92-05I

Subspace Identification Algorithms for Bilinear and 2-D System Identification

José Ramos
Institute for Land and Water Management
and
ESAT Laboratory
Katholieke Universiteit Leuven
Vitaldecosterstraat 102
3000, Leuven, Belgium

ESAT/SISTA Report 1992-05

Abstract

Subspace algorithms that rely on robust numerical linear algebra are becoming increasingly important in areas such as array processing, mobile telephones, system identification, etc. The class of linear subspace system identification algorithms have already shown to be successful for industrial as well as environmental applications. These subspace identification algorithms use input/output data directly, contrary to other classical state-space identification algorithms that use Markov parameters. The advantages of the subspace algorithms are the automatic structure identification (system order), geometrical insights (notions of angle between subspaces), and the fact that they rely on robust numerical procedures (singular value decomposition). In this paper we extend the linear subspace identification algorithm to more complex dynamical systems such as bilinear and 2-D state space models. It is shown that the new algorithms are simple extensions of the linear subspace algorithm. Potential applications of the new class of algorithms may include biology, econometrics, image processing, and environmental engineering.

1 Introduction

Consider the linear, discrete, time invariant, multivariable system with state space representation

$$x_{k+1} = Ax_k + Bu_k \quad (1)$$
$$y_k = Cx_k + Du_k \quad (2)$$

where $u_k \in \Re^m$, $y_k \in \Re^l$, *and* $x_k \in \Re^n$ *denote, respectively, the input, output, and state vector at time* k. *The dimension of* x_k *corresponds to the minimal system order* (n). *Furthermore,* A, B, C, *and* D *are unknown system matrices of appropriate dimensions, to be identified (up to a similarity transformation) by means of recorded Input/Output sequences* $\{u_1\ u_2\ \ldots\ u_N\}$ *and* $\{y_1\ y_2\ \ldots\ y_N\}$.

Let the past (H_1) and future (H_2) input/output Hankel matrices be defined as

$$H_1 = \left[\begin{array}{c} U_1 \\ \hline Y_1 \end{array}\right] = \left[\begin{array}{cccc} u_1 & u_2 & \cdots & u_j \\ u_2 & u_3 & \cdots & u_{j+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_i & u_{i+1} & \cdots & u_{i+j-1} \\ \hline y_1 & y_2 & \cdots & y_j \\ y_2 & y_3 & \cdots & y_{j+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_i & y_{i+1} & \cdots & y_{i+j-1} \end{array}\right] \tag{3}$$

and

$$H_2 = \left[\begin{array}{c} U_2 \\ \hline Y_2 \end{array}\right] = \left[\begin{array}{cccc} u_{i+1} & u_{i+2} & \cdots & u_{i+j} \\ u_{i+2} & u_{i+3} & \cdots & u_{i+j+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i} & u_{2i+1} & \cdots & u_{2i+j-1} \\ \hline y_{i+1} & y_{i+2} & \cdots & y_{i+j} \\ y_{i+2} & y_{i+3} & \cdots & y_{i+j+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i} & y_{2i+1} & \cdots & y_{2i+j-1} \end{array}\right] \tag{4}$$

where $j \gg i$. Let the state vector sequences X_1 and X_2 be defined as

$$X_1 = [x_1\ x_2\ \ldots\ x_j]$$
$$X_2 = [x_{i+1}\ x_{i+2}\ \ldots\ x_{i+j}]$$

Then, it is shown in [2] that under certain mild conditions, the following properties hold:

$$\begin{aligned} &i. && \text{rank}\{H_1\} = mi + n \\ &ii. && \text{rank}\{H_2\} = mi + n \\ &iii. && \text{rank}\{\left[\begin{array}{c} H_1 \\ H_2 \end{array}\right]\} = 2mi + n \\ &iv. && \text{span}_{\text{row}}\{X_2\} = \text{span}_{\text{row}}\{H_1\} \cap \text{span}_{\text{row}}\{H_2\} \end{aligned}$$

Thus, any basis for the above intersection (iv) can be chosen as a valid state vector sequence X_2', with the basis vectors as the consecutive row vectors. Notice that X_2' is unique up to a coordinate transformation in the state space (i.e., $X_2' = TX_2$ for some nonsingular

transformation T). Once X_2' is known, the system matrices are identified by solving an (overdetermined) system of linear equations:

$$\begin{bmatrix} x'_{i+2} & \cdots & x'_{i+j} \\ y_{i+1} & \cdots & y_{i+j-1} \end{bmatrix} = \begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix} \begin{bmatrix} x'_{i+1} & \cdots & x'_{i+j-1} \\ u_{i+1} & \cdots & u_{i+j-1} \end{bmatrix} \tag{5}$$

with $A' = TAT^{-1}$, $B' = TB$, $C' = CT^{-1}$, $D' = D$.

