

Identification of Multivariable Linear Parameter-Varying Systems Based on Subspace Techniques

Vincent Verdult and Michel Verhaegen

Faculty of Applied Physics, University of Twente
P.O. Box 217, NL-7500 AE Enschede, The Netherlands
V.Verdult@tn.utwente.nl, M.Verhaegen@tn.utwente.nl

Abstract

The paper presents a subspace type of identification method for multivariable linear parameter-varying systems in state space representation with affine parameter dependence. It is shown that a major problem with subspace methods for this kind of systems is the enormous dimensions of the data matrices involved. To overcome the curse of dimensionality, we suggest to use only the most dominant rows of the data matrices in estimating the model. An efficient selection algorithm is discussed that does not require the formation of the complete data matrices, but can process them row by row.

1 Introduction

Subspace identification is by now a well-accepted method for identification of multivariable linear systems [1], [2]. Recently, subspace methods have been developed to handle certain classes of nonlinear systems, like Wiener [3], [4], Hammerstein [5] and bilinear systems [6], [7], [8], [9], [10]. In this paper we present a subspace identification method for linear parameter-varying (LPV) systems with affine parameter dependence. These systems can be represented as

$$\begin{aligned}x_{k+1} &= A_0 x_k + [A_1, A_2, \dots, A_s](p_k \otimes x_k) \\ &\quad + B_0 u_k + [B_1, B_2, \dots, B_s](p_k \otimes u_k) + K e_k \quad (1) \\ y_k &= C x_k + D u_k + e_k \quad (2)\end{aligned}$$

where \otimes denotes the Kronecker product, $x_k \in \mathbb{R}^n$ represents the unknown state, $u_k \in \mathbb{R}^m$ is the input, $y_k \in \mathbb{R}^\ell$ is the output, $p_k \in \mathbb{R}^s$ is the time-varying parameter vector, and $e_k \in \mathbb{R}^\ell$ is an unknown white noise sequence which is independent of u_k and p_k . The identification problem that we consider is: given measurements of the input u_k , the output y_k and the parameter vector p_k , determine the matrices $A_0, A_1, \dots, A_s, B_0, B_1, \dots, B_s, C$ and D .

The LPV model can be thought of as a weighted combination of linear models. The weighting functions are then the elements of the parameter vector p_k . This allows us to use the LPV model to estimate local linear state space models [11]. We do not require the weighting functions to be slowly time-varying. Hence, we can also model systems in which the parameters vary quite fast. We also do not need full state measurement as in earlier work reported in [12].

The LPV model can also be used for identification of bilinear systems, if we take the parameter vector p_k equal to the input u_k . For bilinear state space systems, the matrices B_1, \dots, B_s are equal to zero. This can easily be taken into account in the LPV identification method presented below. In fact, the theory for the LPV subspace method presented in this paper is an extension of the theory for bilinear systems developed by Favoreel [7].

A major problem with subspace methods for bilinear and LPV systems is the enormous dimensions of the data matrices involved. The number of rows in the data matrices grows exponentially with the order of the system. Already for very low order systems the matrices become too large to be handled on a standard computer. For bilinear systems this has already been observed by Favoreel [7]. He has pointed out some possible directions to reduce this curse of dimensionality, but their viability still needs to be proven. In this paper we suggest a novel approach to tackle the dimensionality problem. We present a procedure to select a subset of the most dominant rows, that does not require the formation of the complete data matrices, but can process them row by row. Since only a subset of rows is used, the identified LPV model is an approximate model. If the performance of this LPV model is not satisfactory, the model can be improved using a nonlinear parameter optimization based identification procedure, like the one presented by Lee [13]. Because of the nonconvex nature of such a nonlinear optimization based procedure, a good initial guess of the model is required. This is exactly what our method can provide.

The paper is organized as follows. In section 2 the theory for subspace identification of LPV systems is explained. Section 3 describes a selection procedure to reduce the number of rows in the data matrices. Section 4 discusses the implementation of this selection procedure and section 5 presents a multivariable identification example.

