TR R165–94
Juni 1994

Some Experience with the MOESP Class of Subspace Model Identification Methods in identifying the BO105 Helicopter

Michel Verhaegen
Andras Varga

DLR Oberpfaffenhofen
German Aerospace Research Establishment
Institute for Robotics and System Dynamics
Control Design Engineering Group (Prof. G. Grübel)
D–82234 Wessling

Abstract

This technical note reports on some of the experience gained with analyzing some of the flight test data of the BO105 Helicopter. This data was supplied by J. Kaletka of the DLR Braunschweig and served as a benchmark example in the AGARD Working group WG-18. The results reported here show that the MOESP [1] approach allows to reliably detect the order and structure of the underlying linear state space model, to concatenate different individually collected data sets and to accurately reconstruct the output sequences using the identified model.

1. Remarks on the data acquisition and commonly used identification approaches.

The data analyzed in this brief technical note corresponds to some of flight test data used as a benchmark example in the AGARD Working Group WG-18. The purpose of this WG was to evaluate and compare the at that time current state of the art identification tools in identifying Rotorcraft systems, such as helicopters. In this brief note, we restrict to the flight test data that was kindly supplied by Mr. J. Kaletka of the DLR in Braunschweig and that was recorded during flight tests with the BO105 helicopter. The data set included 2 sets of four data files. Each set corresponds to a particular type of input excitation used, namely frequency sweep inputs and the 3211 DLR Braunschweig input [J. Kaletka, 1991]. Each set then further consists of four data files, each characterizing the excitation of one particular control input. An overview of the available data is given in Table 1.

TYPE OF INPUT	EXCITED CONTROL INPUT
Set 1 frequency sweep input	longitudinal stick lateral stick pedal collective
Set 2 3211 DLR Braunschweig input	longitudinal stick lateral stick pedal collective

Table 1: Overview of the different analyzed data sets.

Of the recorded data sequences all four input quantities were selected as inpnt of the system to be identified, but we only selected the qnantities listed in Table 2 as output quantities.

About the recorded output quantities we remark that the airspeed components used in the identification process were not those obtained by direct measurement but those reconstrncted from inertial data [J. Kaletka, 1991].

For an extensive discussion of the features characterizing this data, we refer to [J. Kaletka, 1991]. Here we only restrict attention to the following *three features*, which are relevant to our present study:

1. Accuracy of the measurements: It was indicated in [J. Kaletka, 1991], that "*accurate mea-*

[1] The abbreviation MOESP stands for Multivariable Output-Error State sPace Subspace Model Identification approach and was originally developed in [Verhaegen and Dewilde, 1992a, Verhaegen and Dewilde, 1992b, Verhaegen, 1993a, Verhaegen, 1993c] and [Verhaegen, 1994a].

QUANTITY	SYMBOL	UNIT
Longitudinal component of the airspeed	u	m/s
Lateral compenent of the airspeed	v	m/s
Vertical component of the airspeed	w	m/s
Roll angle	Φ	rad
Pitch angle	Θ	rad
Roll rate	p	rad/s
Pitch rate	q	rad/s
Yaw rate	r	rad/s

Table 2: List of the output quantities used in the idenfitication runs.

surements are an indispensable prerequisite for reliable system identification". Inspection of the input data sequences gives the impression that some of the sequences are heavily corrupted by uoise, probably due to the vibration of the helicopter. We illustrate this assertion with a plot of the input data records present in **set 2** — longitudinal input (see Table 1) in figure 1. Especially for the pedal and collective input the records look extremely noisy. When these quantities do not correspond to the actual system input, but need to be considered as measurement errors we have the so-called errors-in-variables problem in system identification. This type of problem generally results in biased estimates in most commonly used identification schemes when no special pre-cautions are taken into account. These pre-cautions generally require the assumption that the errors are white noise sequences of known variance.

2. Multiple or concatenated data sequences: Due to the helicopter *instabilities* it is typical that the length of a single data run is limited. As a consequence, the data set does only contain a short period of oscillatory modes which have long time constants, such as e.g. the phugoid mode. Therefore, it was indicated in [J. Kaletka, 1991] that the identification method should be able to identify a *single* model from different test runs. This is known as *multiple* or *concatenated* data sequence analysis. However, when applying well established parameter optimization schemes, such as those described in [AGARD LS-178], to these concatenated data sets, care needs to be taken such that all data sets have *practically the same* initial flight test condition and helicopter and instrumentation status.

3. Decreasing the sample rate: Even when limiting the number of inputs to 4 and the number of outputs to 8, as indicated in table 2. the number of measurements is extremely large. Therefore, in order to allow the data to be incorporated within MATLAB [Moler et al, 1987], even when using a powerful workstation, the number of data points need to be reduced. This is achieved by lowering the sample rate to $0,05$ seconds instead of the sampling rate of 0.01 seconds used during the collection of the data. For this sampling period, it was assumed that the direct feedthrough matrix of the quadruple of system matrices is equal to zero.

In [AGARD LS-178], a number of critical problems that may occur in the analysis of the helicopter data have been listed. Of these problems, we draw attention to the following two:

1. Treatment of linear time-invariant (LTI) models: Though the coupled six degree of freedom rigid body model of the helicopter is inherently non-linear, most of the methods

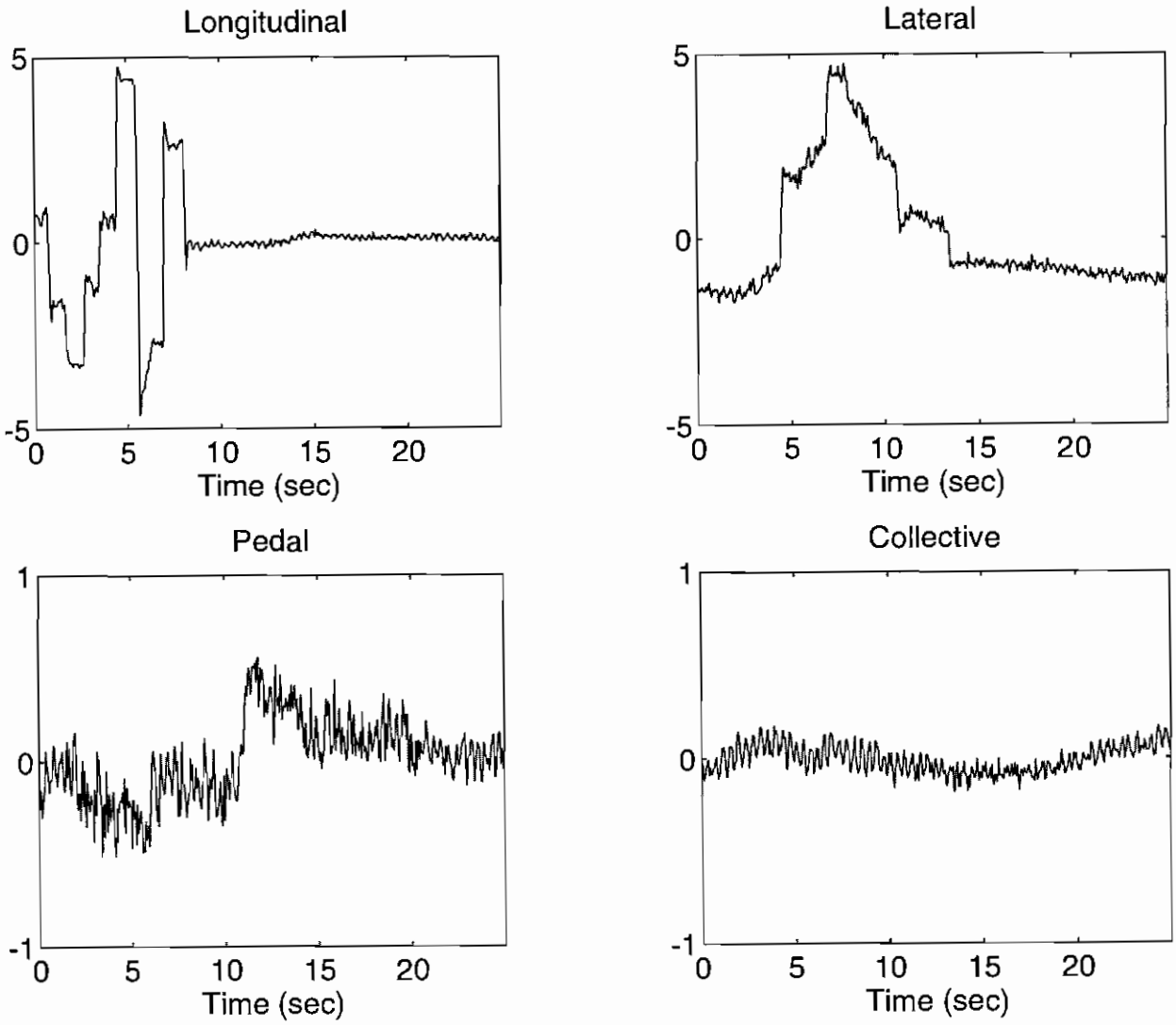

Figure 1: Input data records belonging to set 2 — longitudinal excitation (see Table 1).

analyzed in [AGARD LS-178] treated the LTI identification problem. This restriction will also be imposed in the present note.

2. The model structure: The methods evaluated in [AGARD LS-178] belong to the class of parametric model identification (PMI) schemes. The main feature characterizing this class of identification methods is that they optimize a performance index, generally a suitable chosen norm of the residuals, with respect to a priori selected parameters in a particular model. A selected set of parameters is called the model structure. That this is not a trivial problem at all was clearly pointed out in [[J. Kaletka, 1991], p. 9-7.]. The main problem is that one can not identify all the parameters in a (continuous) state representation of fixed (a priori selected) order. Therefore, a decision needs to be made on which parameters should be fixed and at what value. The latter values may be either zero or the magnitudes determined for example during wind tunnel measurements. However, that this is a very delicate problem is illustrated by the large variations of the number of parameters that were estimated by the different teams cooperation in the AGARD WG-18. More precisely the number varied from 58 to 30. It is remarked that since on the one hand this structure selection process consumes a significant part of the total analysis time of the identification

problem and on the other hand critically may influence the outcome of the identification process, the proposal of a simple and robust way to select the model structure will be a great asset to the (aeronautical) identification community.

