Boston SIAM Conf 1986

A minimal partial realization algorithm for data with relative errors.

MARC VAN BAREL AND ADHEMAR BULTHEEL *

RUNNING TITLE: mpr and relative errors

* Katholieke Universiteit Leuven, Department of Computer Science, Leuven, Belgium.

Abstract. In this paper an algorithm is presented to compute a minimal partial realization (mpr) of the form $C(z) \cdot A(z)^{-1}$ given N $(m \times n)$-Markov parameters which are not exactly known ($C(z)$ and $A(z)$ are polynomial matrices, respectively $(m \times n)$ and $(n \times n)$). The algorithm is based on the singular value decomposition of certain block-Hankel matrices containing the Markov parameters. Insight is given in the errors between the given Markov parameters and those of the computed mpr $C(z) \cdot A(z)^{-1}$. It is shown that the method is a special case of the total linear least squares technique for identification and parameter estimation and that it can also be used for model reduction. Finally a numerical example is presented.

1. Introduction

The notation used will be for discrete stable linear systems but everything can also be applied to continuous linear systems.

The problem considered here is very important in linear systems theory. Given a finite sequence of N matrices $H_i \in \mathbb{R}^{m \times n}$, $i = 1, \ldots, N$, called Markov parameters, find an $(m \times n)$ polynomial matrix $C(z)$ and a nonsingular $(n \times n)$ polynomial matrix $A(z)$ such that :

$$C(z) \cdot A(z)^{-1} = H_1 z^{-1} + H_2 z^{-2} + \cdots + H_N z^{-N} + O_{-}(z^{-N-1}) \qquad (1)$$

with $C(z)$ and $A(z)$ right coprime and the degree of the determinant of $A(z)$ (deg det $A(z)$) as small as possible. (An analoguous definition can be given for the left polynomial matrix fraction representation $B(z)^{-1} \cdot D(z)$). $C(z) \cdot A(z)^{-1}$ is the transfer function of a discrete linear system. The coefficients of the power series of $C(z) \cdot A(z)^{-1}$ around $z = \infty$ form the impulse response of the system. The deg det $A(z)$ is equal to the minimal number of states to realize the system. So (1) means that we are looking for a transfer function representation of a system

* Katholieke Universiteit Leuven, Department of Computer Science, Leuven, Belgium.

with a minimal number of states such that the first N impulse response coefficients are H_i, $i = 1, \ldots, N$. These Markov parameters H_i are supposed to be measured with a certain relative precision. (See [1] and [3] for the single-input single-output case with absolute errors on the Markov parameters).

In Section 2 the problem is transformed and solved under the assumption that the Markov parameters are exactly known. Also the connection is given with the total linear least squares technique used in [5]. Section 3 contains an algorithm to solve the inexact problem. In Section 4 insight is given into the errors between the Markov parameters of the computed system and the given Markov parameters. Section 5 shows a numerical example. Section 6 gives some notes and conclusions on the effectiveness, the numerical stability and the adaptation to the model reduction problem of the algorithm of section 3.

2. Solution of the problem when the Markov parameters are exactly known

We denote the j-th column of $A(z)$ by $a_j(z) = a_{0,j} + a_{1,j}z + \cdots + a_{d_j,j}z^{d_j}$ with $a_{i,j} \in \mathbb{R}^n$ and $a_{d_j,j} \neq 0$. d_j is called the degree of the polynomial vector $a_j(z)$ (we can define the degree of the zero polynomial vector as $-\infty$). The matrix $HD_A = [a_{d_1,1} \;\; a_{d_2,2} \;\cdots\; a_{d_n,n}]$ is called the highest degree coefficient matrix of the polynomial matrix $A(z)$. When HD_A is nonsingular we say that $A(z)$ is column reduced. In this case deg det $A(z) = \sum_1^n d_j$. To transform the problem we need the definition of a unimodular matrix. A square polynomial matrix $U(z)$ is unimodular when det $U(z)$ is a nonzero constant. Every polynomial matrix $A(z)$ can be transformed into a column reduced polynomial matrix by right multiplication with a unimodular matrix : $A'(z) = A(z) \cdot U(z)$ and $HD_{A'}$ is nonsingular. This means that we can look for $C(z) \cdot A(z)^{-1}$ with $A(z)$ column reduced and $\sum_1^n d_j$ minimal satisfying (1).

The denominator columns $a_j(z)$, called solution vectors of level N given N Markov parameters, can be found as solutions of the following set of homogeneous linear equations with a block-Hankel coefficient matrix [4] :

$$\begin{bmatrix} H_1 & \cdots & H_{d+1} \\ \vdots & & \vdots \\ H_{N-d} & \cdots & H_N \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad H_{N,d} \cdot a(z) = 0. \tag{2}$$

Note that we use $a(z)$ and $[a_0^T \cdots a_d^T]^T$ as equivalent notations. The corresponding column of the numerator polynomial matrix can be computed as :

$$\begin{bmatrix} 0 & 0 & \cdots & H_1 \\ \vdots & \vdots & & \vdots \\ 0 & H_1 & \cdots & H_d \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} c_{d-1} \\ \vdots \\ c_0 \end{bmatrix} \quad \text{or } H'_{N,d} \cdot a(z) = c^*(z). \tag{3}$$

We can rewrite this as

$$\begin{bmatrix} H'_{N,d} & I \\ H_{N,d} & 0 \end{bmatrix} \cdot \begin{bmatrix} a(z) \\ -c^*(z) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} .$$

Note the similarity between this problem for the single input- single output case and the parameter estimation problem [5]. I.e. given an input $u(z)$ and an output $y(z)$, find the

parameters of a transfer function of degree d that matches the given data as close as possible. Their algorithm uses the total linear least squares technique but their article gives no insight about the errors between the given output $y(z)$ and the output $\tilde{y}(z)$ of the computed system when the same input $u(z)$ is applied. In our problem the input $u(z) = 1$ so the coefficients of the output power series $y(z)$ are the Markov parameters.
So (2) indicates that we have to study the structure of the kernels of the block-Hankel matrices $H_{N,d}$. We put the solution vectors of level p, $a_j^p(z)$, given p Markov parameters with $p = 1, \ldots, N$ in a two-dimensional table called the solution table. See table 1.

level \ degree	0	1	2	⋯	
0	$a_1^0(z) \ldots a_n^0(z)$				
1		$a_1^1(z)$	⋯	$a_n^1(z)$	
2					
3					
⋮					
p			$a_1^p(z)$	⋯	$a_n^p(z)$
⋮					
N				$a_1^N(z)$ ⋯	$a_n^N(z)$