2 LPV Subspace Identification Method

First, we introduce some notation to explain the subspace method. In the definitions below, we use the symbol \odot to denote the Khatri-Rao product, which is a column wise Kronecker product for two matrices with an equal number of

columns.

$$\begin{aligned}
X_k &:= [x_k, x_{k+1}, \dots, x_{k+N-1}] \\
X_{j|j} &:= \begin{bmatrix} X_j \\ P_j \odot X_j \end{bmatrix} \\
X_{k+j|j} &:= \begin{bmatrix} X_{k+j-1|j} \\ P_{k+j} \odot X_{k+j-1|j} \end{bmatrix} \\
Y_{j|j} &:= Y_j \\
Y_{k+j|j} &:= \begin{bmatrix} Y_{k+j} \\ Y_{k+j-1|j} \\ P_{k+j} \odot Y_{k+j-1|j} \end{bmatrix} \\
\mathcal{Y}_{j|j} &:= Y_j \\
\mathcal{Y}_{k+j|j} &:= \begin{bmatrix} Y_{k+j} \\ \mathcal{Y}_{k+j-1|j} \end{bmatrix} \\
\tilde{P}_{j|j} &:= P_j \\
\tilde{P}_{k+j|j} &:= \begin{bmatrix} \tilde{P}_{k+j-1|j} \\ P_{k+j,1} \\ P_{k+j,1} \odot \tilde{P}_{k+j-1|j} \\ \vdots \\ P_{k+j,m} \\ P_{k+j,m} \odot \tilde{P}_{k+j-1|j} \end{bmatrix}
\end{aligned}$$

where $P_{k,i}$ denotes the i th row of P_k . We define the extended input v_k as $v_k := [u_k^T, (p_k \otimes u_k)^T]^T$. The matrices P_k , U_k , V_k , Y_k and E_k are defined similar to X_k ; $P_{k+j|j}$, $U_{k+j|j}$, $V_{k+j|j}$ and $E_{k+j|j}$ similar to $Y_{k+j|j}$.

The key to subspace identification is the derivation of the data equations. These equations describe the relations between the previously defined matrices. They are formulated in the following two lemmas. The proof of these lemmas is a straightforward evaluation of the LPV system equations (1) and (2) and will be omitted here, for the sake of brevity.

Lemma 1 State equations

$$\begin{aligned}
X_{k+j} &= \Delta_k^x X_{k+j-1|j} + \Delta_k^v V_{k+j-1|j} \\
&\quad + \Delta_k^y Y_{k+j-1|j} \quad (3)
\end{aligned}$$

$$X_{k+j|j} = \begin{bmatrix} X_j \\ \tilde{P}_{k+j|j} \odot X_j \end{bmatrix} \quad (4)$$

where

$$\begin{aligned}
\Delta_1^x &:= [(A_0 - KC), A_1, A_2, \dots, A_s] \\
\Delta_k^x &:= [(A_0 - KC)\Delta_{k-1}^x, A_1\Delta_{k-1}^x, \dots, A_s\Delta_{k-1}^x] \\
\Delta_1^v &:= [(B_0 - KD), B_1, B_2, \dots, B_s] \\
\Delta_k^v &:= [\Delta_1^v, (A_0 - KC)\Delta_{k-1}^u, A_1\Delta_{k-1}^u, \dots, A_s\Delta_{k-1}^u] \\
\Delta_1^y &:= K \\
\Delta_k^y &:= [\Delta_1^y, (A_0 - KC)\Delta_{k-1}^y, A_1\Delta_{k-1}^y, \dots, A_s\Delta_{k-1}^y]
\end{aligned}$$

Lemma 2 Output equation

$$\begin{aligned}
\mathcal{Y}_{k+j|j} &= H_k^x X_{k+j-1|j} + H_k^v V_{k+j-1|j} \\
&\quad + H_k^e E_{k+j-1|j} + G_k^u U_{k+j} + G_k^e E_{k+j} \quad (5)
\end{aligned}$$

