

## Combined Deterministic and Stochastic System Identification and Realization: DSR—a subspace approach based on observations

DAVID DI RUSCIO†

*Keywords: System identification, subspace methods, stochastic and deterministic systems, state-space methods, time series analysis*

A numerically stable and general algorithm for identification and realization of a complete dynamic linear state space model, including the system order, for combined deterministic and stochastic systems from time series is presented. A special property of this algorithm is that the innovations covariance matrix and the Markov parameters for the stochastic sub-system are determined directly from a projection of known data matrices, without e.g. recursions of non-linear matrix Riccati equations. A realization of the Kalman filter gain matrix is determined from the estimated extended observability matrix and the Markov parameters. Monte Carlo simulations are used to analyze the statistical properties of the algorithm as well as comparing with existing algorithms.

### 1. Introduction

System identification can be defined as the problem of building mathematical models of systems based on observed data. Traditionally a set of model structures with some free parameters are specified and a prediction error (PE) criterion measuring the difference of the observed outputs and the model outputs is optimized with respect to the free parameters. In general, resulting in a non linear optimization problem in the free parameters even when a linear time invariant model is specified. A tremendous amount of research has been reported, resulting in the so called prediction error methods (PEM).

In our eyes the field of subspace identification, Larimore (1983, 1990), Verhagen (1994), Van Overschee and De Moor (1994), Di Ruscio (1994), not only resolves the problem of system identification but also deals with the additional problem of structure identification. In subspace identification methods a data matrix is constructed from certain projections of the given system data. The observability matrix for the system is extracted as the column space of this matrix and the system order is equal to the dimension of the column space.

Descriptions of the advantages of subspace identification methods over traditional PE methods can be found in Viberg (1996) and in Van Overschee (1995).

The method for system identification and state space model realization which is presented in this work is believed to be a valuable tool for analyzing and modeling of observed input and output data from a wide range of systems, in particular combined deterministic and stochastic dynamical systems. One particular industrial application is presented in Di Ruscio and Holmberg (1996). Only linear algebra is applied in order to estimate a complete linear time invariant state space model.

---

Received 14 February 1996.

†Department of Process Automation, Telemark Institute of Technology, N-3914 Porsgrunn, Norway.

The remainder of the paper is organized as follows. Section 2 gives a definition of the system and the problem considered in this work. In Section 3 the data is organized into data matrices which satisfy an extended state space model or matrix equation. Section 4 shows how the system order and the model matrices can be extracted from the known data matrices. A numerically stable and efficient implementation is presented in Section 5. Section 6 gives a comparison of the method presented in this work with other published methods. Numerical examples and Monte Carlo simulations are presented in Section 7 and some concluding remarks follow in Section 8.

## 2. Preliminary definitions

### 2.1. System definition

Assume that the underlying system can be described by a discrete-time, time invariant, linear state space model (SSM) of the form

$$x_{k+1} = Ax_k + Bu_k + Ce_k \quad (1)$$

$$y_k = Dx_k + Eu_k + e_k \quad (2)$$

where the integer  $k \geq 0$  is discrete-time,  $x \in \mathbb{R}^n$  is the state vector with initial value  $x_0$ ,  $y \in \mathbb{R}^m$  is the system output,  $u \in \mathbb{R}^r$  is the system input,  $e \in \mathbb{R}^m$  is an unknown innovations process of white noise, assumed to be covariance stationary, with zero mean and covariance matrix  $E(e_k e_k^T) = \Delta$ . The constant matrices in the SSM are of appropriate dimensions.  $A$  is the state transition matrix,  $B$  is the external input matrix,  $C$  is the Kalman gain matrix,  $D$  is the output matrix and  $E$  is the direct control input to output (feed-through) matrix. We will assume that  $(D, A)$  is an observable pair.

The innovations model, Equations (1) and (2), is discussed in e.g. Faurre (1976).