The computation of the state vector sequence is done by means of an svd of the concatenation of H_1 and H_2. In order to save space, we defer the computation of the state vector sequence for a later section. In section 2, we present the subspace identification algorithm for bilinear systems. In section 3, we present the 2-D subspace identification algorithm.

2 Iterative Bilinear Subspace Algorithm

Consider the general bilinear, discrete, time invariant, MIMO state space model

$$x_{k+1} = Ax_k + Bu_k + E[u_k \otimes x_k] \tag{6}$$

$$y_k = Cx_k + Du_k + F[u_k \otimes x_k] \tag{7}$$

where $u_k \in \Re^m$, $x_k \in \Re^n$, and $y_k \in \Re^\ell$ denote, respectively, the input, state, and output at time k. $E = [E_1, E_2, \ldots, E_m]$, $F = [F_1, F_2, \ldots, F_m]$, and A,B,C,D are system matrices of appropriate dimensions (note that $E_i \in \Re^{n \times n}$ and $F_i \in \Re^{\ell \times n}$). The tensor product we will adopt here is the one defined by the Kronecker product, i.e.,

$$a \otimes b \equiv \begin{bmatrix} a_1 b \\ \vdots \\ a_r b \end{bmatrix} \tag{8}$$

where $a \in \Re^r$, $b \in \Re^s$.

Contrary to linear systems, the impulse response for bilinear systems can well be approximated as a finite sum p of Volterra kernels of increasing orders, i.e.,

$$\begin{aligned} y_k = {} & h^{[0]}_{k_0} U[k_0] + \sum_{k_1} h^{[1]}_{k_1} U[k_0, k_1] + \sum_{k_1} \sum_{k_2} h^{[2]}_{k_1,k_2} U[k_0, k_1, k_2] \\ & + \ldots + \sum_{k_1} \sum_{k_2} \cdots \sum_{k_p} h^{[p]}_{k_1,k_2,\ldots,k_p} U[k_0, k_1, k_2, \ldots, k_p] \end{aligned} \tag{9}$$

where $h^{[p]}_{k_1,k_2,\ldots,k_p}$ denotes the p^{th} order Volterra kernel, $U[k_0, k_1, k_2, \ldots, k_p]$ denotes the product of the present input with "p" past inputs lagged respectively by $(k_1, k_2, \ldots, k_p)$, where $k_0 = 0$. Thus, the identification of bilinear systems via a direct input/output approach, seems a formidable computational task. This is perhaps the reason for adopting recursive algorithms, or the more recently common Walsh functions-based algorithms. Impulse response-based algorithms, such as those proposed in [6], require knowledge of the Volterra kernels or Markov parameters. However, this is a rather restrictive assumption.

Following the works of [6, 7] on bilinear systems theory, we study here the cascade structure of these systems. Without loss of generality we restrict our attention to bilinear systems that are stable and minimal. We can decompose (5)-(6) as the sum of p subsystems (i.e., number

of terms in the Volterra expansion)

$$\begin{aligned} x_{k+1}^0 &= Ax_k^0 + Bu_k \\ y_k^0 &= Cx_k^0 + Du_k \end{aligned}$$

$$\begin{aligned} x_{k+1}^1 &= Ax_k^1 + Bu_k + E[u_k \otimes x_k^0] \\ y_k^1 &= Cx_k^1 + Du_k + F[u_k \otimes x_k^0] \\ &\vdots \\ x_{k+1}^p &= Ax_k^p + Bu_k + E[u_k \otimes x_k^{p-1}] \qquad (10) \\ y_k^p &= Cx_k^p + Du_k + F[u_k \otimes x_k^{p-1}] \qquad (11) \end{aligned}$$

where x_k^p and y_k^p correspond respectively to the state and output of the p^{th} subsystem at time k. One can easily check that $x_0^p = x_0$ for all subsystems.