where

$$\begin{aligned}
H_1^x &:= \begin{bmatrix} CA_0 & CA_1 & \dots & CA_s \\ C & 0 & \dots & 0 \end{bmatrix} \\
H_k^x &:= \begin{bmatrix} CA_0\bar{\Delta}_{k-1}^x & CA_1\bar{\Delta}_{k-1}^x & \dots & CA_s\bar{\Delta}_{k-1}^x \\ H_{k-1}^x & 0 & \dots & 0 \end{bmatrix} \\
H_1^v &:= \begin{bmatrix} CB_0 & CB_1 & \dots & CB_s \\ D & 0 & \dots & 0 \end{bmatrix} \\
H_k^v &:= \begin{bmatrix} CB_0 & CB_1 & \dots & CB_s & CA_0\bar{\Delta}_{k-1}^v & CA_1\bar{\Delta}_{k-1}^v & \dots & CA_s\bar{\Delta}_{k-1}^v \\ G_{k-1}^u & 0 & \dots & 0 & H_{k-1}^v & 0 & \dots & 0 \end{bmatrix} \\
H_1^e &:= \begin{bmatrix} CK_0 & CK_1 & \dots & CK_s \\ I_\ell & 0 & \dots & 0 \end{bmatrix} \\
H_k^e &:= \begin{bmatrix} CK_0 & CK_1 & \dots & CK_s & CA_0\bar{\Delta}_{k-1}^e & CA_1\bar{\Delta}_{k-1}^e & \dots & CA_s\bar{\Delta}_{k-1}^e \\ G_{k-1}^e & 0 & \dots & 0 & H_{k-1}^e & 0 & \dots & 0 \end{bmatrix} \\
G_k^u &:= \begin{bmatrix} D \\ 0 \end{bmatrix} \\
G_k^e &:= \begin{bmatrix} I_\ell \\ 0 \end{bmatrix} \\
\bar{\Delta}_1^x &:= [A_0, A_1, \dots, A_s] \\
\bar{\Delta}_k^x &:= [A_0\bar{\Delta}_{k-1}^x, A_1\bar{\Delta}_{k-1}^x, \dots, A_s\bar{\Delta}_{k-1}^x] \\
\bar{\Delta}_1^v &:= [B_0, B_1, \dots, B_s] \\
\bar{\Delta}_k^v &:= [\bar{\Delta}_1^v, A_0\bar{\Delta}_{k-1}^v, A_1\bar{\Delta}_{k-1}^v, \dots, A_s\bar{\Delta}_{k-1}^v] \\
\bar{\Delta}_1^e &:= [K_0, K_1, \dots, K_s] \\
\bar{\Delta}_k^e &:= [\bar{\Delta}_1^e, A_0\bar{\Delta}_{k-1}^e, A_1\bar{\Delta}_{k-1}^e, \dots, A_s\bar{\Delta}_{k-1}^e]
\end{aligned}$$

Now, we state the basic theorem for LPV subspace identification.

Theorem 1 Define

$$\begin{aligned}
W_{j,0} &:= \begin{bmatrix} V_{j-1|0} \\ Y_{j-1|0} \end{bmatrix} \\
Z_{k,j,0} &:= \begin{bmatrix} U_{k+j} \\ V_{k+j-1|j} \\ \tilde{P}_{k+j-1|j} \odot \begin{bmatrix} V_{j-1|0} \\ Y_{j-1|0} \end{bmatrix} \end{bmatrix}
\end{aligned}$$

Given the following QR factorization

$$\begin{bmatrix} W_{j,0} \\ Z_{k,j,0} \\ \mathcal{Y}_{k+j|j} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} \quad (6)$$

Under the assumption that $[W_{j,0}^T, Z_{k,j,0}^T]^T$ has full row rank, the noise e_k differs from zero, the system is observable and j is such that $\Delta_j^x = 0$, we have

$$\lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \Gamma_k X_j = \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \bar{R}(:, 1 : n_w) W_{j,0} \quad (7)$$

where

$$\Gamma_k := H_k^x(:, 1 : n), \quad \bar{R} := [R_{31}, R_{32}] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^{-1}$$

and n_w is the total number of rows in $W_{j,0}$.

Proof: Since $\Delta_j^x = 0$ equation (3) yields

$$X_j = \Delta_j^v V_{j-1|0} + \Delta_j^y Y_{j-1|0} = \Psi_j^w W_{j,0} \quad (8)$$

where $\Psi_j^w := [\Delta_j^v, \Delta_j^y]$. Therefore with equations (4) and (5) we can write

$$\begin{aligned} \mathcal{Y}_{k+j|j} &= \Gamma_k \Psi_j^w W_{j,0} + (\Lambda_k \otimes \Psi_j^w) \left(\tilde{P}_{k+j-1|j} \odot W_{j,0} \right) \\ &\quad + H_k^u U_{k+j-1|j} + H_k^e E_{k+j-1|j} + G_k^u U_{k+j} + G_i^e E_{k+j} \quad (9) \end{aligned}$$

where Λ_k is such that $H_k^e = [\Gamma_k, \Lambda_k]$. With (6) we can write

$$\begin{aligned} \mathcal{Y}_{k+j|j} [Q_1^T, Q_2^T] &= [R_{31}, R_{32}] \\ &= [\Gamma_k \Psi_j^w, G_k^u, H_k^v, (\Lambda_k \otimes \Psi_j^w)] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \\ &\quad + H_k^e E_{k+j-1|j} [Q_1^T, Q_2^T] + G_k^u E_{k+j} [Q_1^T, Q_2^T] \quad (10) \end{aligned}$$

