The Hankel realization method in PRIMAL

Gerlings, P.H.M.

Award date:
1988

VAKGROEP SYSTEEM-
EN REGELTECHNIEK

NR-1485P (1987-12-01) P.H.M. Gerlings

THE HANKEL REALIZATION METHOD IN PRIMAL

P.H.M. Gerlings

AFSTUDEERVERSLAG

Begeleider: Ir. R.J.P. van der Linden
Afstudeerhoogleraar: Prof.ir. O. Rademaker

SUMMARY

This report treats the implementation of the Hankel realization method in PRIMAL.(="Package for Real-time Interactive Modelling, Analysis, and Learning")

The Hankel realization method computes a state space model, based upon estimated, measured or known impulse responses. The number of parameters is reduced significantly, without much loss of performance. The performance is measured by the relative output simulation error.

This realization method is implemented in HANKEL in such a way that, having an impulse response and input-output data, the relative output simulation errors are computed for a range of dimensions. This is done at low extra computational cost, compared to the effort needed to compute a single realization.

Several variants of the Hankel realization method are tested on a broad spectrum of practical data.

The performance of the models achieved by estimating an impulse response and subsequent realization, compared to models generated by an Instrumental Variable method, proves to be good.

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF SYMBOLS

Symbol	formula	size	description
A	1.3	n x n	System matrix
a_i	2.1	1	Coefficient
AIC4	4.1	1	Dimension selection criterion
ARMS(k,l)	4.2	1	Absolute root mean square error of mkl over Mpar Markov parameters
B	1.3	n x p	Input-distribution matrix
C	1.3	q x n	Output matrix
D	1.3	q x p	Input-output matrix
E_p	2.8	nc x p	= $[\ I_p\ \vdots\ \emptyset\ ]^T$
E_q	2.7	q x nr	= $[\ I_q\ \vdots\ \emptyset\ ]$
G	2.6	n x nc	Controllability matrix
H	2.2	nr x nc	Hankel matrix
Happ	3.1	1	Relative approximation error of the Hankel matrix.
HB	3.7		Hankel matrix composed of vec(M_i)
H_k	2.16	nr x nc	Approximate Hankel matrix (rank k)
Hs	2.10	nr x nc	Shifted Hankel matrix
H:U<i>Y<j>		1	Response of output <j> due to a pulse on input <i> of a HANKEL realization.
I_k	2.10	k x k	k-th dimensional unity matrix
M_i	1.1	q x p	Markov parameter with time index i
mkl	1.1	1	Response of output k as a result of a pulse on input l
Mpar	4.1	1	Number of Markov parameters
MR_i	4.2	q x p	Markov parameter computed from a HANKEL realization (time-index i)
$\underline{m}_t$	3.4	q	= $[m1t\ \ldots\ mqt]^T$
M:U<i>Y<j>		1	Estimated response of output<j> due to a pulse on input<i> (MARKOV)
n	1.3	1	Dimension state space model
N		1	Number of data points used
nc	2.2	1	Number of columns Hankel matrix
nr	2.2	1	Number of rows Hankel matrix
p	1.1	1	Number of system inputs
q	1.1	1	Number of system outputs
Q	2.6	nr x n	Observability matrix
r	2.1	1	Order of a system
RRMS(k,l)	4.2	1	Relative root mean square error of mkl over Mpar Markov parameters
S/N			Signal power to the noise power
T	2.11	n x n	Transformation matrix
$\underline{u}(k)$	1.3	p	Input vector with time-index k
U	2.12	nr x nr	Left singular vectors
V	2.12	nc x nc	Right singular vectors
vec(M_i)	3.6	q.p x 1	Vector Markov parameter
$\underline{x}(k)$	1.3	n	State vector with time index k
$\underline{y}(k)$	1.3	q	Output vector with time index k

LIST OF SYMBOLS

Symbol	formula	size	description
δ_i	2.12	1	i-th singular value
σ_j	3.8	1	Relative simulation error for output j
Σ	2.12		Diagonal matrix containing the singular values
Σ_k	2.16	k x k	Diagonal matrix containing the first k singular values of Σ.
$(\ )^+$			Pseudo-inverse of a matrix
$(\ )^T$			Transposed of a vector or matrix
$(\ )_s$			Elements of the matrix are shifted over p columns to the left
$\emptyset$			Part of a matrix filled by zeros

1. INTRODUCTION.

First the PRIMAL project is described in paragraph 1.1. Then the available identification methods in PRIMAL are recited.(1.2) It appears that PRIMAL lacks a realization method.(1.3) The last paragraph (1.4) gives an overview of the structure of this report.

1.1. THE PRIMAL PROJECT.

In the subfaculty Systems and Control of the department of Physics of the University of Technology Eindhoven (TUE) a software package, called PRIMAL, is developed.
PRIMAL stands for "Package for Real-time Interactive Modelling, Analyses, and Learning". This software package is a tool for the experimenter on the often difficult trajectory from enigeering an experiment to implementing a controller. A few important steps on this trajectory are: imposing test signals on a process, measuring of in- and outputs, preconditioning of data, system identification and controller design. For all these steps tools are available in PRIMAL.

A list of some characteristics of the PRIMAL software package:

1. Interactive structure.
   The experimenter has always the possibility to intervene in a running experiment or active analysis. It is also possible to examine intermediate results.
2. Ability of imposing test signals on the process inputs.
3. Measuring in- and outputs and parallel processing of the already acquired data.
4. A set of tools ("applications") is available for the different tasks: signal analysis, identification, controller design, etc.
5. Multi-tasking structure.
   The applications, computer programs, are independent and may run in parallel. Many applications can use active datasets. This means one application is reading from the dataset, in order to use preliminary results or already available data, while another writes to it.
6. A user friendly interface.
7. Extensive graphical facilities.
   This enables the experimenter to attain a quick view on a huge amount of data.
8. The package (written in FORTRAN-77) is easily transportable to other computer systems.
9. New applications can be easily added to the package.

1.2. IDENTIFICATION METHODS IMPLEMENTED IN PRIMAL.

All identification methods in PRIMAL implemented have a firm theoretical or practical background. The available methods are presented in table 1.1.
They can be divided is into Single Input-Single Output (SISO) methods and methods that can handle systems with more than one input and/or more than one output. (MIMO)

Table 1.1 The identification methods available in PRIMAL.

Application	Model	description
EMM	SISO, ARMA	Extended Matrix Method
RPE	SISO, ARMA	Recursive Prediction Error Methods
GUIDORZI	MIMO, State Space	Method of Guidorzi
TRANSFER	MIMO, Transfer function	Estimating Transfer Function
MARKOV	MIMO, Impulse response	Estimating Markov par.
IVM	MIMO, ARMA	Instrumental Variable Method
ORDERTST	MIMO	Prediction Error Ordertest

These identification methods are compared, using simulation data and practical data by Berben [1987]. The relative output simulation error is used as performance criterion.

The Markov results were always among the best. On practical data the difference between Markov and the other identification methods was impressive. However Markov, estimating the impulse response of a system, is less parsimonious with its parameters than e.g. IVM.

1.3. THE HANKEL REALIZATION METHOD.

By using a realization method, e.g. the Hankel realization method, the number of parameters is reduced significantly. The objective of a realization method is to do this without much loss of model accuracy. The Hankel realization method derives a low dimensional state space model from the estimated, measured or known impulse response of a process. State space models are often used for simulation purposes and in modern control techniques.

The impulse response of a system consists of a sequence of Markov parameters. A Markov parameter is a (q x p)-matrix :

$$M_i = \begin{vmatrix} m11 & \ldots. & m1p \\ m21 & & \\ \vdots & & \vdots \\ \vdots & & \\ mq1 & \ldots. & mqp \end{vmatrix}_i \qquad (1.1)$$

with: q = number of outputs
p = number of inputs
i = time-index
mkl = response of output k as a result of a pulse on input l

Then the impulse response is presented by :

$$\{ M_k \} \quad k = 0,1,2, \ldots. \qquad (1.2)$$

From this sequence of Markov parameters a state space model is derived. A state space model has the following structure:

$$\begin{aligned} \underline{x}(k+1) &= A\,\underline{x}(k) + B\,\underline{u}(k) \\ \underline{y}(k) &= C\,\underline{x}(k) + D\,\underline{u}(k) \end{aligned} \qquad (1.3)$$

with: k = time-index
n = dimension state space model
$\underline{x}$ = state space vector (n)
$\underline{u}$ = input vector (p)
$\underline{y}$ = output vector (q)
A = system matrix (n x n)
B = input matrix (n x p)
C = output matrix (q x n)
D = input-output matrix (q x p)

The Hankel realization theory was published firstly by Ho and Kalman [1966]. They accomplished an exact realization of finite-dimensional linear systems, given exact data. An extension to noisy data was issued by Zeiger and McEwen [1974].

The computed state space models will be compared with the models estimated by the available identification methods. The relative output simulation error will be used as performance criterion.

The Hankel realization method is often presented as a model reduction method. In case of model reduction a high dimensional model of a system is approximated by a low dimensional model. This report however is restricted to the Hankel method in realization.

1.4. THE STRUCTURE OF THIS REPORT.

In this paragraph a global overview on the structure of this report is given.
The realization theory is described in chapter 2. In chapter 3 three dimension tests are described.
The implementation of the theory in computer programs is described in chapter 4.
In order to test the programs a number of datasets is used (chapter 5), on which first an impulse response is estimated. (chapter 6)
The results of testing the dimension tests are described in chapter 7.
The realization algorithm is tested (chapter 8), and the performance of the models achieved by estimating an impulse response and a subsequent realization are compared to the performance of models generated by an Instrumental Variable method. (chapter 10)

The final conclusions of this report are summarized in chapter 11.

2. REALIZATION THEORY.

In this chapter the theory of the Hankel realization method is presented. This theory was introduced firstly by Ho and Kalman in 1966. The described algorithm was published firstly by Zeiger and McEwen [1974]. Later on several modifications were published by, amongst others. Damen and Hajdasinski [1979;1982] and Kung [1979].
In the first paragraph a few theorems and assumptions are set out. Paragraph 2.2 gives an outline of the Hankel realization theory. The influence of noisy data is explained in paragraph 2.3. Four implementation variants are described in 2.4.

2.1. THEORETICAL PRELIMINARIES.

The systems under consideration must satisfy the following assumptions:

- linear
- time-invariant
- discrete-time
- stable
- causal

The impulse response of a MIMO system consists of a sequence of Markov parameters : $\{ M_k \}$ $k = 0,1,2, \ldots$

Theorem 1.
The sequence of Markov parameters $\{M_k\}$ for $k = 0,1,2..$ has a finite dimensional realization $\{A,B,C,D\}$ if and only if there exist an integer r and constants a_i such that:

$$M_{r+j} = \sum_{i=1}^{r} a_i M_{r+j-i} \qquad \text{for all } j > 0 \qquad (2.1)$$

The minimum value of r is called the realizability index.

In this report r is also called the order of the system.

The Hankel matrix is composed of Markov parameters and has a special structure:

$$H = \begin{vmatrix} M_1 & M_2 & M_3 & \ldots & M_j \\ M_2 & M_3 & \ldots & & \\ M_3 & & & & \\ \vdots & & & & \\ \vdots & & & & \\ M_i & & \ldots & & M_{i+j-1} \end{vmatrix} \qquad (2.2)$$

Theorem 2.

If the sequence of Markov parameters $\{M_k\}$ for k=0,1,.. has a finite dimensional realization {A,B,C,D}, with realizability index r, then the minimal dimension n of the state space model satisfies:

$$\text{rank } [\ H\ ] = n$$

and : (2.3)

$$n \leq r \,.\, \min(p,q)$$

provided that $i \geq r$ and $j \geq r$ (see (2.2))

In this report n will be called the dimension of the system.

Because of the linear dependence between the Markov parameters it is clear that any Hankel matrix, provided $\min(i,j) \geq r$, has the same dimension n.

If a state space model is known, it is very easy to compute its Markov parameters. The following two relations are derived directly from the state space model description.

$$\begin{aligned} M_0 &= D \\ M_k &= C\, A^{(k-1)}\, B \qquad k \geq 1 \end{aligned} \tag{2.4}$$

From the above equation it is very obvious that the D matrix equals M_0, the Markov parameter on t=0. Therefore not much attention is paid to the D matrix.

The already mentioned special structure of the Hankel matrix and its rank (theorem 2) are essential to the theoretical justification of the following algorithm.

2.2. THE HANKEL REALIZATION METHOD.

A finite-dimensional system, with dimension n, and exact data are assumed. The Hankel matrix is composed of Markov parameters, resulting in equation (2.2):

$$H = \begin{vmatrix} M_1 & M_2 & M_3 & \cdots & M_j \\ M_2 & M_3 & \cdots & & \\ M_3 & & & & \\ \vdots & & & & \\ \vdots & & & & \\ M_i & & \cdots\cdots & & M_{i+j-1} \end{vmatrix} \tag{2.2}$$

The dimensions of the Hankel matrix are :

block dimensions : (i x j)
total dimensions : (nr x nc)
with : nr = i.q
nc = j.p

Assume that the dimensions of the Hankel matrix are large enough. ($nr \geq n$ and $nc \geq n$)

Substituting relation (2.4) into the Hankel matrix results in:

$$H = \begin{vmatrix} CB & CAB & \dots & CA^{(j-1)}B \\ CAB & & & \\ \vdots & & & \\ \vdots & & & \\ CA^{(i-1)}B & \dots & & CA^{(i+j-2)}B \end{vmatrix} \qquad (2.5)$$

This matrix can be decomposed into two matrices:

$$H = \begin{vmatrix} C \\ CA \\ \vdots \\ \vdots \\ CA^{(i-1)} \end{vmatrix} \; | \; B \;\; AB \;\; \dots \;\; A^{(j-1)}B \; | \qquad (2.6)$$

$$H = Q \cdot G$$

So the Hankel matrix is decomposed in a controllability matrix (G, n x nc) and an observability matrix (Q, nr x n), both having rank n.
From the above decomposition it is possible to calculate the state space matrices.
As readily can be seen :

> C is the first block of Q

$$C = E_q \cdot Q \qquad (2.7)$$

> B is the first block of G

$$B = G \cdot E_p \qquad (2.8)$$

> A follows from the relation :

$$Hs = Q \cdot A \cdot G$$

so :

$$A = Q^{+} \cdot Hs \cdot G^{+} \qquad (2.9)$$

with :
- Q^{+} = Pseudo-inverse of the matrix Q
- G^{+} = Pseudo-inverse of the matrix G
- Hs = Hankel matrix shifted over one block column(≡ one column of Markov parameters) to the left.

$$Hs = \begin{vmatrix} M_2 & M_3 & M_4 & \dots & M_{j+1} \\ M_3 & M_4 & \dots & & \\ M_4 & & & & \\ \vdots & & & & \\ M_{i+1} & & \dots\dots & & M_{i+j} \end{vmatrix} \qquad (2.10)$$

$$E_p = \begin{vmatrix} I_p \\ \hline \emptyset \end{vmatrix} \quad \text{(nc x p) matrix}$$

$$E_q = | \; I_q \; : \; \emptyset \; | \quad \text{(q x nr) matrix}$$

I_k = k-th dimensional unity matrix

Given the foregoing reasoning it is clear that the next aim is to determine a decomposition of the Hankel matrix. Any decomposition of H into two matrices with full rank n will show the same structure as in equation (2.6) with some minimum realization $\{A^*, B^*, C^*, D^*\}$, because a state space model is a non-unique description of a system.[van den Hof, 1982]
An equivalence transformation is defined as a transformation of the state vector $\underline{x}$, with a non-singular matrix T, which doesn't alter the input-output behaviour of the model in any sense. The following equivalence transformation, for every nonsingular matrix T (n x n), exists:

$$\underline{x}^* = T\,\underline{x}$$

and consequently:

$$\begin{array}{lll} G^* = T \cdot G & & Q^* = Q \cdot T^{-1} \\ A^* = T \cdot A \cdot T^{-1} & & B^* = T \cdot B \\ C^* = C \cdot T^{-1} & & D^* = D \end{array} \qquad (2.11)$$

It is possible to use the singular value decomposition (SVD) of the Hankel matrix as a tool to achieve such a decomposition.

$$H = U \Sigma V^T \qquad (2.12)$$

with : U = Left singular vectors
Σ = Diag $(\delta_1, \ldots, \delta_n, 0, \ldots, 0)$
with $\delta_1 \geq \delta_2 \geq \ldots \delta_n > 0$
V = Right singular vectors

Because all elements of the diagonal matrix Σ are greater or equal to zero the following decomposition is possible:

$$\begin{aligned} H &= (\, U\, \Sigma^{\frac{1}{2}} \,)\, (\, \Sigma^{\frac{1}{2}}\, V^T \,) \\ &= \quad Q \quad \cdot \quad G \end{aligned} \qquad (2.13)$$

And analogously:

$$\begin{aligned} H^+ &= V\, \Sigma^+\, U^T \\ &= (\, V\, (\Sigma^+)^{\frac{1}{2}} \,)\, (\, (\Sigma^+)^{\frac{1}{2}}\, U^T) \\ &= \quad G^+ \quad \cdot \quad Q^+ \end{aligned} \qquad (2.14)$$

with : $\Sigma^+ = \text{diag}\,(\delta_1^{-1}, \ldots, \delta_n^{-1}, 0, \ldots, 0)$

A short recapitulation of the algorithm:

> starting-point:
the impulse response $\{M_k\}$ for $k = 0,1,2...$

> compose the Hankel matrix : H (see (2.2))

> Compute the singular value decomposition of the Hankel matrix:

$$H = U \Sigma V^T$$

> Construct the state space matrices :

$$\begin{aligned} A &= (\Sigma^+)^{\frac{1}{2}} U^T Hs V (\Sigma^+)^{\frac{1}{2}} \\ B &= \Sigma^{\frac{1}{2}} V^T E_p \\ C &= E_q U \Sigma^{\frac{1}{2}} \\ D &= M_0 \end{aligned} \qquad (2.15)$$

2.3. EXTENSIONS TO THE THEORY FOR NOISY MARKOV PARAMETERS.

If the system has a finite dimension and the Markov parameters are exact, then the foregoing algorithm will give an exact realization. This means that both models, Markov parameter model and state space model, will have the same behaviour under all circumstances.
However in practical situations the Markov parameters are often estimated from noisy data.

A severe implication of this fact is that the theoretical justification of the algorithm is no longer valid. This is caused by two facts:

1. The rank of the Hankel matrix doesn't equal the dimension of the state space model.
2. The shifted Hankel matrix will either lack the correct rank (should equal n) or the correct structure.(Hankel structure)

In case of exact data and a n-dimensional system only the first n singular values will be non-equal to zero. In the noisy case all singular values will be non-equal to zero.
The noise introduces a new problem : choosing an acceptable dimension for the realization. This problem will be examined in chapter 3.
In order to construct a state space model of dimension k, only the first k singular values are taken into account. (dimension k: k singular values) In other words Σ in (2.15) is substituted by Σ_k, defined as:

$$\Sigma_k = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_k, 0, \ldots, 0)$$

Then the following approximate Hankel matrix can be computed:

$$H_k = U \Sigma_k V^T \qquad (2.16)$$

This matrix H_k is the k-th dimensional least squares approximation of the original Hankel matrix. [Damen and Hajdasinski, 1979]
Because the Hankel matrix consists of Markov parameters, this is also a least squares estimation of the impulse response of the system. But not all Markov parameters appear with the same frequency in the Hankel matrix, due to its special structure. (see 2.2) Consequently not all Markov parameters are regarded as equally important in this approximation. The first and the last Markov parameter used, appear only once in the Hankel matrix. The weighting of the Markov parameters in the Hankel matrix (block square) is shown in the picture beneath.

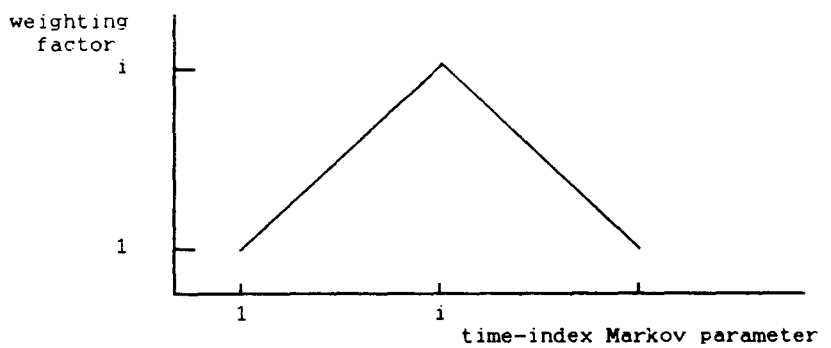


fig 2.1 Weighting of the Markov parameters in the least squares approximation of the block square Hankel matrix.

Because of the least squares approximation of the original Hankel matrix the realization algorithm may be sensitive to magnitude differences between the various entries in the Markov parameters. This is elucidated by the following example.

Suppose:> 2 input- 2 output system

then $$M_i = \begin{vmatrix} m11 & m12 \\ m21 & m22 \end{vmatrix}_i$$

> $abs(m11)_i \cong 10.abs(m21)_i$
> 10 % amplitude noise on the elements of the Markov parameters

In this case the impulse response from input 1 to output 2 (m21) will disappear in the noise of m11. This is caused by the (absolute) least squares approximation of the Hankel matrix. Therefore this realization algorithm won't approximate m21 very well.

So it is necessary that the input and output signals are properly scaled before estimating the impulse responses and using this realization algorithm. However in case of MIMO systems it isn't always possible to perform this scaling successfully, e.g. if the number of entries in a Markov parameter (p.q) exceeds the number of scaling possibilities (p+q).

2.4. IMPLEMENTED VARIANTS OF THE HANKEL REALIZATION METHOD

In the noisy case theoretical justification of the realization algorithm is lost because of the already mentioned incorrect rank of the Hankel matrix and the incorrect rank or the incorrect structure of the shifted Hankel matrix.
The shifted Hankel matrix is the instrument for the development of variants on the Hankel realization method. In all variants described here the matrices B,C and D are identical. The differences arise in the system matrix A, which is computed from the shifted Hankel matrix (Hs) (see 2.15).

There are two characteristic points of difference between the methods:

- Has the shifted Hankel matrix the correct rank or the correct Hankel structure ?
  It is possible to construct a shifted Hankel matrix with both the correct structure and the correct rank.[Damen and Hajdasinski, 1982] This can be done at the cost of a considerable computational effort.
- What is used to fill the last block column of the shifted Hankel matrix ?

Four variants have been tested. A description follows of the handling of the shifted Hankel matrix (Hs) by the four variants. We assume that more singular values than the desired state space dimension k are available.

a. Bosgra - van Zee [1979].

In this case the shifted Hankel matrix is filled with the measured or estimated impulse response. This implicates that an extra Markov parameter is needed. (M_{i+j})

$$Hs = \begin{vmatrix} M_2 & M_3 & M_4 & \cdots & M_{j+1} \\ M_3 & M_4 & \cdots & & \\ M_4 & & & & \\ \vdots & & & & \\ \vdots & & & & \\ M_{i+1} & & \cdots\cdots & & M_{i+j} \end{vmatrix} \qquad (2.17)$$

This shifted Hankel matrix has the Hankel structure but

lacks the correct rank. In the noise free case with a finite dimensional system this variant will give an exact realization.

b. Damen - Hajdasinski [1979].