Let

$$u_k^p = \left[\begin{array}{c} u_k \\ \hline u_k \otimes x_k^{p-1} \end{array} \right]$$

then we can look at the p^{th} subsystem as an independent linear system driven by the inputs $u_k^p \in \Re^{m(n+1)}$, i.e.,

$$\begin{aligned} x_{k+1}^p &= Ax_k^p + \bar{B}u_k^p \qquad (12) \\ y_k &= Cx_k^p + \bar{D}u_k^p + e_k^p \qquad (13) \end{aligned}$$

where $\bar{B} = [B \mid E]$, $\bar{D} = [D \mid F]$, and $e_k^p = y_k - y_k^p$ is the error due to the fact that we do not observe y_k^p but y_k. One can show that as $p \to \infty$, $e_k^p \to 0$. Thus we can iteratively apply the algorithm of [2, 3] to identify the individual subsystems from known inputs $[u_1^p, u_2^p, \cdots, u_N^p]$ and outputs $[y_1, y_2, \cdots, y_N]$.

The basis of the iterative subspace algorithm is to determine a state vector sequence which, upon convergence, corresponds to a basis for the intersection of some subspaces H_1^p and H_2^p.

To fix ideas, we first need the input/output equations which follow from (12) – (13)

$$Y_1 = \Gamma X_1^p + H_T U_1^p + E_1^p \qquad (14)$$

$$Y_2 = \Gamma X_2^p + H_T U_2^p + E_2^p \qquad (15)$$

where H_T is a block Toeplitz matrix with the Markov parameters

$$H_T = \begin{bmatrix} \bar{D} & 0 & 0 & \dots & 0 \\ C\bar{B} & \bar{D} & 0 & \dots & 0 \\ CA\bar{B} & C\bar{B} & \bar{D} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{i-2}\bar{B} & CA^{i-3}\bar{B} & CA^{i-4}\bar{B} & \dots & \bar{D} \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix}$$

Y_1, and Y_2 are as defined previously. Similar to H_1, H_2, U_1, U_2, X_1, and X_2 in (3)-(4), we can define H_1^p, H_2^p, U_1^p, U_2^p, X_1^p, and X_2^p, using u_k^p and x_k^p instead of u_k and x_k. Finally, E_1^p and E_2^p are defined similarly to U_1^p and U_2^p using e_k^p instead.

The way to compute the intersection of H_1^p and H_2^p is by means of an SVD of their concatenation, i.e.,

$$\begin{aligned} \left[\begin{array}{c} H_1^p \\ H_2^p \end{array}\right] &= U_H S_H V_H^T \\ &= \left[\begin{array}{c|c} U_{11} & U_{12} \\ \hline U_{21} & U_{22} \end{array}\right] \left[\begin{array}{c|c} S_{11} & 0 \\ \hline 0 & 0 \end{array}\right] V_H^T \end{aligned} \tag{16}$$

From

$$U_{12}^T H_1^p = -U_{22}^T H_2^p$$

it follows that the row space of $U_{12}^T H_1^p$ equals the required intersection. However, $U_{12}^T H_1^p$ contains $2li - n$ row vectors[1], only n of which are linearly independent[2]. Thus, it remains to select n suitable combinations of these row vectors. In [2], an additional SVD of $U_{12}^T U_{11} S_{11}$ was used to compute an $n \times 2li - n$ matrix $\mathcal{T}$ such that

$$X_2^p = \mathcal{T} U_{12}^T H_1^p$$

Assuming that at the p^{th} iteration $X_2^p \approx X_2$, then the system matrices can be computed by solving an overdetermined system of equations. The algorithm is finally summarized as follows:

1. Initialization step:
   - Form data records $\{u_1, \cdots, u_N\}$ and $\{y_1, \cdots, y_N\}$
   - compute the initial state vector sequence

     $$X^0 = \{\ x_1^0 \mid x_2^0 \mid \cdots \mid x_{2i+j-1}^0\ \}$$

     by applying steps $(2) - (4)$ with $u_k^0 = u_k$ and $\alpha = 0$
2. for $\alpha = 1 : p$, form augmented inputs

   $$u_k^\alpha = \left[\begin{array}{c} u_k \\ \hline u_k \otimes x_k^{\alpha-1} \end{array}\right]$$

   compute H_1^α, H_2^α and the orthogonal complement

   $$\left[\begin{array}{c|c} U_{12}^T & U_{22}^T \end{array}\right] \left[\begin{array}{c} H_1^\alpha \\ H_2^\alpha \end{array}\right] = 0$$