Note that for N approaching infinity the matrix $\frac{1}{N} [W_{j,0}^T, Z_{k,j,0}^T]^T [E_{k+j-1|j}^T, E_{k+j}^T]$ will go to zero. Therefore, we have

$$\lim_{N \rightarrow \infty} \frac{1}{N} \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} [E_{k+j-1|j}^T, E_{k+j}^T] = 0$$

The lower triangular matrix is invertible due to the full rank assumption on $[W_{j,0}^T, Z_{k,j,0}^T]^T$, therefore

$$\lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} [E_{k+j-1|j}^T, E_{k+j}^T] = 0$$

Now (10) can be written as

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} [\Gamma_k \Psi_j^w, G_k^u, H_k^u, (\Lambda_k \otimes \Psi_j^w)] &= \\ \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} [R_{31}, R_{32}] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^{-1} & \end{aligned}$$

Using equation (8) we obtain

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \Gamma_k X_j &= \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \Gamma_k \Psi_j^w W_{j,0} \\ &= \lim_{N \rightarrow \infty} \frac{1}{\sqrt{N}} \tilde{R}(:, 1 : n_w) W_{j,0} \end{aligned}$$

This completes the proof. \square

According to theorem 1 the state sequence can be recovered as the row space of the right hand side of equation (7). Given the following singular value decomposition

$$\tilde{R}(:, 1 : n_w) W_{j,0} = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$$

where Σ_1 is a diagonal matrix containing the dominant singular values, the state sequence can be estimated as $\hat{X}_j = \Sigma_1^{1/2} V_1^T$. Note that the number of dominant singular values provides an estimate for the order of the system. Finally, the system matrices are determined by solving the following two least squares problems.

$$\begin{aligned} \hat{X}_{j+1} &\approx [A_0, \dots, A_s, B_0, \dots, B_s] \begin{bmatrix} \hat{X}_j \\ P_j \odot \hat{X}_j \\ U_j \\ P_j \odot U_j \end{bmatrix} \\ Y_j &\approx [C, D] \begin{bmatrix} \hat{X}_j \\ U_j \end{bmatrix} \end{aligned}$$

Table 1: Total number of rows of the matrices $W_{j,0}$ and $Z_{k,j,0}$; $s = 3, m = l, j = k$.

	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$	$\ell = 5$
$k = 2$	105	210	315	420	525
$k = 3$	1701	3402	5103	6804	8505
$k = 4$	27285	54570	81855	109140	136425
$k = 5$	436821	873642	1310463	1747284	2184105

3 Reducing the Dimension of the Data Matrices

The subspace identification method presented in the previous section is not very useful in practice, because of the huge dimensions of the data matrices involved. The number of rows in the matrices $W_{j,0}$ and $Z_{k,j,0}$ grows exponentially with the order of the system. Hence, already for relatively low order systems the amount of memory required to compute the QR factorization exceeds the limits of what is currently available on the average desktop computer. Table 1 shows the number of rows of the data matrices $W_{j,0}$ and $Z_{k,j,0}$ as a function of system dimensions for a square system with three time-varying parameters ($p_k \in \mathbb{R}^3$).

Equally important, to be able to perform the QR decomposition the number of columns in both $W_{j,0}$ and $Z_{k,j,0}$ needs to be larger than the total number of rows of $W_{j,0}$ and $Z_{k,j,0}$ together. Since, the number of columns equals the number of data samples, an enormous amount of data is required to fulfill this condition.

These problems can be overcome if we do not use all the rows of $W_{j,0}$ and $Z_{k,j,0}$ to compute the QR factorization. This will of course be at the expense of introducing an error, so that we are only able to compute an approximation of the original system. However, this approximation can be used to initialize a nonlinear parameter optimization based identification procedure as in [13].