In case of Damen-Hajdasinski the shifted Hankel matrix is filled by the k-th dimensional approximation of the Hankel matrix. So :

$$H_k = U \Sigma_k V^T \qquad (2.16)$$

$$\text{with } \Sigma_k = \text{diag}(\delta_1,\ldots,\delta_k,0,\ldots,0)$$

Hs is achieved by shifting the matrix H_k over one block column to the left.
This leaves us with the problem of filling the last block column of the shifted Hankel matrix. This is done by copying the last but one block column and shifting it one block row upwards.
The last element in this matrix is filled with the estimated, known or measured Markov parameter M_{i+j}.
So:

$$Hs = \begin{vmatrix} 11\mu_2 & 12\mu_3 & 13\mu_4 & \cdots\cdots & 1j\mu_j & 2j\mu_{j+1} \\ 21\mu_3 & 22\mu_4 & & \cdots\cdots & 2j\mu_{j+1} & \vdots \\ 31\mu_4 & & & & \vdots & \vdots \\ \vdots & & & & \vdots & \\ \vdots & & & & & ij\mu_{i+j-1} \\ i1\mu_{i+1} & & \cdots\cdots & & ij\mu_{i+j-1} & M_{i+j} \end{vmatrix} \qquad (2.18)$$

with : $kl\mu_s$ = the kl-th block element of Hs containing an approximation of the s-th Markov parameter.

This shifted Hankel matrix has in practical cases, with noise, rank k+1. So Hs lacks both the correct rank and the correct structure. However in the noise-free case and a finite-dimensional system this method gives an exact realization.

c. Structural.

The reconstructed Hankel matrix of the previous method does not have the Hankel structure. This structure can be restored by averaging along contra-diagonal blocks. This appears to be the least squares approximation of the reconstructed Hankel matrix H_k [Damen and Hajdasinski, 1982]. A more practical view is: on different entries of H_k, normally different, approximations ($kl\mu_s$) of the same Markov parameter are present. By averaging the different approximations the Hankel structure can be restored. As element right below M_{i+j} is used.(2.19)

$$Hs = \begin{vmatrix} \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_j & \mu_{j+1} \\ \mu_3 & \mu_4 & & \cdots & \mu_{j+1} & \\ \mu_4 & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \mu_{i+j-1} \\ \mu_{i+1} & & & \cdots & \mu_{i+j-1} & M_{i+j} \end{vmatrix} \qquad (2.19)$$

with : μ_s = approximation of the s-th Markov parameter.

This shifted Hankel matrix lacks the correct rank but has the correct structure. In the finite-dimensional noise free case this will also yield an exact realization.

d. Kung [1979].

For the computation of the system matrix A Kung uses a shifted version of the controllability matrix (G) or the observability matrix (Q). This however is equivalent to the following algorithm using a shifted Hankel matrix. (for a proof see Appendix 1) The shifted Hankel matrix Hs is in this case a shifted version of the k-th dimensional approximation of the original Hankel matrix.(H_k see 2.16) The last block column however is filled with zeros.
So:

$$Hs = \begin{vmatrix} 11\mu_2 & 12\mu_3 & 13\mu_4 & \cdots & 1j\mu_j & \emptyset \\ 21\mu_3 & 22\mu_4 & & \cdots & 2j\mu_{j+1} & \emptyset \\ 31\mu_4 & & & & \vdots & \\ \vdots & & & & \vdots & \vdots \\ \vdots & & & & & \\ i1\mu_{i+1} & & & \cdots & ij\mu_{i+j-1} & \emptyset \end{vmatrix} \qquad (2.20)$$

with : $kl\mu_s$ = the kl-th block element of Hs containing an approximation of the s-th Markov parameter.

This matrix has the correct rank but lacks the correct structure.
If we look at this matrix as an approximation of the impulse response, it is clear that adding a block column of zeros can be seen as an approximation of the Markov parameters $\{M_k\}$ $k=j+1,\ldots,i+j$ by zero at these entries of the shifted Hankel matrix. Consequently it is obvious that this will cause extra errors if these Markov parameter differ considerably from zero.
In contrast with the three previous methods the Kung algorithm won't always produce an exact realization in the finite-dimensional noise free case, due to the last block column with zeros in the shifted Hankel matrix.

If we consider the four realization variants, it strikes that the method of Kung won't always produced an exact realization in the noise free case. But this realization variant will always, also in the noisy case, give a stable state space model. This is proven in appendix 2. For the other realization methods this couldn't be proven.

3. ORDER/DIMENSION TESTS.

As already described in chapter 2 it is necessary to select a dimension for the state space model before a realization is performed. The dimension tests mentioned in literature, in the context of the Hankel realization algorithm, are often based on the singular values of the Hankel matrix. Some of these tests are described in paragraph 3.1.
An alternative method is proposed by Backx [1987]. This method is described in paragraph 3.2.
By means of an order or dimension test the user should be able to select a satisfying order or dimension. A dimension will be called satisfactory if a higher dimension doesn't considerably improve the approximation of the system under consideration. The relative output simulation error, defined in paragraph 3.3, will serve as a measure for this approximation.

3.1. SINGULAR VALUES OF THE HANKEL MATRIX.

The Hankel matrix is composed of the Markov parameters as in equation (2.2). According to theorem 2 in chapter 2 for the noise free case with a finite-dimensional system the following equation applies:

$$\text{rank } [\, H \,] = n \qquad (2.3)$$

provided that the dimensions of the Hankel matrix are large enough.

After the singular value decomposition only the first n singular values are unequal to zero. This is illustrated in figure 3.1. If the data contains noise, all singular values will be unequal to zero. Assuming relatively low level white noise on the Markov parameters results in the following picture.

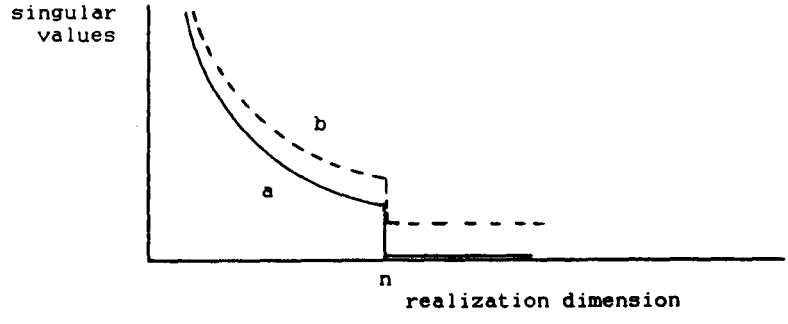


fig 3.1. The singular values of the Hankel matrix as a function of the dimension.
a. exact data, finite dimension: n.
b. noisy case, finite dimension: n.

The n-th singular value will be much larger than the (n+1)-th singular value. If the noise level is higher, the gap between the n-th singular value and the (n+1)-th singular value will disappear.
Another method of indicating the dimension in a figure is plotting the (i+1)-th singular value divided by its predecessor or in other words :(δ_{i+1} / δ_i) as function of the dimension. This is done in figure 3.2 for the finite-dimensional noisy case.

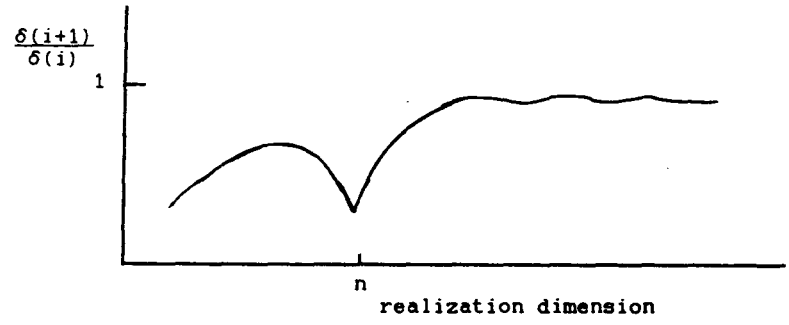


fig 3.2. The singular values of the Hankel matrix divided by their predecessor. (δ_{i+1} / δ_i) in the finite dimensional noisy case as function of the dimension.

This figure shows a dip for the true dimension n of the system. For higher dimensions the quotient will approach one.

A second selection method considers the relative error of the least squares approximation (Happ) of the original Hankel matrix (euclidian norm).

$$Happ = \frac{\| H - H_k \|}{\| H \|} = \text{sqrt}\left(\sum_{i=n+1}^{s} \delta_i^2 \Big/ \sum_{i=1}^{s} \delta_i^2 \right) \qquad (3.1)$$

with : $s = \min(nr, nc)$

If this relative error is beneath a certain value, the model is accepted. This certain value must be based on the noise level of the Markov parameters. Because not all Markov parameters are equally weighted (see figure 2.1), it is difficult to determine this value. If Happ is plotted versus the dimension a similar path as in figure 3.1 for the singular values is expected.

3.2. SINGULAR VALUES OF A MATRIX CONSISTING OF VECTOR MARKOV PARAMETERS.

The alternative method proposed by Backx [1987] selects a realizability index, which is defined by theorem 1 from chapter 2.

$$M_{r+j} = \sum_{i=1}^{r} a_i M_{r+j-i} \qquad \text{for all } j > 0 \qquad (2.1)$$

The minimum value of r is called the realizability index. If this r is known an upper limit for the dimension n is known, because of theorem 2 of chapter 2:

$$n \leq r.\min(p,q) \qquad (2.3)$$

Relation (2.1) may be applied to every individual element of the Markov parameters. A Markov parameter is defined in (1.1) as:

$$M_i = \begin{vmatrix} m11 & \ldots. & m1p \\ \vdots & & \vdots \\ mq1 & \ldots. & mqp \end{vmatrix}_i \qquad (1.1)$$

Applying equation (2.1) to the individual elements leads to:

$$\begin{aligned} m11_{r1+j} &= \sum_{i=1}^{r1} a_i \, m11_{r1+j-i} \quad \text{for all } j > 0 \\ &\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ mqp_{rs+j} &= \sum_{i=1}^{rs} a_i \, mqp_{rs+j-i} \quad \text{for all } j > 0 \end{aligned} \qquad (3.3)$$

The individual realizability index rt indicates the complexity of the transfer function from one input to one output. Therefore their values may differ for t = [1..s]. The realizability index r of the system will equal the highest individual one.

Computing r can be done the following way.
Define :

$$\underline{m}_t := \begin{vmatrix} m1t \\ \vdots \\ mqt \end{vmatrix} \qquad (3.4)$$

which is a column of the Markov parameter. Then a Markov parameter can be written as:

$$M_i = (\underline{m}1 \;\; \underline{m}2 \;\; \ldots. \;\; \underline{m}p)_i$$

Define a vector Markov parameter as:

$$\text{vec}(M_i) := \left| \begin{array}{c} \underline{m}1 \\ \underline{m}2 \\ : \\ \underline{m}p \end{array} \right|_i \qquad (3.5)$$

The matrix HB. having a Hankel structure can be constructed:

$$HB = \left| \begin{array}{cccc} \text{vec}(M_1) & \text{vec}(M_2) & .... & \text{vec}(M_j) \\ \text{vec}(M_2) & \text{vec}(M_3) & & \\ \text{vec}(M_3) & & & : \\ : & & & \\ \text{vec}(M_i) & ..... & & \text{vec}(M_{i+j-1}) \end{array} \right| \qquad (3.6)$$

Substitution of the foregoing definitions, (3.4) and (3.5) into this matrix HB (3.6) gives:

$$HB = \left| \begin{array}{ccccc} (k=1) & (k=2) & (k=3) & ..... & (k=j) \\ m11 & m11 & & & \\ : & : & & & \\ mq1 & mq1 & & & \\ & & & & \\ : & : & & & \\ & & & & \\ mqp & mqp & & & \\ \hline (k=2) & (k=3) & & & \\ m11 & & & & \\ : & & & & : \\ mq1 & & & & : \\ & & & & : \\ : & & & & \\ & & & & \\ mqp & & & & \\ \hline (k=3) & & & & \\ : & & & & \\ & & & & \\ : & & & & \\ \hline (k=i) & & ..... & & k=i+j-1 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ \hline \end{array} \right| \qquad (3.7)$$

If the singular value decomposition is applied to this matrix then, in the noise free case with finite order, only the first r singular values will be non-zero. The number of singular values indicates the number of independent columns. As can be seen the number of independent columns is determined by the maximum of rt, which is the realizability index (or order) of the system. This method is illustrated in figure 3.3. Line a applies for the noise free case and line b for the noisy case, both having a finite order.

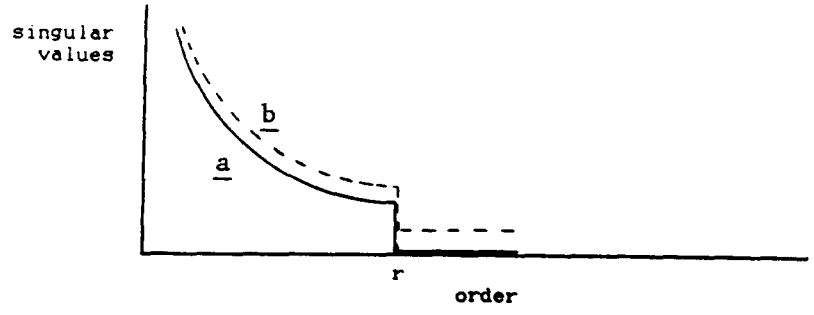


fig 3.3 The singular values of the HB matrix (3.7) as a function of the order.
a. noise free case, finite order
b. noisy case, finite order

In this case also the singular values divided by their predecessor can be presented, analogously to figure 3.2. The dip will now point at the order of the system.

A comparison of this method with the singular value decomposition of the Hankel matrix can now be made.

A first remark is that there are many situations in which they will give the same sequence of singular values. Normally both matrices, Hankel (2.2) and HB (3.7), are constructed block square, which means that the number of block rows equals the number of block columns (or i=j). Then the two methods only differ for MIMO systems. The second method is expected to give more explicit results, because less independent directions are searched in a higher dimensional space.

A second remark is that the first method yields just what we want: a dimension for the state space model. The other method yields an order.

3.3. RELATIVE OUTPUT SIMULATION ERRORS.

None of the two previous tests provides a direct link between the simulation performance of a model and its dimension. However this is what an experimenter often wants to know. Therefore a third dimension test is implemented. This test delivers the relative output simulation error as function of the realization dimension. This method seems very crude, because realizations are computed for several dimensions and for each realization a simulation is performed over a data range [k1,k2]. But this can be implemented rather nicely as will be shown in chapter 4. It is however clear that this method will demand more computational overhead than the foregoing methods.

The relative output simulation error for output i is defined as :

$$\sigma_i = \sum_{k=k1}^{k2} (e_i(k) - \tilde{e}_i)^2 \; / \; \sum_{k=k1}^{k2} (y_i(k) - \tilde{y}_i)^2 \qquad (3.8)$$

with : [k1,k2] data range
$y_i(k)$ = data sample k for output i
$e_i(k)$ = difference between model and process for output i at sample k
$\tilde{}$ = averaged value of the quantity over the data range [k1,k2]

The relative output errors are presented in a graph as function of the realization dimension. This graph relates directly the dimension of a realization to its performance. For a finite-dimensional case with exact data this will give line a of figure 3.4. If dimension n is reached the realization will be exact. and consequently the relative output error will be zero.

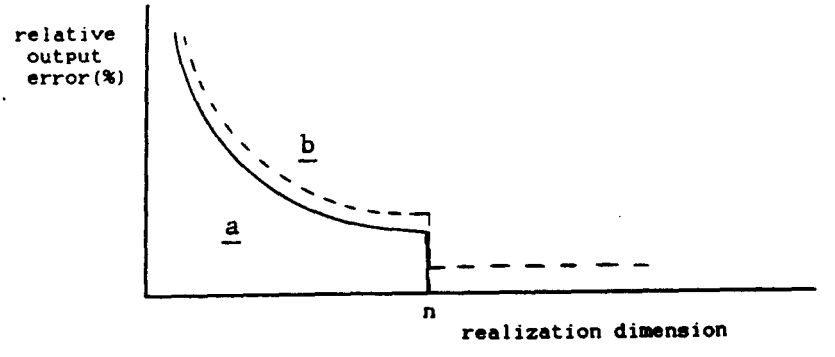


fig 3.4 Relative output simulation errors as function of the dimension of the realized model.
a. in the finite dimensional noise free case.
b. in the finite dimensional noisy case.

If the data contain noise influences the relative output simulation error will stabilize at a certain level. This is illustrated by line b in figure 3.4.

4. IMPLEMENTATION.

In this chapter the implementation of the foregoing theory in computer programs (PRIMAL applications) is described. The dimension test using the singular values of the Hankel matrix is implemented as HANKDIM. The order test described in paragraph 3.2 is implemented as HANKDIM2. In HANKEL all four variants of the Hankel realization method. described in chapter 2, are implemented. Also the dimension test based on the relative output simulation errors is included in HANKEL.

The implementation of HANKDIM and HANKDIM2 is discussed in paragraph 4.1.
The implementation of the realization algorithm (HANKEL) is described in paragraph 4.2.
For each program the required inputs, the user choices, and the outputs are described.
In case of the realization algorithm (HANKEL) a schematic view of the program structure is presented.

4.1. IMPLEMENTATION OF HANKDIM AND HANKDIM2.

Both HANKDIM and HANKDIM2 are implemented straightforwardly. Given the impulse response of the system the singular values of the Hankel matrix (HANKDIM, equation (2.2)) or the HB-matrix (HANKDIM2, equation (3.7)) are computed.
On output a text dataset and a dataset containing the singular values are created. An example of a text dataset is given in appendix 3 for HANKDIM, and in appendix 4 for HANKDIM2.

The application HANKDIM/HANKDIM2 has the following parameters which can be adjusted by the user:

1. The number of Markov parameters taken into account.
2. The shape of the Hankel matrix (in case of HANKDIM2 the HB-matrix). Normally a block square matrix is used. This means that the number of block rows equals the number of block columns.
3. The (sub)system examined.
   Having a 3 input-3 output system, it is possible to examine a subsystem consisting of the first two inputs and the last two outputs.
   This opens the possibility of examining one particular transfer from one input to one output.

4.2. IMPLEMENTATION OF HANKEL.

The application HANKEL can be used in two different ways. If only the impulse response of the system (Markov parameters) is available a straightforward realization (see chapter 2) is computed.
If in addition also a dataset, containing input-output data of the system under survey, is available then the most efficient model within a dimension range is selected.

Now a more detailed overview of the application HANKEL will be given. A schematic view of the program structure is presented in figure 4.1. The application will be discussed going through this scheme from top to bottom.

The parameter input of the application is represented by the first two blocks in figure 4.1.
The parameters which may or must be specified by the user:

1. In case of one input dataset (containing the Markov parameters):
   - The dimension of the realization.
   - The number of Markov parameters used.
   - One of the four realization variants described in paragraph 2.4.
   - The dimensions of the Hankel matrix. Normally a block square Hankel matrix is used.
   - The examined (sub)system. (see paragraph 4.1)

2. If a dataset with input-output data of the system is present, some additional parameters must be specified:
   - Instead of one realization dimension a range of realization dimensions. With realizations of these dimensions a simulation is performed.
   - A data range to perform the simulation on.
   - The signals serving as model inputs and the signals to be used as reference for the model outputs.

The following step in the algorithm is the singular value decomposition of the Hankel matrix. (see fig 4.1)

Next, the algorithm splits up:
1. If only Markov parameters are available (one input dataset) a realization is computed with the specified dimension and written to an output dataset.
2. If also a dataset, containing input-output signals of the system under survey, is available over a range of dimensions the following is done:
   - a model is computed.
   - for each stable model the relative output simulation error is computed (equation (3.8)). The simulation is performed using the specified input signals as model inputs and comparing the model outputs with the reference signals. These

specified signals may be selected from the input-output dataset used for estimation of the impulse response (best fit), or a cross validation dataset.

If a realization is unstable the program continues with the next dimension. (see fig 4.1) The resulting simulation error as a function of the dimension may be examined graphically.

The most efficient dimension will be selected using an extension of the modified AIC criterion [Freeman, 1985]. The criterion used weights the achieved accuracy (σ_i^2 summed over the outputs) and the number of data used for the realization (p.q.Mpar) against the size of the state space model (n(n+p+q)+p.q) :

$$AIC4 = p.q.Mpar.\ln\{ \sum_{i=1}^{q} \sigma_i^2 \} + 4.[n.(n+p+q) + p.q] \qquad (4.1)$$

The realization with the selected dimension is written to an output dataset.

As can be seen from figure 4.1 it requires only one singular value decomposition to examine realizations over a range of dimensions. This has enormous computational advantages, because the singular value decomposition of a large Hankel matrix is very time consuming.

Now the paths of the algorithm meet.(figure 4.1)

The eigenvalues of the realization are computed. The impulse response of the realization, computed by using equation(2.4), is compared to the initial impulse response. Both impulse responses can be examined graphically (together). An absolute (ARMS) and a relative root mean square error (RRMS) are defined for each entry in the Markov parameters as:

$$ARMS(i,j) = \sqrt{(1/Mpar) \cdot \sum_{k=0}^{Mpar} \{ MR_k(i,j) - M_k(i,j) \}^2} \qquad (4.2)$$

$$RRMS(i,j) = SQRT(Mpar).ARMS(i,j) \Big/ \sqrt{\sum_{k=0}^{Mpar} \{ M_k(i,j) \}^2}$$

with : i denotes the i-th output
j denotes the j-th input
MR_k k-th Markov parameter computed from the realization

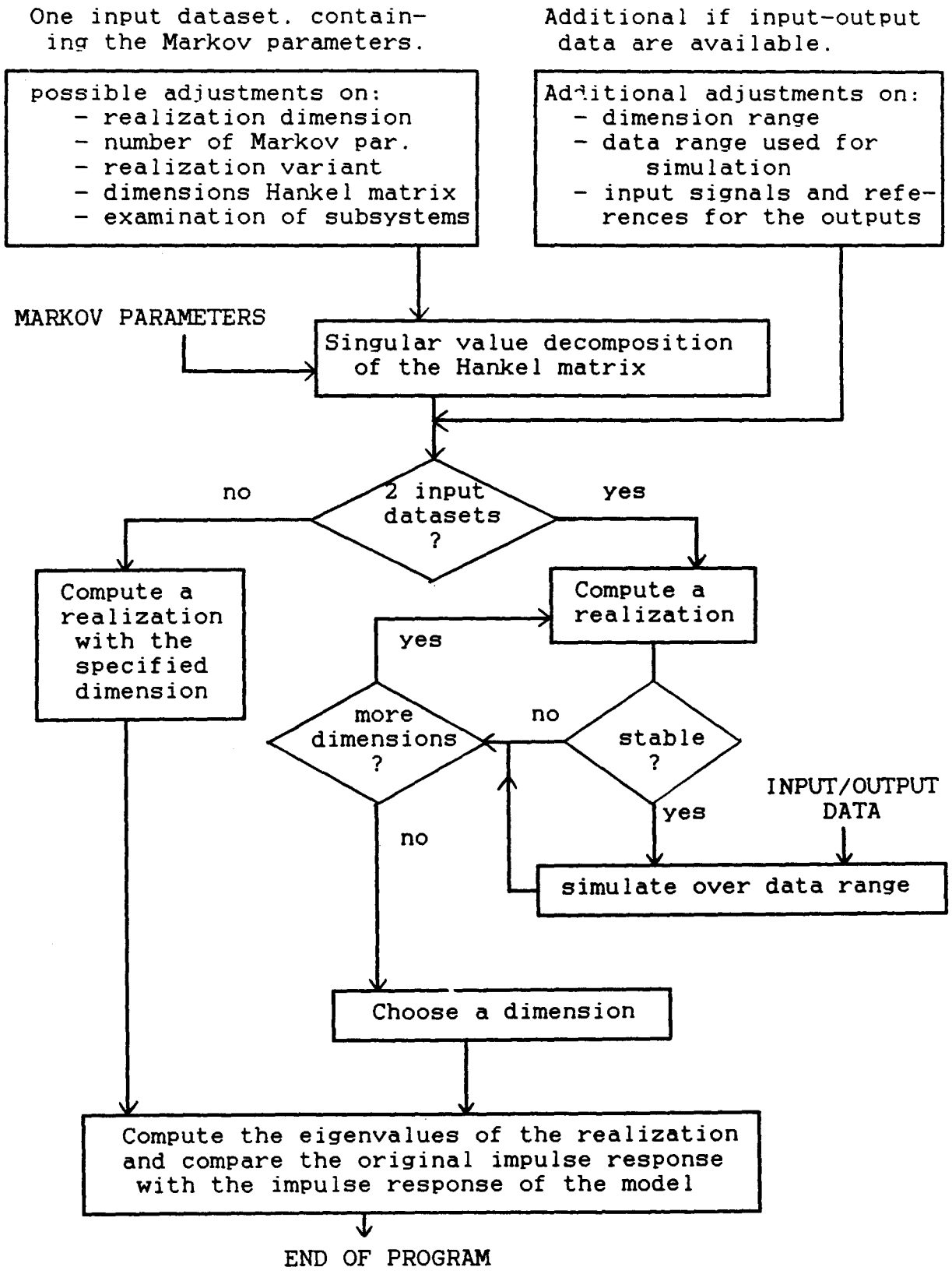


Fig. 4.1 Schematic overview of the structure of the HANKEL application.