Table 1: Solution table

Because of the special nesting of the Hankel matrices $H_{p,d}$, it should be clear that if $a(z)$ is in Ker $H_{p,d}$, then $a(z)$ will also be in Ker $H_{p+1,d+1}$ and Ker $H_{p,d+1}$. Moreover $za(z)$ will also be in Ker $H_{p,d+1}$ and these relations are of course transitive. Thus Ker $H_{p,d}$ will contain Ker $H_{p,j}$, $j=0,1,\ldots,d-1$, Ker $H_{p-1,j}$, $j=0,1,\ldots,d-1$, Ker $H_{p-2,j}$, $j=0,1,\ldots,d-2$, etc. and shifted versions of these. The part of Ker $H_{p,d}$ that is not contained in all these will be spanned by the solution vectors of level p and degree d. By using this reasoning recursively, it will be clear that it is possible to construct a basis for Ker $H_{p,d}$ from the solution vectors that can be found in the solution table inside the triangle with vertices $(p,d),(p,0),(p-d,0)$. See table 2.
The solution vectors on the diagonal through $(p,d),(p-d,0)$ are said to have potential degree $\sigma=d$ (with respect to level p). More generally, the solution vectors on the diagonal $(p,d-k),(p-d+k,0)$ have potential degree $\sigma=d-k$ (w.r.t. level p). Thus a basis for Ker $H_{p,d}$ can be found from the solution vectors having a potential degree at most d. More precisely, the following theorem can be proved.

Theorem 1 [4]

A basis $BH_{p,d}$ for Ker $H_{p,d}$ is given by

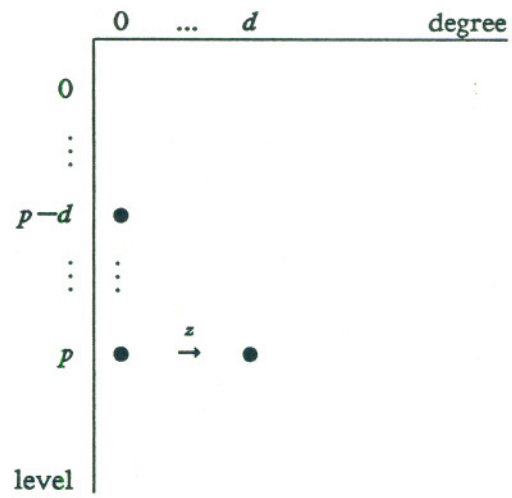


Table 2

$$BH_{p,d} = \cup\, z^{d-\sigma_j^i} a_j^i(z)$$

where the union is over all solution vectors $a_j^i(z)$ of potential degree $\sigma_j^i = d_j^i - i + p$ at most d. (d_j^i is the degree of $a_j^i(z)$).
This means that dim Ker $H_{p,d}$ is equal to the number of solution vectors of potential degree at most d (w.r.t. level p), and this is equal to the number of solution vectors in the triangle depicted in table 2.

If $H(z) = \sum_1^\infty H_k z^{-k}$ is a rational matrix function, then Ker $H_{N,d}$ will stabilize from a certain N on. I.e.

$$\text{Ker } H_{N,d} = \text{Ker } H_{N',d} \quad \forall\, N' \geqslant N.$$

This means that the solution vectors stabilize too. I.e.

$$a_j^{N'}(z) = a_j^N(z) \quad \forall\, N' \geqslant N.$$

From now on, we shall suppose that N is large enough to have reached this point of stabilization (see table 3). It then follows from theorem 1 that

$$\dim \text{Ker } H_{N,d+1} = \dim \text{Ker } H_{N,d} + k + j$$

if k is the number of solution vectors of level N of degree $\leqslant d$ and j is the number of solution vectors of level N of degree $d+1$. This means that the solution vectors have reached a final value.
Note that the nested structure of these bases can be used to construct a recursive algorithm to compute the solution vectors ([2] and [4]). This algorithm can not be changed into a robust one when the Markov parameters are inaccurate but a robust one can be designed as we shall presently do.
Here we propose a nonrecursive algorithm to compute the solution vectors of level N based on the singular value decomposition. Recall that every matrix $A \in \mathbb{R}^{p\times q}$ of rank $s < \min(p,q)$ has a singular value decomposition (SVD) : $A = U \cdot \Sigma \cdot V^T$ with $U \in \mathbb{R}^{p\times p}$ and $V \in \mathbb{R}^{q\times q}$ orthogonal matrices and Σ a diagonal $(p\times q)$ matrix with diagonal elements (singular values) $\sigma_1 \geqslant \sigma_2$

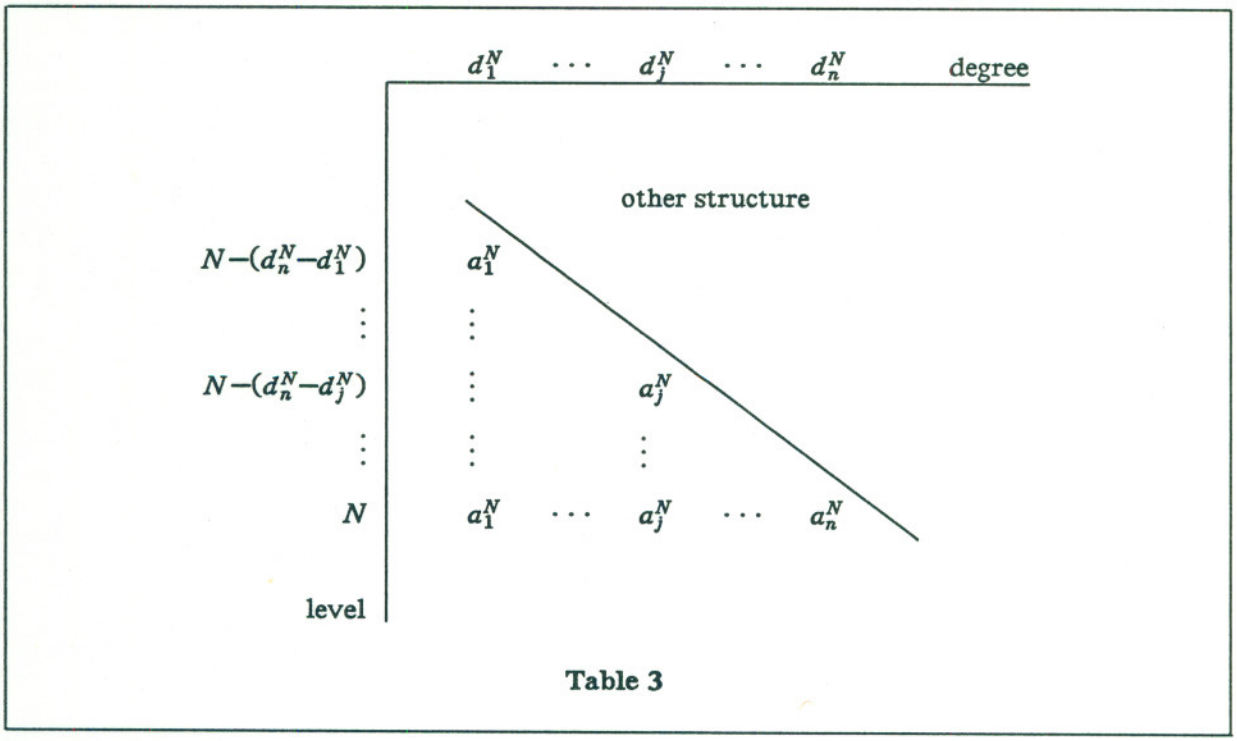


Table 3

$\geqslant \sigma_s > \sigma_{s+1} = \cdots = \sigma_{\min(p,q)} = 0$. The last $q-s$ columns of V form an orthonormal basis for the nullspace of A. The algorithm to compute the solution vectors is as follows :

