SNR $= \infty$	100.0	100.0	100.0	99.9	99.9	99.9
	VAF($y_k, y_k^{(p)}$)					
	Output 1	Output 2	Output 3			
$N = 1000$, SNR $= \infty$	54.0	34.5	15.7			
$p = 5$	Algorithm 2			Algorithm 2 + Tikhonov reg. + GCV		
	Output 1	Output 2	Output 3	Output 1	Output 2	Output 3
$N = 500$, SNR $= 40$	66.5	67.2	67.6	90.5	90.6	90.8
$N = 500$, SNR $= \infty$	99.4	99.4	99.4	99.2	99.3	99.3
$N = 1000$, SNR $= 40$	17.2	19.1	21.0	94.8	95.0	95.1
$N = 1000$, SNR $= \infty$	100.0	100.0	100.0	100.0	100.0	100.0
	VAF($y_k, y_k^{(p)}$)					
	Output 1	Output 2	Output 3			
$N = 1000$, SNR $= \infty$	70.8	67.5	56.8			

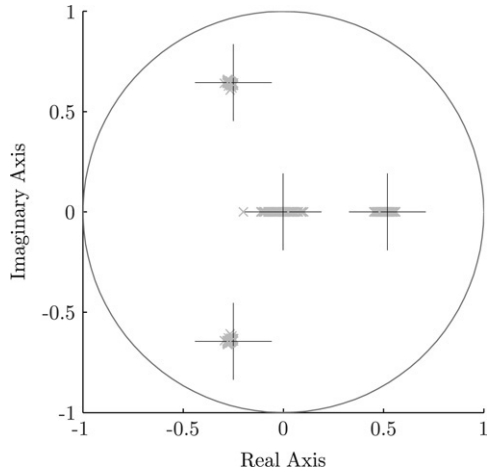

Fig. 2. Eigenvalues of the estimated $A^{(i)}$ matrices in the complex plane, for 100 experiments and $M = 100$. The big crosses correspond to the real values of the eigenvalues of the matrices.

consistent results but a small bias arises due to the approximation made in the algorithm.

7.3. Bilinear identification example

In the previous two examples we demonstrated a number of key features of the proposed algorithm. In this example we demonstrate the effect of the past window using a low-order open-loop bilinear model. As indicated in this article a small p will lead in general to biased estimates while for large p, even without noise, the variance will increase due to the curse of dimensionality. The bilinear system is given by:

$$\left[A^{(1)} \quad A^{(2)}\right] = \left[\begin{array}{cc|cc} 0.5 & 0.5 & 0.2 & 0.2 \\ -0.5 & 0.5 & -0.2 & 0.2 \end{array}\right],$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \end{bmatrix}.$$

The input is generated by filtering a zero-mean white-noise sequence with a fourth-order low pass Butterworth filter that has a cutoff frequency of 0.8 times the Nyquist frequency. Because we work with a bilinear system the scheduling is given by $\mu_k = \begin{bmatrix} 1, & u_k^T \end{bmatrix}^T$. To investigate the sensitivity of the identification algorithm with respect to the realization of the input and noise, a Monte Carlo simulation with 100 runs was carried out with $N = 500$. Compared with the previous two examples the complexity of this example is small, which enables us to clearly demonstrate the trade-off between a large p and a small p. In Fig. 3 we demonstrate the effect of p on the amount of approximation we introduce and the consequence for the identification algorithm. The top figure presents the VAF between y_k and $y_k^{(p)}$, defined in (9), which basically indicates the amount of approximation we incorporated in the algorithm. The system is uniformly exponentially stable and therefore the approximation error will converge to zero if p goes to infinity. In the plot in the middle the VAF is given between the signal generated by the estimated model, using noise free data, and the validation data. In this figure we see that for really small p we have a biased estimate, while for large p the variance increases due to the fact that we look for a minimum-norm solution because the number of unknowns exceeds the number of data points. In the bottom figure, noise with an SNR of 15 dB is added to the identification data and we clearly see a larger variance for a larger p. This simulation example illustrates the trade-off problem for the identification for both bilinear and LPV systems. With increasing complexity it is expected that it is more worthwhile to pick a small p, as already seen in the first example. However, if the bias is too large, the biased estimate may serve as an initial estimate for optimization based identification algorithms (Lee & Poolla, 1999), which was already suggested in Verdult and Verhaegen (2002).