In literature the Hankel realization method is often propagated as a model reduction method. [amongst others Kung.1979] The HANKEL application is implemented in such a way that it is possible to use a state space model as input instead of an impulse response. In this case an user specified number of Markov parameters of the initial state space model are computed using equation (2.4).

5. THE DATASETS USED FOR TESTING.

In this chapter a short characterization of the datasets used for testing the order/dimension tests and the Hankel realization algorithm is given.

The data can be divided into two sets (between parentheses the name of the dataset is mentioned) :

1. simulated data (see paragraph 5.1):
   - Aström process (AST1, AST2)
   - Carriere data (CARRIERE)
2. practical data (see paragraph 5.2):
   - Climate chamber data (CLIMCH)
   - Picos data (PICOS)
   - Feeder data (FEED1, FEED2)

Simulated data are used for comparison with the exact model. Since PRIMAL is designed for practical data, the applications are tested on practical data.

The main characteristics of the datasets are shown table 5.1. If a quality is unknown for a dataset a dash is used.

Table 5.1 An overview of the main characteristics of the used datasets.

Dataset	origin	p	q	n	N	noise*	S/N#
AST1	simulation	1	1	2	1000	white	10 dB
AST2	simulation	1	1	2	1000	coloured	10 dB
CARRIERE	simulation	3	2	5	1000	coloured	20 dB
CLIMCH	experiments	1	1	-	234	-	-
PICOS	experiments	2	2	-	2041	-	-
FEED1	experiments	3	1	-	912	-	-
FEED2	experiments	3	2	-	912	-	-

* Additive output noise
\# Signal to noise ratio

5.1 SIMULATED DATASETS.

a. Aström process (AST1, AST2)

This is a series of SISO-simulated data. The Aström process can be represented by the following state space model:

$$\underline{x}(k+1) = \begin{vmatrix} 1.5 & -0.7 \\ 1 & 0 \end{vmatrix} \underline{x}(k) + \begin{vmatrix} 1 \\ 0 \end{vmatrix} u(k)$$

$$y(k) = \begin{vmatrix} 1 & 0.5 \end{vmatrix} \underline{x}(k) \qquad (5.1)$$

This system (n=2) has two complex poles :0.75 ± i 0.371. As input signal white Gaussian noise (zero mean) is used.

From this state space model two datasets are extracted. The first (AST1) has additive white Gaussian noise (zero mean) on the outputs. The second dataset (AST2) has coloured output noise. [Berben, 1987] In both cases the signal to noise ratio is 10 dB.

b. Carriere data (CARRIERE)

This MIMO test system is developed by Carrière [1984]. It is represented by the following state space model :

$$\underline{x}(k+1) = \begin{vmatrix} 0.08 & & \emptyset & & \\ & 0.9 & & & \\ & & 0.1 & & \\ & \emptyset & & 0.92 & \\ & & & & 0.95 \end{vmatrix} \underline{x}(k) + \begin{vmatrix} 1.0 & 0.0 & -1.0 \\ 1.0 & 1.7 & 0.0 \\ 0.0 & 1.0 & -1.5 \\ 7.0 & 0.0 & 8.0 \\ 0.0 & 7.0 & 5.0 \end{vmatrix} \underline{u}(k) \qquad (5.2a)$$

$$\underline{y}(k) = \begin{vmatrix} 4.0 & 3.0 & 0.0 & 0.0 & 0.0 \\ 9.0 & 1.0 & 8.0 & -0.5 & 0.5 \end{vmatrix} \underline{x}(k) + \begin{vmatrix} 0.1 & 0.3 & 0.5 \\ 0.2 & 0.7 & 0.4 \end{vmatrix} \underline{u}(k) \qquad (5.2b)$$

Some characteristics of this system are:
- dimension 5
- order 3
- 3 inputs and 2 outputs
- 3 'slow' eigenvalues and 2 'fast' eigenvalues, both the 'slow' and the 'fast' eigenvalues are close to each other
- the variances of the states are approximately of the same magnitude.

As input signal white gaussian noise (zero mean) is used. The data sequence has additive coloured noise on the outputs.[Carrière,1984] The signal to noise ratio is 20 dB.

5.2 PRACTICAL DATASETS.

a. Climate chamber data (CLIMCH)

This data is recorded at an experimental set up by our subfaculty. In the set up a climate chamber [Driessen,1987] is heated by means of an under-floor heating. The dynamic relation of interest is the relation between the inlet water temperature of the under-floor heating (TWI) and the temperature in the chamber (TR). During the experiments the temperature outside the chamber is kept constant. Figure 5.1 provides a schematic view of the climate chamber.

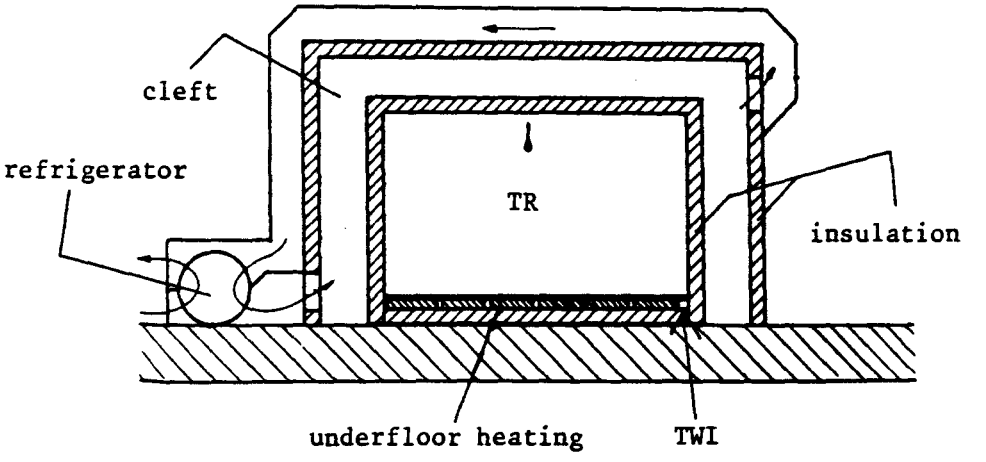


fig. 5.1 A schematic view of the climate chamber.

b. Picos data (PICOS)

This data is recorded at a glass production plant of Philips (Winschoten) where quartz tubes are manufactured. The shape of the tube is determined by the following two important parameters:

- average tube diameter (D)
- average tube wall thickness (WD)

For controlling of these two conditions the drawing speed of the glass (V) and the pressure (P) are used. [Backx, 1987] A schematic overview is given in fig 5.2.

Some characteristics of the data are:

- 2 inputs - 2 outputs system
- existence of non-linearities, so the model will only be valid near some working point.

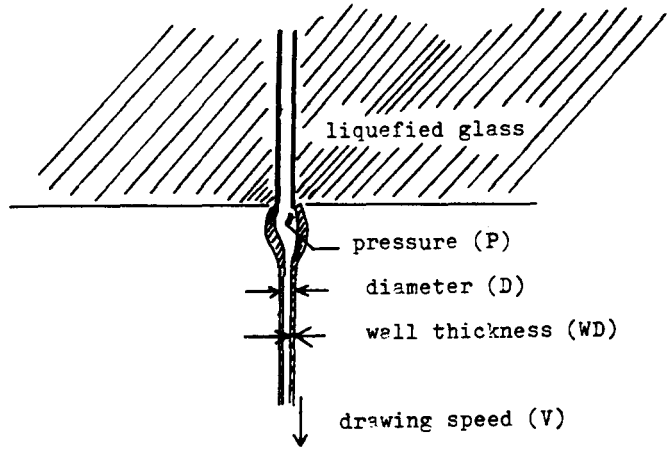


fig 5.2. A schematic view of the tube glass production process.

c. Feeder data (FEED1, FEED2)

This data sequence is recorded during a pilot project with PRIMAL at a glass-feeder (Philips, Eindhoven). A feeder is the final part of a process that is used for producing glass. It is designed to realize a homogeneous temperature distribution of the glass at a specified level. [Backx, 1987] In figure 5.3 a schematic view of the glass-feeder is given.

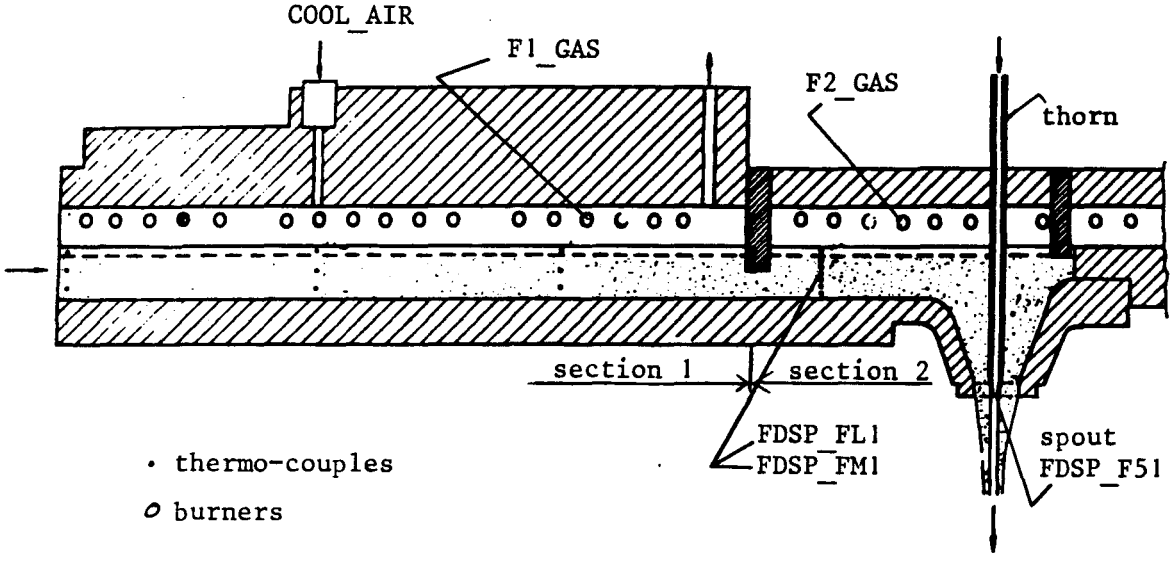


fig 5.3 A schematic view of a glass-feeder.

Two systems of the feeder are examined. In both systems the gas flow of the burners (F1_GAS of section 1 and F2_GAS of section 2) and the flow of the cooling air in section 1 (COOL_AIR) are used as the three inputs.
In the first system (FEED1) the glass temperature in the centre of the spout (FDSP_F51) is used as output.
The second system (FEED2) has two temperatures in section 2 as outputs. (FDSP_FM1 and FDSP_FL1)

6. ESTIMATED IMPULSE RESPONSES ON THE DATASETS.

The first step to a state space model computed by a realization method is estimating the impulse response.

The PRIMAL application MARKOV estimates, by minimizing the one-step-ahead prediction error, the parameters in the following model structure [van der Linden, 1987]:

$$\underline{y}(k) = M_0\,\underline{u}(k) + M_1\,\underline{u}(k-1) + \ldots + M_{Mpar}\,\underline{u}(k-Mpar) \qquad (6.1)$$

The impulse responses are shown in this chapter. Table 6.1 connects the datasets with the corresponding figures. The number of data points used for the Markov parameter estimation is indicated by N.
Further the relative output simulation errors, defined by equation (3.8), of the impulse response models are computed. These simulations are performed in best fit unless indicated otherwise.

Table 6.1 The relative output simulation errors (σ_i) of the by MARKOV estimated impulse responses on the test datasets.

dataset	p	q	Mpar	N	simulation range	σ_i (%) *out 1	out 2	figure
AST1	1	1	40	1000	[100,1000]	8.7	-	fig 6.1a
AST2	1	1	40	1000	[100,1000]	8.5	-	fig 6.1b
CARRIERE	3	2	40	1000	[100,1000]	0.9	1.3	fig 6.2
CLIMCH	1	1	60	234	[50,234]	1.2	-	fig 6.5
cross#					[50,218]	1.4	-	
PICOS	2	2	99	2041	[100,2041]	9.2	4.4	fig 6.3
cross					[100,2041]	13.8	4.8	
FEED1	3	1	65	912	[92,912]	1.4	-	fig 6.4
FEED2	3	2	65	912	[92,912]	1.3	6.1	fig 6.6

\# cross validation of the model estimated above
* out = output

Now follow the figures containing the impulse responses. Above the figures M: U<i>_Y<j> is placed. This indicates the by MARKOV estimated response of output Y<j> due to a pulse on input U<i>.

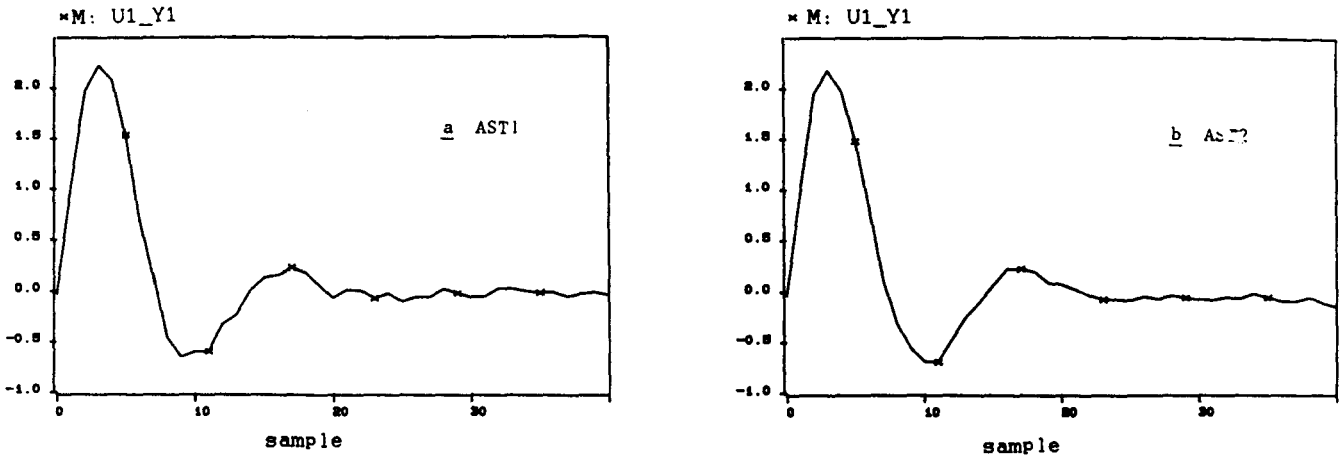


fig 6.1 The estimated impulse responses for the Aström process.
a AST1 (Mpar = 40)
b AST2 (Mpar = 40)

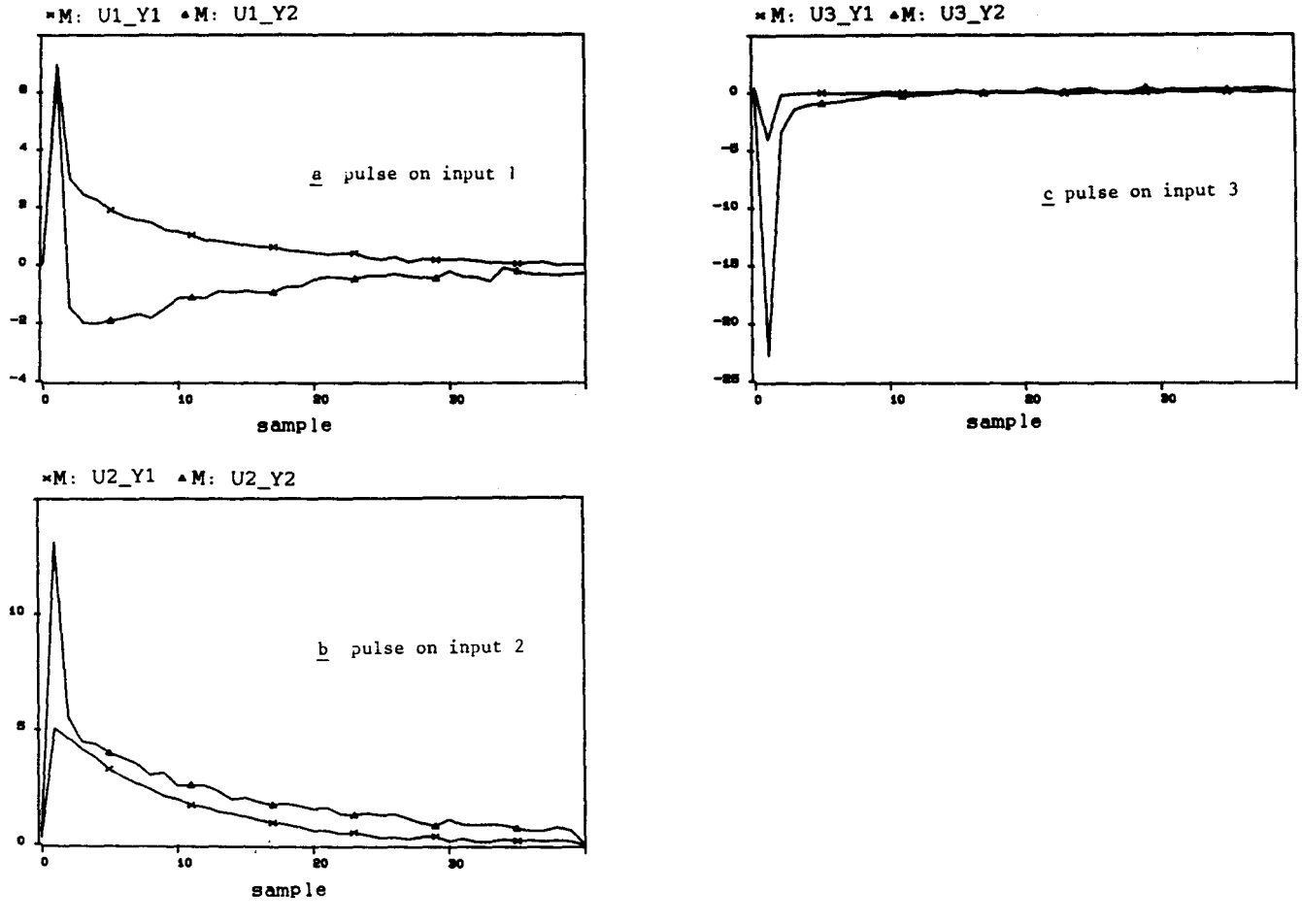
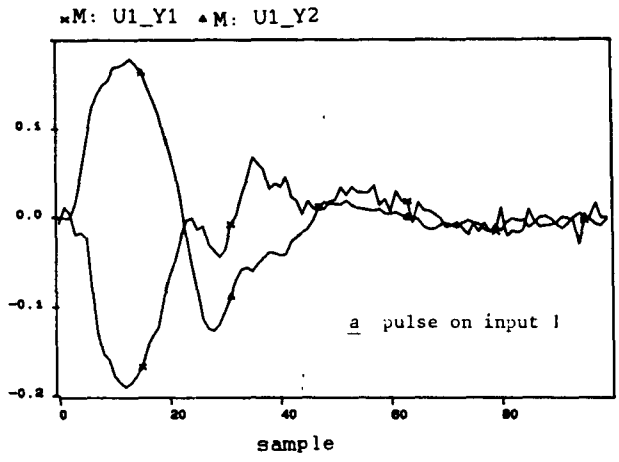


fig 6.2 Estimated impulse responses for CARRIERE. (Mpar = 40)
a Response to a pulse on input 1. (M:U1_Y1, M:U1_Y2)
b Response to a pulse on input 2. (M:U2_Y1, M:U2_Y2)
c Response to a pulse on input 3. (M:U3_Y1, M:U3_Y2)



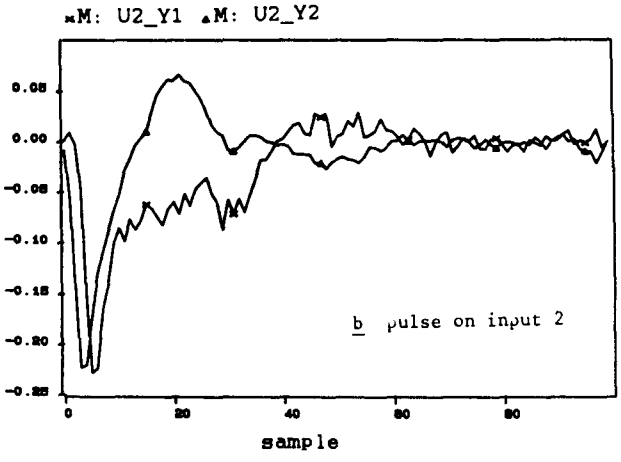
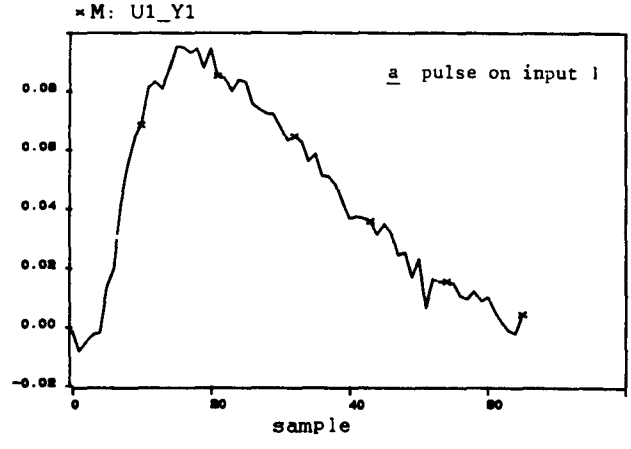
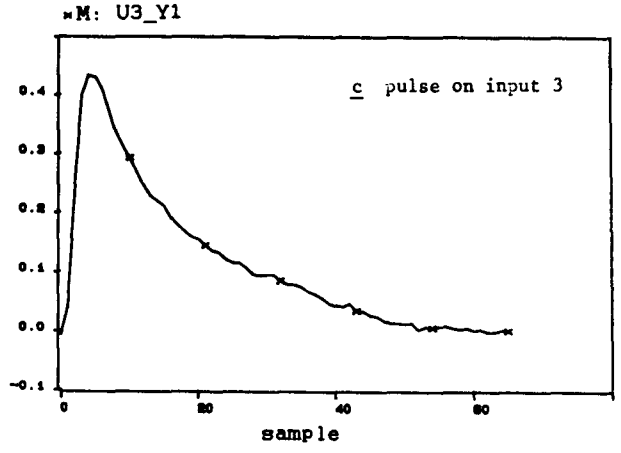


fig 6.3 Estimated impulse responses for PICOS. (Mpar = 99)
a Response to a pulse on input 1. (M:U1_Y1, M:U1_Y2)
b Response to a pulse on input 2. (M:U2_Y1, M:U2_Y2)





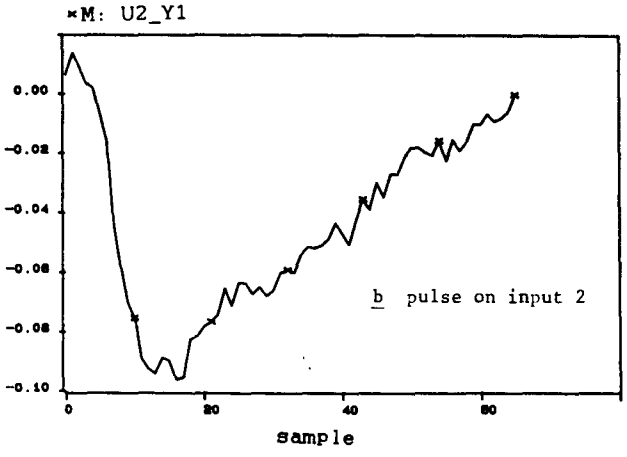


fig 6.4 Estimated impulse response for FEED1. (Mpar = 65)
a Response to a pulse on input 1. (M:U1_Y1)
b Response to a pulse on input 2. (M:U2_Y1)
c Response to a pulse on input 3. (M:U3_Y1)

As can be seen the scaling on the vertical axis differs considerably for the three figures. In an absolute sense the response due to a pulse on input 3 is about 4 times as big as the response due to a pulse on one of the other inputs.