2. Basic aspects of the MOESP class of SMI schemes.

The goal of the present study is to apply the recently developed class of MOESP Subspace Model Identification (SMI) schemes to the data sets described in the previous section. Before doing that, we first recall and extend the main features of one variant of the MOESP class, namely the so-called PO scheme. This abbreviation stands for the ordinary MOESP scheme extended with instrumental variables constructed from Past Output quantities. The scheme was originally proposed in [Verhaegen, 1994a] and we refer the interested reader to this publication for a derivation of this scheme and an analysis of the consistency in identifying state space models given in so-called innovation form. In this brief note, we restrict to recalling the following features of this schemes:

- the model structure selection
- the key theorem describing the operation of the PO scheme
- the calculation of the pair $[A, C]$ of the quadruple of system matrices $[A, B, C, D]$ of a discrete LTI state space model
- the calculation of the pair $[B, D]$
- Concatenating data sets and dealing with non-zero initial conditions
- an alternative way to compute the pain $[B, D]$.

In the following subsections, we will subsequently treat these items.

2.1. Model structure selection.

Given the linear state space model:

$$x_{k+1} = Ax_k + Bu_k \quad \text{with} \quad x_k \in \mathbb{R}^n,\ u_k \in \mathbb{R}^m \tag{1}$$

$$y_k = Cx_k + Du_k \quad \text{with} \quad y_k \in \mathbb{R}^\ell \tag{2}$$

and the input-output (i/o) data records:

$$\{u_j\ u_{j+1}\ \cdots\ u_{j+N-1}\} \quad \{y_j\ y_{j+1}\ \cdots\ y_{j+N-1}\} := \{y_k\}_{k=j}^{j+N-1}$$

the following structured relationship holds between the block-Hankel matrices $U_{j,s,N}$ and $Y_{j,s,N}$ constructed from the i/o data:

$$Y_{j,s,N} = \Gamma_s X_{j,N} + H_s U_{j,s,N} \tag{3}$$

In this equation, the following quantities are defined:

$$U_{j,s,N} = \begin{pmatrix} u_j & u_{j+1} & \cdots & u_{j+N-1} \\ u_{j+1} & u_{j+2} & \cdots & u_{j+N} \\ \vdots & & \ddots & \\ u_{j+s-1} & u_{j+s} & \cdots & u_{j+N+s-2} \end{pmatrix}$$

(and a similar definition of the block-Hankel matrix $Y_{j,s,N}$),

$$H_s = \begin{pmatrix} D & 0 & & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & & 0 \\ \vdots & & & \ddots & \vdots \\ CA^{s-2}B & & & \cdots & D \end{pmatrix} \quad \Gamma_s = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{pmatrix}$$

and the state sequence $\begin{pmatrix} x_j & x_{j+1} & \cdots & x_{j+N-1} \end{pmatrix}$. The triple of indices (j, s, N) of the Hankel matrices determine their size and what part of the i/o sequences is stored in them. More precisely, the first index j refers to the time index of the first element of the i/o record, the second index determines the number of block rows of the Hankel matrix and the third index is the number of columns.

Eq. (3) is a model representation in the true sense of relating input and output sequences. The local state space model (1-2) is implicitly contained in this *global* model representation. The structure of this global model, referred to as the data equation in [Verhaegen and Dewilde, 1992a], is determined by **only** two parameters:

1. the Hankel matrix dimension parameter s, briefly indicated by the *SMI dimensioning parameter*. When the length of the i/o record is fixed, it fully determines the size (i.e. number of rows and columns) of the Hankel matrices.

2. the order n of the state space model. This parameter determines the number of columns of the extended observability matrix Γ_s (and hence also the number of rows of the state sequence $X_{j,s}$).

Given a particular *i/o* sequence, the model structure selection problem for the SMI schemes analyzed in this note corresponds to a selection of an appropriate pair (s, n). In Theorem 1, it is clarified that essentially only one parameter needs to be specified by the user, namely the SMI dimensioning parameter s. A selection of this parameter only requires a rough estimate of the underlying system order.

2.2. The Basic Theorem of the PO scheme.

The PO scheme addresses the identification of MIMO state space models given in the so-called innovation form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ z_k &= Cx_k + Du_k + v_k \end{aligned} \tag{4}$$

where the process noise w_k and measurement noise v_k are zero-mean white noise sequences, which are statistically independent from the deterministic input u_k and have the covariance

matrix:

$$E[\begin{pmatrix} w_k \\ v_k \end{pmatrix}(w_j^T \;\; v_j^T)] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \quad \text{for } k=j$$
$$= 0 \quad \text{for } k \neq j$$

For this class of systems, we have the following Theorem. The Theorem was stated in [Verhaegen, 1994a] and we refer the reader interested in its proof to consult this publication.

Theorem 1 *Let the process noise w_k and the measurement noise ν_k of the system (4) be discrete zero-mean white noise, independent of the input u_j for all k, j and of the initial state x_0, let the input u_k to the system (1-2) be sufficiently persistently exciting [2] and let the following RQ factorization be given:*

$$\begin{pmatrix} U_{s+1,s,N} \\ U_{1,s,N} \\ Z_{1,s,N} \\ Z_{s+1,s,N} \end{pmatrix} = \begin{pmatrix} R_{11}^N & & & \\ R_{21}^N & R_{22}^N & & \\ R_{31}^N & R_{32}^N & R_{33}^N & \\ R_{41}^N & R_{42}^N & R_{43}^N & R_{44}^N \end{pmatrix} \begin{pmatrix} Q_1^N \\ Q_2^N \\ Q_3^N \\ Q_4^N \end{pmatrix} \tag{5}$$

then:

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Z_{s+1,s,N}(Q_2^N)^T = \lim_{N\to\infty} \frac{1}{\sqrt{N}} \Gamma_s X_{s+1,N}(Q_2^N)^T \tag{6}$$

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Z_{s+1,s,N}(Q_3^N)^T = \lim_{N\to\infty} \frac{1}{\sqrt{N}} \Gamma_s X_{s+1,N}(Q_3^N)^T \tag{7}$$

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Z_{1,s,N}(Q_1^N)^T = \lim_{N\to\infty} (\frac{1}{\sqrt{N}} \Gamma_s X_{1,N}(Q_1^N)^T + \frac{1}{\sqrt{N}} H_s(R_{21}^N)) \tag{8}$$

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Z_{1,s,N}(Q_2^N)^T = \lim_{N\to\infty} (\frac{1}{\sqrt{N}} \Gamma_s X_{1,N}(Q_2^N)^T + \frac{1}{\sqrt{N}} H_s(R_{22}^N)) \tag{9}$$

$$\lim_{N\to\infty} \frac{1}{\sqrt{N}} Z_{s+1,s,N}(Q_1^N)^T = \lim_{N\to\infty} (\frac{1}{\sqrt{N}} \Gamma_s X_{s+1,N}(Q_1^N)^T + \frac{1}{\sqrt{N}} H_s(R_{11}^N)) \tag{10}$$

□

The matrices $Z_{j,s,N}$ in the RQ factorization are the Hankel-matrices constructed from the outputs z_k of the model (4).

From this Theorem the equations to compute (estimate) the quadruple of system matrices $[A, B, C, D]_T$ (up to a similarity transformation T) have been derived in [Verhaegen, 1994a]. A summary of these equations is given in the next two subsections.

2.3. Computing the pair $[A, C]_T$.

Let us consider the SVD of the compound matrix $\begin{bmatrix} R_{42}^N & R_{43}^N \end{bmatrix}$, i.e.:

$$\frac{1}{\sqrt{N}} \begin{bmatrix} R_{42}^N & R_{43}^N \end{bmatrix} = \ell_s \left(\begin{array}{c|c} \overset{n}{U_n} & \overset{\ell s-n}{U_n^\perp} \end{array} \right) \left(\begin{array}{c|c} \overset{n}{S_n} & \overset{\ell s-n}{0} \\ \hline 0 & S_2 \end{array} \right) \begin{pmatrix} (V_n)^T \\ (V_n^\perp)^T \end{pmatrix} \tag{11}$$

[2] The notion of persistency of excitation in the context of SMI is defined in [Verhaegen and Dewilde, 1992a]. Roughly speaking it states that the inpnt has properly excited the controllable and observable modes of the system.

then Eqs. (6-7) of Theorem 1 show that when the input is sufficiently persistently exciting, which guarantees the matrix $X_{s+1,N}\left[\begin{array}{cc}(Q_2^N)^T & (Q_3^N)^T\end{array}\right]$ to be of rank n, and when

$$\boxed{s > n}, \tag{12}$$

the column space of the matrix U_n is a consistent estimate of that of Γ_s. The constraint in Eq. (12) indicates that s can be specified based on a rough estimate of n. When s is properly selected, the extended observability matrix has rank n, and as a consequence their will be n non-zero singular values in the SVD in Eq. (11) when $N \to \infty$. This shows that once s is specified by the user, the algorithm supplies him with information on the selection of the order of the underlying system.

The knowledge of the column space of the extended observability matrix Γ_s allows to compute the system matrices A_T and C_T exploiting the shift structure of this matrix as follows:

$$U_n(1:\ell(s-1),:)A_T = U_n(\ell+1:\ell s,:) \tag{13}$$

$$C_T = U_n(1:\ell,:) \tag{14}$$

In these two equations we have adopted the standard MATLAB [3] notation to select a subpart of a matrix [Moler et al, 1987].