8. Conclusions

In this paper we presented an open-loop and closed-loop LPV subspace identification method which is an extension of LTI subspace closed-loop identification methods. The methodology from closed-loop LTI subspace identification is used to 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

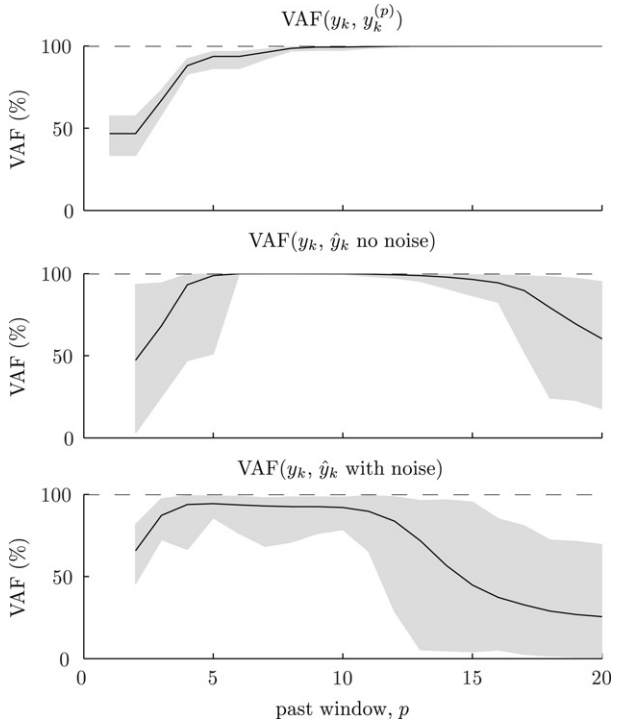

Fig. 3. The mean value of the VAF over 100 experiments (solid line) is presented with respect to the past window size. The VAF values of the 100 experiments are within the grey confidence region.

kernel method was proposed. A computational efficient representation of the kernel is presented which makes the approach numerical attractive. Furthermore, we showed that if there is structure in the scheduling, then the computational complexity reduces even more. The algorithm was illustrated with three simulation examples.

Acknowledgements

The authors wish to thank J.K.R. Rice, D.A. Joosten, R. Hallouzi, P. Massioni, B.A. Kulcsar, I. Houtzager and four anonymous reviewers for their comments and suggestions. This research is supported by the Technology Foundation STW, applied science division of NWO and the technology program of the Dutch Ministry of Economic Affairs.

References

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

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

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

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

Borges, J., Verdult, V., Verhaegen, M., & Botto, M. A. (2004). Separable least squares for projected gradient identification of composite local linear state-space models. In *Proceedings of the 16th international symposium on mathematical theory of networks and systems*.

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

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

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

Eggleston, D. M., & Stoddard, F. S. (1987). *Wind turbine engineering design*. Van Nostrand Reinhold.

Espinoza, M., Suykens, J. A. K., & De Moor, B. (2006). Fixed-size least squares support vector machines: A large scale application in electrical load forecasting. *Computational Management Science*, *3*(2), 113–129.

Favoreel, W., De Moor, B., & Van Overschee, P. (1999). Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, *44*(6), 1157–1165.

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

Giarre, L., Bauso, D., Falugi, P., & Bamieh, B. (2006). LPV model identification for gain scheduling control: An application to rotating stall and surge control problem. *Control Engineering Practice*, *14*(4), 351–361.

Golub, G. H., & van Loan, C. F. (1996). *Matrix computations*. The John Hopkins University Press.

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

Hench, J. J., & Laub, A. J. (1994). Numerical solution of the discrete-time periodic Riccati equation. *IEEE Transactions on Automatic Control*, *39*(6), 1197–1210.

Jansson, M. (2005). A new subspace identification method for open and closed loop data. In *Proceedings of the 16th IFAC world congress*.

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

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

Liu, K. (1997). Identification of linear time-varying systems. *Journal of Sound and Vibration*, *206*(4), 487–505.

Lovera, M. (1997). Subspace identification methods: Theory and applications. *Ph.D. thesis.* Politecnico Di Milano.

Lovera, M., & Mercere, G. (2007). Identification for gain-scheduling: Balanced subspace approach. In *Proceedings of the American control conference*.

Nemani, M., Ravikanth, R., & Bamieh, B. A. (1995). Identification of linear parametrically varying systems. In *Proceedings of the 34th IEEE conference on decision and control*.

Previdi, F., & Lovera, M. (2004). Identification of nonlinear parametrically varying models using separable least squares. *International Journal of Control*, *77*(16), 1382–1392.

Rugh, W. (1996). *Linear system theory*. Upper Saddle River.

Santos, P. L. d., Ramos, J. A., & de Carvalho, J. L. M. (2005). 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*.

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

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

Sima, D. M. (2006). *Regularization techniques in model fitting and parameter estimation. Ph.D. thesis*. Katholieke Universiteit Leuven.