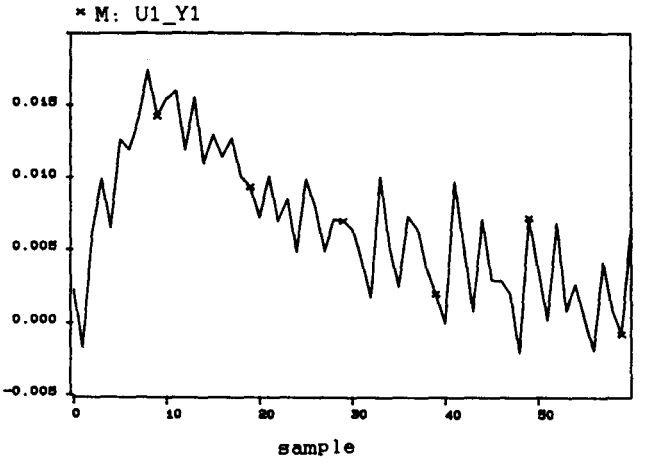
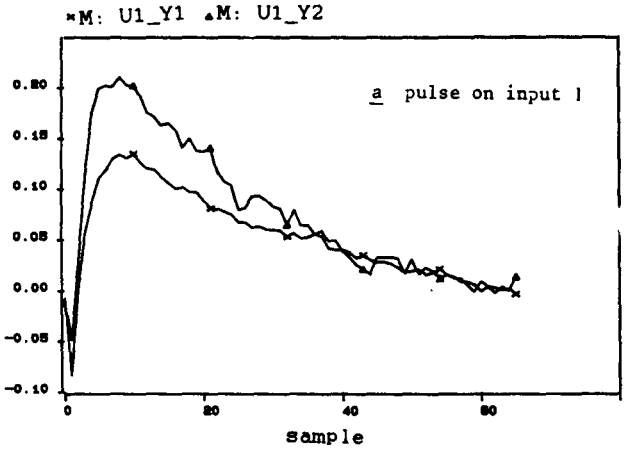
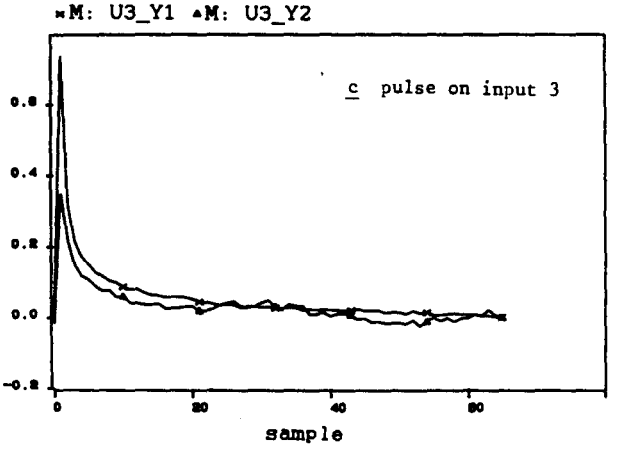


fig 6.5 The estimated impulse response for CLIMCH. (Mpar = 60)

In this case the Markov parameters are oscillating. The relative output error of the impulse response model is low, and as later on will be shown the realization of this impulse response performs well, also in case of cross validation.





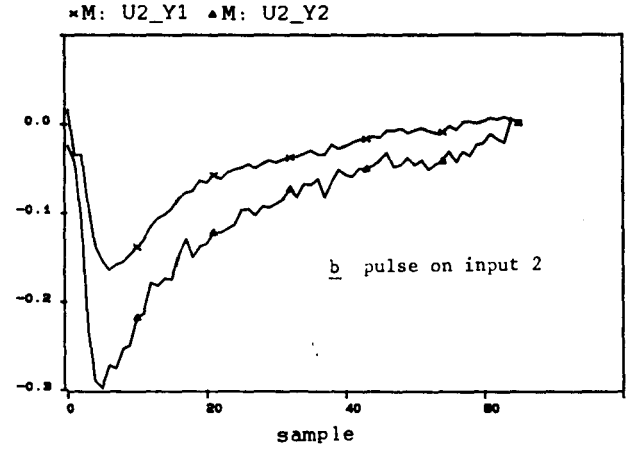


fig 6.6 Estimated impulse responses for FEED2. (Mpar = 65)
a Response to a pulse on input 1.(M:U1_Y1, M:U1_Y2)
b Response to a pulse on input 2.(M:U2_Y1, M:U2_Y2)
c Response to a pulse on input 3.(M:U3_Y1, M:U3_Y2)

7. TESTING OF THE ORDER/DIMENSION TESTS.

In this chapter two order/dimension tests, HANKDIM and HANKDIM2, are tried out. The dimension test HANKDIM computes the singular values of the Hankel matrix (see paragraph 3.1). The order test HANKDIM2 computes the singular values of the matrix HB (see paragraph 3.2). The two methods are compared to each other in this chapter.

During the tests the following three criteria will be examined:

1. δ_i as function of the order/dimension i.
2. δ_{i+1}/δ_i as function of the order/dimension i.
3. Happ as function of the dimension i.

Points of examination are:

1. The clarity of a choice. In other words how easy is it to select a dimension from the figures.
2. Sensitivity of the selected dimension for the number of Markov parameters taken into account.
3. The correctness of the selected order or dimension. This is very obvious for simulated datasets. On practical systems the relative output simulation error as function of the dimension is used as a measure. A comparison with the simulation errors will be made later on.

In paragraph 7.1 the results of tests on simulated datasets are described. The results for practical data are described in paragraph 7.2.

7.1. ORDER/DIMENSION TESTS ON SIMULATED DATA.

In order to test the sensitivity for the number of Markov parameters taken into account, tests have been performed for 40, 30, 20 and 10 parameters of the Markov parameter sequence of the simulated datasets.

Datasets on which HANKDIM and HANKDIM2 are identical, as AST1 and AST2, are referred to by HANKDIM(2).
Tests of HANKDIM(2) on AST1 and AST2 render in all cases a clear, and correct dimension 2. The singular values (δ_i) and the relative approximation error of the Hankel matrix (Happ) also indicate this dimension 2.

The results for the CARRIERE dataset are shown in figure 7.1, for HANKDIM, and in 7.2, for HANKDIM2.

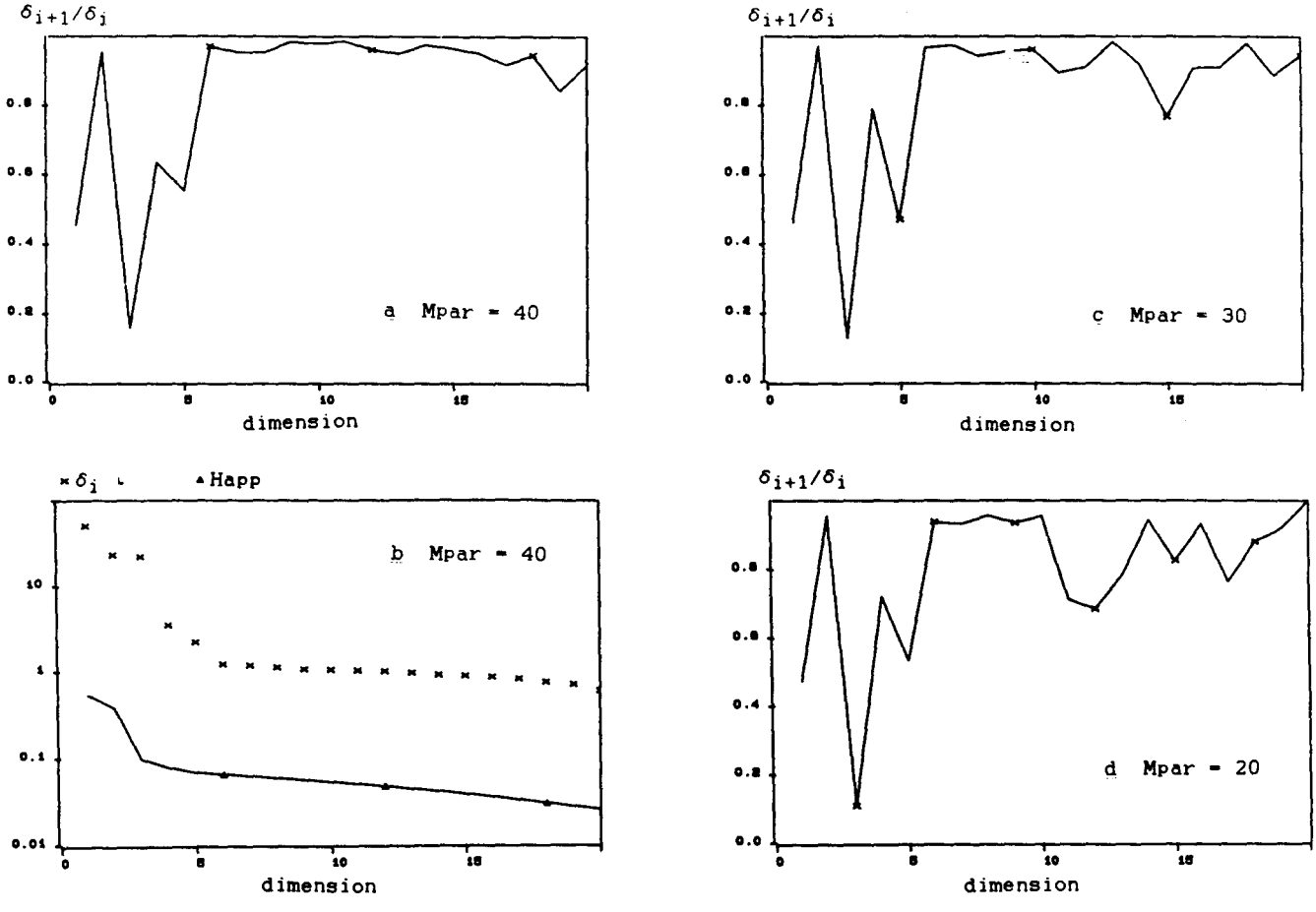


fig 7.1. HANKDIM on CARRIERE.
> δ_{i+1} / δ_i versus dimension for 40 (a), 30 (c), and 20 (d) Markov parameters used.
> δ_i and Happ as function of the dimension on a logarithmic scale (b).

It is clear that HANKDIM indicates the correct dimension (5) if (δ_{i+1} / δ_i) is plotted, though also a dimension 3 can be selected. The singular values δ_i also indicate a dimension 3 or 5. The relative approximation error of the Hankel matrix indicates a dimension 3.

The results of applying HANKDIM2 to the CARRIERE dataset are shown in figure 7.2.

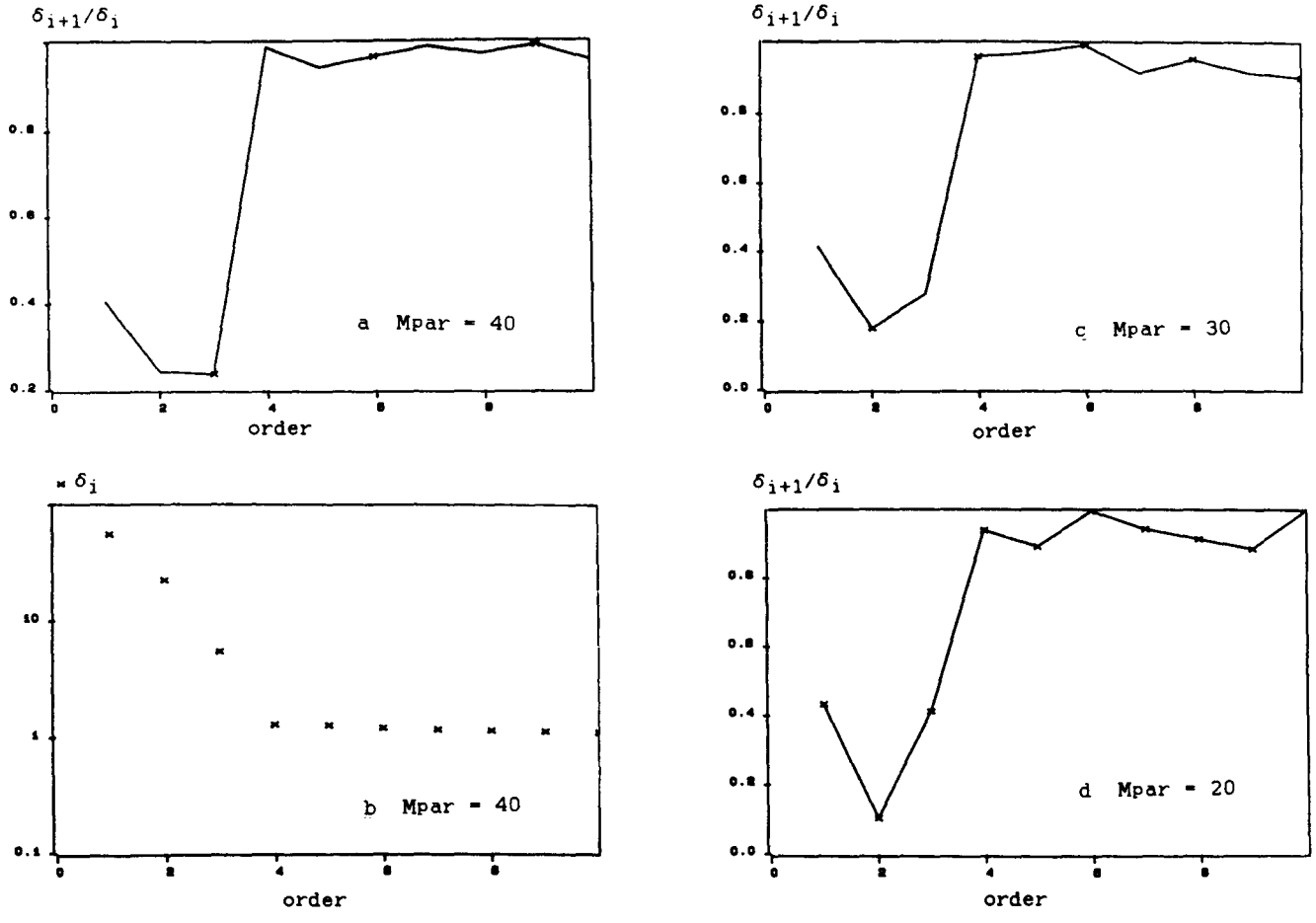


fig 7.2. HANKDIM2 on CARRIERE.
> δ_{i+1} / δ_i versus dimension for 40 (a), 30 (c), and 20 (d) Markov parameters used.
> δ_i as function of the dimension on a logarithmic scale (b)

HANKDIM2 renders the correct order 3. If 20 Markov parameters are used an order 2 could be chosen. The singular values (δ_i) clearly indicate an order 3. (Mpar=40)

Some conclusions that can be drawn, from the tests of HANKDIM and HANKDIM2 on simulated data are :

- On this simulated data both HANKDIM and HANKDIM2 give good results. The sensitivity for the number of Markov parameters taken into account is low.

- In all cases the singular values as function of the dimension can be used for selecting a dimension.

The noise level on the CARRIERE data is very low.(S/N=20dB) For AST1 and AST2 the noise level is considerably higher. (S/N = 10 dB)

7.2. ORDER/DIMENSION TESTS ON PRACTICAL DATA.

The tests executed on practical data are shown in table 7.2.

Table 7.2 Tests of HANKDIM/HANKDIM2 on practical datasets.

data	method	Markov parameters	figure
CLIMCH	HANKDIM(2)	60, 40, 30, and 20	fig. 7.3
PICOS	HANKDIM	99, 70, and 50	-
	HANKDIM2	99, 70, and 50	-
FEED1	HANKDIM(2)	65, 50, 35, and 20	-
FEED2	HANKDIM	65, 50, 35, and 20	fig. 7.4
	HANKDIM2	65, 50, 35, and 20	fig. 7.5

Now the results will be discussed using the subsequent figures as an illustration.
First the tests on CLIMCH will be discussed. (figure 7.3)

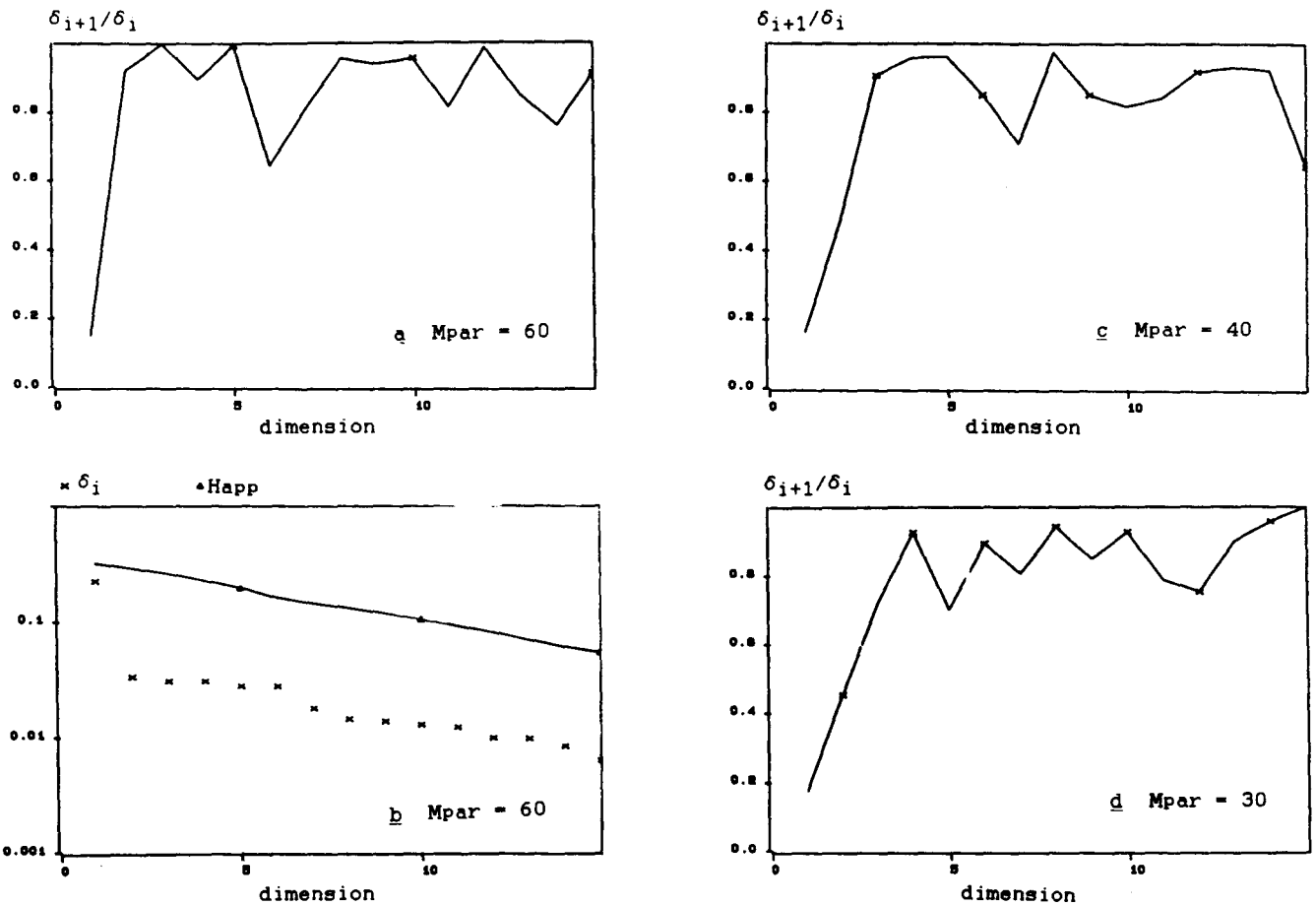


fig 7.3. HANKDIM(2) on CLIMCH
- > δ_{i+1} / δ_i versus dimension with 60 (a), 40 (c), and 30 (d) Markov parameters taken into account.
- > δ_i and Happ versus dimension on a logarithmic scale with 60 Markov parameters (b).

A dimension 1 is a clear choice if 60 Markov parameters are used. This is reinforced by the singular values plotted as function of the dimension. (fig 7.3b) If less than 40 Markov parameters are used it is possible to consider a dimension 2, but a dimension 1 is indicated most clearly.

On the next two pages the results of HANKDIM and HANKDIM2 on the 3 input - 2 output system FEED2 will be presented.

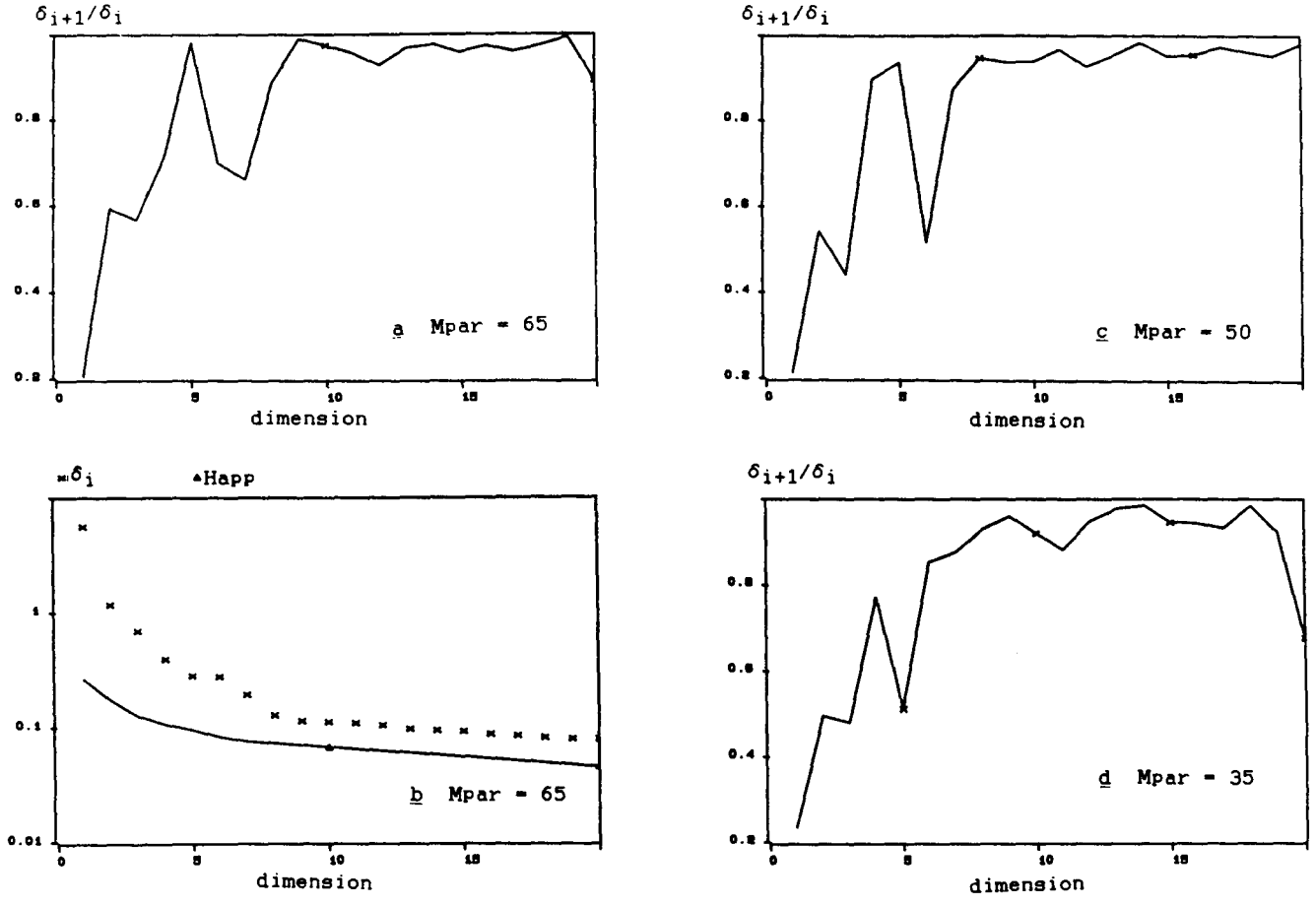


fig 7.4. HANKDIM on FEED2
> δ_{i+1} / δ_i versus dimension with 65 (a), 50 (c), and 35 (d) Markov parameters taken into account.
> δ_i and Happ versus dimension on a logarithmic scale with 65 Markov parameters (b).