3. compute the SVD of

   $$U_{12}^T U_{11} S_{11} = U_q S_q V_q^T$$

[1] dimension of U_{12}
[2] dimension of the intersection

4. *compute the state vector sequences*

$$X_1^\alpha = -U_q^T U_{22}^T H_1^\alpha$$

$$X_2^\alpha = U_q^T U_{12}^T H_1^\alpha$$

and form augmented sequence

$$X^\alpha = \{\ x_1^\alpha \mid x_2^\alpha \mid \cdots \mid x_{2i+j-1}^\alpha\ \}$$

5. *repeat steps* $(2)-(4)$ *until convergence* $(i.e.,\ X^p \approx X)$

6. *let*

$$P = \left[\begin{array}{c|c|c} x_{i+2} & \cdots & x_{i+j} \\ \hline y_{i+1} & \cdots & y_{i+j-1} \end{array}\right]$$

$$Q = \left[\begin{array}{c|c|c} x_{i+1} & \cdots & x_{i+j-1} \\ \hline u_{i+1} & \cdots & u_{i+j-1} \\ \hline u_{i+1} \otimes x_{i+1} & \cdots & u_{i+j-1} \otimes x_{i+j-1} \end{array}\right]$$

then compute the system matrices by solving the overdetermined system of equations

$$P = \left[\begin{array}{c|c|c} A & B & E \\ \hline C & D & F \end{array}\right] \cdot Q$$

For additional algorithmic details we refer the reader to [2], where it is shown how an explicit computation of X_2 *can be avoided, resulting in a considerable computational saving.*

3 2-D Subspace Algorithm

Consider a stable and causal 2-D digital filter with the following proper transfer function

$$H(z_1, z_2) \;=\; \frac{N(z_1, z_2)}{D_h(z_1) D_v(z_2)} \tag{17}$$

where $D_h(z_1)$ *and* $D_v(z_2)$ *are polynomials of degree* n_h *in* z_1 *and* n_v *in* z_2*, respectively. A 2-D digital filter that satisfies the above transfer function can be described by a variant of Roesser's local state-space model [4]*

$$\begin{bmatrix} x_{i+1,j}^h \\ x_{i,j+1}^v \end{bmatrix} \;=\; \underbrace{\left[\begin{array}{c|c} A_1 & A_2 \\ \hline 0 & A_4 \end{array}\right]}_{A} \begin{bmatrix} x_{i,j}^h \\ x_{i,j}^v \end{bmatrix} + \underbrace{\left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right]}_{B} u_{i,j} \tag{18}$$

$$y_{i,j} \;=\; \underbrace{\left[\begin{array}{c|c} C_1 & C_2 \end{array}\right]}_{C} \begin{bmatrix} x_{i,j}^h \\ x_{i,j}^v \end{bmatrix} + D u_{i,j} \tag{19}$$

where $u_{i,j} \in \Re^m$, $x^h_{i,j} \in \Re^{n_h}$, $x^v_{i,j} \in \Re^{n_v}$, *and* $y_{i,j} \in \Re^\ell$ *denote, respectively, the input, local horizontal state, local vertical state, and output at the* $(i,j)^{th}$ *spatial location of a finite field* $\mathcal{F}$. *The matrices* $[A_1, A_2, A_4, B_1, B_2, C_1, C_2, D]$ *are of appropriate dimensions, to be identified (up to a similarity transformation) from a given I/O pair* $\{u_{0,0}, u_{0,1}, \ldots, u_{0,M}, u_{1,0}, u_{1,1}, \ldots, u_{N,M}\}$ *and* $\{y_{0,0}, y_{0,1}, \ldots, y_{0,M}, y_{1,0}, y_{1,1}, \ldots, y_{N,M}\}$. *We refer to* (16)-(17) *as a 2-D CRSD discrete, time invariant, mimo state-space model [5]. If the dimension of the local state* $(n = n_h + n_v)$ *is minimal, we say the system is minimal of order* n.