2.4. Computing the pair $[B_T, D]$.

Using the corresponding matrices in the R-factor of the RQ factorization in Theorem 1, the Eqs. (8-10) of Theorem 1 are denoted compactly as:

$$\begin{aligned}\frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{31}^N & R_{32}^N & R_{41}^N\end{array}\right] &= \Gamma_s \frac{1}{\sqrt{N}}\left[\begin{array}{ccc}X_{1,N}(Q_1^N)^T & X_{1,N}(Q_2^N)^T & X_{s+1,N}(Q_1^N)^T\end{array}\right] \\ &+ H_s \frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{21}^N & R_{22}^N & R_{11}^N\end{array}\right] + O_N(\epsilon)\end{aligned} \tag{15}$$

where $O_N(\epsilon)$ is a matrix of appropriate dimensions and of ϵ-norm for finite N and vanishing when $N \to \infty$.

When the column spaces of the matrix Γ_s and its orthogonal complement are respectively equal to that of U_n and $U_n^\perp$, then we can multiply Eq. (15) on the left by $(U_n^\perp)^T$, and obtain:

$$(U_n^\perp)^T \frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{31}^N & R_{32}^N & R_{41}^N\end{array}\right] = (U_n^\perp)^T H_s \frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{21}^N & R_{22}^N & R_{11}^N\end{array}\right] + O_N(\epsilon) \tag{16}$$

Caused by the sufficiently persistence of excitation of the input, the matrix $\left[\begin{array}{ccc}R_{21}^N & R_{22}^N & R_{11}^N\end{array}\right]$ has a right pseudo-inverse. Multiplying Eq. (16) on the right by this inverse, yields:

$$(U_n^\perp)^T \frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{31}^N & R_{32}^N & R_{41}^N\end{array}\right] \left(\frac{1}{\sqrt{N}}\left[\begin{array}{ccc}R_{21}^N & R_{22}^N & R_{11}^N\end{array}\right]\right)^\dagger = (U_n^\perp)^T H_s + O_N(\epsilon)$$

If we denote the left hand side of this equation by $\Xi \in R^{(\ell s-n)\times m_1 s}$, then this equation can be written as:

$$\Xi = (U_n^\perp)^T H_s + O_N(\epsilon) \tag{17}$$

[3] MATLAB® is a registered trademark of Mathworks, Inc.

As indicated in [Verhaegen and Dewilde, 1992a], this is an equation which is linear in the matrix pair (B_T, D) once the columns space of Γ_s is known. To see this, we rewrite this equation as (see Eq. (45) of [Verhaegen and Dewilde, 1992a]):

$$\begin{array}{c} \ell s-n \\ \ell s-n \\ \vdots \\ \ell s-n \end{array} \begin{pmatrix} \overset{m}{\Xi(:,1:m)} \\ \Xi(:,m+1:2m) \\ \vdots \\ \Xi(:,m(s-1)+1:ms) \end{pmatrix} =$$

$$\underbrace{\begin{pmatrix} \overset{\ell}{U_n^\perp(\ell(s-1)+1:\ell s,:)^T} & \cdots & & \overset{\ell}{U_n^\perp(1:\ell,:)^T} \\ 0 & U_n^\perp(\ell(s-1)+1:\ell s,:)^T & \cdots & U_n^\perp(\ell+1:2\ell,:)^T \\ 0 & 0 & & \vdots \\ 0 & \cdots & 0 & U_n^\perp(\ell(s-1)+1:\ell s,:)^T \end{pmatrix}}$$

$$\times \left(\begin{array}{c|c} U_n((s-2)\ell+1:(s-1)\ell,:) & 0 \\ \vdots & \vdots \\ \hline U_n(1:\ell,:) & 0 \\ \hline 0 & I_\ell \end{array}\right) \begin{pmatrix} B_T \\ D \end{pmatrix} \qquad (18)$$

As mentioned in [Verhaegen, 1994a], the number of operations required in constructing this set of equations is proportional to s^3. In this technical note, we briefly discuss an alternative implementation that reduces the computational complexity (of constructing the set of equations alone) by a factor s. This more efficient implementation is based on the following two ideas:

1. First we compress the matrix $\left[\begin{array}{cccc} U_n^\perp(\ell(s-1)+1:\ell s,:)^T & \cdots & U_n^\perp(\ell+1:2\ell,:)^T & U_n^\perp(1:\ell,:)^T \end{array}\right]$ to an upper-trapezoidal matrix by means of orthogonal transformation Q. This is denoted as:

$$Q^T \left[\begin{array}{cccc} U_n^\perp(\ell(s-1)+1:\ell s,:)^T & \cdots & U_n^\perp(\ell+1:2\ell,:)^T & U_n^\perp(1:\ell,:)^T \end{array}\right] =$$

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} & \cdots & r_{1s} \\ 0 & r_{22} & \cdots & r_{2k} & \cdots & r_{2s} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & r_{kk} & \cdots & r_{ks} \end{bmatrix}$$

where $r_{ij} \in \mathbb{R}^{\ell \times \ell}$ for $i \neq k$ and $r_{kj} \in \mathbb{R}\ \mu \times \ell$ with $\mu = min_k(\ell s - n - (k-1)\ell)$ under the constraint that $(\ell s - n - (k-1)\ell) > 0$. This orthogonal transformation can be applied to all the block-rows of Eq. (18), and reduces the underbraced matrix in that equation to a

matrix of the following form (illustrated for $s = 4, k = 3$):

$$\left[\begin{array}{cccc}
r_{11} & r_{12} & r_{13} & r_{14} \\
0 & r_{22} & r_{23} & r_{24} \\
0 & 0 & r_{33} & r_{34} \\
\hline
0 & r_{11} & r_{12} & r_{13} \\
0 & 0 & r_{22} & r_{23} \\
0 & 0 & 0 & r_{33} \\
\hline
0 & 0 & r_{11} & r_{12} \\
0 & 0 & 0 & r_{22} \\
0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & r_{11} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{array}\right] \tag{19}$$

It is remarked that all the matrices r_{ii} have full row rank.

2. Second we compress the matrix in Eq. (19) to an upper triangular matrix (again using orthogonal transformations). Here we exploit both the already close to upper triangular structure of this matrix as well as the repetition of the different submatrices. This is again outlined for the example illustrated in Eq. (19). We first start off to compress the matrices in the full rectangular boxes to an upper triangular matrix. Since in all three cases, these matrices are the same, the computations need to be performed only once. Second we continue with the dashed boxes and finally the dashed-dotted box. This way of operating on the data both reduces the computational complexity and the storage. The computed orthogonal transformations need to be applied to the left hand side of Eq. (18).

Apart from efficiently constructing the set of equations (18), the above procedure will also speed up the actual (least-squares) solution of this set of equations.

2.5. Concatenating data sets and dealing with initial conditions.

From the model representation (3), we immediately conclude that this representation holds for arbitrary non-zero initial conditions. Therefore, non-zero initial conditions have *no effect* at all on the calculations of the quadruple $[A, B, C, D]_T$.

Also concatenating different data sets introduces no additional problems. This is illustrated for two different data batches, namely:

$$\{u_k^1, y_k^1\}_{k=1}^{N_1} \quad \text{and} \quad \{u_k^2, y_k^2\}_{k=1}^{N_2}$$

With the first i/o data set, we obtain the data equation:

$$Y_{1,s,N_1-s+1}^1 = \Gamma_s X_{1,N_1-s+1}^1 + H_s U_{1,s,N_1-s+1}^1$$

and with the second i/o data set, we have:

$$Y_{1,s,N_2-s+1}^2 = \Gamma_s X_{1,N_2-s+1}^2 + H_s U_{1,s,N_2-s+1}^2$$

The Hankel matrices in these two data equations are non-empty when both N_1 and N_2 are larger then or equal to s ($2s$ for the PO scheme). Both data equations can easily be combined

as follows:

$$\left[\begin{array}{c|c} Y^1_{1,s,N_1-s+1} & Y^2_{1,s,N_2-s+1} \end{array} \right] = \Gamma_s \left[\begin{array}{c|c} X^1_{1,N_1-s+1} & X^2_{1,N_2-s+1} \end{array} \right] + H_s \left[\begin{array}{c|c} U^1_{1,s,N_1-s+1} & U^2_{1,s,N_2-s+1} \end{array} \right]$$

The structure of this equation is equal to that of the original data equation (3) apart from the fact that the concatenated Hankel matrices of the i/o data are no-longer Hankel. However, since this property is not exploited in the derivation of the PO algorithm, we still can use the main body of this algorithm when starting with an RQ factorization of the following matrix:

$$\left[\begin{array}{c|c} U^1_{s+1,s,N_1-2s+1} & U^2_{s+1,s,N_2-2s+1} \\ U^1_{1,s,N_1-2s+1} & U^2_{1,s,N_2-2s+1} \\ Y^1_{1,s,N_1-2s+1} & Y^2_{1,s,N_2-2s+1} \\ Y^1_{s+1,s,N_1-2s+1} & Y^2_{s+1,s,N_2-2s+1} \end{array} \right]$$

2.6. An alternative way to compute the pair $[B_T, D]$ and the initial conditions.

It has been observed experimentally [Verhaegen and Babuska, 1994b] that when the first s Markov parameters of the system to be identified are small in magnitude in combination with the presence of modes with large time constants, the calculation of the matrix pair $[B_T, D]$ may become sensitive. Such (critical) condition also hold for the identification problem of the BO105 helicopter. Though the true nature of this problem is not yet fully understood, it was observed in the experiments performed in [Verhaegen and Bahuska, 1994b] that another way of computing this pair yields more accurate results under such circumstances. This alternative way is presented in this section under the assumption that the matrix D equals zero.

Let the i,j-th entry of the matrix B be denoted by b_{ij}, then we can rewrite the state equation (1) as:

$$x_{k+1} = Ax_k + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u_k(1)b_{11} + \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} u_k(1)b_{21} + \cdots + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} u_k(1)b_{n1} + \cdots +$$

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} u_k(m)b_{nm}$$

Here $u_k(i)$ denotes the i-th component of the vector u_k.