The selected dimension depends on the number of Markov parameters taken into account. The selected dimension varies between 5 (for Mpar=35) and 7 (for Mpar=65). The relative approximation error (Happ) doesn't indicate a clear dimension. The singular values do indicate also a dimension 7.(fig 7.4a, Mpar=65)

The results of HANKDIM2 on the same dataset are shown in figure 7.5.

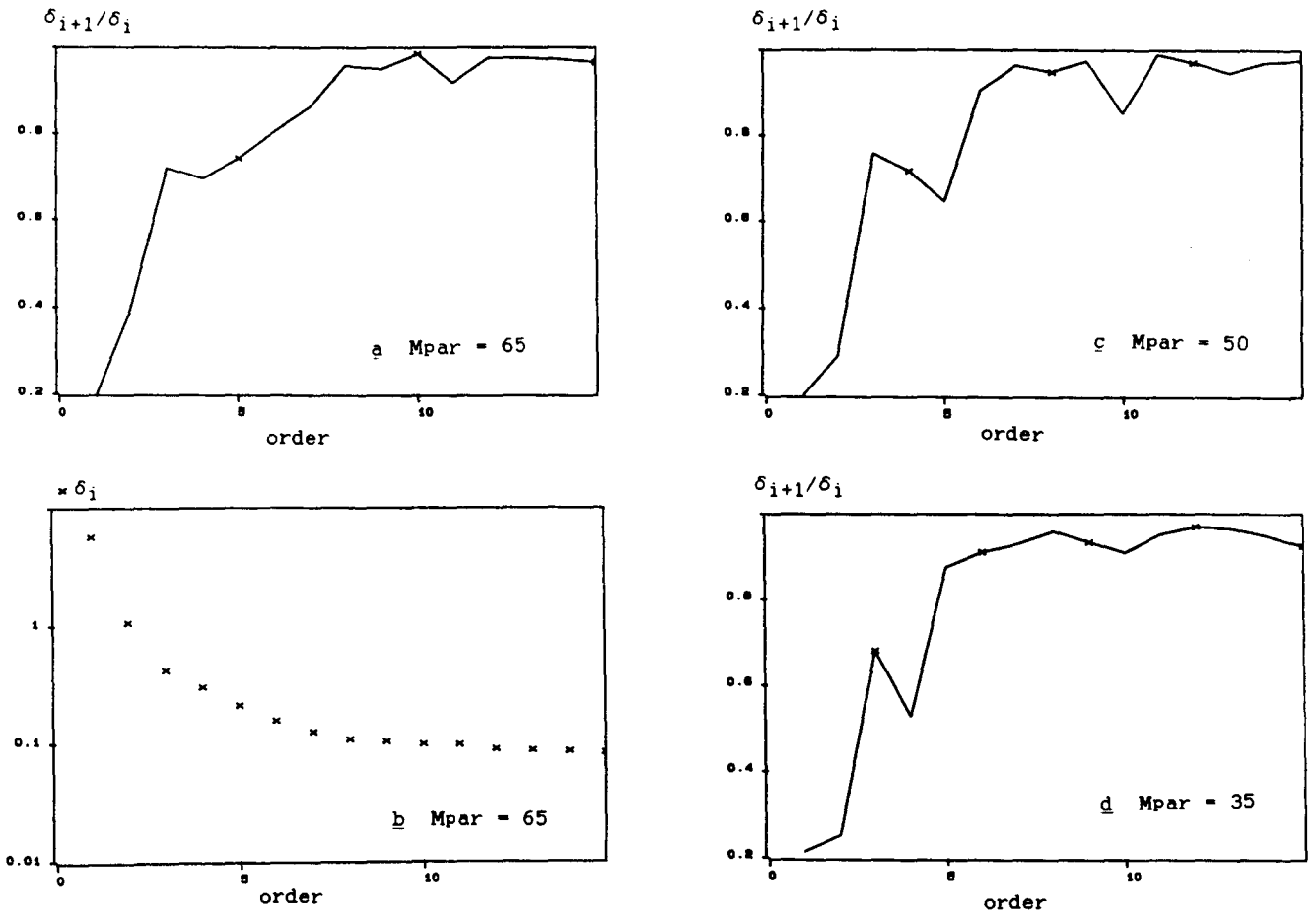


fig 7.5. HANKDIM2 on FEED2

- > δ_{i+1} / δ_i versus dimension with 65 (a), 50 (c), and 35 (d) Markov parameters taken into account.
- > δ_i versus dimension on a logarithmic scale with 65 Markov parameters taken into account (b).

For 65 Markov parameters its difficult to select an order (figure 7.5a). If less Markov parameters are taken into account an order 4 or 5 would be selected. In this case the singular values don't indicate an order clearly. The selected order and the clearness of the results depends on the number of Markov parameters taken into account.

The main results of all tests are recited in table 7.3.

Table 7.3. Results of HANKDIM and HANKDIM2 on practical data.

data	method	selected dimensions
CLIMCH	HANKDIM(2)	dimension 1, may be 2 for Mpar ≤ 40
PICOS	HANKDIM	A dimension 8 or 9 , though a dimension 3 isn't impossible
	HANKDIM2	An order 6 to 8 (implies dim ≤ 16)
FEED1	HANKDIM(2)	dimension 3 or 4, clear choice
FEED2	HANKDIM	choices vary from 5 to 7
	HANKDIM2	more clear results for less Markov parameters, order 4 to 5 (implies dimension ≤ 8 to 10)

Some conclusions that can be drawn, from tests of HANKDIM and HANKDIM2 on practical data, are :

- In many cases there isn't a clear choice for the dimension or the order. (see fig 7.5 for HANKDIM2)
- The selected dimension may depend on the number of Markov parameters taken into account. (sensitivity) (see fig 7.4 for HANKDIM)
- The expected more enunciated figures for HANKDIM2 don't appear. In cases where it is possible to select a clear order, it is also possible to select a clear dimension. (compare figure 7.4 with figure 7.5)
- The singular values (δ_i) or the relative approximation error of the Hankel matrix (Happ) as function of the dimension don't give much additional information about the dimension to be chosen.(see fig 7.4b and fig 7.5b)

7.3. CONCLUSIONS.

In this chapter we have tried out two order/dimension tests. implemented as HANKDIM and HANKDIM2. The dimension test HANKDIM computes the singular values of the Hankel matrix (paragraph 3.1) and the order test HANKDIM2 the singular values of the HB-matrix (paragraph 3.2).

The conclusions that can be drawn are:

1. Often, on practical data. it is not possible to select a clear dimension.

2. The number of Markov parameters taken into account may influence the selected order/dimension.

3. In case of simulated datasets a correct order or dimension is selected. On practical data not much can be said about the correctness of the selected order or dimension.

4. The singular values divided by their predecessor (δ_{i+1}/δ_i) gives best results. Specifically on the practical data the other two criteria don't enable an order/dimension choice.

A final conclusion on the performance of these order/ dimension tests is drawn after the relative output simulation errors are computed. (Chapter 9)

8. TESTING OF THE HANKEL REALIZATION ALGORITHM.

In this chapter the results of testing the HANKEL realization algorithm are described. Properties that are tested on:

1. The influence of the number of Markov parameters used on:
   - > the output errors and the selected dimension.
   - > the shape of the impulse response of the realization. (paragraph 8.1)
2. The sensitivity of the output errors and selected dimension for the length of the simulation range. (paragraph 8.2)
3. Cross validation results of Hankel realizations. (paragraph 8.3)
4. Differences between the four implemented variants of the Hankel realization algorithm described in paragraph 2.4. (paragraph 8.4)
5. The objective of a realization method defined in paragraph 1.3. (paragraph 8.5)

All properties mentioned above are tested on almost every dataset described in chapter 5. For each property some representative examples are presented.

The conclusions that can be drawn from testing HANKEL are summarized in paragraph 8.6.

8.1 SENSITIVITY FOR THE NUMBER OF MARKOV PARAMETERS.

In this paragraph the sensitivity of the output simulation errors and the dimension choice for the number of Markov parameters taken into account is tested. Also the influence on the impulse response of the realization is shown.

The first example uses the dataset AST1. On AST1 HANKEL is applied using 40, 30, 20, and 10 Markov parameters. (figure 8.1.)

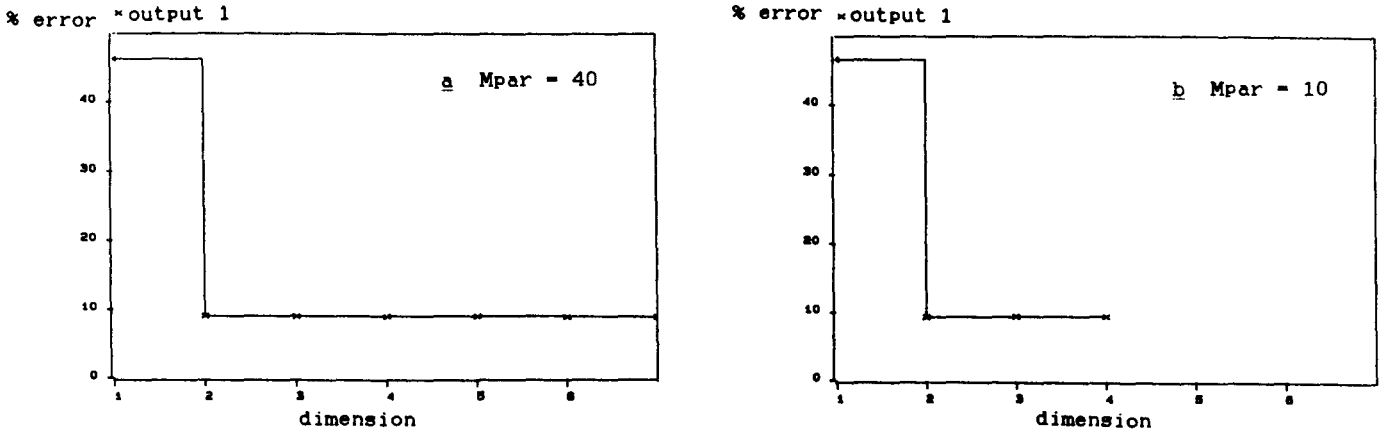


fig 8.1 Relative simulation errors versus dimension.(AST1)
(simulation range [100,1000]; Bosgra-van Zee)
a Markov parameters : 40.
b Markov parameters : 10.

The selected dimension is unshaken by the varying number of Markov parameters taken into account. In table 8.1 the relative output simulation errors as function of the dimension are stated. In this table also the relative root mean square error of the impulse response is recited(RRMS equation (4.2)).

Table 8.1 Relative simulation errors for testing HANKEL with a varying number of Markov parameters for AST1.
(simulation range [100,1000]; Bosgra-van Zee)

Relative output simulation error(%)	Number of Markov parameters (AST1) 40	30	20	10
dimension : 1	46.3	46.3	46.7	46.6
dimension : 2	9.1	9.1	9.1	9.3
dimension : 3	9.0	9.1	9.1	9.4
dimension : 4	9.0	9.0	9.0	9.4
RRMS (dim : 2 (%)) (Mpar = 40)	6.5	6.5	6.5	8.1

The impulse response is approximated equally well for 40.30 and 20 Markov parameters. (Table 8.1) For 10 Markov parameters the result is a little worse.(figure 8.2)

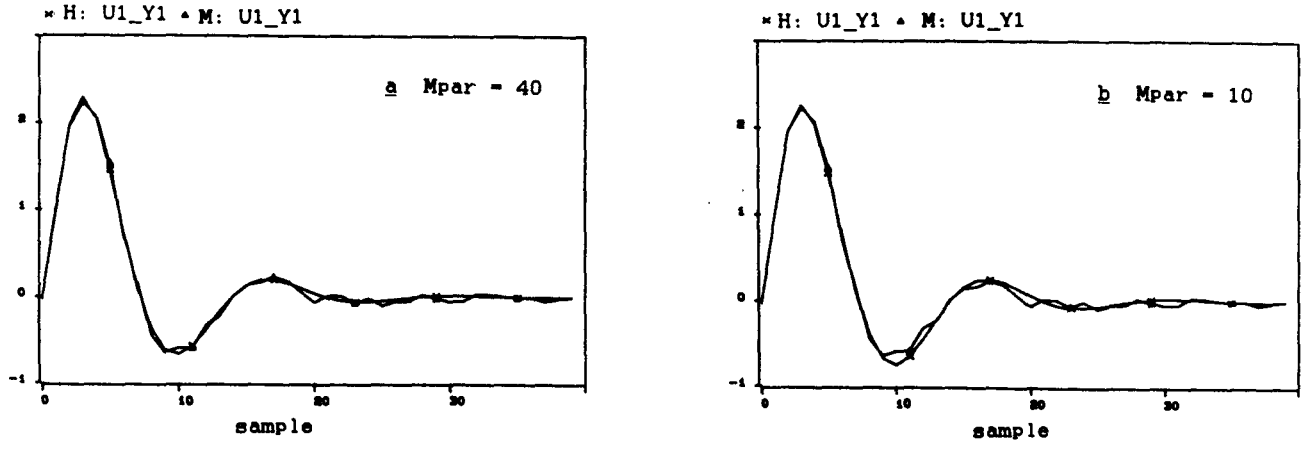


fig 8.2 Input impulse response (M: U1_Y1) compared with the impulse response of a realization (H: U1_Y1). (Bosgra-van Zee; realization dimension : 2)
a Number of Markov parameters used : 40.
b Number of Markov parameters used : 10.

The differences between the two impulse responses are very small. Some differences appear in the tail of the impulse response (sample 10 to 40). This isn't surprising because H: U2_Y2 of figure 8.2b is computed using only the first 10 Markov parameters.

In the next example the dataset FEED1 is used. On this dataset HANKEL is applied using 65, 50, 35, and 20 Markov parameters. (figure 8.3)

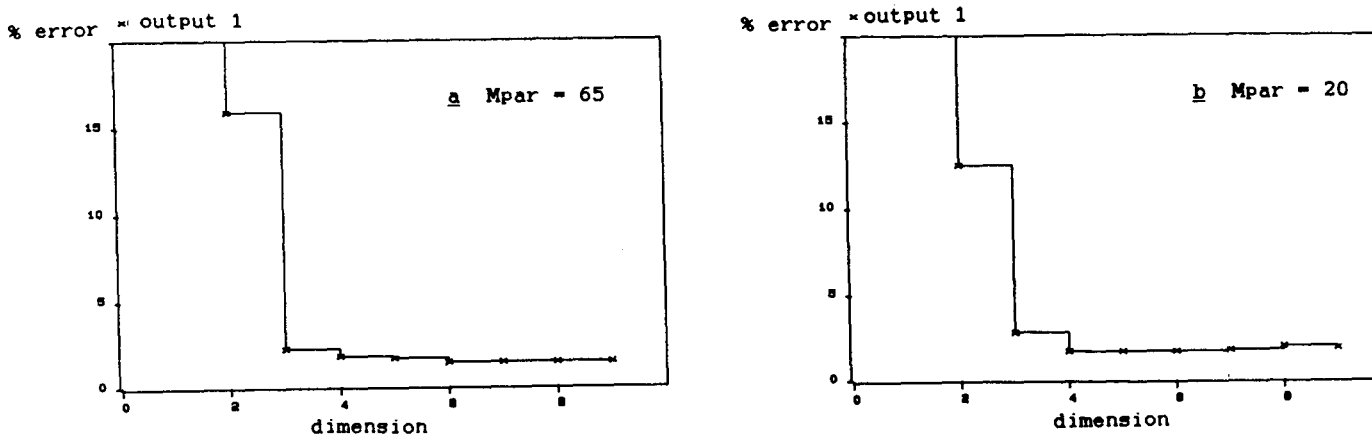


fig 8.3 Relative simulation error versus dimension.(FEED1) (simulation range [92.912]; Bosgra-van Zee)
a Number of Markov parameters used : 65.
b Number of Markov parameters used : 20.

The resulting dimension choice may be a little influenced, because of doubts between dimension 3 and 4. The relative simulation errors are recorded in table 8.2.

Table 8.2 Relative simulation errors of testing HANKEL with a varying number of Markov parameters on FEED1. (simulation range [92.912]; Bosgra-van Zee)

Relative output simulation error(%)	Number of Markov parameters (FEED1) 65	50	35	20
dimension : 1	33.5	32.9	32.3	34.1
dimension : 2	15.9	15.4	14.2	12.5
dimension : 3	2.2	2.1	2.0	2.8
dimension : 4	1.7	1.8	1.6	1.6
dimension : 5	1.6	1.8	1.7	1.7

The differences between the simulation errors for the varying number of Markov parameters are small. Only dimension 3 for 20 Markov parameters shows a small deviation. The differences between the impulse responses of a realization using 20 Markov parameters and a realization using 65 Markov parameters are illustrated in figure 8.4 for input 1 (dimension 4).

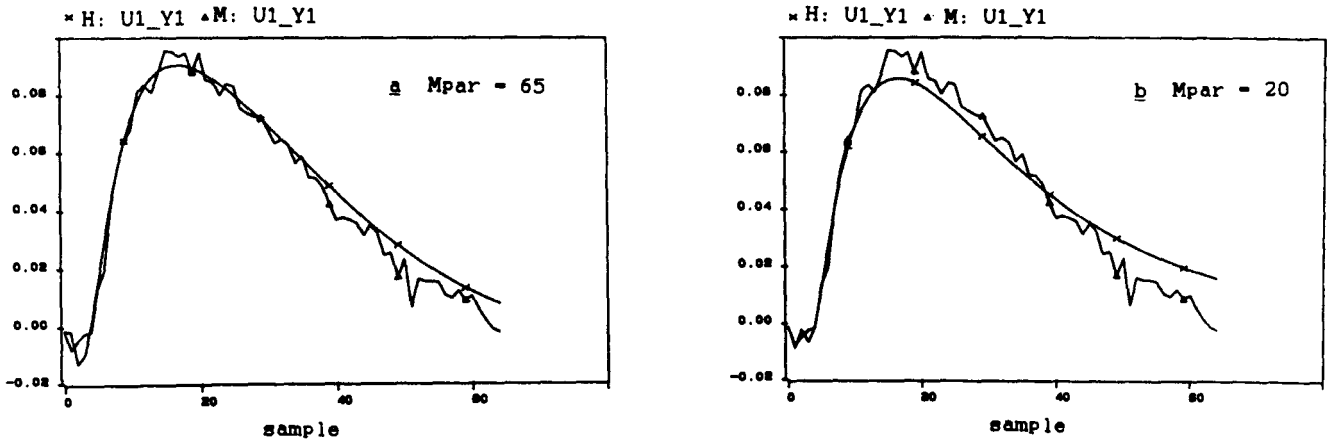


fig 8.4 Input impulse response (M: U1_Y1) compared with the impulse response of a realization.(H: U1_Y1) (Bosgra-van Zee; dimension : 4, FEED1)
a Number of Markov parameters used : 65.
b Number of Markov parameters used : 20.

The tail of the impulse response of the realization using 20 Markov parameters is flatter and its peak value is lower. At the start of the impulse response H:U1_Y1 in figure 8.4b follows the input impulse response better compared to figure 8.4a. This different behaviour is expected because in 8.4b only the first 20 Markov parameters are taken into account.

Conclusion :

For the datasets tested can be concluded that the number of Markov parameters taken into account :
- does not influence the selected dimension and the relative output simulation errors;
- has some influence on the shape of the impulse response;

provided that this number of Markov parameters is not too small.

8.2 SENSITIVITY FOR THE LENGTH OF THE SIMULATION RANGE.

In this paragraph the sensitivity of the dimension choice and the output errors for the number of simulation points used is examined.
The first example is a test performed on AST2. In this case 300, 500. and 1000 simulation points are used. In each case the first 100 samples are excluded from the error computation. (figure 8.5)

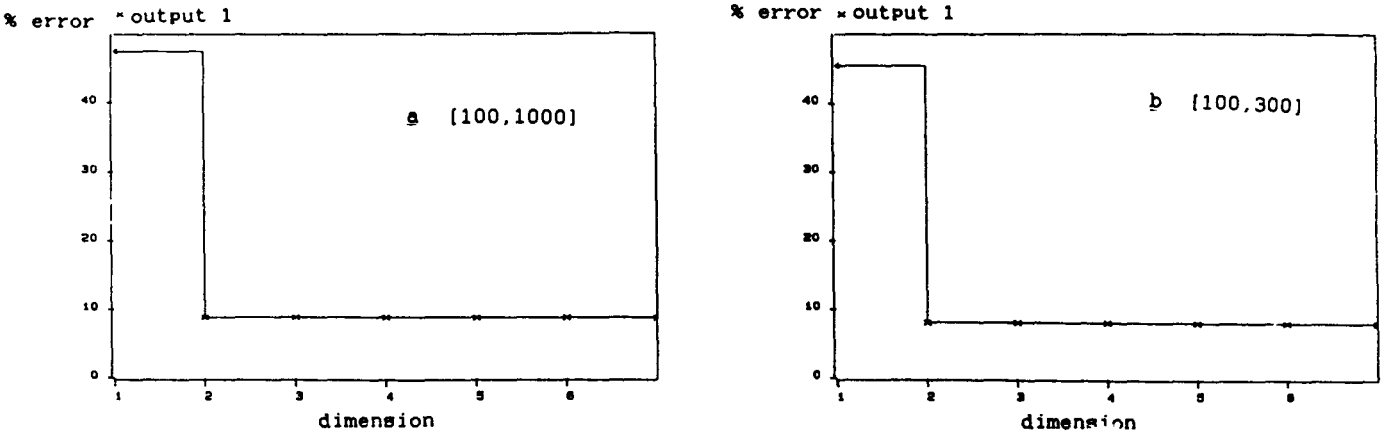


fig 8.5 Relative simulation errors versus dimension.(AST2)
(40 Markov parameters; Bosgra-van Zee)
a Simulation range: [100,1000].
b Simulation range: [100,300].

It is clear that the length of the simulation range doesn't have any influence on the selected dimension. The corresponding percentages are recorded in table 8.3.

Table 8.3 Relative simulation errors (%) of testing Hankel with a varying number of simulation points on AST2. (40 Markov parameters; Bosgra-van Zee)

Relative output simulation error	simulation range of AST2 [100,1000]	[100,500]	[100,300]
dimension : 1	47.5	44.5	45.5
dimension : 2	8.8	7.3	8.0
dimension : 3	8.8	7.3	8.0
dimension : 4	8.8	7.3	8.0

The relative output errors are slightly influenced by the number of simulation points used.