First, we explain how to reduce the number of rows in $W_{j,0}$. Then we use a similar procedure to reduce the number of rows in $Z_{k,j,0}$. Let the matrix $\hat{W}_{j,0}$ contain a subset of the rows of $W_{j,0}$ and the matrix $\hat{Z}_{k,j,0}$ contain a subset of the rows of $Z_{k,j,0}$. Then the reduced QR factorization is defined as:

$$\begin{bmatrix} \hat{W}_{j,0} \\ \hat{Z}_{k,j,0} \\ \mathcal{Y}_{k+j|j} \end{bmatrix} = \begin{bmatrix} \hat{R}_{11} & 0 & 0 \\ \hat{R}_{21} & \hat{R}_{22} & 0 \\ \hat{R}_{31} & \hat{R}_{32} & \hat{R}_{33} \end{bmatrix} \begin{bmatrix} \hat{Q}_1 \\ \hat{Q}_2 \\ \hat{Q}_3 \end{bmatrix}$$

We approximate the state sequence given in equation (8) as $X_j \approx \hat{\Psi}_j^w \hat{W}_{j,0}$, where $\hat{\Psi}_j^w$ contains the columns of the matrix Ψ_j^w that correspond to the rows of $\hat{W}_{j,0}$. To determine the row of $W_{j,0}$ that has the largest contribution to the matrix X_j , we would like to solve the following set of least squares problems and compute the corresponding residuals.

$$\min_{\Psi_j(:,i)} \|X_j - \Psi_j(:,i) W_{j,0}(i,:)\|_F^2 \quad (11)$$

The row with the smallest residual has the largest contribution to the matrix X_j and should hence be selected. In practice, of course we cannot solve the least squares problems (11), because the state is unknown. Therefore, we propose to solve the following set of problems instead.

$$\min_{\Psi_j^c(:,i)} \left\| Y_j - \Psi_j^c(:,i) W_{j,0}(i,:) \right\|_F^2 \quad (12)$$

where $\Psi_j^c := C\Psi_j$. Since C does not have full column rank, the ordering of the residuals obtained by solving (12) can be different. The explanation is that for the full row rank matrices M_i and M_j , the relation $\|CM_i\| \leq \|CM_j\|$ does not necessarily imply that $\|M_i\| \leq \|M_j\|$. In practice this will not be a big problem as the example in section 5 illustrates.

To select the next row of $W_{j,0}$ we have to take into account the effect of the previously selected row. This means that we have to recompute the residuals and then select again the row that yields the smallest residual. More specifically, we can use the following procedure [14] to select the rows:

Take $W^1 := W_{j,0}$ and $Y^1 := Y_j$. For i ranging from 1 to n_w we perform the following steps

1. Select the row of W^i that has the smallest residual when solving

$$\min_{\Psi^q} \left\| Y^i - \Psi^q W^i(q,:) \right\|_F^2$$

2. Interchange the i th row of W^i with the row selected in the previous step, by the permutation matrix Π^i .
3. Compute a Householder rotation H^i such that

$$\Pi^i \begin{bmatrix} T^i & 0 \\ P^i & W^i \end{bmatrix} H^i = \begin{bmatrix} T^{i+1} & 0 \\ P^{i+1} & W^{i+1} \end{bmatrix}$$

where $T^i \in \mathbb{R}^{(i-1) \times (i-1)}$, and $P^i \in \mathbb{R}^{(n_w-i+1) \times (i-1)}$.

4. Apply H^i to Y^i :

$$\begin{bmatrix} \bar{Y}^i & Y^i \end{bmatrix} H_w^i = \begin{bmatrix} \bar{Y}^{i+1} & Y^{i+1} \end{bmatrix}$$

where $\bar{Y}^i \in \mathbb{R}^{(i-1) \times (i-1)}$.

The rows of $W_{j,0}$ can now be ordered according to their contribution to Y_j , by applying the permutation matrices Π^i , i.e. $\Pi^{n_w} \cdot \Pi^{n_w-1} \dots \Pi^1 \cdot W_{j,0}$. Next, we have to decide on how many rows we use. This will determine the accuracy of the approximation. A condition that can be used to decide on the number of rows is the norm of the residual. After applying the permutation matrices we take the \hat{n}_w first rows and store them into the matrix $\hat{W}_{j,0}$. This matrix with a reduced number of rows will now be used instead of $W_{j,0}$.