1) Take $H_{N,0}$, $H_{N,1}$, $\cdots$ until the nullspace is different from $\{0\}$. Suppose this occurs for H_{N,d_1} :

$$H_{N,d_1} = \begin{bmatrix} U_1^{d_1} \\ U_2^{d_1} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^{d_1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{d_1 T} \\ V_2^{d_1 T} \end{bmatrix} \text{ with } \begin{cases} U_1^{d_1} \in \mathbb{R}^{m \times (N-d_1)m} \\ \Sigma_1^{d_1} \in \mathbb{R}^{s \times s} \text{ nonsingular} \end{cases}$$

(Here and below, a partitioning of some U matrix will be by its first m rows and the rest of it, the partitioning of a Σ matrix is with respect to its significant and nonsignificant values and the partitioning of a V matrix is conformal with the corresponding Σ.) Thus Ker H_{N,d_1} has dimension $(d_1+1)n-s$ and therefore there are $(d_1+1)n-s$ solution vectors of degree d_1 [4]. We can take the columns of V_2 as these solution vectors.

2) Suppose we have already made a choice for the first k solution vectors of degree smaller than d. If $k=n$ the algorithm ends. $H_{N,d-1}$ can be written as :

$$H_{N,d-1} = \begin{bmatrix} U_1^{d-1} \\ U_2^{d-1} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^{d-1} & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{d-1 T} \\ V_2^{d-1 T} \end{bmatrix} \text{ with } \begin{cases} U_1^{d-1} \in \mathbb{R}^{m \times (N-d+1)m} \\ \Sigma_1^{d-1} \in \mathbb{R}^{r \times r} \text{ nonsingular} \end{cases}$$

The number of columns of V_2^{d-1} i.e. dim Ker $H_{N,d-1}$ is equal to the number of solution vectors in the triangle with vertices $(N,0)$, $(N,d-1)$, $(N-d+1,0)$ of the solution table. Suppose this number is l. It then follows from theorem 1 and the structure of table 3 that dim Ker $H_{N,d}$ is $l+k+j$ where k is the number of solution vectors of level N of degree $<d$ and j (possibly $j=0$) is the number of solution vectors for level N of degree d. Now let us construct Ker $H_{N,d}$. Therefore we write $H_{N,d}$ as follows :

$$H_{N,d} = \left[\begin{array}{c|c} \begin{bmatrix} H_1 \\ \vdots \\ H_{N-d} \end{bmatrix} & \\ \hline & \overline{H}_{N,d-1} \end{array}\right]$$

where $\overline{H}_{N,d-1}$ is $H_{N,d-1}$ without its first block row. Thus

$$H_{N,d} = \underbrace{\left[\begin{array}{c|c} \begin{bmatrix} H_1 \\ \vdots \\ H_{N-d} \end{bmatrix} & U_2^{d-1} \cdot \begin{bmatrix} \Sigma_1^{d-1} & \\ 0 & \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \end{array}\right]}_{(4)} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1^T} \\ 0 & V_2^{d-1^T} \end{bmatrix}.$$

Suppose the SVD of (4) is

$$U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix}.$$

Then

$$H_{N,d} = \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1'^T & \\ & V_2'^T \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \right] \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1^T} \\ 0 & V_2^{d-1^T} \end{bmatrix}$$

$$= U' \cdot \begin{bmatrix} \Sigma_1' & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} V_1'^T & 0 \\ V_2'^T & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1^T} \\ 0 & V_2^{d-1^T} \end{bmatrix}}_{(5)}.$$

(5) is orthogonal, hence the singular value decomposition of $H_{N,d}$ can be written as :

$$H_{N,d} = U' \cdot \begin{bmatrix} \Sigma_1' & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \left[\begin{array}{cc} \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1^T} \end{bmatrix} & \\ \begin{bmatrix} 0 \end{bmatrix} & V_2^{d-1^T} \end{array}\right]$$

or

$$H_{N,d} = \begin{bmatrix} U_1^d \\ U_2^d \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1^d & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} V_1^{d^T} \\ V_2^{d^T} \end{bmatrix}$$

with

$$\begin{cases} U_1^d \text{ the first } m \text{ rows of } U' \\ U_2^d \text{ the last } (N-d-1)m \text{ rows of } U' \\ \Sigma_1^d = \Sigma_1' \\ V_1^{dT} = V_1'^T \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1T} \end{bmatrix} \end{cases}$$

$$V_2^{dT} = \begin{bmatrix} V_2'^T \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^{d-1T} \end{bmatrix} \\ 0 \qquad V_2^{d-1T} \end{bmatrix}.$$

The lower part of V_2^{dT} shows that Ker $H_{N,d-1}$ is a part of Ker $H_{N,d}$. It thus contains l rows. The upper part will give the other $k+j$ basis vectors of Ker $H_{N,d}$. Thus if the number of columns of

$$\begin{bmatrix} I & 0 \\ 0 & V_1^{d-1} \end{bmatrix} \cdot [V_2']$$

is equal to $k+j$, there are j solution vectors of degree d (see table 3). Choose j columns of these $k+j$ columns such that the highest degree coefficient matrix of the solution vectors of level N computed so far has full column rank.

3) Repeat 2) until there are n solution vectors. The numerator polynomial matrix can then be computed by using (3).

3. The solution of the inexact problem

Suppose now that the Markov parameters H_i, $i = 1, \ldots, N$ are not known exactly but are only given within a relative accuracy ϵ for each entry which we assume to be the same for all the entries of the parameters.

We have to adapt the algorithm of section 2 because the relative errors on the Markov parameters will move certain singular values σ_i from zero to a value which is not zero but which is to be considered "numerically zero", given the errors on the Markov parameters.

Before we state the algorithm to solve the inexact problem we note a property of the singular value decomposition. We shall say that a matrix B is in an (absolute) ϵ-neighbourhood of a matrix A if the difference between the corresponding entries of A and B is bounded by ϵ. Suppose the SVD of the $(m \times n)$ matrix A is : $A = U \cdot \Sigma \cdot V^T$ or $A = \sum_{i=1}^{min(m,n)} \sigma_i u_i v_i^T$ with σ_i the i-th singular value and u_i, v_i the i-th column of U respectively V. Suppose $k < \min(m,n)$ is such that $\sigma_k > \epsilon$ and $\sigma_{k+1} \leqslant \epsilon$. We know that

$$\min_{rank\ \tilde{A} \leqslant k} \|A - \tilde{A}\| = \|\text{diag}(\sigma_{k+1}, \ldots, \sigma_{min(m,n)})\|$$

with $\|\cdot\|$ an orthogonally invariant matrix norm.