### 2.2. Problem definition

The problem investigated in this paper is to identify a state space model, including the system order ( $n$ ), for both the deterministic part and the stochastic part of the system, i.e. the quadruple matrices  $(A, B, D, E)$  and the double matrices  $(C, \Delta)$ , respectively, directly from known system input and output data vectors (or time series) defined as

$$\left. \begin{array}{ll} u_k & \forall \quad k = 0, \dots, N-1 \\ y_k & \forall \quad k = 0, \dots, N-1 \end{array} \right\} \text{Known data vectors}$$

In continuous time systems the matrix  $E$  in Equation (2) is usually zero. This is not the case in discrete time systems due to sampling. However,  $E$  can be forced to be zero by including a structure constraint. This will be commented later.

### 2.3. Matrix definitions

Associated with the SSM, Equations (1) and (2), we make the following definitions:

- The extended observability matrix ( $O_i$ ) for the pair  $(D, A)$  is defined as

$$O_i \stackrel{\text{def}}{=} \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{i-1} \end{bmatrix} \in \mathbb{R}^{im \times n} \quad (3)$$

where subscript  $i$  denotes the number of block rows. The matrix  $O_i$  is denoted the extended observability matrix when the number of block rows  $i$  is greater than

the minimal number of block rows needed in the observability matrix, in order to check if the system is observable.

- The reversed extended controllability matrix ( $C_i^d$ ) for the pair  $(A, B)$  is defined as

$$C_i^d \stackrel{\text{def}}{=} [A^{i-1}B \ A^{i-2} \dots B] \in \mathbb{R}^{n \times ir} \quad (4)$$

where subscript  $i$  denotes the number of block columns.

A matrix  $C_i^s$  for the pair  $(A, C)$  is defined similar to Equation (4), i.e. with  $C$  substituted for  $B$  in the above definition.

- The lower block triangular Toeplitz matrix ( $H_i^d$ ) for the quadruple matrices  $(D, A, B, E)$

$$H_i^d \stackrel{\text{def}}{=} \begin{bmatrix} E & 0 & 0 & \dots & 0 \\ DB & E & 0 & \dots & 0 \\ DAB & DB & E & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}B & DA^{i-3}B & DA^{i-4}B & \dots & E \end{bmatrix} \in \mathbb{R}^{im \times ir} \quad (5)$$

where subscript  $i$  denotes the number of block rows.

A lower block triangular Toeplitz matrix  $H_i^s$  for the quadruple matrices  $(D, A, C, F)$  is defined as

$$H_i^s \stackrel{\text{def}}{=} \begin{bmatrix} F & 0 & 0 & \dots & 0 \\ DC & F & 0 & \dots & 0 \\ DAC & DC & F & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}C & DA^{i-3}C & DA^{i-4}C & \dots & F \end{bmatrix} \in \mathbb{R}^{im \times im} \quad (6)$$

where  $F = I$  for the output model formulation, Equation (2).

#### 2.4. Notation

The projection  $A/B$  of two matrices  $A$  and  $B$  is defined as  $AB^T(BB^T)^\dagger B$  where  $\dagger$  denotes the Moore-Penrose pseudo-inverse of a matrix.

### 3. Extended state space model

The state space model, Equations (1) and (2), can generally be written as the following extended state space model (ESSM) (Di Ruscio (1994))

$$Y_{k+1|L} = \tilde{A}Y_{k|L} + \tilde{B}U_{k|L+1} + \tilde{C}E_{k|L+1} \quad (7)$$

where the known output and input data matrices  $Y_{k|L}$  and  $U_{k|L+1}$  are defined as follows

$$Y_{k|L} \stackrel{\text{def}}{=} \begin{bmatrix} y_k & y_{k+1} & y_{k+2} & \dots & y_{k+K-1} \\ y_{k+1} & y_{k+2} & y_{k+3} & \dots & y_{k+K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{k+L-1} & y_{k+L} & y_{k+L+1} & \dots & y_{k+L+K-2} \end{bmatrix} \in \mathbb{R}^{Lm \times K} \quad (8)$$