As a nice side effect, this selection procedure already yields the first part of the reduced QR factorization:

$$\hat{W}_{j,0} = \begin{bmatrix} \hat{R}_{11} & 0 \end{bmatrix} \hat{Q}_1$$

The next step is to reduce the number of rows in the matrix $Z_{k,j,0}$. For this we use a similar procedure as for the selection of the rows of $W_{j,0}$. As a selection criterion we use

$$\min_{\Phi^q} \left\| \mathcal{Y}^i - \Phi^q Z^i(q,:) \right\|_F^2$$

where $\mathcal{Y}^1 := \mathcal{Y}_{k+j|j} \hat{Q}_1^T$, which is inspired by equation (9). We can continue applying Householder rotations to compute \hat{R}_{21} and \hat{R}_{22} .

Since we do not need the matrix \hat{R}_{33} in the subsequent computations, there is no need to complete the reduced QR factorization. The matrices \hat{R}_{31} and \hat{R}_{32} are obtained by applying all the Householder rotations used in the selection procedures for $W_{j,0}$ and $Z_{k,j,0}$ to the matrix $\mathcal{Y}_{k+j|j}$.

Now an approximation of the state can be computed from the SVD of the matrix

$$\hat{R}(:, 1 : \hat{n}_w) W_{j,0}$$

where

$$\hat{R} := \begin{bmatrix} \hat{R}_{31} & \hat{R}_{32} \end{bmatrix} \begin{bmatrix} \hat{R}_{11} & 0 \\ \hat{R}_{21} & \hat{R}_{22} \end{bmatrix}^{-1}$$

and \hat{n}_w is the total number of rows in $\hat{W}_{j,0}$. The system matrices can be computed in a similar way as described in section 2.

4 Implementation Issues

An efficient implementation of the selection algorithm described above is of paramount importance. Due to the huge dimensions of the data matrices, it is not feasible to store the complete matrices $W_{j,0}$ and $Z_{k,j,0}$ into memory. Therefore, we propose to construct only one row at a time. After evaluating the associated value for the residual, the row is replaced by the next one. In this way, we only need to store the residuals. If we come across a row that has the lowest residual thus far, we store it. We also store the Householder rotations associated with the selected rows. The implementation of this method might look like this:

Select \hat{n}_w rows from $W_{j,0} \in \mathbb{R}^{n_w \times N}$. The matrix $R \in \mathbb{R}^{\hat{n}_w \times \hat{n}_w}$ is the result from the reduced QR factorization. The matrix $S \in \mathbb{R}^{\hat{n}_w \times N}$ contains the selected rows from $W_{j,0}$. Define the index set $\mathcal{I}^1 = \{1, 2, \dots, n_w\}$ for $q \in \{1, 2, \dots, \hat{n}_w\}$

1. Select the row from \mathcal{I}^q that has the smallest residual for $k \in \mathcal{I}^q$
 - (a) Construct the k th row of $W_{j,0}$:
 $W_k^1 = W_{j,0}(k, 1 : N)$
 - (b) for $i \in \{1, 2, \dots, q-1\}$
 - Apply all previously computed Householder rotations $W_k^{i+1} = W_k^i H^i$

(c) Compute the residuals

$$r_k^q = \min_{\phi_k^q} \|Y_q(:, q : N) - \phi_k^q W_k^q(:, q : N)\|_F^2$$

Note that this can be done efficiently by first computing a Householder rotation Q_k^q such that $W_k^q(:, q : N)Q_k^q = [P_k^q, 0_{(q+1) \times N}]$ with $P_k^q \in \mathbb{R}$ and then evaluating

$$r_k^q = \|Y_q(:, q : N)Q_k^q(:, 2 : N - q + 1)\|_F^2$$

(d) if $r_k^q \leq r_i^q$ for all $i \in \mathcal{I}^q$ then row k has the smallest residual thus far and we select this row, that is, we make the q th row of S equal to W_k^q and we take $\mathcal{I}^{q+1} = \mathcal{I}^q \setminus \{k\}$ as the new index set.

2. Compute a Householder rotation H^q such that $S(q, :)H^q = [R^q, 0_{(q+1) \times N}]$ with $R^q \in \mathbb{R}^{1 \times q}$.
3. The q th row of the R matrix is constructed as $[R^q, 0_{(q+1) \times w}]$.
4. Apply the rotation H^q to Y^q : $Y^{q+1} = Y^q H^q$.

For an efficient implementation, the Householder matrices H^q should not be explicitly formed. It is much more efficient to use and store the associated Householder vectors [15].