Smola, A. J., & Scholkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th international conference on machine learning*.

Suykens, J. A. K., van Gestel, T. V., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. World Scientific.

Takahashi, K., & Massaquoi, S. G. (2007). Neuroengineering model of human limb control — gainscheduled feedback control approach. In *Proceedings of the 46th IEEE conference on decision and control.*

Tóth, R., Felici, F., Heuberger, P. S. C., & Van den Hof, P. M. J. (2007a). Discrete time LPV I/O and state-space representations, differences of behavior and pitfalls of interpolation. In *Proceedings of the European control conference*.

Tóth, R., Heuberger, P. S. C., & Van den Hof, P. M. J. (2007b). LPV system identification with globally fixed orthonormal basis functions. In *Proceedings of the 46th IEEE conference on decision and control.*

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems*. Kluwer Academic Publishers.

van Wingerden, J. W. (2008). *Control of wind turbines with 'smart' rotors: Proof of concept & LPV subspace identification. Ph.D. thesis*. Delft University of Technology.

van Wingerden, J. W., & Verhaegen, M. (2007). Subspace identification of bilinear systems using a dedicated input sequence. In *Proceedings of the 46th IEEE conference on decision and control.*

van Wingerden, J. W., & Verhaegen, M. (2008a). Subspace identification of MIMO LPV systems: The PBSID approach. In *Proceedings of the 47th IEEE conference on decision and control.*

van Wingerden, J. W., & Verhaegen, M. (2008b). Subspace identification of multivariable LPV systems: A novel approach. In *Proceedings of the IEEE multi-conference on systems and control.*

van Wingerden, J. W., Felici, F., & Verhaegen, M. (2007). Subspace identification of MIMO LPV systems using piecewise constant scheduling sequence with hard/soft switching. In *Proceedings of the European control conference*.

van Wingerden, J. W., Houtzager, I., Felici, F., & Verhaegen, M. (2009). Closed-loop identification of the time-varying dynamics of variable-speed wind turbines [Special issue on Wind turbines: New challenges and advanced control solutions]. *International Journal of Robust and Nonlinear Control*, *19*(1), 4–21.

Varga, A. (2005). On solving discrete-time periodic Riccati equations. In *Proceedings of the 16th IFAC world congress*.

Verdult, V. (2002). *Nonlinear system identification; A state-space approach. Ph.D. thesis.* University of Twente.
Verdult, V., & Verhaegen, M. (2002). Subspace identification of multivariable linear parameter-varying systems. *Automatica*, *38*(5), 805–814.
Verdult, V., & Verhaegen, M. (2005). Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, *41*(9), 1557–1565.
Verhaegen, M., & Verdult, V. (2007). *Filtering and system identification: An introduction*. Cambridge University Press.
Zhou, K., Doyle, J. C., & Glover, K. (1996). *Robust and optimal control*. New Jersey: Upper Saddle River.

Jan-Willem van Wingerden was born in Ridderkerk, The Netherlands, in 1980. In December 2004, he received his M.Sc. degree Cum Laude in mechanical engineering from the Delft University of Technology, The Netherlands. In November 2008, he received his Ph.D. degree Cum Laude from the Delft Center of Systems and Control, Delft University of Technology. His main research interests include LPV identification, subspace identification, 'Smart' structures, and control and identification of wind turbines.

Michel Verhaegen received an engineering degree in aeronautics from the Delft University of Technology, The Netherlands, in August 1982, and the doctoral degree in applied sciences from the Catholic University Leuven, Belgium, in November 1985. During his graduate study, he held a research assistantship sponsored by the Flemish Institute for scientific research (IWT). From 1985 to 1994, he has been research fellow of the U.S. National Research Council (NRC), affiliated with the NASA Ames Research Center in California, and of the Dutch Academy of Arts and Sciences, affiliated with the Network Theory Group of the Delft University of Technology. During the period 1994–1999, he was an Associate Professor of the control laboratory of the Delft University of Technology and became appointed as full professor at the faculty of Applied Physics of the University of Twente in the Netherlands in 1999. From 2001 on Prof. Verhaegen moved back to the University of Delft and is now a member of the Delft Center for Systems and Control. Prof. Verhaegen has held short sabbatical leaves at the University of Uppsala, McGill, Lund and the German Aerospace Research Center (DLR) in Munich and is participating in several European Research Networks. His main research interest is the interdisciplinary domain of numerical algebra and system theory. In this field he has published over 100 papers. Current activities focus on the transfer of knowledge about new identification, fault tolerant control and data-driven controller design methodologies to industry. Application areas include smart structures, adaptive optics, and vehicle mechatronics.