Let us define the past $\{H_1\}$ *and future* $\{H_2\}$ *input/output data matrices as follow*

$$H_1 = \left[\frac{U_1}{Y_1}\right]$$

$$H_2 = \left[\frac{U_2}{Y_2}\right]$$

$$U_1 = \begin{bmatrix} U_{i,[0|j]} & \cdots & U_{i+s-1,[0|j]} \\ U_{i+1,[0|j]} & \cdots & U_{i+s,[0|j]} \\ \vdots & \ddots & \vdots \\ U_{i+r-1,[0|j]} & \cdots & U_{i+r+s-2,[0|j]} \end{bmatrix}$$

$$Y_1 = \begin{bmatrix} Y_{i,j} & Y_{i+1,j} & \cdots & Y_{i+s-1,j} \\ Y_{i+1,j} & Y_{i+2,j} & \cdots & Y_{i+s,j} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{i+r-1,j} & Y_{i+r,j} & \cdots & Y_{i+r+s-2,j} \end{bmatrix}$$

$$U_2 = \begin{bmatrix} U_{i+r,[0|j]} & \cdots & U_{i+r+s-1,[0|j]} \\ U_{i+r+1,[0|j]} & \cdots & U_{i+r+s,[0|j]} \\ \vdots & \ddots & \vdots \\ U_{i+2r-1,[0|j]} & \cdots & U_{i+2r+s-2,[0|j]} \end{bmatrix}$$

$$Y_2 = \begin{bmatrix} Y_{i+r,j} & Y_{i+r+1,j} & \cdots & Y_{i+r+s-1,j} \\ Y_{i+r,j} & Y_{i+r+1,j} & \cdots & Y_{i+r+s,j} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{i+2r-1,j} & Y_{i+2r,j} & \cdots & Y_{i+2r+s-2,j} \end{bmatrix}$$

where $\{Y_1, Y_2\} \in \Re^{r\ell \times s}$ *and* $\{U_1, U_2\} \in \Re^{r(j+1)m \times s}$, *and* $s \gg r$. *From* (16) – (17), *one can easily derive the following subspace equations (also known as block processing form of* (18) – (19))

$$Y_1 = \Gamma_h X^h_1 + H_T U_1 \tag{20}$$
$$Y_2 = \Gamma_h X^h_2 + H_T U_2 \tag{21}$$
$$X^h_2 = A^r_1 X^h_1 + \Delta_{r,j} U_1 \tag{22}$$

where

$$H_T = \left[\, \mathcal{H}_v \mid H_h \,\right]$$

$$\Gamma_h = \begin{bmatrix} C_1 \\ C_1 A_1 \\ \vdots \\ C_1 A_1^{r-1} \end{bmatrix}$$

$$\mathcal{H}_v = \begin{bmatrix} C_2 A_4^{j-1} B_2 & C_2 A_4^{j-2} B_2 & \cdots & C_2 B_2 \\ C_2 A_2 A_4^{j-1} B_2 & C_2 A_2 A_4^{j-2} B_2 & \cdots & C_2 A_2 B_2 \\ \vdots & \vdots & \vdots & \ddots \\ C_2 A_1^{r-2} A_2 A_4^{j-1} B_2 & C_2 A_1^{r-2} A_2 A_4^{j-2} B_2 & \cdots & C_2 A_1^{r-2} A_2 B_2 \end{bmatrix}$$

$$\Delta_{r,j} = \left[\; A_1^{r-1} \mathcal{B}_j \;\middle|\; A_1^{r-2} \mathcal{B}_j \;\middle|\; \cdots \;\middle|\; \mathcal{B}_j \; \right]$$

$$\mathcal{B}_j = \left[\; A_2 \Delta_j \;\middle|\; B_1 \; \right]$$

$$\Delta_j = \left[\; A_4^{j-1} B_2 \;\middle|\; A_4^{j-2} B_2 \;\middle|\; \cdots \;\middle|\; B_2 \; \right]$$