A further improvement can be obtained by observing that there is no need to recompute the residuals completely at each step. After computing the residuals in the first step, the residuals for the following steps can be obtained with the update rule:

$$r_k^q = r^{q-1} - \|Y_q(:, q : N)Q_k^q(:, 1)\|_F^2$$

where r^{q-1} is the residual of the row selected in the previous step.

With this implementation, it is possible to handle systems of moderate dimensions, because the data matrices do not need to be stored. However the computation time increases drastically. This is mainly due to the need to construct one row at a time. Depending on its place in the data matrix, the construction of a row involves several times N multiplications. It is possible to reduce this computational burden a bit by stepping through the rows in a special order. For example, if we are processing the rows of $U_{j-1|0}$, we could, after processing a row in the first part of the matrix (that is: $U_{j-2|0}$), store the row, and use it to first compute the rows in the third part of the matrix (that is: $P_{j-1} \odot U_{j-2|0}$) which can be obtained by element-wise multiplication with a row from P_{j-1} . This requires only N multiplications, in contrast with several times N when the row needs to be constructed from scratch.

The computation time can also be reduced by shrinking the set of rows to be processed at each step. We discard some

rows that have the largest residuals. We shrink the set of rows at every step with a constant factor, such that in the final step the number of rows left to choose from, equals the total number of rows that have been selected. Although, we do not have any guarantee on not losing important rows, it turned out to work quite well for the example presented below.

Finally, the initial set of rows to be considered can be reduced by discarding the ones that are formed by more than h multiplications. This in fact boils down to discarding the terms with order higher than h . It can be argued that because of the limited number of samples available, higher order terms cannot be computed accurately anyway.

5 Example

We have applied the described method to identify a fourth order LPV system with three outputs, two inputs and three time-varying parameters.

The time-varying parameter vector p_k and the input signal u_k are generated as:

$$p_k = \begin{bmatrix} \rho_k \\ \frac{1}{2} \sin(\frac{2\pi k}{100}) \rho_k \\ \frac{1}{2} \cos(\frac{2\pi k}{100}) \rho_k \end{bmatrix} \quad u_k = \begin{bmatrix} G(q)\mu_k + 4H(q)\nu_k \\ \xi_k \end{bmatrix}$$

where ρ_k is a uniformly distributed random sequence with unit variance, μ_k , ν_k and ξ_k are zero mean white Gaussian noise sequences of unit variance, q denotes the shift operator, $G(q) := 0.75 + 1.05q^{-1} + 0.15q^{-2}$, and $H(q)$ is a second-order low-pass Butterworth filter with a cut-off frequency of one-fifth of the sample frequency. We simulate the system for 1000 samples, and use the generated data to estimate the system matrices. Measurement noise of unit variance has been added to the simulated output signal.

The block size of the data matrices has been chosen to be $k = j - 1 = 5$. For this identification problem, the data matrix $W_{j,0}$ contains 3751 rows and the matrix $Z_{k,j,0}$ 957187 rows. To reduce the problem we discard all the rows that contain more than five multiplications. This reduces the number of rows to 2860 for $W_{j,0}$ and 66225 for $Z_{k,j,0}$. We use the described selection procedure to select the 28 most dominant rows from the remaining rows of $W_{j,0}$ and the 66 most dominant rows from the remaining rows of $Z_{k,j,0}$. Selecting the rows and computing an LPV model takes approximately 20 hours with Scilab [16] running on a 400 MHz Pentium II.

We simulate the identified model using both the data set used for identification and a fresh data set of another 1000 samples (without noise). The quality of the output signals generated by the identified LPV model, is measured using the variance accounted for (VAF), which is defined as

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

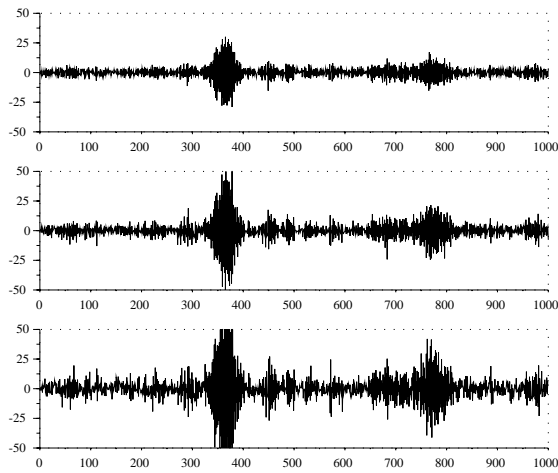


Figure 1: Output signals of the example LPV system.

where \hat{y}_k denotes the estimated output signal, and $\text{var}(\cdot)$ denotes the variance of a quasi-stationary signal. The VAF on the data used for identification is 92.33 %, the VAF on the fresh data set is 70.92 %.