A candidate for the matrix $\tilde{A}$ is $\tilde{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. Since the difference between the elements of A and $\tilde{A}$ is bounded by ϵ, $\tilde{A}$ will be in an absolute ϵ-neighbourhood of A. If the gap between σ_k

and σ_{k+1} is sufficiently large, i.e. if $\sigma_k > \sqrt{mn}\,\epsilon$, then there can be no matrix in an ϵ-neighbourhood of A of rank less than k. $\tilde{A}$ will be a matrix with the largest possible dimension for its kernel that is in an ϵ-neighbourhood of A. If the gap between σ_k and σ_{k+1} is not that large, $\tilde{A}$ may still be a matrix of smallest possible rank that is in the ϵ-neighbourhood and if it is not, we use it as an estimate for it anyway. Instead of having the same order of magnitude for the absolute difference between every element of A and B we want to have the same order of magnitude ϵ for the *relative* difference between the elements of A and B. Such a matrix B is said to be in a relative ϵ-neighbourhood of A. Therefore the matrix A has to be scaled such that all its elements have the same order of magnitude α and then the singular value decomposition can be applied to this scaled matrix. The singular values will be different for the scaled and for the unscaled matrix. By setting the singular values less then ϵ equal to zero and by using the inverse scaling we get a matrix $\tilde{A}$ of rank k' approximating the matrix A with the same order of magnitude for the relative error between the corresponding elements. It will be a matrix which is in practical applications a fairly good estimate for a matrix with the largest possible kernel which is in a relative ϵ-neighbourhood of A. More precisely, suppose that there exist diagonal matrices $D_i = \mathrm{diag}(d_{i,1}, d_{i,2}, \ldots)$, $i = 1,2$ such that the entries of the matrix $D_1 A D_2$ have the same order of magnitude. I.e. $|d_{1,i} a_{i,j} d_{2,j}| = \alpha_{i,j} \approx \alpha$. If it has a SVD $D_1 A D_2 = \sum \sigma_i u_i v_i^T$, then $\bar{A} = \sum_1^{k'} \sigma_i u_i v_i^T$ will approximate $D_1 A D_2$ such that $|d_{1,i} a_{i,j} d_{2,j} - \bar{a}_{i,j}| \leqslant \sigma_{k'+1}$. Hence if we set $\tilde{A} = D_1^{-1} \bar{A} D_2^{-1}$, then $|a_{i,j} - \tilde{a}_{i,j}| \leqslant \sigma_{k'+1} |d_{1,i}^{-1} d_{2,j}^{-1}| = \sigma_{k'+1} |a_{i,j}| \alpha_{i,j}^{-1} \approx \sigma_{k'+1} |a_{i,j}| \alpha^{-1}$. This shows that if the i,j-entry is not well scaled, i.e. if $\alpha_{i,j}$ is substantially smaller than the average α, then the approximation of this entry may have a substantially larger relative error as is indicated by the factor $\alpha_{i,j}^{-1}$.

Note that if the columns of V span the kernel of $\bar{A}$, then the columns of $D_2 V$ will span the kernel of $\tilde{A}$.

To obtain the scaling of the matrix A, we compute D_1 and D_2 by the following algorithm:

repeat 5 times

- compute the Euclidean norm of each row of A and divide this row by the norm ⇒ the length of this row is one
- do the same thing with each of the n columns of A.

This makes the value $\alpha \approx |d_{1,i} a_{i,j} d_{2,j}|$ of order 1.

This leads to the following algorithm for the inexact problem (with the same assumption concerning the structure of the solution vectors as in section 2) :

1) First of all we search for solution vectors of degree 0 :

$$\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_N \end{bmatrix} \cdot [a_0] = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{or} \quad H_{N,0} \cdot a_0 = 0.$$

The nullspace of $H_{N,0}$ is approximated by Ker $\tilde{H}_{N,0}$ with $\tilde{H}_{N,0}$ in a relative ϵ-neighbourhood of $H_{N,0}$ and which is obtained by SVD of the scaled version of $H_{N,0}$ as

explained above.

Suppose

$$D_1 \cdot H_{N,0} \cdot D_{2,0} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{6}$$

where the diagonal elements of Σ_2 are numerically zero, i.e. at most ϵ (the relative error on the data). The partitioning of the matrices is as agreed upon in the previous section. Then we approximate the solution vectors of degree 0 by the columns of

$$D_{2,0} \cdot V_2.$$

Note that if V_2 is empty, then there are no solution vectors of degree 0.

2) Now we should construct solution vectors of degree 1. Thus we have to build $H_{N,1}$, scale it and compute its SVD. Since our algorithm is based on the nesting of the kernels of $H_{N,d}$, we should build Ker $H_{N,1}$ such that Ker $H_{N,0}$ appears as a subspace. If we completely rescaled $H_{N,1}$ and computed its SVD, then the solution vectors of degree 0 (if there are any) will not be found explicitly in the expression for Ker $H_{N,1}$. They even may not be in this kernel. Thus if we don't want to loose the information we have already gathered about our system from the computed SVD, we shouldn't throw it away. So we use the computed SVD (6) in our search for the solution vectors of degree 1. So the method to compute the solution vectors of degree 1 when we know the SVD (6) is as follows :
Because $H_{N,1}$ is a block-Hankel matrix, it can be written as : (see section 2)

$$H_{N,1} = \begin{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{N-1} \end{bmatrix} \vdots \\ D_1'^{-1} \cdot U_2 \cdot \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \cdot \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \cdot D_{2,0}^{-1} \end{bmatrix}.$$

D_1' is equal to D_1 without its first m rows and columns.

$$H_{N,1} = D_1'^{-1} \cdot \Bigg[\overbrace{\underbrace{D_1' \cdot \begin{bmatrix} H_1 \\ \vdots \\ H_{N-1} \end{bmatrix}}_{(8)} \cdot D_{2,1} \vdots U_2 \cdot \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} \vdots U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix}}^{(7)} \Bigg] \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} \cdot \begin{bmatrix} D_{2,1}^{-1} & 0 \\ 0 & D_{2,0}^{-1} \end{bmatrix}.$$