$$H_h = \left[\begin{array}{c|ccc} D & 0 & 0 & \cdots \; 0 \\ \hline C_1 B_1 & \mathcal{D}_j & 0 & \cdots \; 0 \\ C_1 A_1 B_1 & C_1 \mathcal{B}_j & \mathcal{D}_j & \cdots \; 0 \\ \vdots & \vdots & \vdots & \ddots \; \vdots \\ C_1 A_1^{r-2} B_1 & C_1 A_1^{r-3} \mathcal{B}_j & C_1 A_1^{r-4} \mathcal{B}_j & \cdots \; \mathcal{D}_j \end{array}\right]$$

$$\mathcal{D}_j = \left[\; C_2 \Delta_j \;\middle|\; D \; \right]$$

$$X_1^h = \begin{bmatrix} x_{i,j}^h & x_{i+1,j}^h & \cdots & x_{i+s-1,j}^h \end{bmatrix}$$

$$X_2^h = \begin{bmatrix} x_{i+r,j}^h & x_{i+r+1,j}^h & \cdots & x_{i+r+s-1,j}^h \end{bmatrix}$$

$$X_1^v = \begin{bmatrix} x_{i,j}^v & x_{i+1,j}^v & \cdots & x_{i+s-1,j}^v \end{bmatrix}$$

$$X_2^v = \begin{bmatrix} x_{i+r,j}^v & x_{i+r+1,j}^v & \cdots & x_{i+r+s-1,j}^v \end{bmatrix}$$

Under the conditions that $\text{rank}\{X_2^h\} = n_h$ *and* $\text{span}_{\text{row}}\{X_2^h\} \cap \text{span}_{\text{row}}\{U_2^\perp\} = \oslash$, *we have the following properties which are similar to those in [2]*

$$\begin{array}{lrcl} i. & \text{rank}\{H_1\} & = & rm(j+1) + n_h \\ ii. & \text{rank}\{H_2\} & = & rm(j+1) + n_h \\ iii. & \text{rank}\{\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}\} & = & 2rm(j+1) + n_h \\ iv. & dim\{H_1 \cap H_2\} & = & n_h \\ v. & \text{span}_{\text{row}}\{X_2^h\} & = & \text{span}_{\text{row}}\{H_1\} \cap \text{span}_{\text{row}}\{H_2\} \end{array}$$

Thus, any basis for the above intersection can be transformed into a valid state vector sequence X_2^h. *First, we need to redefine the orthogonal complement as follows*

$$\left[U_{12}^T \mid U_{22}^T \right] \stackrel{\text{def}}{=} \left[T_1 \mid T_2 \mid T_3 \mid T_4 \right]$$

where the partitioning corresponds to the input/output parts in H_1 *and* H_2, *respectively. Explicit formulas for the* T*'s can be derived from the above* I/O *equations* (18) – (20), *i.e.*,

$$\begin{aligned}
&\left[T_1 \mid T_2 \mid T_3 \mid T_4 \right] \\
&= \mathcal{A} \left[A_1^r \Gamma_h^\dagger H_T - \Delta_{r,j} \mid -A_1^r \Gamma_h^\dagger \mid -\Gamma_h^\dagger H_T \mid \Gamma_h^\dagger \right] \\
&\quad + \mathcal{B} \left[0 \mid 0 \mid -\Gamma_h^\perp H_T \mid \Gamma_h^\perp \right] \\
&\quad + \mathcal{C} \left[-\Gamma_h^\perp H_T \mid \Gamma_h^\perp \mid 0 \mid 0 \right]
\end{aligned} \tag{23}$$

where, $\mathcal{A}, \mathcal{B}$, *and* $\mathcal{C}$ *are unknown matrices of appropriate dimensions, and* $\Gamma_h^\perp$ *is the orthogonal complement of* Γ_h *such that*

$$\Gamma_h^\perp \Gamma_h = 0$$

See e.g. [3] for a similar derivation. One can easily show that

$$\begin{aligned}
T_4^\dagger (T_2 T_4^\dagger T_3 - T_1) &= \Gamma_h \Delta_{r,j} \\
&= \mathcal{H}_h
\end{aligned} \tag{24}$$

where $Z\mathcal{H}_h$ *is the horizontal Hankel matrix defined in [5], where* Z *is a matrix with ones on the opposite main diagonal. Similarly, one can show that*

$$\begin{aligned}
T_4^\dagger T_3 &= -H_T \\
&= -\left[\mathcal{H}_v \mid H_h \right]
\end{aligned} \tag{25}$$