$$U_{k|L+1} \stackrel{\text{def}}{=} \begin{bmatrix} u_k & u_{k+1} & u_{k+2} & \dots & u_{k+K-1} \\ u_{k+1} & u_{k+2} & u_{k+3} & \dots & u_{k+K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{k+L-1} & u_{k+L} & u_{k+L+1} & \dots & u_{k+L+K-2} \\ u_{k+L} & u_{k+L+1} & u_{k+L+2} & \dots & u_{k+L+K-1} \end{bmatrix} \in \mathbb{R}^{(L+1)r \times K} \quad (9)$$

The unknown data matrix  $E_{k|L+1}$  of innovations noise vectors is defined as

$$E_{k|L+1} \stackrel{\text{def}}{=} \begin{bmatrix} e_k & e_{k+1} & e_{k+2} & \dots & e_{k+K-1} \\ e_{k+1} & e_{k+2} & e_{k+3} & \dots & e_{k+K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_{k+L-1} & e_{k+L} & e_{k+L+1} & \dots & e_{k+L+K-2} \\ e_{k+L} & e_{k+L+1} & e_{k+L+2} & \dots & e_{k+L+K-1} \end{bmatrix} \in \mathbb{R}^{(L+1)m \times K} \quad (10)$$

The scalar integer parameter  $L$  defines the number of block rows in the data matrices and the ESSM model matrices. The number of columns in  $Y_{k|L}$ ,  $U_{k|L+1}$  and  $E_{k|L+1}$  are  $K = N - L - k + 1$ . Each column in these matrices can be interpreted as extended output, input and noise vectors, respectively.  $K$  can be viewed as the number of samples in these extended time series. We also have that  $L < K < N$ .  $L$  is the only necessary parameter which has to be specified by the user.  $L$  is equal to the number of block rows in the extended observability matrix ( $O_L \in \mathbb{R}^{Lm \times n}$ ), which will be determined by the algorithm. For a specified  $L$ , the maximum possible order of the system to be identified, is  $n \leq Lm$  (if  $\text{rank}(D) = m$ , i.e.  $m$  independent outputs), or  $n \leq Ld$  where  $1 \leq d = \text{rank}(D) \leq m$ , i.e.  $d$  independent output variables.

The parameter  $L$  can be interpreted as the identification horizon. This means that  $L$  is the horizon used to recover the present state space vector  $x_k$ .

The matrices in the extended stated space model, Equation (7), are related to the underlying state space model matrices as follows

$$\tilde{A} = O_L A (O_L^T O_L)^{-1} O_L^T \quad (11)$$

$$\begin{aligned} \tilde{B} &= [OB - \tilde{A}E_1 \quad E_1 - \tilde{A}E_2 \quad E_2 - \tilde{A}E_3 \quad \dots \quad E_{L-1} - \tilde{A}E_L \quad E_L] \\ &= [O_L B \quad H_L^d] - \tilde{A} [H_L^d \quad 0_{Lm \times r}] \end{aligned} \quad (12)$$

$$\begin{aligned} \tilde{C} &= [O_L C - \tilde{A}F_1 \quad F_1 - \tilde{A}F_2 \quad F_2 - \tilde{A}F_3 \quad \dots \quad F_{L-1} - \tilde{A}F_L \quad F_L] \\ &= [O_L C \quad H_L^s] - \tilde{A} [H_L^s \quad 0_{Lm \times m}] \end{aligned} \quad (13)$$

The matrices  $E_i$  and  $F_i$ ,  $i = 1, \dots, L$ , are block columns in the Toeplitz matrices  $H_L^d$  and  $H_L^s$  defined in Equations (5) and (6), i.e.

$$H_L^d = [E_1 \quad E_2 \quad \dots \quad E_L] \quad (14)$$

$$H_L^s = [F_1 \quad F_2 \quad \dots \quad F_L] \quad (15)$$

The importance of the ESSM, Equation (7), is that the state vector preliminary is eliminated from the problem. Hence, the number of unknown is reduced. The ESSM also give us the relationship between the data matrices and the, at this stage, unknown, model matrices.