That the influence of the number of simulation points may be considerable is shown in the next example. In this example FEED2 is used. (figure 8.6)

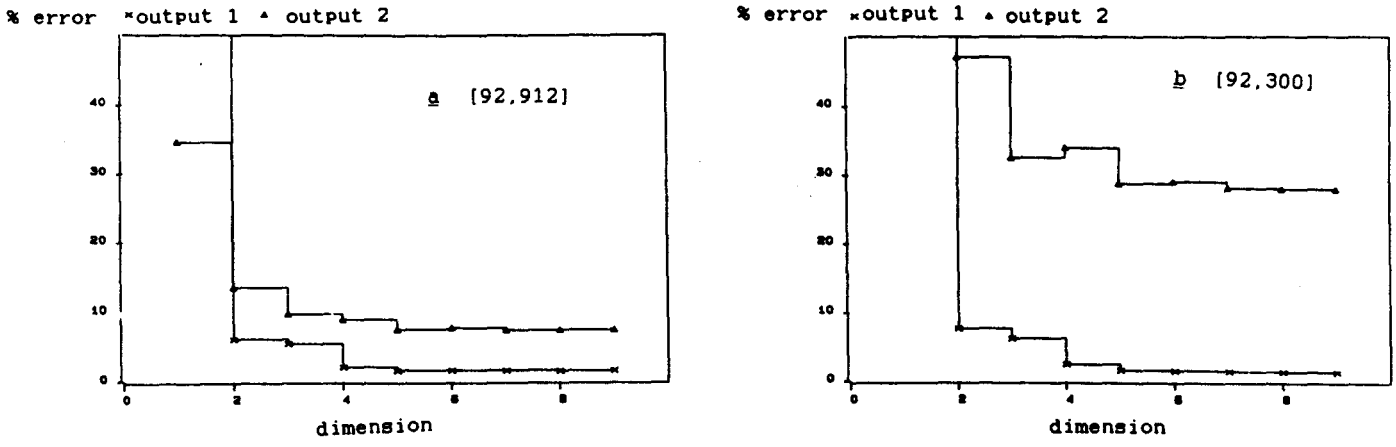


fig 8.6 Relative simulation error versus dimension.(FEED2)
(35 Markov parameters; Bosgra-van Zee)
a Simulation range: [92,912].
b Simulation range: [92,300].

The output error of output 2 is badly influenced, by reducing the length of the simulation range. This is caused by a sensor reaching its limit between data point 240 and 260.
The relative importance has increased due to the smaller simulation range in figure 8.6b.

A representative example for the behaviour of the other datasets is presented in figure 8.7.

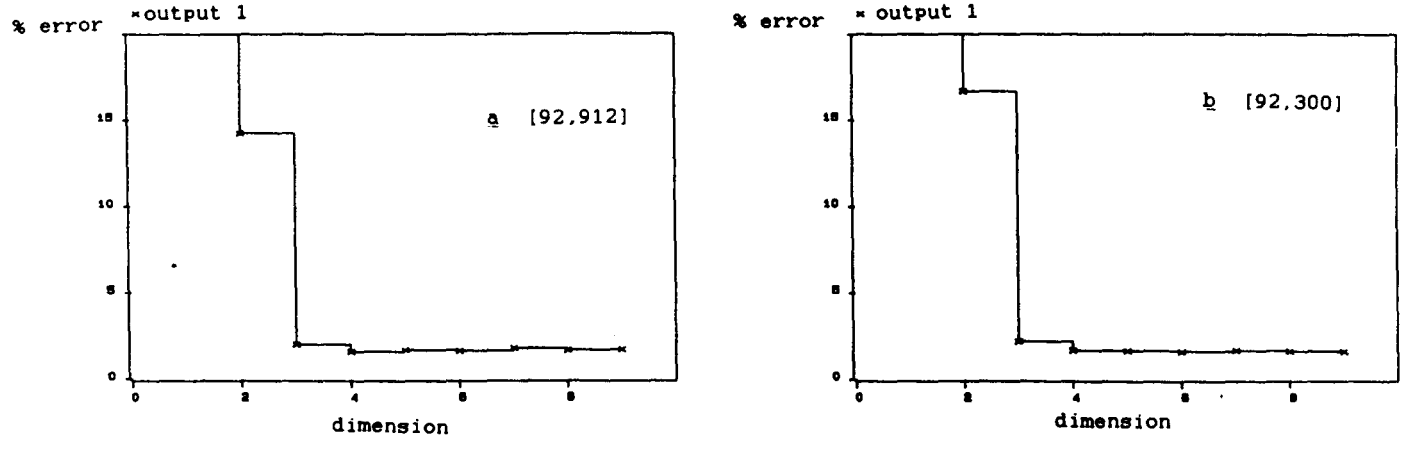


fig 8.7 Relative simulation error versus dimension.(FEED1)
(35 Markov parameters; Bosgra-van Zee)
a Simulation range : [92,912].
b Simulation range : [92,300].

The relative output errors are hardly influenced by the number of data points used for the simulation. This is affirmed by table 8.4.

Table 8.4 Relative simulation errors of testing HANKEL with a varying number of simulation points on FEED1.(35 Markov parameters; Bosgra-van Zee)

relative output simulation errors	simulation range of FEED1 [92,912]	[92,500]	[92,300]
dimension : 1	32.3	30.3	25.2
dimension : 2	14.2	15.7	16.7
dimension : 3	2.0	1.9	2.2
dimension : 4	1.6	1.5	1.7
dimension : 5	1.7	1.6	1.7

Conclusion:

For the datasets tested can be concluded that:
- The number of simulation points taken into account hardly influences the relative output simulation errors and the selected dimension, provided that at least a few hundreds are used.

8.3 CROSS VALIDATION OF HANKEL REALIZATIONS.

In this paragraph the cross validation results of the HANKEL realizations is described. This will be illustrated by the CLIMCH (figure 8.8) and PICOS dataset.(figure 8.9)

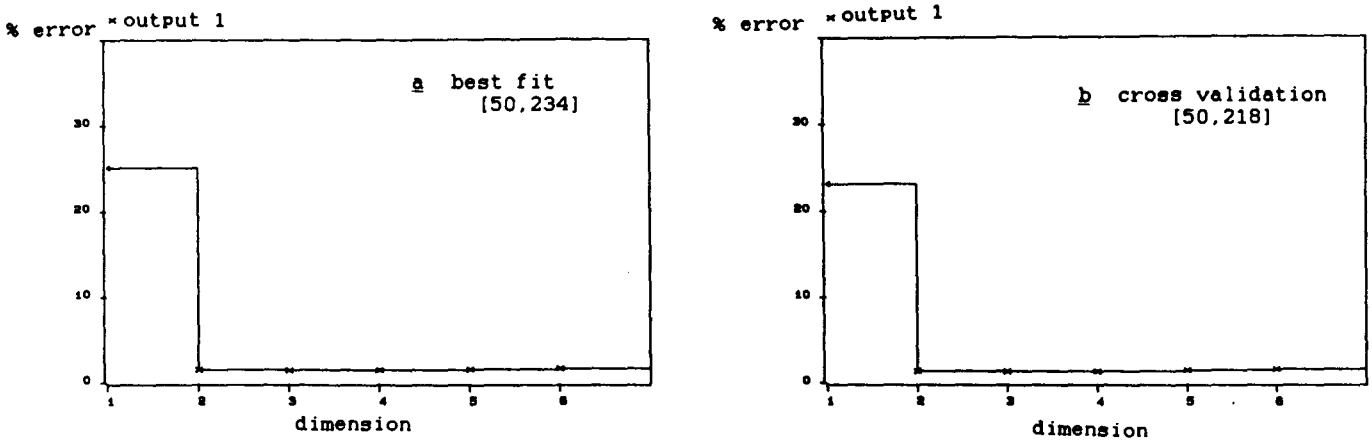


fig 8.8 Relative simulation error versus dimension.(CLIMCH) (Bosgra-van Zee. 40 Markov parameters)
a best fit , simulation range : [50,234].
b cross validation, simulation range : [50.218].

The results in the cross validation are even a little better than in the best fit. If the impulse response model is cross validated then the result is a little worse than the best fit.(table 8.5)

Table 8.5 Relative output errors of the MARKOV impulse response model and HANKEL realizations in best fit and cross validation on CLIMCH.
(Bosgra-van Zee, 40 Markov parameters)

Relative output simulation error (%)	best fit [50,234]	cross validation [50,218]
dimension : 1	25.2	23.1
dimension : 2	1.6	1.4
dimension : 3	1.5	1.4
dimension : 4	1.5	1.4
MARKOV : impulse response model (Mpar=60)	1.2	1.4

In the second example a cross validation dataset for PICOS is used. (figure 8.9)

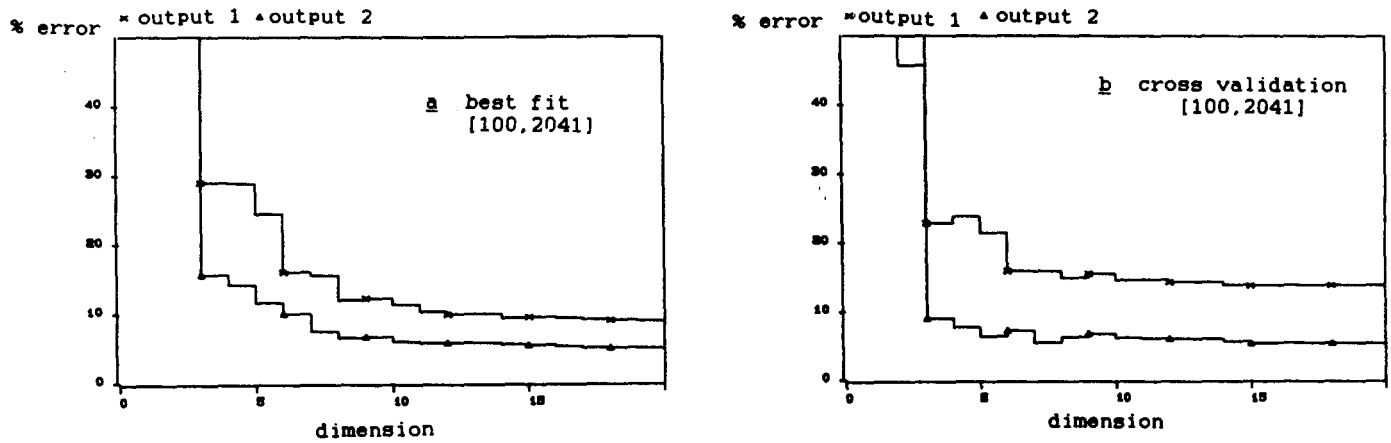


fig 8.9 Relative simulation error versus dimension.(PICOS)
(Bosgra-van Zee, 99 Markov parameters)
a best fit , simulation range : [100,2041].
b cross validation, simulation range : [100,2041]

A dimension 6 or 7 is high enough for the description of the PICOS dataset. For higher dimensions the relative simulation error in the cross validation doesn't lower anymore. (fig 8.9b) This in contrast with the relative output error in the best fit (for output 1) where a dimension 8 seems more suitable. The simulation errors are

smaller in cross validation than in the best fit (dimension smaller than 6).(table 8.6)

Table 8.6 Relative simulation errors of the MARKOV impulse response model and HANKEL realizations in best fit and cross validation on PICOS.
(Bosgra-van Zee, simulation range : [100,2041], 99 Markov parameters used)

Relative output simulation error (%)	output 1		output 2	
	fit*	cross#	fit*	cross#
dimension : 4	28.9	23.8	14.2	7.7
dimension : 5	24.5	21.4	11.6	6.6
dimension : 6	16.1	15.9	10.0	7.2
dimension : 7	15.6	15.8	7.5	5.4
dimension : 8	12.0	14.8	6.6	6.2
MARKOV : impulse response model(Mpar=99)	9.2	13.8	4.4	4.8

* best fit
\# cross validation

In best fit a too high dimension may be chosen easily for the PICOS dataset.
At the other datasets this did not happen, because always a more enunciated dimension choice was possible.

Conclusion :

For the datasets tested can be concluded that the models achieved by estimating an impulse response (MARKOV) and subsequent realization (HANKEL) perform well in cross validation.

8.4 COMPARING THE FOUR IMPLEMENTATION VARIANTS WITH EACH OTHER.

In this paragraph the four implementation variants (Bosgra-van Zee, Damen-Hajdasinski, Kung, and Structural) are compared to each other. These four implementation variants are described in paragraph 2.4.
The first example uses the CARRIERE dataset.(figure 8.10)

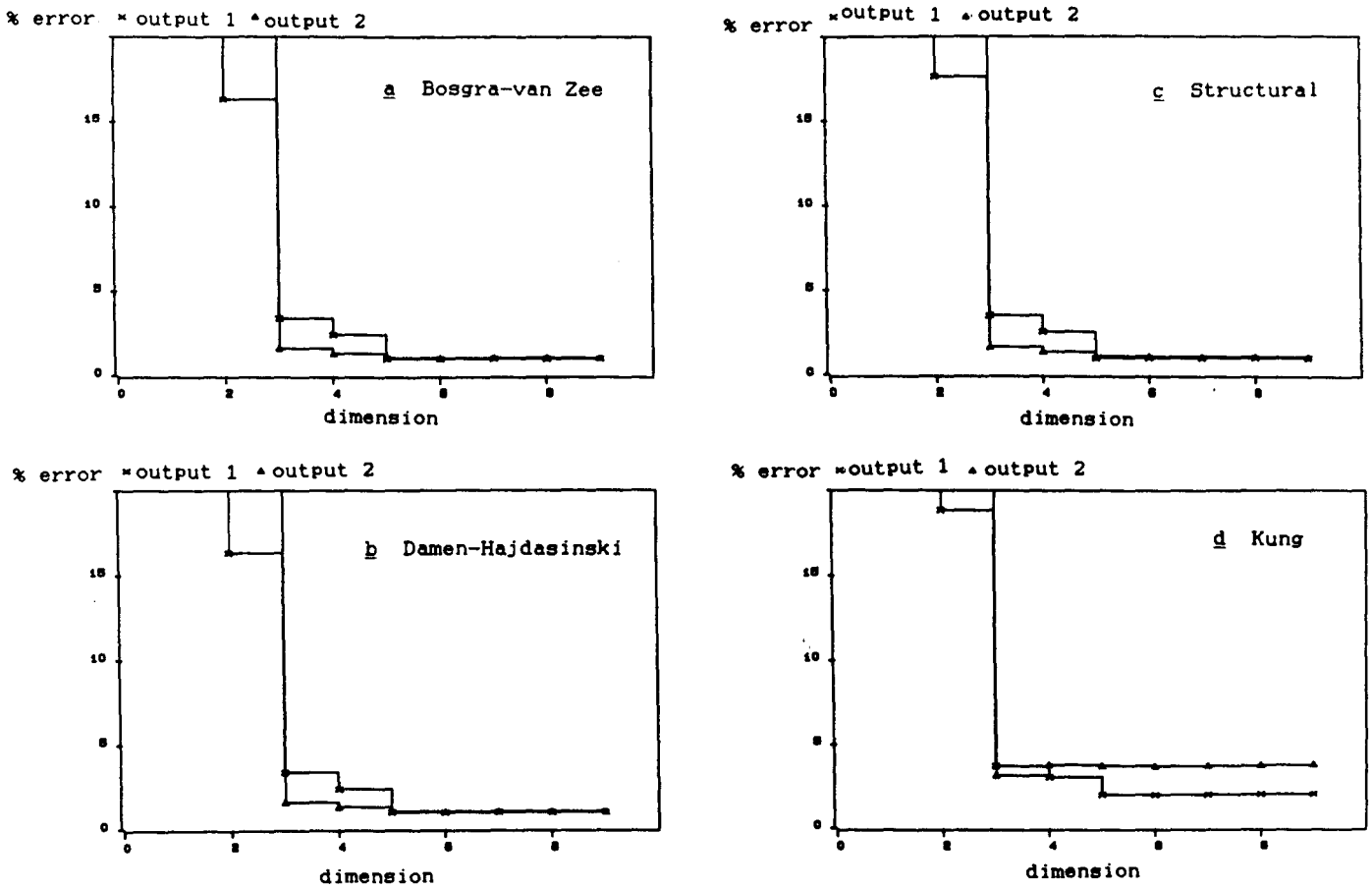


fig 8.10 Relative simulation error vs. dimension.(CARRIERE)
(30 Markov par.; simulation range [100,1000])
a Bosgra-van Zee. c Structural.
b Damen-Hajdasinski. d Kung.

The differences between the three first variants are negligible. (see figure 8.10a, b, and c; table 8.7). The fourth method, Kung, performs worse.

Table 8.7 Relative simulation errors of the HANKEL realizations on CARRIERE.
(Mpar = 30: simulation range : [100,1000]).

dim*	output 1 (CARRIERE)				output 2 (CARRIERE)			
	Bosgra	Damen	Kung	Struc	Bosgra	Damen	Kung	Struc
1	44.5	44.7	45.4	45.4	46.3	46.1	45.4	45.3
2	16.3	16.3	18.8	17.6	44.1	43.9	42.9	43.5
3	3.4	3.4	3.7	3.5	1.6	1.6	3.1	1.6
4	2.4	2.4	3.1	2.5	1.3	1.3	3.8	1.4
5	1.0	1.0	2.0	1.0	1.0	1.0	3.7	1.1
6	1.0	1.0	2.0	1.0	1.0	1.0	3.7	1.1

* realization dimension

The impulse response from input 2 to output 2 for Bosgra-van Zee and Kung are shown in figure 8.11. (realization dimension 5)

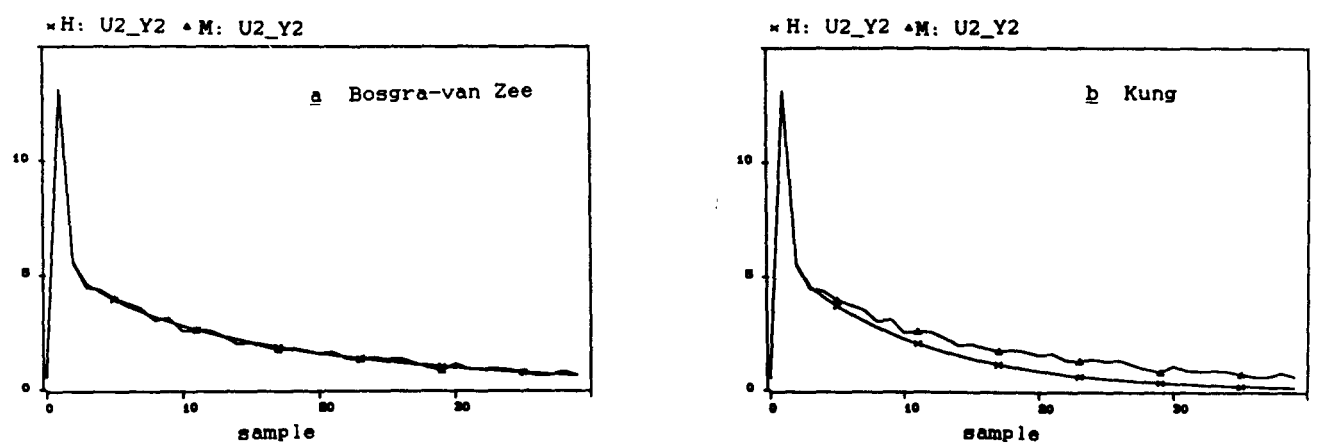


fig 8.11 Input impulse response (M: U2_Y2) compared with the impulse response of a realization (H: U2_Y2). (Mpar=30; realization dimension 5)
a Bosgra-van Zee.
b Kung.

The impulse response of the realization according to Kung deviates considerably more from the by MARKOV estimated impulse response than the impulse response belonging to the realization according to Bosgra-van Zee. The tail of figure 8.11b bends to zero, instead of following the estimated impulse response. This is evidently caused by the zero approximation described in equation (2.20). If the Markov parameter sequence is shortened, the performance of Kung realization worsens.

The second example of this paragraph concerns the CLIMCH data. (figure 8.12)

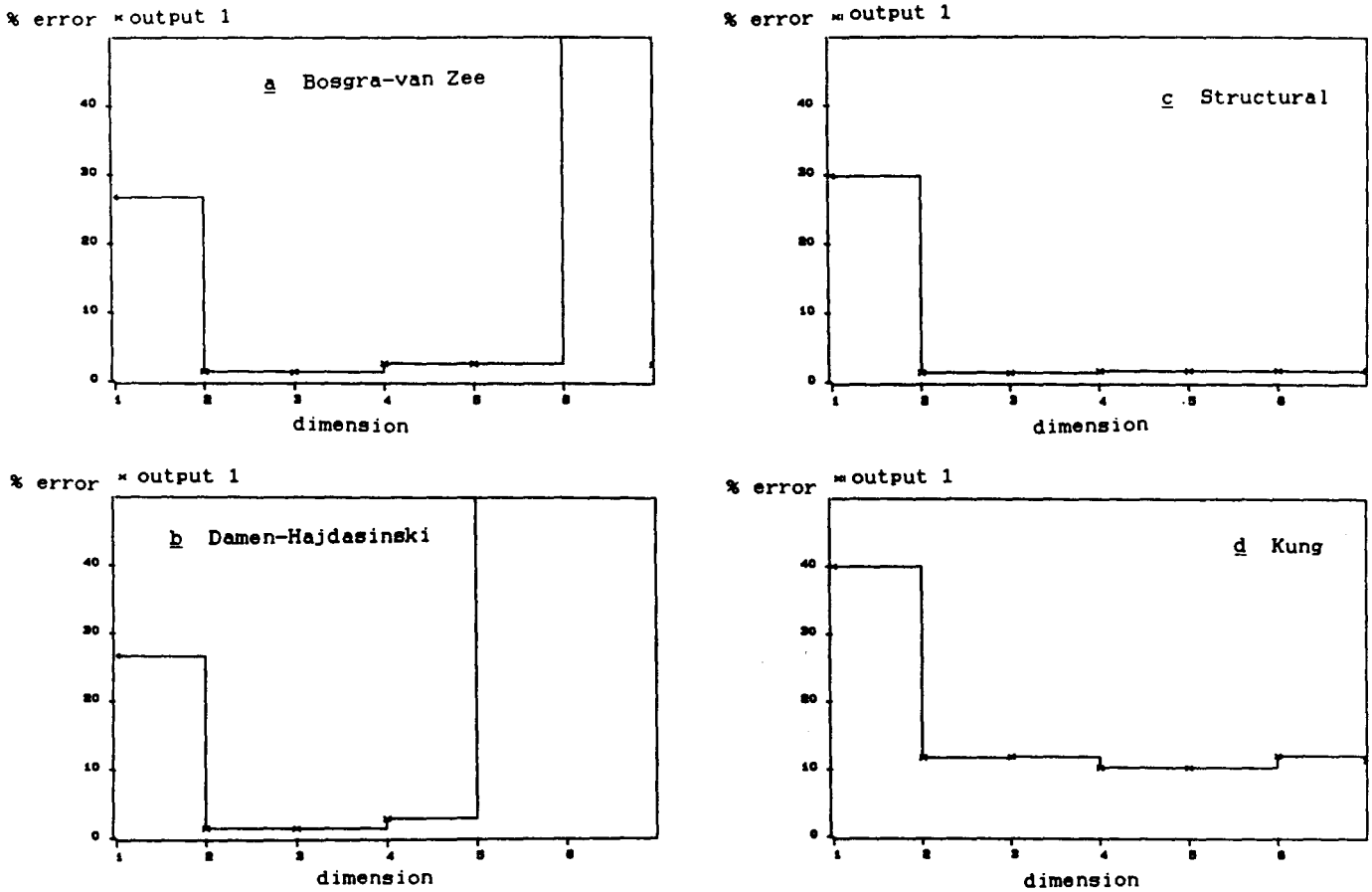


fig 8.12 Relative simulation error vs. dimension.(CLIMCH)
(30 Markov par.; simulation range [100,1000])
a Bosgra-van Zee. c Structural.
b Damen-Hajdasinski. d Kung.