Based on this full representation, we define the following subsystems:

$$\begin{aligned} x^{ij}_{k+1} &= Ax^{ij}_k + {}_i \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} u_k(j) \\ y^{ij}_k &= Cx^{ij}_k \end{aligned}$$

The output of these subsystems can be computed (assuming zero initial conditions) once the

pair (A, C) is known. The relationship between these outputs and the output of the original state space model (1-2) is:

$$y_k = CA^{k-1}x_0 + \left[\begin{array}{cccc|cccc|c} y_k^{11} & y_k^{21} & \cdots & y_l^{n1} & y_k^{22} & \cdots & y_k^{n2} & \cdots \end{array}\right] \left[\begin{array}{c} b_{11} \\ b_{21} \\ \vdots \\ b_{n1} \\ \hline b_{12} \\ \vdots \\ b_{n2} \\ \hline \vdots \end{array}\right]$$

Again we observe that the entries of the matrix B and the initial conditions appear linear in this equation. Therefore, they can be estimated by solving a linear least squares problem. Three remarks are in place here:

1. Solving for the unknown elements of the B-matrix via the above outlined least-squares problems will yield consistent estimates when the true pair (A, C) is used in the calculations for the case the underlying system is in state space innovation form. When we know this pair up to a similarity transformation then the result also holds for the entries of a matrix B_T for the same similarity transformation.

2. The simple formulation of estimating the entries of B via the solution of a linear least squares problem allows to incorporate numerical reliable column pivoting strategies to improve the condition number of the least squares and therefore also of the solution of that problem.

3. The above results can be extended to cope with concatenated data sets. Of course in that case we have to take possibly different initial conditions into account.

3. Some experimental results.

The PO scheme discussed in the previous section is applied to the data sets described in section 1. The only data pre-processing performed on the data is detrending, that is to remove the non-zero mean from all the data records. The results discussed are:

1. The selection of the model structure.

2. The reconstruction of the output sequences using the estimated state space quadruple $[\hat{A}, \hat{B}, \hat{C}, 0]_T$ and the recorded input quantities.

3. The translation of the estimated discrete time model to the continuous time case with the physically defined quantities as state variables.

3.1. Structure selection.

The selection of the SMI dimensioning parameter s is based on the inequality listed in Eq. (12). For the present example a good indication of the order of the underlying system can be derived from "first principles". Based on these principles, the six degrees of freedom motion of the

helicopter can be described by an 8-th order (continuous) state space model. From this rough estimate of the order, the SMI dimensioning parameter was set to:

$$\boxed{s = 15}$$

The second parameter determining the model structure is the order of the (discrete) state space model. As outlined in Theorem 1, in the limit $N \to \infty$, the order of the underlying system is equal to the number of non-zero singular values in the SVD (11) when only process and measurement noise are present. However for the helicopter their is great evidence that just more than that is present, for we have:

1. the high order dynamics of the vibration modes and
2. the non-linear dynamics.

When both phenomena are (statistically) independent from the input signals they will disappear in the limit in the same way the process noise and measurement noise disappear [Verhaegen, 1994a]. However, as we have mentioned in section 1, see figure 1, the input records are contaminated by high frequency components which are probably due to these high order vibration modes of the helicopter. Furthermore, it is also not realistic to assume independency between the non-linear dynamics and the input signals.

Therefore, we conclude that both conditions violate the assumptions stipulated in the derivation of the PO scheme. Therefore, we conclude that the helicopter data serves as a good test example to verify the robustness of this scheme.

To demonstrate the capabilities of the MOESP approach in handling concatenated data sets, we process all the available data batches listed in Table 1. This is done one after the other and the obtained singular values are plotted in Figures 2 and 3. More precisely, in Figure 2 we plot:

1. $\times$ - points: The singular values obtained after processing the data records in **Set 1** — longitudinal of Table 1.
2. o - points: The singular values obtained after processing the data records in **Set 1** — longitudinal and lateral of Table 1.
3. $+$ - points: The singular values obtained after processing the data records in **Set 1** — longitudinal, lateral and pedal of Table 1.

In Figure 3, we display the singular values obtained after processing of all the available data records.

From these figures we make the following observation:

1. When only processing the data records in **Set 1** — longitudinal of Table 1, we do not observe a clear gap between the 8-th and 9-th singular value. See Figure 2 ($\times$ - points). From this batch alone we would probably set the order equal to 4 or 5.
2. From the curve indicated by the $+$'s in Figure 2, we begin to observe a clear gap between the 8 and 9 singular value, a possible indication that the data "contains" an 8-th order *linear* system.

3. When processing all the available data batches, the gap between the 8-th and 9-th singular value has even become more transparent.

Based on these observations we may conclude that:

1. The PO scheme is *robust* with respect to the presence of (weak) non-linear and high order dynamics. This fact reflects the dominant mode extraction capabilities of the MOESP class of identification schemes, already highlighted in [Verhaegen and Dewilde, 1992a].

2. The PO scheme allows to detect the correct order of the six degree of freedom (rigid body) model of the helicopter. In addition it shows that the used input sequences are adequate (maybe not optimal) to identify this part of the underlying system.

3. The concatenation of data batches is necessary to identify the six degree of freedom model.

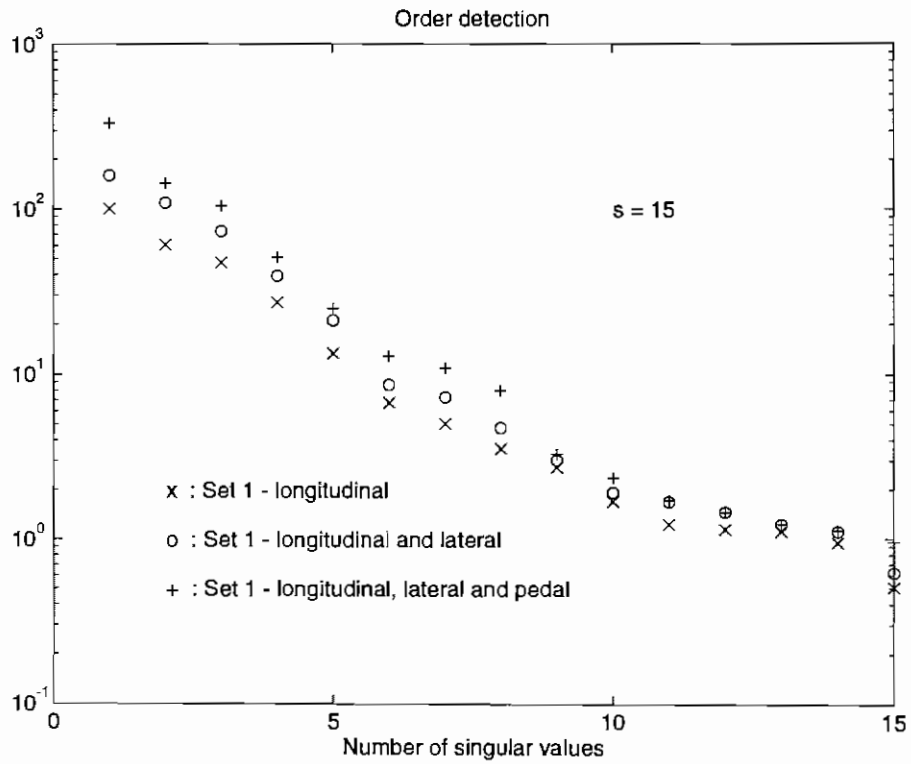

Figure 2: Singular values computed by the PO scheme applied to three sets of concatenated data records belonging to the set with frequency sweep type of inputs.

3.2. Model estimation and preliminary model validation.

By concatenating all available data records, we approximated the extended observability matrix Γ_{15} as was indicated in Eq. (11). From this approximated column space, we compute the pair $(A, C)_T$ by solving the equations Eqs. (13- 14).

When using the procedure described in section 2.4 to compute the B_T matrix, the restruction of the output using the estimated quadruple $[A, B, C, 0]_T$ showed a too high system gain. However, when using the alternative scheme described in section 2.6, far more accurate reconstructed outputs could be obtained. Since, the computational burden of the present MATLAB routine

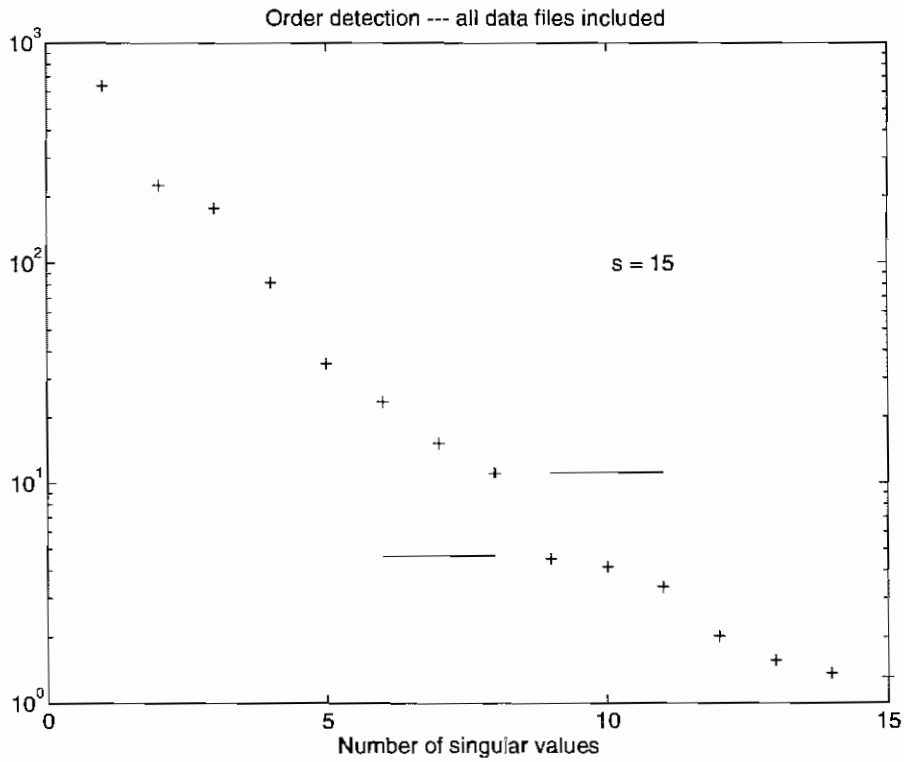

Figure 3: Singular values computed by the PO scheme applied to a concatenation of all available data records listed in Table 1.