$D_{2,1}$ is computed such that the columns of (8) have length one.
We suppose that this method will give us a matrix (7) which is sufficiently well scaled. At the end of this section we shall come back to the question whether this is true or not.
We compute the SVD of (7) and so we can rewrite $H_{N,1}$ as :

$$H_{N,1} = D_1'^{-1} \cdot \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ 0 & \Sigma_2' \end{bmatrix} \cdot \begin{bmatrix} V_1'^T \\ V_2'^T \end{bmatrix} \vdots U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \right] \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} \cdot \begin{bmatrix} D_{2,1}^{-1} & 0 \\ 0 & D_{2,0}^{-1} \end{bmatrix}$$

$$= D_1'^{-1} \cdot \left[U' \cdot \begin{bmatrix} \Sigma_1' & 0 \\ & \\ 0 & \Sigma_2' \end{bmatrix} \vdots \; U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \right] \cdot \underbrace{\begin{bmatrix} V_1'^T & 0 \\ V_2'^T & 0 \\ 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1^T \\ 0 & V_2^T \end{bmatrix} \cdot \begin{bmatrix} D_{2,1}^{-1} & 0 \\ & \\ 0 & D_{2,0}^{-1} \end{bmatrix}}_{(9)} .$$

If we set U_2' to be the last columns of U_2 such that $U_2[0\ \Sigma_2]^T = U_2'\Sigma_2$ and define $U = [U' : U_2']$ and $V^T = (9)$, then

$$D_1' \cdot H_{N,1} \cdot \begin{bmatrix} D_{2,1} & 0 \\ 0 & D_{2,0} \end{bmatrix} = U \cdot \begin{bmatrix} \Sigma_1' & 0 & 0 \\ 0 & \Sigma_2' & 0 \\ 0 & 0 & \Sigma_2 \end{bmatrix} \cdot V^T$$

Note that V is an orthogonal matrix and the columns of U' are orthonormal but U_2' is not orthogonal, nor are columns of U' orthogonal to columns of U_2'. So the above decomposition doesn't give a SVD of the scaled version of the matrix $H_{N,1}$.

Nevertheless we can find an approximation for the solution vectors of degree 1 from this decomposition. Let the diagonal elements of Σ_2' be numerically zero. Suppose the number of these elements is $k+j$ where k is the number of solution vectors of degree 0. Then there are j solution vectors of degree 1.

We choose these solution vectors as column vectors of

$$\begin{bmatrix} D_{2,1} & 0 \\ 0 & D_{2,0} \end{bmatrix} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1 \end{bmatrix} \cdot V_2' \qquad (10)$$

such that the matrix of highest degree coefficients of all the solution vectors found so far has full column rank. To this end we use the QR decomposition with partial column pivoting. We construct a matrix HD whose first k columns are the highest degree coefficients (*hdc*) of the solution vectors of degree 0. These columns can not be pivoted. The following $k+j$ columns are the *hdc* of the solution vectors of degree 1. These columns can be pivoted. So we get :

$$HD = [\underbrace{HD_1}_{k \text{ columns}} \mid \underbrace{HD_2}_{k+j \text{ columns}}].$$

After the QR decomposition with partial column pivoting :

$$Q^T \cdot HD \cdot \underset{\substack{\\ \\ k \quad k+j}}{\begin{bmatrix} I & 0 \\ 0 & J \end{bmatrix}} = \underset{\substack{\\ \\ k \quad j \quad k}}{\left[\begin{array}{c|cc} R & X' & X'' \\ 0 & R' & Y'' \\ 0 & 0 & R'' \end{array}\right]} .$$

Suppose the $(k+i)$-th column of $HD \cdot \begin{bmatrix} I & 0 \\ 0 & J \end{bmatrix}$ is the $(k+j_i)$-th column of HD then the j solution vectors of degree 1 are the columns j_i, $i = 1, \ldots, j$ of (10). Note that we don't need to compute the whole QR decomposition. It is enough to compute the j_i. Note that we could use $\left[\begin{array}{c|c} R & X' \\ 0 & R' \end{array}\right]$ and Q^T in the next step to compute the solution vectors of degree 2.

3) The same procedure can be repeated to obtain solution vectors of degree 2,3,... until we have n solution vectors.
Note that in each step the row scaling matrix D_1 is loosing its first m diagonal elements and the column scaling matrix is extended with n new diagonal elements $D_{2,i}$ on top.

Now, let us come back to the question whether the matrix (7) is properly scaled. It requires some simple algebra to verify that

$$U_2 \cdot \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} = \left(D_1' \cdot \begin{bmatrix} H_2 \\ \vdots \\ H_N \end{bmatrix} \cdot D_{2,0} - U_2 \cdot \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix} \cdot V_2^T \right) \cdot V_1 \, .$$

So that

$$(7) = \underbrace{D_1' \cdot \begin{bmatrix} H_1 & H_2 \\ \vdots & \vdots \\ H_{N-1} & H_N \end{bmatrix} \cdot \begin{bmatrix} D_{2,1} & 0 \\ 0 & D_{2,0} \end{bmatrix}}_{(7')} \cdot \begin{bmatrix} I & 0 \\ 0 & V_1 \end{bmatrix} .$$

If N is large enough we can safely suppose that H_1 and H_N do not contribute substantially to the length of the columns of the central matrix in (7') (which is $H_{N,1}$). This implies that the length of the rows and columns of $D_1' \cdot [H_2^T \ \cdots \ H_N^T]^T \cdot D_{2,0}$ is approximately 1. The scaling of the first block column is a bit more difficult. Indeed, the scaling of the rows is decalated by one block. This means that the row scaling of H_k in $H_{N,0}$ is used to scale H_{k-1} in $H_{N,1}$. Now, if we suppose that the Markov parameters are only slowly varying, i.e. if H_{k-1} has the same order of magnitude as H_k, then there is no problem in this. As a consequence, $D_{2,1}$ will be almost equal to $D_{2,0}$. So, under these assumptions the length of the rows and columns of (7') will have an order of magnitude equal to 1. This shows that (7') is indeed properly scaled. Therefore, the SVD of (7) can be safely done because the transformation of (7') into (7) goes by orthogonal matrix multiplication which doesn't change the singular values.
When the previous assumption is not valid we should look for other more complicated scaling schemes. Take for example a scalar system which oscillates, so that H_k is approximately zero while H_{k-1} has a normal value. In that case, the row scaling factor of H_k will be large to scale it up in the first step. In the second step, this scaling factor will be used to scale H_{k-1} which makes it very large so that this value will dominate the complete column. The column scaling by $D_{2,1}$ will kill all the other elements in the first column. For such situations the following improved scaling procedure seemed to work well in practical situations. When computing the row scaling, we compare the length of a row with the length of some neighbouring rows. We use the maximum of the lengths as a scaling factor for the current row. This means that if a Markov parameter is almost zero compared to neighbouring Markov parameters, we don't scale it up too much to avoid difficulties in the next step. If at the tail of the Markov parameter sequence all the values become small, then they are still scaled up as they should.