We used the LPV model obtained from the subspace identification algorithm as an initial guess for the nonlinear parameter based optimization described in [13]. After 190 iterations, the optimization was finished and the VAF values were 99.13 % and 99.53 % for the identification data and fresh data, respectively. It is no use showing the output signals, because you cannot tell the difference between the real and simulated outputs from such a graph.

6 Conclusion

We have discussed a subspace identification method for multivariable linear parameter-varying state space systems. The original method cannot be used in practice, because the number of rows in the data matrices used, grows exponentially with the order of the system. To address this problem we have presented an algorithm that selects the most important rows from the data matrices and uses this subset to estimate a model of the LPV system. This algorithm processes the rows one by one, so that there is no need to construct the entire data matrices. By means of an example it has been demonstrated that an enormous reduction in the number of rows can be achieved without deteriorating the quality of the estimated model much.

Acknowledgement

This work was supported by the Dutch Technology Foundation (STW) under project number DEL55.3891.

References

[1] P. Van Overschee and B. De Moor, “N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems,” *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.

[2] M. Verhaegen, “Identification of the deterministic part of MIMO state space models given in innovations form from input-output data,” *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.

[3] D. Westwick and M. Verhaegen, “Identifying MIMO Wiener systems using subspace model identification methods,” *Signal Processing*, vol. 52, pp. 235–258, 1996.

[4] C. T. Chou and M. Verhaegen, “Identification of Wiener models with process noise,” in *Proceedings of the 38th IEEE Conference on Decision and Control*, (Phoenix, Arizona), pp. 598–603, Dec. 1999.

[5] M. Verhaegen and D. Westwick, “Identifying MIMO Hammerstein systems in the context of subspace model identification methods,” *International Journal of Control*, vol. 63, no. 2, pp. 331–349, 1996.

[6] W. Favoreel, B. De Moor, and P. Van Overschee, “Subspace identification of bilinear systems subject to white inputs,” *IEEE Transactions on Automatic Control*, vol. 44, pp. 1157–1165, June 1999.

[7] W. Favoreel, *Subspace Methods for Identification and Control of Linear and Bilinear Systems*. PhD thesis, Leuven Catholic University, Leuven, Belgium, Nov. 1999.

[8] V. Verdult, M. Verhaegen, C. T. Chou, and M. Lovera, “Efficient and systematic identification of MIMO bilinear state space models,” in *Proceedings of the 37th IEEE Conference on Decision and Control*, (Tampa, Florida), pp. 1260–1265, Dec. 1998.

[9] V. Verdult and M. Verhaegen, “Subspace identification of MIMO bilinear systems,” in *Proceedings of the European Control Conference*, (Karlsruhe, Germany), Sept. 1999.

[10] H. Chen and J. Maciejowski, “Subspace identification method for combined deterministic-stochastic bilinear systems,” in *Proceedings of the IFAC Symposium on System Identification, SYSID*, (Santa Barbara, California), June 2000.

[11] T. A. Johansen and R. Murray-Smith, “The operating regime approach to nonlinear modelling and control,” in *Multiple Model Approaches to Modelling and Control* (R. Murray-Smith and T. A. Johansen, eds.), ch. 1, pp. 3–72, London: Taylor & Francis, 1997.

[12] M. Lovera, M. Verhaegen, and C. T. Chou, “State space identification of MIMO linear parameter varying models,” in *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems*, (Padua, Italy), pp. 839–842, July 1998.

[13] L. H. Lee, *Identification and Robust Control of Linear Parameter-Varying Systems*. PhD thesis, University of California at Berkeley, Berkeley, California, 1997.

[14] M. Verhaegen, “The minimal residual QR-factorization algorithm for reliably solving subset regression problems,” Tech. Rep. NASA TM 100021, NASA, Ames Research Center, Moffett Field, California, Sept. 1987.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, Maryland: The Johns Hopkins University Press, third ed., 1996.

[16] C. Gomez, ed., *Engineering and Scientific Computation with Scilab*. Boston: Birkhäuser, 1999. <http://www-rocq.inria.fr/scilab/>.