This paper is concerned with the problem of reconstructing the system order and system matrices in the state space model, (1) and (2), from the known data matrices  $Y_{k|L}$  and  $U_{k|L+1}$  which satisfy Equation (7). We refer to Di Ruscio (1994) and (1995) for a proof of the above results, which are the basis for the method presented in this work.

Note that the matrices  $H_L^d$  and  $H_L^s$  satisfy the matrix equation

$$Y_{k|L} = O_L X_k + [H_L^d \quad 0_{Lm \times r}] U_{k|L+1} + [H_L^s \quad 0_{Lm \times m}] E_{k|L+1} \quad (16)$$

and

$$X_k = [x_k \ x_{k+1} \ x_{k+2} \ \dots \ x_{k+K-1}] \in \mathbb{R}^{n \times K} \quad (17)$$

is a matrix of state vectors. Equation (16) is frequently used in other subspace identification methods, e.g. in Van Overschee and De Moor (1994) and Verhagen (1994).

#### 4. System identification and realization

##### 4.1. Identification and realization of system dynamics

The basic step in the algorithm is to identify the system order and the extended observability matrix from known data. In order to do so we will in this section derive an autonomous matrix equation from which the system dynamics can be identified. We will show how the system order  $n$ , the extended observability matrix  $O_L$  are identified. A realization for the system matrices  $A$  and  $D$  and the ESSM transition matrix  $\tilde{A}$  are then computed.

The term  $\tilde{B}U_{k|L+1}$  can be removed from Equation (7) by post-multiplying with a projection matrix  $U_{k|L+1}^\perp$  such that  $U_{k|L+1}U_{k|L+1}^\perp = 0$ . The projection matrix can e.g. be defined as follows

$$U_{k|L+1}^\perp = I_{K \times K} - U_{k|L+1}^T(U_{k|L+1}U_{k|L+1}^T)^{-1}U_{k|L+1} \quad (18)$$

Hence,  $U_{k|L+1}^\perp$  is the orthogonal projection onto the null-space of  $U_{k|L+1}$ . A numerically well posed way of computing the projection matrix is by use of the singular value decomposition (SVD). The projection matrix is given by the left singular vectors of  $U_{k|L+1}$  which is orthogonal to the null-space. However, in order to solve the complete system identification and realization problem, then it is more convenient to use the QR decomposition for computing the projection, as will be shown in Section 5. Note that a projection matrix onto the null-space of  $U_{k|L+1}$  exists if the number of columns  $K$  in the data-matrices satisfy  $K > L + 1$ .

Post-multiplying Equation (7) with the projection matrix  $U_{k|L+1}^\perp$  gives

$$\begin{aligned} & Y_{k+1|L} - Y_{k+1|L}U_{k|L+1}^T(U_{k|L+1}U_{k|L+1}^T)^{-1}U_{k|L+1} \\ &= \tilde{A}(Y_{k|L} - Y_{k|L}U_{k|L+1}^T(U_{k|L+1}U_{k|L+1}^T)^{-1}U_{k|L+1}) \\ &+ \tilde{C}(E_{k|L+1} - E_{k|L+1}U_{k|L+1}^T(U_{k|L+1}U_{k|L+1}^T)^{-1}U_{k|L+1}) \end{aligned} \quad (19)$$

Note that per definition the last noise term in equation (19) is zero as the number of samples approaches infinity, i.e.

$$\lim_{K \rightarrow \infty} \tilde{C} \frac{1}{K} E_{k|L+1} U_{k|L+1}^T = 0 \quad (20)$$

Hence, we have the following result

$$Y_{k+1|L}U_{k|L+1}^\perp = \tilde{A}Y_{k|L}U_{k|L+1}^\perp + \tilde{C}E_{k|L+1} \quad (21)$$