4. The correspondence between the given Markov parameters and these of the computed transfer function

For each solution vector $a_j(z)$ of degree d_j we can write :

$$\begin{bmatrix} & & & H_1 \\ & H_1 & \cdots & H_{d_j} \\ H_1 & H_2 & \cdots & H_{d_j+1} \\ \vdots & & & \vdots \\ H_{N-d_j} & H_{N-d_j+1} & \cdots & H_N \end{bmatrix} \cdot \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ \vdots \\ a_{d_j,j} \end{bmatrix} = \begin{bmatrix} c_{d_j-1,j} \\ \vdots \\ c_{0,j} \\ D_{1,d_j+1}^{-1} \cdot \sigma_j \cdot u_{j,d_j+1} \\ \vdots \\ D_{1,N}^{-1} \cdot \sigma_j \cdot u_{j,N} \end{bmatrix} \begin{matrix} \leftarrow z^{d_j-1} \\ \vdots \\ \leftarrow z^0 \\ \leftarrow z^{-1} \\ \vdots \\ \leftarrow z^{-N+d_j} \end{matrix} \qquad (11)$$

with

σ_j (numerically zero) the singular value corresponding to $a_j(z)$
$[u_{j,d_j+1}^T \cdots u_{j,N}^T]^T$ the corresponding left singular vector with each $u_{j,i} \in \mathbb{R}^m$
$\begin{bmatrix} D_{1,1} & & \\ & & D_{1,N} \end{bmatrix} = D_1$ with $D_{1,i} \in \mathbb{R}^{m \times m}$.

Note that (11) is in fact the matrix notation of a power series multiplication. The solution vector can also be written as :

$$\begin{bmatrix} a_{0,j} \\ \vdots \\ a_{d_j,j} \end{bmatrix} = \begin{bmatrix} D_{2,d_j} \cdot v_{0,j} \\ \vdots \\ D_{2,0} \cdot v_{d_j,j} \end{bmatrix} \quad \text{where } D_{2,k} \in \mathbb{R}^{n \times n} \ , \ k = 0,1,\ldots,d_j \qquad (12)$$

are used to compute the solution vectors of degree d_j.

(11) together with (12) gives :

$$H(z^{-1}) - C(z) \cdot A(z)^{-1} = O_-(z^{-N-1}) + \underbrace{(D_{1,d_1+1}^{-1} \cdot [u_{1,d_1+1} \cdots u_{n,d_1+1}] \cdot z^{-d_1-1} + \cdots + D_{1,N}^{-1} \cdot [u_{1,N} \cdots u_{n,N}] \cdot z^{-N}) \cdot \Sigma \cdot}_{(13)}$$

$$\underbrace{(D_{2,0} \cdot [v_{d_1,1} \cdots v_{d_n,n}] + \cdots + D_{2,h} \cdot [v_{d_1-h,1} \cdots v_{d_n-h,n}] \cdot z^{-h})^{-1}}_{(14)}$$

with

$H(z^{-1}) = H_1 z^{-1} + H_2 z^{-2} + \cdots + H_N z^{-N}$
$u_{j,i} = 0$ when $i \leqslant d_j$
$v_{i,j} = 0$ when $i < 0$
$d_1 \leqslant d_2 \leqslant \cdots \leqslant d_n = h$
$\Sigma = \mathrm{diag}(\sigma_1 \cdots \sigma_n)$

Since V_h is invertible, (it is the matrix of highest degree coefficients), the last factor (14) can also

be written as :

$$(I+V_h^{-1}\cdot D_{2,0}^{-1}\cdot D_{2,1}\cdot V_{h-1}z^{-1}+\cdots+V_h^{-1}\cdot D_{2,0}^{-1}\cdot D_{2,h}\cdot V_0\cdot z^{-h})^{-1}\cdot V_h^{-1}\cdot D_{2,0}^{-1},$$

where

$$V_{h-j}=[v_{d_1-j,1}\cdots v_{d_n-j,n}]\ ,\ j=0,\ldots,h.$$

Although we could not prove from these relations that the sequence of the differences between the given system and the computed one behaves like ϵH_k, this is what was numerically observed. Numerical experiments showed e.g. that :

1. The corresponding elements of $D_{2,h}, D_{2,h-1}, \ldots, D_{2,0}$ have the same order of magnitude so that $D_{2,0}^{-1}\cdot D_{2,i}\approx I_{n,n}$.

2. When there are no (almost) multiple poles all elements of the coefficients of the power series around $z=\infty$ of the factor

$$(I+V_h^{-1}D_{2,0}^{-1}D_{2,1}V_{h-1}\cdot z^{-1}+\cdots+V_h^{-1}D_{2,0}^{-1}\cdot D_{2,h}V_0z^{-h})^{-1} \qquad (15)$$

   have order of magnitude 1 and will ultimately decrease.

3. The order of magnitude of the $D_{1,k}^{-1}$ is more or less like the order of magnitude of $H_kD_{2,0}$. (Recall that we scaled to make the magnitude of the entries of order $\alpha=1$.)

4. Whenever the matrix of highest degree coefficients V_h is nearly singular, then V_h^{-1} will be large and this will increase the overall magnitude of the error sequence, so that extra digits will be lost.

5. The absolute error sequence is the convolution of the two Markov parameter sequences related with (13) and (14). Both are decreasing fast enough to make its convolution decrease almost like H_k. The factor Σ makes it proportional to ϵ, but V_h^{-1} may increase the relative error. Anyway, the relative error sequence will be almost constant.

6. When the block-Hankel matrix can not be scaled properly then we can loose up to α digits for the approximation of some entry in a Markov parameter when $\beta.10^{-\alpha}$ is the absolute value of the effectively scaled element at the corresponding entry in the matrix $D_1\cdot H_{N,0}\cdot D_2$.
   This problem occurs when different entries in the transfer function matrix have different dominant poles in modulus.

7. When we consider the single-input single-output case and assume that the Markov parameters are not varying rapidly the scaling can be done very well. In fact one could do without the column scaling.

8. More digits can be lost when there are (almost) multiple poles because then the Markov parameter sequence of (14) will not decrease sufficiently fast or even increase initially and decay only at a later stage.

9. It is clear that the algorithm of section 4 could also be used for model reduction. Instead of defining the concept 'numerically zero' based on the relative error ϵ, it is based on the relative error between the given Markov parameters and the Markov parameters of the approximation to be searched for.

10. It is clear that each step of the algorithm can be implemented in a numerically stable way (The routines DSVDC and DQRDC of LINPACK can be used).

We shall now illustrate these results with an example.

5. Example

All computations were done in double precision on a Microvax II computer. To check the performance of the algorithm described in section 4 we compute the first 50 Markov parameters of the following system with 3 inputs and 2 outputs and denominator column degrees 2, 2 and 4 respectively :

$$\begin{bmatrix} 3.7z-5.4 & -6.3z+4.2 & -2.12z^3+3.5z^2+1.6z+6.7 \\ 5.8z+4 & -2.3z+3.7 & 5.3z^3-8.4z^2+8.4z-2.3 \end{bmatrix} \cdot$$

$$\begin{bmatrix} z^2-0.25 & 0 & 0 \\ 0 & z^2+0.1z-0.12 & 0 \\ 0 & 0 & z^4+0.21z^2-0.01 \end{bmatrix}^{-1} .$$

The first column has poles : -0.5 $+0.5$
The second column has poles : $+0.3$ -0.4
The third column has poles : $+0.2$ -0.2 $+i.0.5$ $-i.0.5$.