IC, that implements this alternative way of estimating the B_T matrix and the initial conditions, is extremely high in comparison with the other calculations, we restrict the demonstration of this assertion using only a very small part of the available data sets. The selected part corresponds to the first $30s$ of the frequency sweep input to the pedals. This part was selected because it shows an excitation of all the input quantities, a necessary requirement when we want to estimate the whole B matrix.

It should be remarked that also for this way of computing the B-matrix more than a single data batch should be considered. This would certainly improve the condition number of the least squares problem formulated in section 2.6. For the present case this condition number was of the order 10^5. This in addition to the incorporation of numerically reliable rank revealing QR factorization schemes [Golub and Van Loan, 1989] will certainly improve the estimated B-matrix (and initial conditions).

The reconstructed output sequences (and the corresponding true measurements) are plotted in Figures 4 and 5. We observe that despite the use of only a small part of the observation sequence and the relatively high condition number of the least squares problem very accurate estimates of the outputs were obtained.

3.3. The estimated model parameters.

The PO scheme estimates a discrete time state space model. The particular basis in the state space is arbitrary. However, since the physically relevant state is equal to the measured output, we can easily transform the obtained state space model into this physically relevant coordinates. When the obtained state space estimate is denoted by $[A_T, B_T, C_T, 0]$, the physically relevant but

similarly equivalent state space model representation is given by $[C_T A_T C_T^{-1}, C_T B_T, I, 0]$. Here we remark that the matrix C_T was very well conditioned, namely it had a condition number of 35.

Finally, we used the matlab command `d2c` to transform the physically relevant, discrete state space model estimate to the continuous time domain. The corresponding A_c and B_c matrices have the following values:

```
Ac =

  Columns 1 through 7

      u         v         w        phi      theta         p         q

  -0.0296    0.0125    0.0715    0.5763  -10.2842   -0.2656   -0.8977
  -0.0052   -0.0601    0.0111    8.5690   -0.5112   -0.7831   -3.2469
  -0.1671   -0.1157   -0.5901    1.4572   -4.0770    1.1626   47.1939
  -0.0032   -0.0074   -0.0009   -0.0627    0.0691    0.9055    0.2750
  -0.0023   -0.0042   -0.0041    0.0175   -0.0055   -0.0269    1.3489
   0.0065    0.0023   -0.0173   -0.3900    0.2117   -0.5453    0.5529
   0.0041    0.0022   -0.0160    0.1739   -0.0256   -0.2537    0.4117
  -0.0046    0.0327   -0.0444    0.2049    0.2536   -1.6949    3.1423

  Column 8

      r

    0.5735
  -40.1725
   -2.4611
   -0.1477
    0.0198
    1.6211
   -0.6387
   -0.5875


Bc =

     long      lat      pedal      col

    0.5479    0.4782    0.0025    0.1231
    0.4952    0.1200    0.0262    1.0185
   -0.4401   -1.4928    0.0615   -1.9804
    0.0041   -0.0138   -0.0146   -0.0292
   -0.0157   -0.0044    0.0028   -0.0607
   -0.0018    0.0358    0.0059   -0.0701
```

```
-0.0015   -0.0139    0.0059   -0.0117
-0.0200   -0.0479    0.0479   -0.0719
```

4. Concluding remarks.

The overall conclusion of this preliminary identification study with the BO105 flight test data using the MOESP class of identification schemes is that this type of identification schemes give very promising results. This is because the method has demonstrated to deal in a robust manner with the practical difficulties of:

- high order flexible (vibration) modes.
- non-linear phenomena.

Here it is remarked that the first effect may be further reduced when an input can be applied to the system that it is not affected by these high order modes. This might e.g. be the case when applying a computer generated input to the test vehicle. Whether this is feasible with future test vehicles and whether this leads to improved identification results still needs to be demonstrated.

The main advantage of this SMI scheme is its *very simple and robust* way of detecting the 'correct' model structure and the *non-iterative* operation of the calculations. Looking at the preliminary reconstructions obtained with the estimated model, these attractive properties in no way seem to reduce the accuracy of the obtained estimate.

The tests described in this paper, have been performed with the m-files listed in Appendix 1. In these implementations no attention has been put on the efficiency of implementation. In this way, these routines are not very friendly to use. However, recently we have started to incorporate some of these tools within the ANDECS [4] Computer Aided Control Environment. For an overview of the capabilities that will soon come available, we refer to Appendices 2 and 3. Making these tools available within ANDECS with special care of the numerical robustness, efficiency in computation and memory use, will make the full potential of new and promising SMI approach available to the (flight test) identification community.

[4] ANDECS® (ANalysis & DEsign of Controlled Systems) is a registrated trademark of DLR

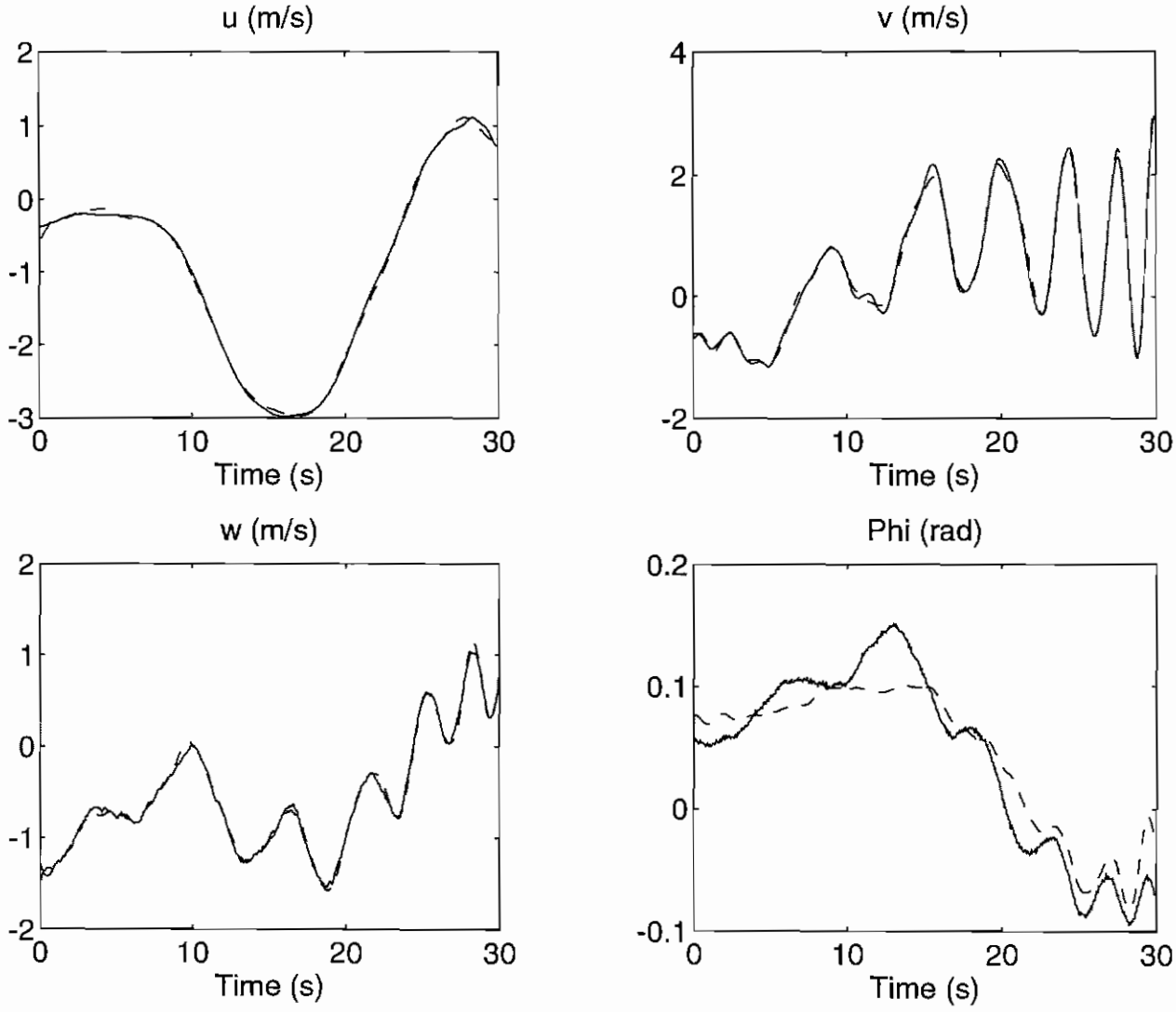

Figure 4: Reconstruction (– – –) and measurements (—) of the first four outputs (listed in Table 2) for the first 30s of the pedal frequency sweep input.