Kung performs again worse than the other three variants. An other feature occurs here. Bosgra-van Zee (dimension 6) and Damen-Hajdasinski (dimension 5,6, and 7) render sometimes unstable models. These appear at unnecessarily high dimensions for this system. This may be caused by noise influences. The same did occur for Structural on another dataset.
Kung never renders an unstable realization, as was expected, because Kung was proven to be stable.(Appendix 2)

Conclusion :
For the datasets tested can be concluded:

- The differences between the first three realization variants (Bosgra-van Zee, Damen-Hajdasinski, and Structural) are very small. The fourth realization method, Kung, performs worse because of the zeros introduced in the shifted Hankel matrix. (see 2.20)
- Sometimes the first three realization methods deliver unstable realizations. However this only occurs if :
  - the number of Markov parameters taken into account is small compared to the impulse response
  - the realization dimension is unnecessarily high for the system under consideration.

  The number of instabilities is very limited.

8.5 HANKEL AS A REALIZATION METHOD.

In this paragraph it is examined whether HANKEL meets the objective: deriving a low dimensional state space model from the estimated impulse responses without much loss of accuracy.
The following table contains the relative simulation errors on all datasets of the by MARKOV estimated impulse response model and a HANKEL realization. The best result, achieved with the realization method of Bosgra-van Zee, is presented.

Table 8.8 Relative simulation errors for the MARKOV impulse response model and a HANKEL realization.
- simulation ranges: table 6.1
- number of parameters of the models: table 8.9.
(Bosgra-van Zee)

Dataset	M_{Mar} $	Mpar	n	MARKOV model *out 1	out 2	HANKEL out 1	out 2
AST1	40	40	2	8.7	-	9.1	-
AST2	40	40	2	8.5	-	8.8	-
CARRIERE	40	40	5	0.9	1.3	1.0	1.1
CLIMCH	60	40	2	1.2	-	1.6	-
#cross	60	40	2	1.4	-	1.4	-
PICOS	99	99	7	9.2	4.4	15.6	7.5
cross	99	99	7	13.8	4.8	15.8	5.4
FEED1	65	35	4	1.4	-	1.6	-
FEED2	65	35	5	1.3	6.1	1.6	7.4

* out = output
\# cross validation with the models estimated on the dataset above
$ M_{Mar} = number of Markov parameters estimated

Not much accuracy is lost during the realization. For PICOS in the best fit some differences arise, but the cross validation indicates that the dimension of the state space model (7) is high enough to describe the process dynamics. In cross validation the differences between HANKEL and MARKOV are small again.

It is always possible to transform a state space model into a (pseudo)-canonical form, therefore the realization can be represented by {n.(p+q) + p.q} parameters.
For the impulse response model the number of parameters is determined by : Mpar.p.q .

Table 8.9 The number of parameters necessary to represent the MARKOV impulse response model and the HANKEL realizations for the diverse datasets.

dataset	p	q	Number of parameters MARKOV	HANKEL
AST1	1	1	40	5
AST2	1	1	40	5
CARRIERE	3	2	240	31
CLIMCH	1	1	60	5
PICOS	2	2	396	32
FEED1	3	1	195	19
FEED2	3	2	390	31

The number of parameters is reduced significantly for all processes.

Conclusion :

For the tested datasets applies:
- HANKEL meets the objective set for a realization method. This objective is: reducing the number of parameters significantly without much loss of accuracy.

8.6 CONCLUSIONS.

In this chapter the HANKEL realization algorithm is tested. This algorithm is described in paragraph 4.2. A summary of the conclusions that can be drawn for the tested datasets is given now:

1. The number of Markov parameters taken into account, provided not too small:
   - does not influence the selected dimension and the relative output simulation errors.
   - has some influence on the shape of the impulse response. (paragraph 8.1)

2. The relative simulation errors and the selected dimension are not sensitive to the number of simulation points, provided at least a few hundreds are used. (paragraph 8.2)

3. Models achieved by estimating an impulse response (MARKOV) and subsequent realization (HANKEL) perform well in cross validation. (paragraph 8.3)

4. Considering the four variants described in paragraph 2.4 can be concluded (paragraph 8.4):
   - The differences between the first three variants (Bosgra-van Zee, Damen-Hajdasinski, and Structural) are negligible.
   - The fourth variant, Kung, performs worse. However always renders a stable realization.
   - In case of the first three variants sometimes an instability occurs. This may happen if:
     - the number of Markov parameters taken into account is small compared to the impulse response
     - the realization dimension is unnecessarily high for the system under consideration

   Since unstable realizations occur rarely it is advised to use one of the first three realization variants.

5. HANKEL is capable of reducing the number of parameters used by an impulse response model significantly, without much loss of accuracy. Thereby the objective set for a realization method is met. (paragraph 8.5)

The general conclusion that can be drawn is that HANKEL is a robust realization algorithm, with good results. (using the relative output simulation error as a measure)

9. COMPARISON OF THE THREE ORDER/DIMENSION TESTS.

In this chapter the results of the order/dimension tests of chapter 7 (HANKDIM2/HANKDIM) are compared to the relative simulation errors produced by the realization algorithm. (HANKEL)

Points of comparison are:
- How easy is it to select a dimension ?
- On simulated data : the correctness of the selected dimension.
- On practical data : the relative simulation errors (HANKEL) are used to check on the possibility of choosing a satisfying dimension using singular values. (HANKDIM/HANKDIM2)

And for HANKEL the results of the automatic dimension choice criterion (AIC4 defined in equation (4.1)) are compared to the dimension selected from the figures.

As in chapter 7 HANKDIM(2) is used to refer to cases where HANKDIM and HANKDIM2 are identical.

simulated datasets

On the datasets AST1 and AST2 both HANKDIM(2) and HANKEL indicate in all tests clearly the correct dimension 2. The results of HANKDIM and HANKEL are compared on the CARRIERE dataset. (figure 9.1.)

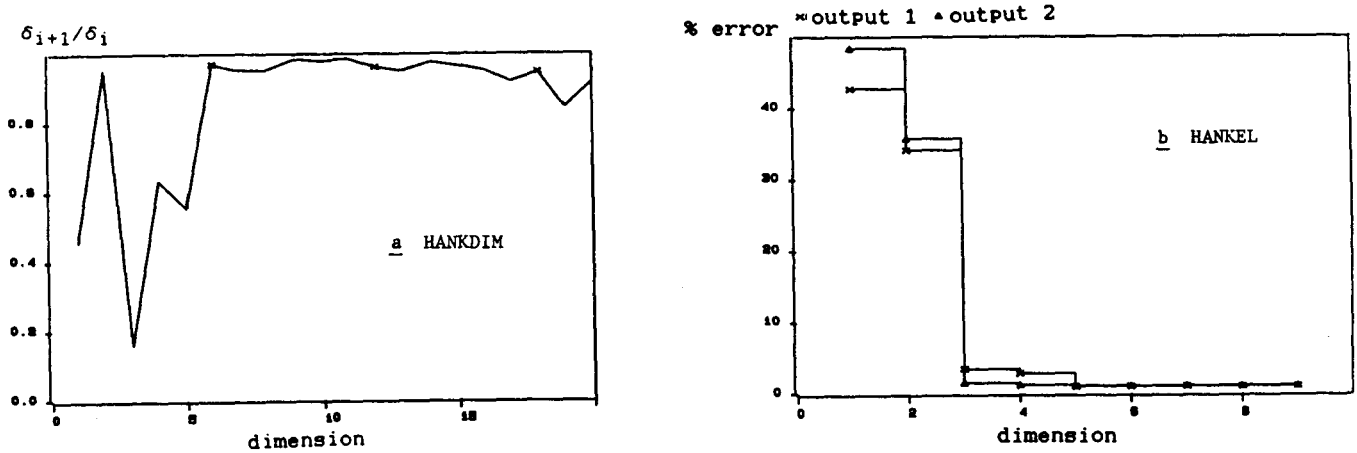


fig 9.1 Dimension tests on CARRIERE.
a HANKDIM (Mpar = 40)
b HANKEL (Mpar=40, [100,1000], Bosgra-van Zee)

The dip a dimension 3 corresponds to an enormous decrease in the relative simulation error. In both cases, HANKDIM and HANKEL. a correct dimension 5 is easily chosen.

Conclusively can be said that on simulated data all methods perform equally well. In all cases it is possible to choose easily the correct dimension/order. The AIC4 chooses the correct dimension for all simulated datasets.

practical data

First two examples are given. Later on all results will be summarized in a table.
The first example concerns a comparison between HANKDIM(2) and HANKEL on the CLIMCH dataset.(figure 9.2)

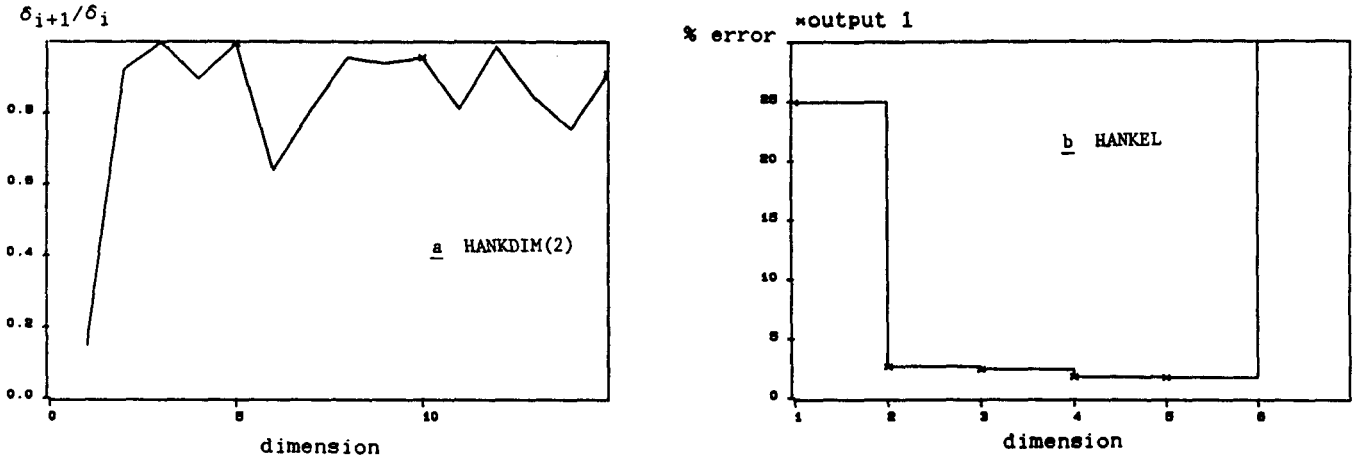


fig 9.2 Dimension test on CLIMCH.
a HANKDIM(2) (Mpar=60)
b HANKEL (Mpar=60, [50,234], Bosgra-van Zee)

HANKDIM(2) indicates a dimension 1, though from the output errors it is clear that dimension 2 must be chosen. In this case HANKDIM(2) failed to indicate a satisfying dimension.

The second example concerns a comparison between HANKDIM2 and HANKEL on FEED2.

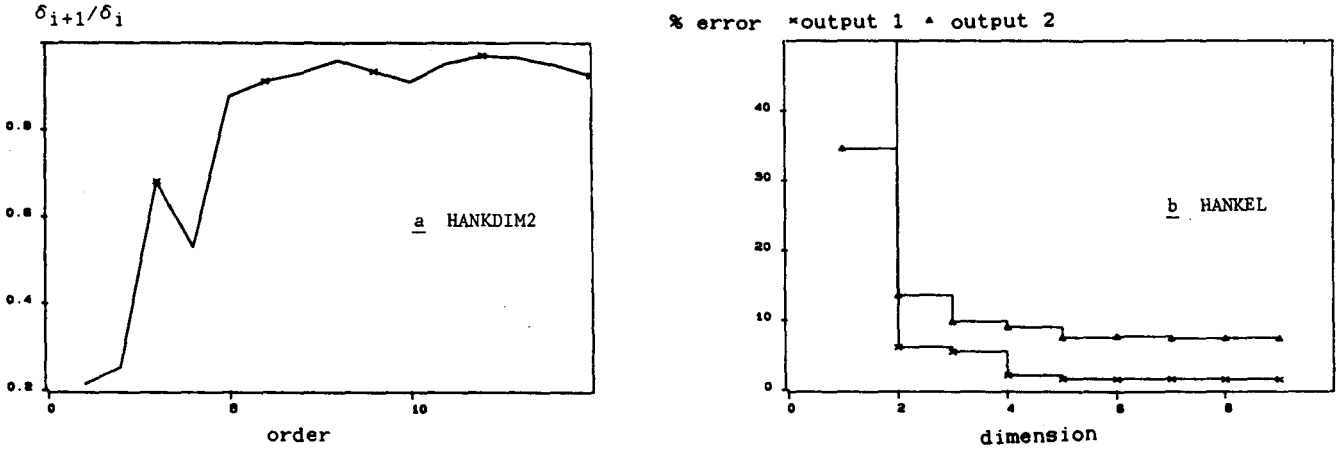


fig 9.3 Order/dimension test on FEED2.
a HANKDIM2 (Mpar = 35)
b HANKEL (Mpar=35, [92,912], Bosgra-van Zee]

Here the disadvantage of selecting an order instead of a dimension is demonstrated.
In this example HANKDIM2 indicates an order 4 for FEED2. An order 4 implicates a dimension smaller than 8(=4.min(p,q)).

The relative simulation errors indicate a dimension 5. Though in this case an order can be chosen easily, it is difficult to indicate the corresponding dimension, because of the inequality (2.3).

In table 9.1 the results on practical datasets are summarized. In fact table 7.3 from chapter 7 is supplemented by the results of HANKEL.

Table 9.1 Results of HANKDIM, HANKDIM2, and HANKEL on practical data.

data	method	dimension chosen
CLIMCH	HANKDIM(2)	dimension 1, may be 2 for Mpar $\leq$ 40
	HANKEL	dimension 2, also chosen by AIC4
PICOS	HANKDIM	A dimension 8 or 9 , though a dimension 3 isn't impossible
	HANKDIM2	An order 6 to 8 (implies dim $\leq$ 16)
	HANKEL	dimension 8, also chosen by AIC4 if Mpar=99, otherwise a dimension 6
FEED1	HANKDIM(2)	dimension 3 or 4, clear choice
	HANKEL	dimension 4, AIC4 : dimension 3 or 4
FEED2	HANKDIM	choices vary from 5 to 7
	HANKDIM2	more clear results for less Markov parameters, order 4 to 5 (implies dimension $\leq$ 8 to 10)
	HANKEL	dimension 4 or 5, AIC4 : 4 or 5

Normally HANKEL renders clear results, making selecting a satisfying dimension easy. From chapter 8 is known that the results of HANKEL are not sensitive to the number of Markov parameters used or to the length of the simulation range taken into account.
Moreover HANKEL gives the opportunity to select a dimension based on simulation results in cross validation. This appears to be of value in case of the PICOS dataset. In all other cases the dimension can be easily chosen in best fit.

conclusions

In this chapter the order test (HANKDIM2) and the dimension test (HANKDIM) are compared to the relative simulation errors as function of the dimension.(HANKEL)

On simulated datasets all tests perform well, giving clear and correct results. In those cases HANKDIM is to be preferred because :

- compared to HANKDIM2 it has the advantage of choosing a dimension instead of an order.
- compared to HANKEL it has the advantage of less computational overhead.

On practical data the performance of HANKDIM and HANKDIM2 is inferior to the performance of HANKEL. Therefore HANKEL is preferred, having the following advantages :

- HANKEL renders clear results and the selected dimension is not sensitive to the number of Markov parameters taken into account or to the number of simulation points used.
- HANKEL gives a direct link between the dimension and the performance of a model.
- It also possible to use a cross validation dataset for the dimension choice.
- The implemented automatic criterion (AIC4) in chooses nearly always the same dimension as would be selected from the graphs.

The disadvantage of HANKEL, the extra computational overhead, is for the greater part abolished by the well functioning of the automatic selection criterion. Because of the structure of the HANKEL algorithm the computation of relative simulation errors for several dimensions doesn't cause much extra computational overhead. (see chapter 4)

10. PERFORMANCE COMPARISON WITH AN INSTRUMENTAL VARIABLE METHOD.

In this chapter the performance of models achieved by estimating an impulse response (MARKOV) and subsequent realization (HANKEL) is compared to the performance of models generated by an Instrumental Variable Method (IVM). Berben [1987] has compared the identification methods available in PRIMAL with each other. The relative output simulation error was used as the performance criterion. The main outcome was that MARKOV and IVM give the best results.

The Instrumental Variable Method (IVM) is an iterative equation error method [Berben, 1987]. IVM estimates the parameters in the following model structure:

$$A(q^{-1})\, y(t) = B(q^{-1})\, u(t) + v(t)$$
$$v(t) = [\, C(q^{-1}) / D(q^{-1}) \,]\, e(t) \qquad (10.1)$$

with: $v(t)$ = equation noise
$e(t)$ = estimation residuals (approximately white noise signal)
$A(q^{-1}), B(q^{-1}), C(q^{-1})$, and $D(q^{-1})$ are polynomials in the backward shift operator q^{-1}

a. simulated data.

In the following table the relative output simulation errors of IVM, MARKOV, and HANKEL on simulated data are recited. HANKEL uses the realization method Bosgra-van Zee.

Table 10.1 Relative simulation errors of IVM, MARKOV, and HANKEL on simulated data. HANKEL uses Bosgra-van Zee. Simulation ranges: table 6.1.

dataset	application		σ_i (%) out 1	out 2
AST1	MARKOV	Mpar=40	8.7	-
	HANKEL	Mused=40, n = 2	9.1	-
	IVM	nA=2,nB=2,nC=2,nD=0	9.1	-
AST2	MARKOV	Mpar=40	8.5	-
	HANKEL	Mused=40, n = 2	8.8	-
	IVM	nA=2,nB=2,nC=3,nD=1	8.9	-
CARRIERE	MARKOV	Mpar=40	0.9	1.3
	HANKEL	Mused=40, n = 5	1.0	1.1
	IVM	nA=2,nB=2,nC=2,nD=2	1.0	1.5

The differences between the three methods on simulated datasets are very small.

b. practical data.

In table 10.2 the results on practical datasets are recorded. For CLIMCH and PICOS also the results in cross validation are recorded. Again HANKEL uses the realization method Bosgra-van Zee.

Table 10.2 Relative simulation errors of IVM, MARKOV, and HANKEL on practical datasets. HANKEL uses Bosgra-van Zee. Simulation ranges: table 6.1.

dataset	appli-cation		output 1 fit*	output 1 cros#	output 2 fit	output 2 cros
CLIMCH	MARKOV	Mpar=60	1.2	1.4	-	-
	HANKEL	Mused=40, n = 2	1.6	1.4	-	-
	IVM	nA=2,nB=2,nC=2,nD=2	2.4	3.2	-	-
PICOS	MARKOV	Mpar=99	9.2	13.8	4.4	4.8
	HANKEL	Mused=99, n = 7	15.6	15.8	7.5	5.4
	IVM	nA=6,nB=4,nC=2,nD=2	18.3	21.3	10.2	7.1
FEED1	MARKOV	Mpar=65	1.4	-	-	-
	HANKEL	Mused=35, n = 4	1.6	-	-	-
	IVM	nA=5,nB=5,nC=3,nD=2	2.9	-	-	-
FEED2	MARKOV	Mpar=65	1.3	-	6.1	-
	HANKEL	Mused=35, n = 5	1.6	-	7.4	-
	IVM	nA=5,nB=5,nC=3,nD=2	2.3	-	9.0	-

* best fit
\# A cross validation is performed for the models, determined on the dataset.

On the tested practical data MARKOV+HANKEL performs always better than IVM.

conclusion

In this chapter the performance of models achieved by estimating an impulse response (MARKOV) and subsequent realization (HANKEL) is compared with models estimated by an Instrumental Variable Method (IVM). The relative output simulation error is used as the performance criterion.

For the tested datasets can be concluded that the MARKOV+HANKEL combination always performs at least equally well compared with IVM.
The MARKOV+HANKEL combination has the advantage of being robust.

Most often the differences between the three models are small.

11. CONCLUSIONS AND SUGGESTIONS.

conclusions:

1. The Hankel realization method (implemented as HANKEL in PRIMAL), which computes a (low-dimensional) state space model from the impulse responses of a system, shows a very good performance in all tests carried out.The number of parameters is reduced significantly, without much loss of model simulation performance.

2. The simulation performance of models achieved by estimating the impulse response (MARKOV) and subsequent realization (HANKEL) compares favorably to models estimated by the approximately optimal Instrumental Variable Method (IVM).
   In many cases the differences in performance are small.

3. Before a realization can be performed a dimension for the state space model must be selected. Three tests, based on the estimated impulse responses, have been tried out.
   The first is a dimension test using the singular values of the Hankel matrix. (paragraph 3.1) The second is an order test using the singular values of a matrix consisting of vector Markov parameters. (paragraph 3.2)
   Both methods perform well on simulated data. On practical data the results are:
   - sensitive for the number of Markov parameters used.
   - often difficult to interpret, which made selecting an order/dimension difficult.

   Therefore a third method, implemented in HANKEL, computes for a range of dimensions the relative output simulation errors (best fit or cross validation), from the impulse response and input-output data. This is done at low extra computational cost, compared to the effort needed to compute a single realization.
   Based on these simulation errors the most efficient model is selected automatically.

   The last technique shows to be superior to the first two methods.

4. Four implementation variants are tested. These variants are known as Bosgra-van Zee, Damen-Hajdasinski, and Kung. [van den Hof,1982] The fourth variant is called Structural (chapter 2). The differences in performance between the three variants Bosgra-van Zee, Damen-Hajdasinski, and Structural are negligible.
   The variant according to Kung performs worse. This method will however always return a stable model.
   The other three variants may render an unstable model if:
   - the number of Markov parameters used is very small compared to the impulse response
   - the realization dimension is unnecessarily high.

   Unstable realizations however rarely occur.

5. The Hankel realization method is, for the tested datasets, not sensitive to the number of Markov parameters taken into account, provided this number is not too small.
   The relative output simulation error as a function of the dimension enables a clear dimension choice. The implemented automatic dimension choice criterion (AIC4) selects nearly always a satisfying dimension.

Suggestions for further research:

1. Model reduction.
   In the literature the Hankel realization method is often referred to as a method for model reduction. Given the good results in realization, tests for model reduction may be executed.
   This can be done easily, because the HANKEL application is implemented in such a way that it is also possible to use a state space model as input instead of an impulse response model.
   The results may be compared to model reduction using balanced state space models. [Pernebo and Silverman, 1982]

2. Dimensions of the Hankel matrix.
   The influence of the dimensions of the Hankel matrix on the performance of the resulting realization isn't tested exhaustively, though preliminary tests indicate that not much improvement can be achieved.

3. Variants of the Hankel realization method.
   Since the differences in performance between the variants (Bosgra-van Zee, Damen-Hajdasinski, and Structural) are negligible, not much improvement in performance is expected from a new variant.

4. Realization methods.
   Other realization methods outside Hankel are e.g. the Page matrix [van den Hof,1982] and the optimal Hankel norm approximation method of Glover. [Glover,1984] But according to Backx [1987] both perform worse than the Hankel realization method.

5. One step method.
   The ultimate models are a result of an estimated impulse response and subsequent realization. It may be interesting to perform an ultimate model estimation in one step. [Backx, 1987]

REFERENCES

Backx, A.C.P.M. (1987)
"Identification of an industrial process: a Markov parameter approach."
Dissertatie drukkerij Wibro, Helmond.

Berben, P.J. (1985)
"Order estimation methods."
Internal report PICOS-R-051, Philips PICOS glass.

Berben, P.J. (1987)
"Een optimale IV-parameterschattingsmethode voor PRIMAL."
M. Sc. Thesis, Group Systems and Control, Department of Physics(NR - 1426P), Eindhoven University of Technology.