Note that the modulus of the dominant poles of the first and third column is 0.5 and of the second column 0.4.
The 50 Markov parameters are corrupted with relative uniform random errors over the interval $[-10^{-8}, 10^{-8}]$. These corrupted Markov parameters are given as input to the algorithm of section 3. The singular values corresponding with the solution vectors are :

$$\begin{cases} \sigma_1 = 6.5 \cdot 10^{-9} \\ \sigma_2 = 5.0 \cdot 10^{-9} \\ \sigma_3 = 5.3 \cdot 10^{-9} \end{cases}$$

In the following figures the absolute values of the variables in function of an index are represented using a logarithmic scale. Figures 1 and 2 illustrate point 3 of the pevious section. Figure 1 contains the entries of D_1^{-1} that correspond to the first output. These are the odd numbered entries of D_1^{-1}. The other numbers displayed on figure 1 are the entries of the Markov parameters corresponding to the first output and the three different inputs, each multiplied with the corresponding element in $D_{2,0}$. Figure 2 gives similar results for the second output.
The similarity in the behaviour of the D_1^{-1} curve and the behaviour of the $H_k D_{2,0}$ curves is striking except for the entries of the second column and for the second entry of the third column. This indicates that the scaling for those entries is not very good.
The dominant pole for the second column is indeed different from the dominant pole for the first and third column. As a consequence, the numbers in the second column of the scaled matrix $D_1 H_{50,0} D_{2,0}$ will not be of magnitute 1 as they are in the first column, but they will

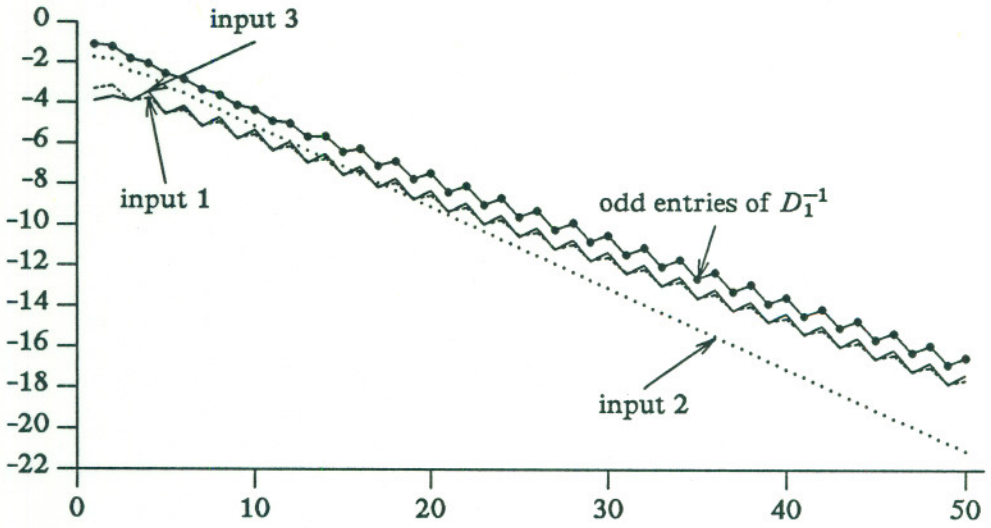


Figure 1: First output

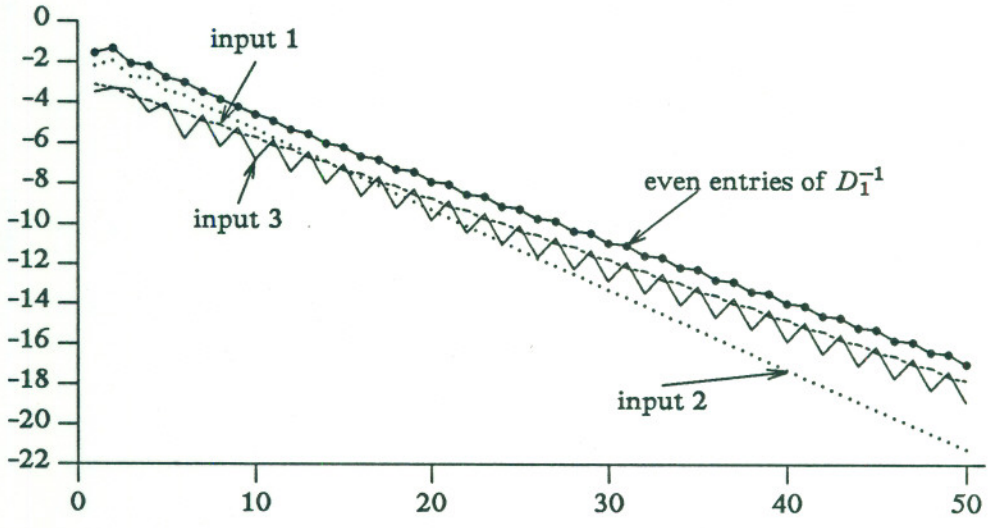


Figure 2: Second output

decrease from about 1 to about 10^{-4}. Therefore one can expect that the relative errors in the approximation of the Markov parameters will increase for the entries in the second column. (See figures 3 and 4)

Another type of bad scaling can be observed for the (2,3) entry. The scaling for the (2,3) entries of the odd Markov parameters is good but for the even Markov parameters, the corresponding entries in the scaled matrix $D_1 H_{50,0} D_{2,0}$ are about $10^{-1.5}$ instead of 1. Therefore we may expect a loss of about 1.5 digits on the (2,3) entries of the even Markov parameters. This is indeed observed in figures 3 and 4.