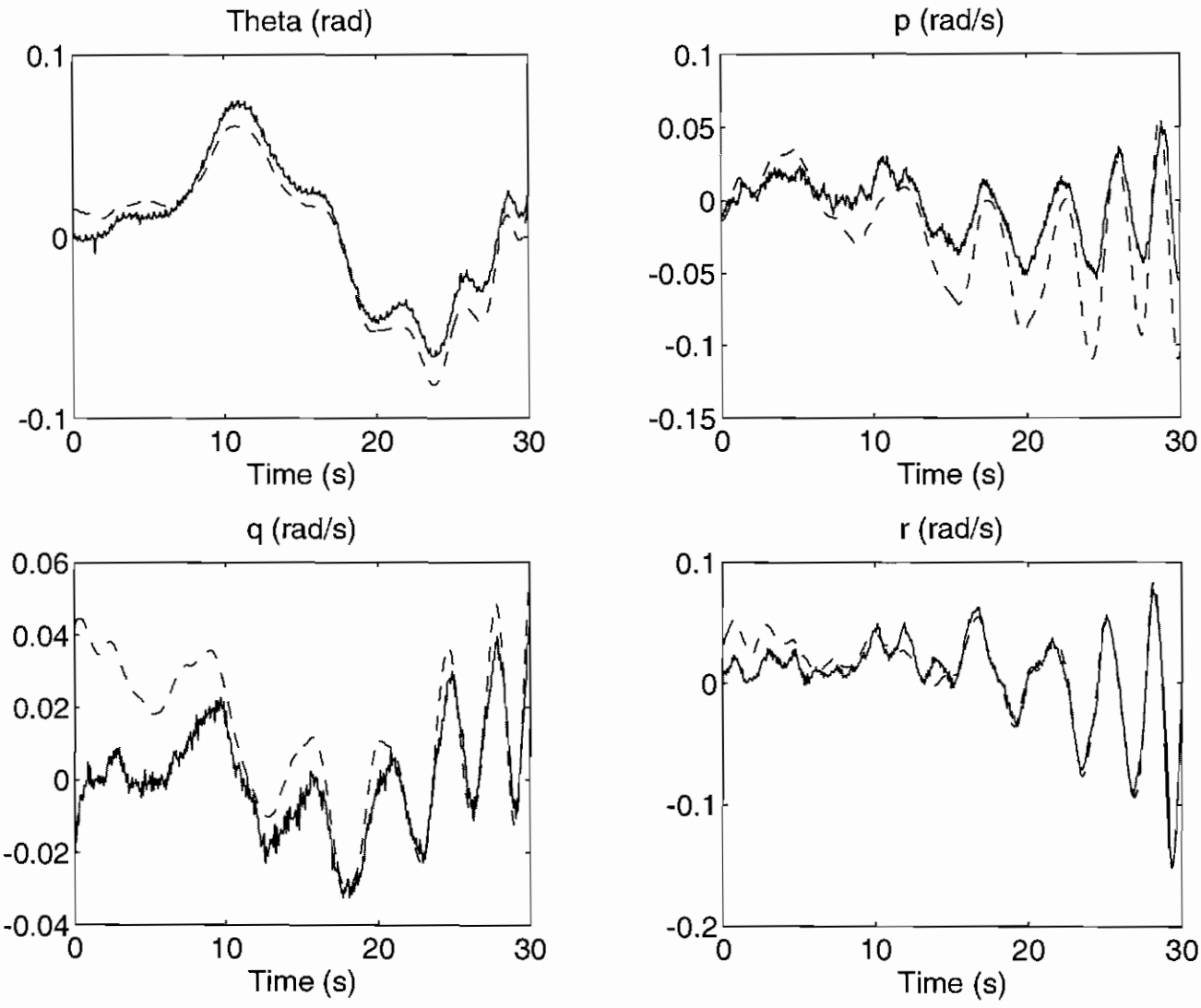

Figure 5: Reconstruction (– – –) and measurements (—) of the last four outputs (listed in Table 2) for the first 30s of the pedal frequency sweep input.

References

KALETKA, J., (1991), BO 105 Identification results, In AGARD LS-178 on *Rotorcraft system identification.*

AGARD LS-178, (1991), AGARD Lecture Series on Rotorcraft System Identification.

GOLUB, G. and LOAN, C. VAN, (1989), Matrix Computations, *The Johns Hopkins University Press.*

LARIMORE, W., (1990), Canonical Variate Analysis in Identification, Filtering and Adaptive Control, In *Proceedings of the 29th IEEE Conf. Decision and Control*, 596–604, Hawaii.

MOLER, C., LITTLE, J., and BANGERT, S., (1987), PRO-MATLAB User's Guide, *The MathWorks Inc.*

OVERSCHEE, P. VAN and MOOR, B. DE, (1994), An Exact Subspace Algorithm for the Identification of Combined Deterministic-Stochastic Systems, In the Special Issue on Statistical Signal Processing and Control of *Automatica*, Vol. 30, No. 1, pp. 75-84.

VERHAEGEN, M., (1990), Identification of MIMO output-error state space models, Technical Report N90.08, Delft University of Technology, Network Theory Section.

VERHAEGEN, M. and DEPRETTERE, E., (1991), Subspace Model Identification, In Deprettere, E.F. and Veen, A.J. van der, editors: *Algorithms and Parallel VLSI architectures*, volume B, 13–32. Elsevier.

VERHAEGEN, M. and DEWILDE, P., (1992a), Subspace Model Identification. Part I: The Output-Error State Space model identification class of algorithms, *Int. J. Control*, **Vol. 56**, No. 5, pp. 1187-1210.

VERHAEGEN, M. and DEWILDE, P., (1992b), Subspace Model Identification. Part II: Analysis of the elementary Output-Error State Space model identification algorithm, *Int. J. Control*, **Vol. 56**, No. 5, pp. 1211-1241.

VERHAEGEN, M., (1993a), Subspace Model Identification. Part III: Analysis of the ordinary Output-Error State Space model identification algorithm, *Int. J. Control*, **Vol. 58**, No. 3, pp. 555-586.

VERHAEGEN, M., (1993b), Identification of the deterministic and stochastic part of MIMO state space models under the presence of process and measurement noise, In Proc. of the *European Control Conference 1993*, Groningen, The Netherlands.

VERHAEGEN, M., (1993c), Application of a Subspace Model Identification Technique to identify LTI systems operating in closed loop, *Automatica*, Vol. 29, No. 4, pp. 1027–1040.

VERHAEGEN, M. (1994a), Identification of the deterministic of MIMO state space models given in innovations form form input-output data, In the Special Issue on Statistical Signal Processing and Control of *Automatica*, Vol. 30, No. 1, pp. 61-74.

VERHAEGEN, M. and BABUSKA, J. (1994b), Diagnosis of lung infections based on pressure-flow records taken from insufflated patients, (*in preparation*).

Appendix 1: m-files implementing the PO scheme.

In this appendix, the m-files are listed that were used in the analysis of the BO105 flight test experiments. Four m-files are listed:

1. The PO_ORDER scheme, which performs the RQ factorization in Eq. (5) with the possibility to concatenate data sets. In addition it supplies the user with information on the order of the underlying system via the singular values in the vector Sn. These correspond to the singular values computed in Eq. (11).

2. The PO_MOD scheme, which calculates the system quadruple $[A, B, C, D]_T$ from the information supplied by the PO_ORDER scheme. The algorithms used are described in sections 2.3 and 2.4.

3. The PO_TST routine, which shows an example of how to use both the PO_ORDER and PO_MOD routines.

4. The IC routine. This routine computes the initial conditions and the matrix B_T as described in section 2.4.

A.1. The PO_ORDER scheme.

```
function [R1C,R2C,Un,Sn,r]=PO_ORDER(u,y,dead_time,s,r);
% PO_ORDER          delivers information about the order of the LTI
%                   state space model and acts as a pre-processor for
%                   PO_MOD. The latter computes the actual
%                   state space quadruple estimate.
%
% model structure when dead_time=0
%                    x(k+1) = Ax(k) + Bu(k) + w(k)
%                    y(k)   = Cx(k) + Du(k) + v(k)
%                   when dead_time>=1
%                    x(k+1) = Ax(k) + Bu(k-dead_time+1) + w(k)
%                    y(k)   = Cx(k)          + v(k)
%                   where w(k),v(k) is zero-mean white noise sequences,
%                   independent of the noise-free input u(k)
%
% [R1C,R2C,Un,Sn]=PO_ORDER(u,y,dead_time,s);
%
% On input,
%   u,y             are the input- and output data sequences, given
%                   as column vectors, u=[u1 u2 ... um] and
%                   y=[y1 y2 ... yL]
%   dead_time       the estimated dead-time in the system
%   s               the dimension parameter that determines the number
%                   of block rows in the processed Hankel matrices
%   r               contains the lower triangular compression from
%                   previously processed input-output (i/o) data records.
```

```
%                   When no previous i/o records have been processed, it
%                   is not specified.
% On return,
%   R1C,
%   R2C,Un          relevant data for the PO_MOD algorithm,
%   Sn              singular values bearing information on the order
%                   of the system
%   r               contains the lower triangular compression based on
%                   the previously and currently processed i/o data records.
%
% The PO_ORDER (PO_MOD) routine corresponds to the PO scheme
% derived and analyzed in VERHAEGEN: "Identification the
% deterministic part of MIMO state space models given in innovations
% form from input-output data, Automatica, 1993.
%
% TESTFILE: the PO_ORDER and PO_MOD routines are
% tested in the m-file PO_TST.m

% Michel Verhaegen June 1992
% Modification June 1994
% copyright (c) 1992 Verhaegen Michel
%
if nargin < 5, r = []; end
[N,m]=size(u);
[N,L]=size(y);
N=N-2*s+1;

if (dead_time==0),
   Y=[];Up=[];Uf=[];
   Up=u(1:N,:);
   Uf=u(s+1:N+s,:);
   for i=(2:s)
       Up=[Up u(i:N+i-1,:)];
       Uf=[Uf u(i+s:N+i+s-1,:)];
   end
   Y=y(1:N,:);
   for i=(2:2*s)
       Y=[Y y(i:N+i-1,:)];
   end,
end,
if (dead_time>=1),
   N=N-dead_time+1;
   Y=[];Up=[];Uf=[];
   Up=u(1:N,:);
   Uf=u(s+1:N+s,:);
   for i=(2:s)
```

```
        Up=[Up u(i:N+i-1,:)];
        Uf=[Uf u(i+s:N+i+s-1,:)];
    end,
    Y=y(dead_time:N+dead_time-1,:);
    for i=(2:2*s),
        Y=[Y y(dead_time+i-1:N+dead_time+i-2,:)];
    end,
end,
r=triu(qr([r';[Uf Up Y]]));r=r(1:2*(m+L)*s,1:2*(m+L)*s)';
R11=r(1:m*s,1:m*s);
R21=r(m*s+1:2*m*s,1:m*s);
R22=r(m*s+1:2*m*s,m*s+1:2*m*s);
R31=r(2*m*s+1:(2*m+L)*s,1:m*s);
R32=r(2*m*s+1:(2*m+L)*s,m*s+1:2*m*s);
R41=r((2*m+L)*s+1:2*(m+L)*s,1:m*s);
R42=r((2*m+L)*s+1:2*(m+L)*s,m*s+1:2*m*s);
R43=r((2*m+L)*s+1:2*(m+L)*s,2*m*s+1:(2*m+L)*s);
[Un,Sn,Vn]=svd([R42 R43]);
Sn = diag(Sn);
R1C=[R21 R22 R11];
R2C=[R31 R32 R41];
%
% END OF THE CALCULATIONS
```