Bosgra, O.H. and Zee, G.A. (1979)
"The use of realization theory in the robust identification of multivariable systems."
Prepr./Proc. 5-th IFAC Symp. Ident. and Syst. Param. Estim. Darmstadt, p. 477-484, 1979.

Carrière, P.M. (1984)
"Identification of multivariable systems and representations in canonical an pseudo-canonical form."
M. Sc. Thesis, Group Measurement and Control, Department of Electrical Engineering, Eindhoven University of Technology.

Damen, A.A.H. and Hajdasinski A.K. (1979)
"Realization of the Markovparameter sequences using the singular value decomposition of the Hankel matrix."
Eindhoven University of Technology, TH-Report 79-E-95.

Damen, A.A.H. and Hajdasinski A.K. (1982)
"Practical tests with different approximate realizations based on the singular value decomposition of the Hankel matrix."
IFAC Identification and Systems Parameter Estimation 1982, Washington D.C., USA 1982.

Driessen, R.M.F. (1987)
"Vorming van het statische en dynamische model van een experimentele klimaatkamer."
Eindhoven University of Technology, NR - 1486E, 1987.

Freeman, T.G. (1985)
"Selecting the best linear transfer function model."
Automatica, Vol. 21, No 4, p. 361-370, 1985.

Glover, K. (1984)
"All optimal Hankel-norm approximations of linear multi-variable systems and their L^{∞}-error bounds."
Int. J. Control, vol.39, no. 6, p. 1115-1193, 1984.

Hajdasinski, A.K. (1980)
"Linear Multivariable systems."
Eindhoven University of Technology, TH-Report 80-E-106.

Ho, L.B. and Kalman, R.E. (1966)
"Effective construction of linear state-variable model from Input-Output functions."
Regelungstechnik, Vol. 14, p. 545-548, 1966.

Hof, P. van den (1982)
"Approximate realization of noisy linear systems: The Hankel and Page matrix approach."
M. Sc. Thesis, Department of Electrical Engineering, Eindhoven University of Technology.

Hof, P. van den (1984)
"Approximate realization of noisy linear multivariable systems."
Journal A, Vol. 25, no 1, 1984 .

Isermann, R. (1974)
"Prozeß-identification."
Springer-Verlag, Berlijn, 1974.

Janssen, P.A. (1987)
"Practical aspects of process identification with PRIMAL; a tool for process identification."
M. Sc. Thesis, Group Systems and Control, Department of Physics(NR -1470P), Eindhoven University of Technology.

Kailath, T. (1980)
"Linear systems."
Prentice-hall, Englewood Cliffs, New Jersey, 1980.

Kung, S. (1979)
"A new identification and model reduction algorithm via singular value decompositions."
Proc. 12th Annual Asilomar Conf. Circuits Syst. Comput., p. 705-714, 1979.

Linden, R.J.P. van der , Renes, W.A. , Janssen, P.A. (1987)
"Interactive on-line signal analysis, system identification, and controller design."
Proceedings 16-th IASTEP International Conference Identification, Modelling, and Simulation, Paris , 1987.

Linden, R.J.P. van der (1987)
"Markov parameter schatting in PRIMAL."
Eindhoven University of Technology, NR - 1484P, 1987.

Moore, B.C. (1981)
"Principal component analysis in linear systems: Controllability, Observability, and Model Reduction."
IEEE Transactions on automatic control, Vol. AC-26, p. 17-32, 1981.

Pernebo, L. and Silverman, L.M. (1982)
"Model reduction via balanced state space representations."
IEEE Transactions on automatic control, Vol. AC-27, no 2,
p. 382-387, 1982.

Renes, W.A. , Linden. R.J.P. van der , Vucht, G.N.M. van ,
Hardam. E. , Mous, S.L.J. (1987)
"Data-acquisition for modelling and control in an industrial
environment."
Proceedings 16-th IASTEP International Conference
Identification, Modelling, and Simulation, Paris , 1987.

Silverman, L.M. and Bettayeb, M. (1980)
"Optimal approximation of linear systems."
Proc. JACC, San Francisco, CA. 1980.

Staar, J. and Vandewalle, J. (1982)
"Singular value decomposition: a reliable tool in the
algorithmic analysis of linear systems."
Journal A, volume 23, p.69-74, 1982.

Strejc, V. (1980)
"State space theory of linear control."
John Wiley & Sons, 1980.

Zeiger, H.P. and McEwen, A.J. (1974)
"Approximate linear realizations of given dimension via
Ho's algorithm."
IEEE Trans. on automatic control AC-19, p. 153, 1974.

APPENDIX 1.

Equivalence between two descriptions of the Kung realization variant.

In this appendix the equivalence between using the controllability or observability matrix to compute the system matrix A and the description of paragraph 2.4. for the method of Kung is proven.

In literature [Kung, 1979] the controllability or the observability matrix is used to compute the system matrix A. For the controllability matrix (G equation (2.6)) :

$$G = \left| B \quad AB \quad .... \quad A^{(j-1)}B \right| \tag{A1.1}$$

Then :

$$A = Gs\ G^{+} \tag{A1.2}$$

with: Gs = G shifted over one block column to the left.
G^{+} is the pseudo-inverse of G

In the noisy case the singular value decomposition of the Hankel matrix is used.

$$H = U\ \Sigma\ V^{T} \tag{2.12}$$

The dimensions of these matrices are:

H : nr x nc
U : nr x r
Σ : r x r
V : nc x r
r = min(nr,nc)

For a k-th dimensional realization only the first k singular values are taken into account. Because the number of singular values have no influence on the following the index k will be omitted.
G is approximated by:

$$G = \Sigma^{1/2}\ V^{T} \tag{A1.3}$$

with $\Sigma = \mathrm{diag}(\delta_1,..,\delta_k,0,\ ..\ 0)$

By substituting (A1.3) into (A1.2) :

$$A = (\Sigma^{1/2}\ V^{T})s\ \ V\ \Sigma^{-1/2}$$

$$= \left| (\Sigma^{1/2}\ V^{T})s \ \vdots\ \emptyset \right| V\ \Sigma^{-1/2} \tag{A1.4}$$

with $\Sigma^{-1/2} = (\Sigma^{+})^{1/2} = \mathrm{diag}(\delta_1^{-1},..,\delta_k^{-1},0,..,0)$

This will be proven to be identical to using the shifted Hankel matrix as defined in (2.20):

$$A^* = \Sigma^{-\frac{1}{2}}\, U^T\, Hs\, V\, \Sigma^{-\frac{1}{2}}$$

$$= \Sigma^{-\frac{1}{2}}\, U^T \left| \; (U\, \Sigma\, V^T)s \;\vdots\; \emptyset \; \right| V\, \Sigma^{-\frac{1}{2}} \qquad \text{(A1.5)}$$

Now follows the proof that A (A1.4) equals A^* (A1.5).

Proof

A satisfying condition for A (A1.4) to be equal to A^* (A1.5) is:

$$\left| \; (U\, \Sigma\, V^T)s \;\vdots\; \emptyset \; \right| = U\, \Sigma^{\frac{1}{2}} \left| \; (\Sigma^{\frac{1}{2}}\, V^T)s \;\vdots\; \emptyset \; \right| \qquad \text{(A1.6)}$$

Define :

$$\begin{aligned} U^* &= U\, \Sigma^{\frac{1}{2}} && (\ nr \times r\ ) \\ V^* &= \Sigma^{\frac{1}{2}}\, V^T && (\ r \times nc\ ) \end{aligned} \qquad \text{(A1.7)}$$

Then :

$$U\, \Sigma\, V^T = U^*\, V^* \qquad \text{(A1.8)}$$

If $\alpha_i := SQRT(\delta_i)$ then U^* and V^* can be written as:

$$U^* = \left| \begin{matrix} \underline{u}1^T \\ \underline{u}2^T \\ \vdots \\ \vdots \\ \underline{u}m^T \end{matrix} \right| \quad \text{with } \underline{u}i^T = \left| \; ui1.\alpha 1 \;\; ui2.\alpha 2 \;\ldots\; uir.\alpha r \; \right| \qquad \text{(A1.9)}$$

$$V^* = \left| \; \underline{v}1 \;\; \underline{v}2 \;\ldots\; \underline{v}n \; \right| \quad \text{with } \underline{v}i = \left| \begin{matrix} \alpha 1.vi1 \\ \alpha 2.vi2 \\ \vdots \\ \vdots \\ \alpha r.vir \end{matrix} \right|$$

If $U\, \Sigma\, V^T$ is expressed in these definitions then:

$$U\, \Sigma\, V^T = \left| \begin{matrix} \underline{u}1^T.\underline{v}1 & \underline{u}1^T.\underline{v}2 & \ldots\ldots & \underline{u}1^T.\underline{v}n \\ \underline{u}2^T.\underline{v}1 & & & \\ \vdots \;\; \vdots & \vdots & & \vdots \;\; \vdots \\ \underline{u}m^T.\underline{v}1 & \underline{u}m^T.\underline{v}2 & \ldots\ldots & \underline{u}m^T.\underline{v}n \end{matrix} \right| \qquad \text{(A1.10)}$$

The over one block column, or p columns to the left, shifted version of the matrix :

$$(U\ \Sigma\ V^T)s = \left|\begin{array}{ccc:c} \underline{u}1^T.\underline{v}2 & \cdots\cdots & \underline{u}1^T.\underline{v}n & \\ \vdots & & \vdots & \emptyset \\ \underline{u}m^T.\underline{v}2 & \cdots\cdots & \underline{u}m^T.\underline{v}n & \end{array}\right| \qquad \text{(A1.11)}$$

Equation (A1.11) is the left hand side of the desired equality (A1.6). The right hand side is examined now. Using the definitions of U^* and V^*.

$$U\ \Sigma^{\frac{1}{2}}\ (\Sigma^{\frac{1}{2}}\ V^T)s = U^*\ (V^*)s =$$

$$= \left|\begin{array}{c} \underline{u}1^T \\ \vdots \\ \underline{u}m^T \end{array}\right| \left|\ \underline{v}2\ \ldots\ \underline{v}n\ \vdots\ \emptyset\ \right| \qquad \text{(A1.12)}$$

$$= \left|\begin{array}{ccc:c} \underline{u}1^T.\underline{v}2 & \ldots & \underline{u}1^T.\underline{v}n & \\ \vdots & & \vdots & \emptyset \\ \underline{u}m^T.\underline{v}2 & \ldots & \underline{u}m^T.\underline{v}n & \end{array}\right|$$

Because equation (A1.12) is identical to equation (A1.11) the desired equality is proven to be true.
As a consequence the equivalence between the two descriptions for the realization method of Kung is proven.

Remark: If we would define $U^* = U\ \Sigma$ and $V^* = V^T$ it then follows that:

$$(U\ \Sigma\ V^T)s = U\ \Sigma\ (V^T)s \qquad \text{(A1.13)}$$

and by combining (A1.6) and (A1.13) :

$$(\ \Sigma^{\frac{1}{2}}\ V^T\ )s = \Sigma^{\frac{1}{2}}\ (V^T)s \qquad \text{(A1.14)}$$

APPENDIX 2.

Proof of the stability of the state space models computed according to the Kung realization variant.

In this appendix the stability of the realizations according to the method of Kung and Damen-Hajdasinski is examined. These methods are described in chapter 2.
It is proven that the realization method of Kung always renders a stable state space model.

The stability of a model is determined by the eigenvalues of the system matrix A.

The matrix A is computed according to (see chapter 2):

$$A = \Sigma^{-\frac{1}{2}} U^T Hs\ V\ \Sigma^{-\frac{1}{2}} \qquad \text{(A2.1)}$$

The matrix dimensions are:

A : k x k
Hs: nr x nc
Σ : r x r
U : nr x r
V : nc x r
r = min(nr,nc)
k ≤ r

with: $\Sigma^{-\frac{1}{2}} = (\Sigma^{+})^{\frac{1}{2}} = \text{diag}(\delta_1^{-1},\ldots,\delta_k^{-1},0,\ldots,0)$

shifted Hankel matrix :
Kung (see (2.20)) :

$$Hs = |\,(U\ \Sigma\ V^T)s \ \vdots\ \emptyset\ |$$

Damen-Hajdasinski (see (2.18)):

$$Hs = |\,(U\ \Sigma\ V^T)s \ \vdots\ Q\ |$$

with Q according to equation (2.18)

These two shifted Hankel matrices can be described as:

$$Hs = |\,(U\ \Sigma\ V^T)s \ \vdots\ P\ | \qquad \text{(A2.2)}$$

with Kung: P = $\emptyset$ (nr x p)
Damen: P = Q (nr x p)
p = number of inputs

Remark : In an approximate realization of dimension k only the (k x k) upper left block of the matrix computed in (A2.1) will contain non zero numbers.
As in appendix 1 the index, indicating a k-th dimensional realization, will be omitted.

The last block column is represented by a (nr x p) matrix P. The stability of matrix A will be examined using Hs of equation (A2.2). Substitution of equation (A2.2) in equation (A2.1) gives:

$$A = \Sigma^{-\frac{1}{2}} U^T \mid (U \Sigma V^T)s \;:\; P \mid V \Sigma^{-\frac{1}{2}} \qquad (A2.3)$$

Using appendix one:

$$\begin{aligned} A &= \Sigma^{-\frac{1}{2}} U^T \mid U \Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}} V^T)s \;:\; P \mid V \Sigma^{-\frac{1}{2}} \\ &= \mid \Sigma^{-\frac{1}{2}} U^T U \Sigma^{\frac{1}{2}} (\Sigma^{\frac{1}{2}} V^T)s \;:\; \Sigma^{-\frac{1}{2}} U^T P \mid V \Sigma^{-\frac{1}{2}} \\ &= \mid (\Sigma^{\frac{1}{2}} V^T)s \;:\; \Sigma^{-\frac{1}{2}} U^T P \mid V \Sigma^{-\frac{1}{2}} \qquad (A2.4) \end{aligned}$$

Define the matrices V1 ([nc-p] x r) and V2 (p x r) as sub blocks of the matrix V (nc x r):

$$V = \left| \begin{matrix} V1 \\ V2 \end{matrix} \right| \qquad (A2.5)$$

Then :

$$\begin{aligned} A &= \mid (\Sigma^{\frac{1}{2}} V^T)s \;:\; \Sigma^{-\frac{1}{2}} U^T P \mid \left| \begin{matrix} V1 \\ V2 \end{matrix} \right| \Sigma^{-\frac{1}{2}} \\ &= (\Sigma^{\frac{1}{2}} V^T)s \; V1 \; \Sigma^{-\frac{1}{2}} + \Sigma^{-\frac{1}{2}} U^T P \; V2 \; \Sigma^{-\frac{1}{2}} \\ &= A1 + A2 \qquad (A2.6) \end{aligned}$$

Interim conclusions :

1. The system matrix A can be computed as the summation of two matrices.
2. The matrix A1 is the system matrix according to the method of Kung.(P = Ø). For Damen-Hajdasinski A2 is added.

Next the stability of the realization variant of Kung is proven.

Proof

The system matrix of the realization method of Kung is given by:

$$A1 = (\Sigma^{\frac{1}{2}} V^T)s \; V1 \; \Sigma^{-\frac{1}{2}} \qquad (A2.7)$$

Using equation (A1.14) from appendix 1:

$$A1 = \Sigma^{\frac{1}{2}} (V^T)s \; V1 \; \Sigma^{-\frac{1}{2}} \qquad (A2.8)$$

This is equivalent to:

$$\Sigma^{-\frac{1}{2}} A1 \; \Sigma^{\frac{1}{2}} = (V^T)s \; V1 \qquad (A2.9)$$

Define V^{*T} as:

$$V^{*T} = \mid (V^T)s \;:\; Ø \mid \qquad (A2.10)$$

Then :

$$\Sigma^{-\frac{1}{2}} A1 \; \Sigma^{\frac{1}{2}} = V^{*T} V \qquad (A2.11)$$

The stability of the matrix A1 is guaranteed if all singular values, and thereby the eigenvalues [Kailath, 1980], of the left hand side of (A2.11) are less or equal one.
If we consider the euclidian norm :

$$\| \Sigma^{-\frac{1}{2}} A1 \Sigma^{\frac{1}{2}} \| = \| V^{*T} V \|$$
$$\leq \| V^{*T} \| \| V \| \qquad (A2.12)$$

Because the matrix V follows from the singular value decomposition of the Hankel matrix it is an orthonormal matrix, and consequently its euclidian norm equals one.
The norm of V^{*T} is the norm of a block from the matrix V.
According to Staar [1982] its euclidian norm will always be less or equal the euclidian norm of V.

So :

$$\| \Sigma^{-\frac{1}{2}} A1 \Sigma^{\frac{1}{2}} \| \leq 1 \qquad (A2.13)$$

Because of equation (A2.13) the stability of the matrix A1 and thereby the stability of the realization according to Kung is proven.

For examination of the stability of the realization variant according to Damen-Hajdasinski, the matrix A2 has to be taken into account too. Therefore the stability of the sum of two matrices has to be examined:

$$\| A \| = \| A1 + A2 \|$$
$$\leq \| A1 \| + \| A2 \| \qquad (A2.14)$$

Then one of the following conditions must be satisfied:

$$\| A1 + A2 \| \leq 1$$
$$\| A1 \| + \| A2 \| \leq 1 \qquad (A2.15)$$

It wasn't possible to prove (A2.15), nor didn't we find a way to determine the last block column in such a way that would approximate the original Hankel matrix in better way, and that would guarantee stability of the realization.

APPENDIX 3.

HANKDIM : AN EXAMPLE OF THE TEXT OUTPUT.

```
HANKDIM Application :  vrs. 1.0

----------Input data----------------------------------------------------
[PETERG.FEED2]MARKOV.MODEL;1          19OCT87   REPFLM1      394 records

----------Use of input data---------------------------------------------

Number of Markov parameters used:  65

Number of inputs  :  3          Number of outputs :  2
        Input  1  : U1                  Output  1  : Y1
        Input  2  : U2                  Output  2  : Y2
        Input  3  : U3

---------- Parameters ------------------------------------------------
Construction method Hankel matrix : BLOCK_SQ

Output dataset title              : 3X2_BL65
```

APPENDIX 4.

HANKDIM2 : AN EXAMPLE OF THE TEXT OUTPUT.

```
HANKDIM2 Application :  vrs. 1.0

----------Input data----------------------------------------------------
[PETERG.FEED2]MARKOV.MODEL;1          19OCT87    REPFLM1        394 records

----------Use of input data---------------------------------------------

Number of Markov parameters used:  65

Number of inputs  :  3          Number of outputs :  2
        Input  1  : U1                  Output  1  : Y1
        Input  2  : U2                  Output  2  : Y2
        Input  3  : U3

---------- Parameters ------------------------------------------------
Construction method HB-matrix : BLOCK_SQ

Output dataset title          : 2I3X2_65
```

APPENDIX 5.

HANKEL : AN EXAMPLE OF THE TEXT OUTPUT.

a. on input: > dataset containing impulse response parameters
> dataset containing input-output data

```
HANKEL Application : REALIZATION  vrs. 1.0

----------Input data----------
[PETERG.FEED3]MARKOV.MODEL;1          19OCT87   REPF51        196 records
[PETERG.FEED1]FILTER.DATA;1           19OCT87   REPTRD        912 records

--------------------Input model----------

Selected model    : SYSTEM

Number of inputs :    3                  Number of outputs :   1
Model type       :   MFD
Markov parameters:    65

-------------------- Model in- and output ----------
Number of inputs  :  3                   Number of outputs :  1
        Input  1  : U1                           Output  1  : Y1
        Input  2  : U2
        Input  3  : U3

-------------------- Reference signals ----------
Input signal   1  : G1             Ref. signal output   1  : F51
Input signal   2  : CA
Input signal   3  : G2


----------Use of Input/Output VECTOR data set----------
Number of samples :   912
Start sample      :     1          Start sample statistics :    92

----------Use of Markov model----------

Markov parameters used for computation :  65
Markov parameters used for comparison  :  65


---------- Parameters ----------

Output dataset title              : I65B0912
Realization method                : BOSGRA
Construction method Hankel matrix: BLOCK_SQ

----------Realization dimensions----------

Lowest realization dimension :   1
Dimension step               :   1
Highest realization dimension:   9
```

Final Results

Dimensions of the Hankel matrix

Number of block rows : 32
Number of block columns : 33
Number of rows : 32
Number of columns : 99

Singular values of the Hankel matrix

Number of singular values : 32

0.49101D+01	0.52482D-01	0.31398D-01	0.22928D-01
0.10374D+01	0.49970D-01	0.30718D-01	0.20861D-01
0.45223D+00	0.38815D-01	0.30484D-01	0.20422D-01
0.20563D+00	0.37160D-01	0.29427D-01	0.19977D-01
0.86970D-01	0.36339D-01	0.28848D-01	0.19128D-01
0.69343D-01	0.35236D-01	0.27497D-01	0.13988D-01
0.53801D-01	0.33856D-01	0.27317D-01	0.13173D-01
0.52877D-01	0.33049D-01	0.24763D-01	0.12761D-01

Relative output simulation errors (%)

Dimension	Output 1
1	33.45
2	15.85
3	2.19
4	1.74
5	1.64
6	1.39
7	1.40
8	1.39
9	1.40

The dimension is chosen according to the AIC4 criterion.

Chosen model dimension : 4

The eigenvalues of the realization

Model is stable.

Number	Real part			Imaginary part	Absolute value
1	0.71086D+00	+	j	0.90638D-01	0.71662D+00
2	0.71086D+00	−	j	0.90638D-01	0.71662D+00
3	0.95981D+00	+	j	0.37297D-01	0.96054D+00
4	0.95981D+00	−	j	0.37297D-01	0.96054D+00

Absolute and relative root mean square errors of the impulse responses

Input	Output	Absolute error	Relative error (%)
1	1	0.53463D-02	9.66
2	1	0.67641D-02	12.59
3	1	0.61812D-02	3.68

b. on input dataset containing impulse response parameters.

```
HANKEL Application : REALIZATION  vrs. 1.0

----------Input data----------------------------------------------------------
[PETERG.FEED3]MARKOV.MODEL:1          19OCT87   REPF51        196 records

--------------------Input model---------------------------------------------

Number of inputs :    3                    Number of outputs :   1
Model type       :   MFD
Markov parameters:    65

----------Use of Markov model-------------------------------------------------

Selected system : SYSTEM

Markov parameters used for computation :  50
Markov parameters used for comparison  :  65

Number of inputs  :  3          Number of outputs :  1
        Input  1  : U1                  Output  1  : Y1
        Input  2  : U2
        Input  3  : U3


---------- Parameters ---------------------------------------------------------

Construction method Hankel matrix : BLOCK_SQ

Realization dimension :    4
Realization method    : BOSGRA
Output dataset title  : 3X1_D4
```

Final Results

Dimensions of the Hankel matrix

Number of block rows : 25
Number of block columns : 25
Number of rows : 25
Number of columns : 75

Singular values of the Hankel matrix

Number of singular values : 25

0.46537D+01	0.46011D-01	0.25481D-01	0.16484D-01
0.92743D+00	0.37775D-01	0.24564D-01	0.14823D-01
0.44443D+00	0.33685D-01	0.23870D-01	0.14481D-01
0.11751D+00	0.31481D-01	0.22567D-01	0.12470D-01
0.67402D-01	0.28793D-01	0.18623D-01	
0.50863D-01	0.28691D-01	0.18126D-01	
0.47364D-01	0.26968D-01	0.17486D-01	

The eigenvalues of the realization

Model is stable.

Number	Real part			Imaginary part	Absolute value
1	0.70795D+00	+	j	0.76806D-01	0.71210D+00
2	0.70795D+00	-	j	0.76806D-01	0.71210D+00
3	0.95952D+00	+	j	0.36535D-01	0.96021D+00
4	0.95952D+00	-	j	0.36535D-01	0.96021D+00

Absolute and relative root mean square errors of the impulse responses

Input	Output	Absolute error	Relative error (%)
1	1	0.74270D-02	13.42
2	1	0.87755D-02	16.33
3	1	0.62056D-02	3.69