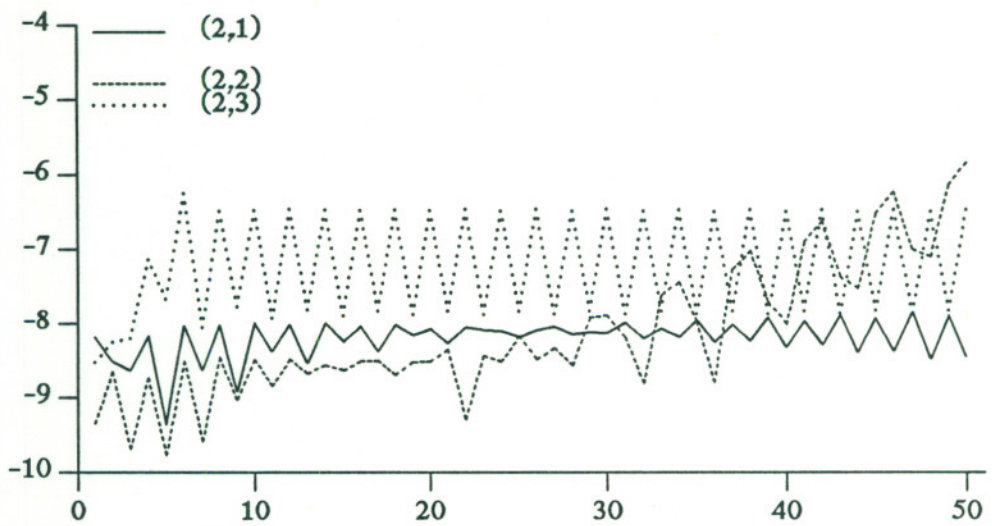


Figure 3: Relative errors for (2,1),(2,2),(2,3) entries.

Figure 3 gives the relative errors for the (2,1), (2,2) and (2,3) entries of the Markov parameters. The curves represented are characteristic for our example. The (2,1) error curve is more or less constant and tells us that the approximation is correct up to about 8 digits. The same behaviour is found for all well scaled entries: (1,1), (2,1) and (1,3). The (2,2) error curve indicates that the approximation is extremely good in the beginning (9 digits) but becomes worse at the end (only 6 digits correct). The (1,2) error curve has a similar behaviour. The error curve for the (2,3) entry is also as we predicted before. The odd Markov parameters have about 8 correct digits for that entry but the even Markov parameters are about 1.5 digits less accurate.

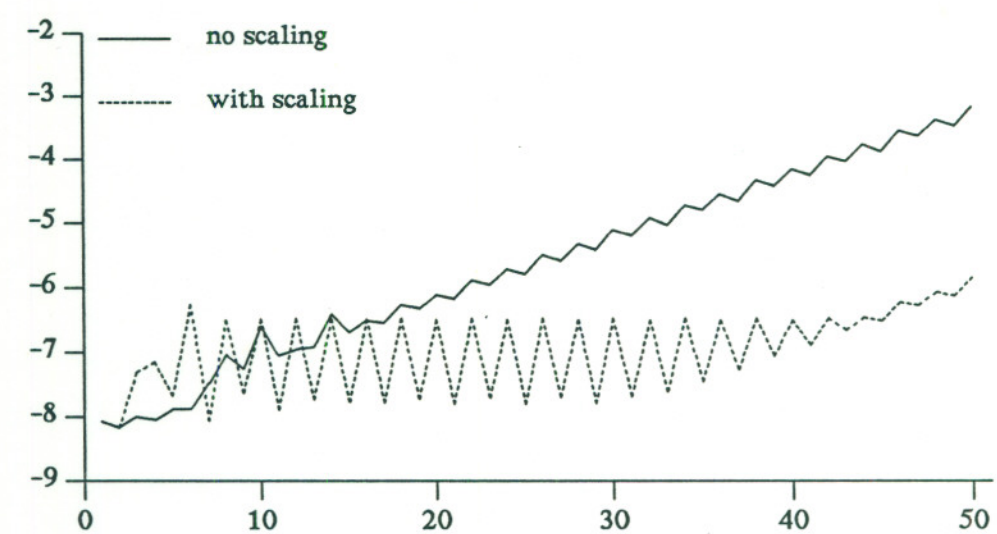


Figure 4: Maximum of relative errors

In figure 4 we plotted the maximum of the relative error of the 6 entries of the Markov

parameters. As you can see, the approximation is correct to somewhat less than 8 digits for the odd Markov parameters up to the 30th. The even Markov parameters have about 6.5 correct digits. This is essentially caused by the error in the (2,3) entry. From Markov parameter 30 on, the effect of the error growth in the entries of the second column becomes apparent for the odd Markov parameters and it grows even above the error in the (2,3) entry of the even Markov parameters from the 46th on.
Finally we also ran the algorithm without scaling. The singular values corresponding with the solution vectors are :

$$\begin{cases} \sigma_1 = 2.4 \cdot 10^{-9} \\ \sigma_2 = 2.9 \cdot 10^{-9} \\ \sigma_3 = 1.8 \cdot 10^{-9} \end{cases}$$

In figure 4 you can also find the maximum of the relative errors obtained in this case. As you can see, it grows well above the relative errors as obtained by the algorithm with scaling. Note however that the absolute error isn't constant either. It decreases but not as fast as the Markov parameters. The reason for this spontaneous attempt of good relative behavour of the error in the unscaled version is explained by the following phenomenon. The error, which is still given by (13)·(14) where we have to set all $D_{i,j} = I$, will decrease due to the decreasing coefficients of the factor (13). In the scaled version, the entries of the left singular vectors appearing in (13) all had a more or less constant magnitude while the factors $D_{1,k}^{-1}$ made the coefficients decrease like $H_k D_{2,0}$. For the unscaled case, the role of the $D_{1,k}^{-1}$ is taken over, be it somewhat less effective, by the left singular vectors themselves. Their entries will roughly decrease like the most slowly decreasing entries in the Markov parameters.

6. Conclusion :

An algorithm is proposed that computes a right matrix fraction description whose Markov parameters approximate the given Markov parameters with a certain relative precision. Theoretically, the denominator columns are found from the kernels of block Hankel matrices based on the given Markov parameters. These kernels can be computed by singular value decomposition. The nesting of these kernels, which can be derived from the solution table, can be made explicit in the singular value decomposition. However, when the data are corrupted by a relative noise of the order of ϵ, we replace the kernels of the block Hankel matrices by the kernels of matrices that are in a relative ϵ-neighbourhood. To compute such a kernel, we have to scale the block Hankel matrices properly. Therefore the first block Hankel matrix is multiplied from both sides with diagonal matrices such that the rows and columns of the product all have length approximately 1. The scaling of the subsequent block Hankels is only approximated to maintain the nesting of the kernels. Numerical experiments showed that this procedure gives the desired results.

REFERENCES

[1] L.S. de Jong, *Numerical aspects of realization algorithms in linear systems theory*, Ph.D. thesis, Technische Hogeschool Eindhoven, 1975.

[2] B.W. Dickinson, M. Morf and T. Kailath, *A minimal realization algorithm for matrix sequences*, IEEE Trans. Automat. control AC-19 (1974), 31-38.

[3] W.B. Gragg and A. Lindquist, *On the partial realization problem*, Linear Algebra Appl. 50 (1983), 277-319.

[4] M. Van Barel and A. Bultheel, *A minimal partial realization algorithm for MIMO systems*, Internal report TW79, K.U.leuven, Department of Computer Science, 1986.

[5] S. Van Huffel and J. Vandewalle, *The use of total linear least squares techniques for identification and parameter estimation*, IFAC Identification and System Parameter Estimation 1985, York, U.K., 1167-1172.