A.2. The PO_MOD scheme.

```
function [A,B,C,D]=PO_MOD(R1C,R2C,Un,dead_time,n,L,m,s,stb);
% PO_MOD            estimates the quadruple of system matrices
%                   of a LTI state space model
%                   using the output of the PO_ORDER routine
%
% model structure   when dead_time=0
%                   x(k+1) = Ax(k) + Bu(k)              + w(k)
%                   y(k)   = Cx(k) + Du(k) + v(k)
%                  when dead_time>=1
%                   x(k+1) = Ax(k) + Bu(k-dead_time+1) + w(k)
%                   y(k)   = Cx(k)          + v(k)
%                  where w(k),v(k) is zero-mean white noise sequences,
%                  independent of the noise-free input u(k)
%
% [A,B,C,D]=PO_MOD(R1C,R2C,Un,dead_time,n,L,m,s);
%
% On input,
%  R1C,
%  R2C,Un         data computed by the PO_ORDER routine
%  dead_time      the dead-time of the state space model
```

```
%  n              the order of the state space model
%  L              the number of outputs
%  m              the number of inputs
%  s              dimension parameter in PO_ORDER routine
%                 of block rows in the matrices R1C, etc.
%
% On return,
%  [A,B]
%  [C,D]          the estimated state space quadruple
%
% The PO_MOD routine corresponds to the PO scheme
% derived and analyzed in VERHAEGEN: "Identification the
% deterministic part of MIMO state space models given in innovations
% form from input-output data, Automatica, 1993.
%
% TESTFILE: the PO_ORDER and PO_MOD routines are
% tested in the m-file PO_TST
%

%    Michel Verhaegen  June 1992
%    copyright (c) 1992 Verhaegen Michel

fprintf('Calculating the A and C matrix \n')
for i=1:s,
    Un((i-1)*L+1:i*L,:)=Un((i-1)*L+1:i*L,:)*stb^(i-1);
end
un1=Un(1:(s-1)*L,1:n);
un2=Un(L+1:s*L,1:n);
A=un1\un2;
C=un1(1:L,:);
%
fprintf('Calculating the B and D matrix \n')
u2=Un(:,n+1:s*L);
K = u2'*R2C/R1C;
Kexpand = K(:,1:m);
for i=(2:s),
    Kexpand = [Kexpand;K(:,(i-1)*m+1:i*m)];
end
% Kexpand is an [(Ls-n)s x m] - matrix
Q = u2';
for i=(2:s),
    Q=[Q;[u2(L*(i-1)+1:L*s,:)' zeros(L*s-n,(i-1)*L)]];
end
% Q is an [(Ls-n)s x Ls] - matrix
if (dead_time==0),
   IG = [eye(L) zeros(L,n);zeros((s-1)*L,L) un1];
```

```
   % IG is an [(L+(Ls-n) x L+n] - matrix
   % solving for the pair [B,D] in total least squares sense:
   [U,S,V]=svd([Q*IG Kexpand]);
   DB = V(:,n+L+1:n+L+m);
   DB = -DB(1:n+L,:)*inv(DB(n+L+1:n+L+m,:));
   % or in least squares sense: DB = (Q*IG)\Kexpand;
   % DB = (Q*IG)\Kexpand;
   D  = DB(1:L,:);
   B  = DB(L+1:L+n,:);
end,
if (dead_time>=1),
   Q = Q(:,L+1:L*s)*un1;
   % solving for the matrix B in total least squares sense:
   [U,S,V]=svd([Q Kexpand],0);
   DB = V(:,n+1:n+m);
   % or in least squares sense: B= Q\Kexpand;
   B  = - DB(1:n,:)*inv(DB(n+1:n+m,:));
   D  = zeros(L,m);
end
%
% END OF THE CALCULATIONS
```

A.3. The PO_TST routine.

```
%
% PO_TST   testfile for PO_ORDER and PO_MOD
%
a=[1 -1.5 0.7];b=[0 1 0.5];
u=randn(200,1);y=filter(b,a,u);
% processing the first batch of data
[R1C,R2C,Un,Sn,r1]=PO_ORDER(u,y,1,10);
u=rand(300,1);y=filter(b,a,u);
% processing the second batch of data
[R1C,R2C,Un,Sn,r2]=PO_ORDER(u,y,1,10,r1);
[A,B,C,D]=PO_MOD(R1C,R2C,Un,1,2,1,1,10);
plot([dimpulse(A,B,C,D,1,20) dimpulse(b,a,20)])
%
pause,
b=[1 1 0.5];y=filter(b,a,u);
[R1C,R2C,Un,Sn]=PO_ORDER(u,y,0,10);
[A,B,C,D]=PO_MOD(R1C,R2C,Un,0,2,1,1,10);
plot([dimpulse(A,B,C,D,1,20) dimpulse(b,a,20)])
```

A.4. The IC routine.

```
function [ye,x0,B]= IC(A,C,u,y);
% IC       estimates the initial conditions and the input matrix B
```

```
%          of the quadruple of system matrices [A,B,C,D] using
%          the knowledge of the pair (A,C).
%
% [ye,x0,B]=IC(A,C,u,y);
%
% On input,
%   (A,C)   the estimated (A,C) pair of the quadruple of system matrices
%   u,y     the input respectively output data of the system to be
%           identified.
%
% On return,
%   ye      the reconstructed output based on the estimated quadruple
%           [A,B,C,0]
%   x0      the estimated initial condition
%   B       the estimated B matrix
%
% The IC routine is described in: VERHAEGEN, VARGA and GRUEBEL: "Some
% experience with the MOESP class of subspace model identification
% methods in identifying the BO105 Helicopter.
%
% Remark: No effort is made to improve the condition number of the
% constructed least-squares problem via e.g. rank revealing QR.

%
% Michel Verhaegen, June 1994
% copyright (c) 1994 Verhaegen Michel
m = size(u,2);
n = size(A,1);
Ext_O = C;
for i=2:size(y,1),
    Ext_O = [Ext_O;C*A^(i-1)];
end
YY = [];
for i=1:m,
    i,
    for j=1:n,
        B=zeros(n,1);B(j,1)=1;
        yh = dlsim(A,B,C,zeros(size(C,1),1),u(:,i));
        yy = [];
        for k=1:size(y,1),
            yy = [yy;yh(k,:)'];
        end
        YY = [YY yy];
    end
end
yy=[];
```

```
for i=1:size(y,1),
%    yy = [yy;y(i,:)'-ye(i,:)'];
    yy = [yy;y(i,:)'];
end
cond([Ext_0 YY]),
X = [Ext_0 YY]\yy;
x0 = X(1:n,1);
B = [];
for i=1:m,
    B = [B X(i*n+1:(i+1)*n,1)];
end
ye = dlsim(A,B,C,zeros(size(C,1),m),u,x0);
% END OF THE CALCULATIONS
```

Appendix 2: The documentation of the RASP implementation of the ordinary MOESP algorithm.

```
SUBROUTINE
Ordinary MOESP identification
Procedure purpose:
  This subroutine estimates from given input and output data sequences,
  the order and the matrices of the linear time-invariant discrete-time
  state space model

                  x(k+1) = Ax(k) + Bu(k)

                  y(k)   = Cx(k) + Du(k) + e(k)

  by using the ordinary MOESP algorithm.

Usage:
   CALL RPIMOE( U, NSMP, M, Y, L, NOBL, TOL, WITHD, CONCT, LAST,
                CTRL, N, A, B, C, D, R, IWORK, RWORK, LRWORK, * )

  U     :  IN, DOUBLE (NSMP,M)
           system input data sequence matrix  U = [u_1 u_2 ... u_m].
           Column  j  of  U  contains the  NSMP  values of the
           j-th input component for consecutive time increments.
  NSMP  :  IN, INTEGER
           number of rows of matrices  U  and  Y  (number of samples)
           for the current data batch.
           For non-sequential processing of the data:
           NSMP >= (M+L+1)*NOBL - 1
           For sequential processing of several data batches:
           NSMP >= NOBL . For LAST = .TRUE., a call with NSMP = 0 is
                          allowed if the upper triangular
                          factor of the QR factorization is already
                          available in the array  R,  and only the
                          system matrices are to be computed.
           The total number of samples when calling with LAST = .TRUE.
           should be at least (M+L+1)*NOBL - 1.
  M     :  IN, INTEGER
           dimension of system input vector.
  Y     :  IN, DOUBLE (NSMP,L)
           system output data sequence matrix  Y = [y_1 y_2 ... y_l].
           Column  j  of  Y  contains the  NSMP  values of the
           j-th output component for consecutive time increments.
  L     :  IN, INTEGER
           dimension of system output vector.
  NOBL  :  IN, INTEGER
```

number of block rows in the processed input and output Hankel matrices. NOBL should be chosen larger than n, the estimated dimension of state vector.