from which the vertical Hankel matrix can be extracted, i.e., $Z\mathcal{H}_v$. *The above formulas reveal the horizontal/vertical decomposition of the 2-D CRSD filter. The identification is done as follows. First, compute the svd of* $\mathcal{H}_h$ *and* $\mathcal{H}_v$, *i.e.*,

$$\mathcal{H}_h = U_h S_h V_h^T$$

$$\begin{aligned}
\mathcal{H}_v &= U_v S_v V_v^T \\
&= (U_v S_v^{\frac{1}{2}})(S_v^{\frac{1}{2}} V_v^T) \\
&= \Gamma_v \Delta_j
\end{aligned}$$

Then compute the balancing transformation

$$\mathcal{T} = \left[(T_2 T_4^\dagger T_3 - T_1) \cdot V_h S_h^{-\frac{1}{2}} \right]^\dagger \tag{26}$$

Alternatively, one can compute a $\mathcal{T}$ *matrix as*

$$\mathcal{T} = S_h^{-\frac{1}{2}} U_h^T T_4^\dagger \tag{27}$$

thus, avoiding the computation of another pseudo-inverse.

Once $\mathcal{T}$ is known, X_2^h is computed, which is known beforehand to correspond to a horizontally balanced realization, i.e.,

$$X_2^h = \underbrace{S_h^{-\frac{1}{2}} U_h^T T_4^\dagger}_{\mathcal{T}} \cdot \left[\begin{array}{c|c} T_1 & T_2 \end{array}\right] \cdot \left[\begin{array}{c} U_1 \\ \hline Y_1 \end{array}\right]$$

Similarly, the vertical state vector sequence can be computed from

$$X_2^v = \Delta_j U_2$$

Finally, the system matrices can be computed from

$$P = \begin{bmatrix} x^h_{i+r+1,j} & \cdots & x^h_{i+r+s-1,j} \\ y_{i+r,j} & \cdots & y_{i+r+s-2,j} \end{bmatrix}$$

$$Q = \begin{bmatrix} x^h_{i+r,j} & \cdots & x^h_{i+r+s-2,j} \\ x^v_{i+r,j} & \cdots & x^h_{i+r+s-2,j} \\ u_{i+r,j} & \cdots & u_{i+r+s-2,j} \end{bmatrix}$$

$$P = \begin{bmatrix} A_1 & A_2 & B_1 \\ C_1 & C_2 & D \end{bmatrix} \cdot Q$$

$$B_2 = \Delta_j \begin{bmatrix} 0 \\ 0 \\ \vdots \\ I_{n_v} \end{bmatrix}$$

$$A_4 = \overleftarrow{\Delta}_j \, \Delta_j^\dagger$$

where $\leftarrow$ denotes a shift of m blocks to the left, with zeros padded on the right.

References

[1] *S.Y. Kung, "A new identification and model reduction algorithm via singular value decomposition," Proc. 12th Asilomar Conf. on Circuits, Systems and Computers, Pacific Grove, CA, pp. 705-714, 1978.*

[2] *M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle, "On- and off-line identification of linear state space models," Int. Journal of Control,* **49**, *No. 1, pp. 219-232, 1989.*

[3] *M. Moonen, J. Ramos, A subspace algorithm for balanced state space system identification, ESAT/SISTA report 1991-07, Kath. Universiteit Leuven, Dept. E.E. , Belgium. Submitted to IEEE Trans Automat. Contr.*

[4] *R.P. Roesser, "A discrete state-space model for image processing," IEEE Trans. Automat. Contr.,* **AC-20**, *No. 1, pp. 1-10, 1975.*

[5] *B. Lashgari, L.M. Silverman, and J.F. Abramatic, "Approximation of 2-D separable in denominator filters," IEEE Trans. Circuits Syst.,* **CAS-30**, *No. 2, pp. 107-121, 1983.*

[6] *Hsu, C.S., Desai, U.B., and C.A. Crawley. 1985. "Reduction and Approximation of Discrete Bilinear Systems." In Applied Digital Control, S.G. Tzafestas, ed. North Holland, Amsterdam.*

[7] *Marszalek, W. 1985. "Analysis of Bilinear Systems with Picard's Method and Block-Pulse Operational Matrices." Journal of the Franklin Institute Nos. 3/4: 105-109.*