TOL : IN, DOUBLE
absolute tolerance used for determining an estimate of the system order. If TOL > 0, the estimate is indicated by the number of singular values greater than or equal to TOL. (Singular values less than TOL are considered zero.) If TOL = 0, the internally computed default value TOL = NOBL*EPS*SV[1] is used, where EPS is the relative machine precision, and SV[1] is the maximal singular value. If TOL < 0, the estimate is indicated by the index of the singular value that has the largest logarithmic gap to its successor. (See Method for the definition of the singular values.)

WITHD : IN, LOGICAL
specifies whether or not a non-zero feedthrough matrix D to be included in the estimated state space representation:
WITHD = .TRUE. means a possibly non-zero D matrix should be estimated.
WITHD = .FALSE. means a zero D matrix is assumed.

CONCT : IN, LOGICAL
specifies whether or not the successive data batches belong to a single experiment:
CONCT = .TRUE. means the current data batch is a continuation of the previous data batch.
CONCT = .FALSE. means there is no connection between the current data batch and the previous ones.

LAST : IN, LOGICAL
specifies whether or not sequential data processing is used, and (for sequential processing) whether or not the current data block is the last one, as follows:
LAST = .TRUE. means non-sequential processing or the last data block in sequential processing.
LAST = .FALSE. means not the last data block in sequential processing.

CTRL : IN, LOGICAL
specifies whether or not the user's confirmation of the system order estimate is desired before the computation of system matrices, as follows:
CTRL = .TRUE. means user's confirmation is desired.
CTRL = .FALSE. means no confirmation is necessary.
If CTRL = .TRUE., a reverse communication routine routine, IMCORD, is called, and, after inspecting the singular values and system order estimate, n, the user may accept n or set a new value.

```
          IMCORD  is not called by the routine if  CTRL = .FALSE..
          CTRL = .TRUE.  should be used in extreme cases only.
N      :  OUT, INTEGER
          estimated order of the system.
A :   OUT, DOUBLE (NOBL-1,NOBL-1)
 the N*N estimated system state matrix A.
          (column dense)
B :   OUT, DOUBLE (NOBL-1,M)
 the N*M estimated system input matrix B.
          (column dense)
C :   OUT, DOUBLE (L,NOBL-1)
 the L*N estimated system output matrix C.
          (column dense)
D      :  OUT, DOUBLE (L,M)
 the L*M estimated system feedthrough matrix D if
          WITHD = .TRUE..
          D is not referenced if  WITHD = .FALSE..
          (column dense)
R :   IN, OUT, DOUBLE ((M*L)*NOBL,(M+L)*NOBL)
          On output, if  LAST = .FALSE. then R
          contains the current upper triangular factor of the QR
          factorization used to compress the data before determining
          the singular values in the  MOESP  algorithm.
          The content of  R  should be preserved between succesive
          calls of  RPIMOE  with  LAST = .FALSE..
          On first input, the content of R is not meaningfull.
IWORK :   OUT, INTEGER(NOBL-1)
          working array.
RWORK :   OUT, DOUBLE (LRWORK)
          working array.
          RWORK(1)  contains the optimal suggested value of LRWORK;
          RWORK(2)  contains an estimate of the reciprocal condition
          number for the final linear algebraic system solved;
          RWORK(3:2+L*NOBL)  contain the singular values used for
          determining the system order.
          On return with error number 4, RWORK(1) contains the
          minimum necessary value of LRWORK.
LRWORK :  IN, INTEGER
          dimension of working array RWORK.
          The value of LRWORK should be at least LRWMIN, where
          LRWMIN = max ( 2*(M+L)*NOBL,
                   2+L*NOBL + max( 4*L*NOBL, 5*L*NOBL-4 ),
                   2+L*NOBL +
                   max( (L*NOBL-1)*NOBL, L+NOBL-1+M )*(L+NOBL-1+M) +
                   max( (NOBL-1)*((NOBL-1)*L+2),
                        (L*NOBL-1)*M*NOBL, (L*NOBL-1)*(L*NOBL-L),
```

```
                    4*(L+NOBL-1+M), 5*(L+NOBL-1+M)-4 ) )
          This is an overestimate computed using the largest and
          smallest values for  n, namely  NOBL-1  and 1, respectively
          (for positive and negative terms, respectively).
          For good performance,  LRWORK  should be large.
  *     :  RETURN 1, target label in case of error (e.g. *1111)

File input/output:
  none

Method:

  Let us denote m = M, l = L, t = NSPM, s = NOBL.

  For non-sequential data acquisition, the  t x (m+l)s  matrix

       [ U'      Y'     ]
         1,s,t   1,s,t

  is constructed, where  U      and  Y      are Hankel matrices
                         1,s,t       1,s,t
  defined in terms of the input and output data /1/.
  A QR factorization is used to compress the data, and then a
  singular value decomposition (SVD) of the principal submatrix
  R_22 := R(ms+1:(m+l)s,ms+1:(m+l)s)  of the upper triangular factor
  R  reveals the order  n  of the system as the number of "non-zero"
  singular values. System matrices are finally computed from the
  right singular vectors of  R_22  and the submatrices  R_11  and
  R_12  of  R.

  For sequential data acquisition, the QR decomposition is done
  sequentially, by updating the upper triangular factor  R.  When
  all data have been compressed, the system order and system
  matrices are computed as in the previous case.

  Literature
  /1/ Verhaegen M., and P. Dewilde
      Subspace Model Identification. Part 1: The output-error state-
      space model identification class of algorithms.
      Int. J. Control, 56, pp. 1187-1210, 1992.
  /2/ Verhaegen M.
      Subspace Model Identification. Part 3: Analysis of the
      ordinary output-error state-space model identification
      algorithm.
      Int. J. Control, 58, pp. 555-586, 1993.
```

Remarks:

- The NSMP argument may vary from a cycle to another in sequential data acquisition (LAST=.FALSE.), but NOBL, M, and L should be kept constant. For efficiency, it is advisable to use NSMP as large as possible. NSMP cannot exceed its initial value, NSMPIN.
- When 100 cycles of sequential data acquisition are completed with LAST = .FALSE., a warning is issued, to prevent for an infinite loop.

Copyright:

1994 - DLR Institut fuer Robotik und Systemdynamik

Life cycle:

1994 MARCH V. Sima, Institut fuer Informatik, Bukarest: coded

Libraries required:

RASP, BLAS (1,2,3), LAPACK

Example:

Consider the discrete-time system (A,B,C,D) with the following matrices:

```
A = ( 1.5   -0.7 )     B = ( 1 )     C = ( 1   0.5 )     D =  0
    ( 1.0    0.0 )         ( 0 )
```

whose output response to random input signals and zero initial state was determined by using the subroutine RPIMIU. The following sequence of statements can be used to estimate the order and the matrices of the original system

```
      DATA   ISEED/ 1, 2, 3, 5 /
      NSMP   = 120
      NOBL   = 10
      M      = 1
      L      = 1
      TOL    = -1.0D0
      WITHD  = .FALSE.
      LAST   = .TRUE.
      CONCT  = .FALSE.
      CTRL   = .FALSE.
      LRWORK = 8100
C     Compute the output response for uniformly distributed
C     input sequence.
      X(1)   = 0.0D0
      X(2)   = 0.0D0
      CALL RPIMIU( NSMP, 2, M, L, 0, 0, A, B, E, C, D, F, X, U,
     *             W, V, ISEED, Y, RWORK, LRWORK, WITHD, 'G',
```

```
     *            'G', 'R', 'R', 'R', 1, *1111 )
C     Perform the system identification.
      CALL RPIMOE( U, NSMP, M, Y, L, NOBL, TOL, WITHD, CONCT, LAST,
     *            CTRL, N, A, B, C, D, R, IWORK, RWORK, LRWORK,
     *            *1111 )

 The estimated system matrices are:

 -                          -                      -
 A = (  .7831   .5606 )     B = ( -3.5468 )     C = ( -.5749   .4382 )
     ( -.2472   .7169 )         ( -2.3707 )

 which coincide with the matrices obtained by applying a similarity
 transformation to the original system.

******
_RASPMODR_CONTROL_IDENT_RPIMOE_MSGE


RPIMOE
-1-
Invalid parameter value on entry.
-2-
Singular value decomposition failed.
-3-
A singular upper triangular matrix was found.
-4-
Not enough working storage. It should be at least //LENG//.
******
```

Appendix 3: Overview of the implementations within ANDECS.

On the basis of the RASP subroutine RPIMOE, we implemented an ANDECS module SSMOE which provides the basic functionalities for the subspace identification approach. Given the input and output trajectories U and Y, this module computes the order and the matrices of the linear time-invariant discrete-time state space model

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + Du_k + e_k$$

by using the ordinary MOESP algorithm [Verhaegen and Dewilde, 1992a].

The following subcommands, implemented to support the user's dialogue in the SSMOE module, are generic for all similar modules for subspace identification:

```
PARAM    - set the parameters for the system identification
PLIST    - specify the listing options
POUT     - specify the output options
NEWBATCH - add a new data batch to the previously processed data
COMPUTE  - compute the system matrices
END      - terminate module
```

The following parameters can be specified by using the PARAM subcommand:

```
BLOCKS - the number of block rows in the input and output Hankel matrices
         Default value: 10
ORDER  - the order of system
         Default value: 0 (that means, the order is interactively
         determined)
D      - specifies whether the feedthrough matrix D in the
         state=space model is zero or not
TOLSVD - tolerance on singular values to determine an estimate of
         system order
         Default value: TOLSVD = 0.0D0 (an internally computed value is used)
         IF TOLSVD < 0, the estimate is computed by searching for the
         maximum  gap between two successive order detection singular values
```

Optionally the order detection singular can be listed by using the PLIST subcommand. By selecting appropriate options of the subcommand POUT, these singular values can be optionally stored together with the resulting system matrices.

The subcommand NEWBATCH is necessary if several data batches (U_i, Y_i), $i = 1 : N$ resulted from different experiments are to be used in a single processing of data.