The noise term  $\tilde{C}E_{k|L+1}$  can be removed from Equation (21) by post-multiplying with  $\frac{1}{K}W_i^T$  where  $W_i$  is defined as a matrix of "instrumental" variables which is uncorrelated with  $E_{k|L+1}$ , i.e. we are seeking for a matrix with the following property

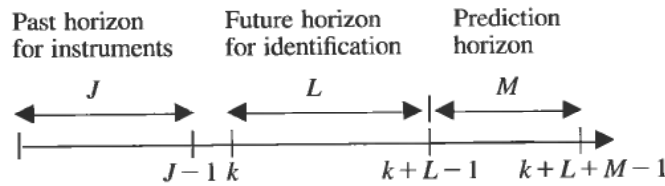


Figure 1. Illustration of horizons involved in the DSR algorithm. Usually  $k = J$ . That is, the end of the past is the beginning of the future.

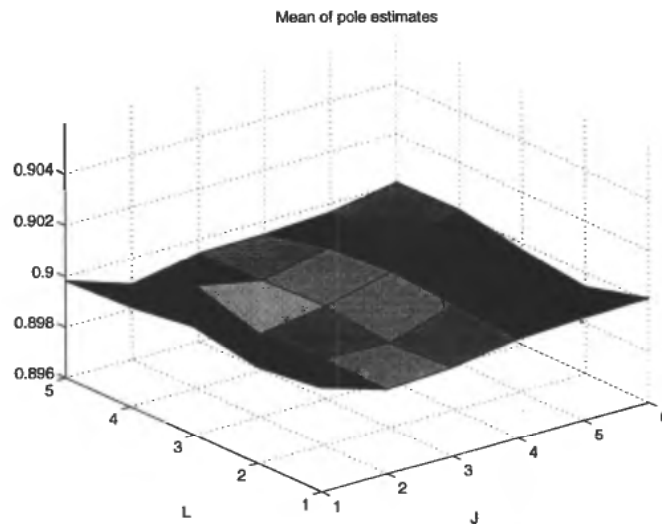


Figure 2. The mean value of the pole estimates for a Monte Carlo simulation with varying past horizon parameter  $J$  and identification parameter  $L$ . The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sum of 5 sinusoid signals ( $u^1$ ). The maximum estimate was 0.9003 for  $L = 3$  and  $J = 4$ . The minimum estimate was 0.8990 for  $L = 5$  and  $J = 2$ . The closest estimate to the actual pole ( $a = 0.9$ ) was 0.89997 for  $L = 3$  and  $J = 5$ .

$$\lim_{K \rightarrow \infty} \frac{1}{K} E_{k|L+1} W_i^T = 0 \quad (22)$$

An additional property is that  $W_i$  should be sufficiently correlated with the informative part in the ESSM in order not to destroy information about e.g. the system order.

An intuitive good choice is to use *past* data as instruments to remove *future* noise. This choice ensures that the instruments is sufficiently correlated with the informative part of the signals and sufficiently uncorrelated with future noise.

Define  $J$  as the number of time instants in the *past* horizon which is used for defining the instrumental variable matrix. Define  $L$  as the number of time instants in the horizon for identifying the state at time instant  $k$ , that is  $x_k$ , as well as the extended observability matrix of the system. Define  $M$  as a prediction horizon. However, we will restrict ourself to the case  $M = 1$  in this work. These horizons are illustrated schematically in Figure 1.

Some alternative instruments,  $W_i$ , for removing the noise and which satisfy Equation (22) are as follows. Define

$$W_i \in \mathbb{R}^{ni \times K} \quad \forall \quad i = 1, 2, 3 \quad (23)$$

where the row dimension  $ni$  is the number of instrumental variables and

$$W_1 = \begin{bmatrix} Y_{0|J} \\ U_{0|J} \end{bmatrix} \quad W_2 = Y_{0|J} \quad W_3 = U_{0|J} \quad \forall l \geq 0 \quad (24)$$

The choice of instruments is unimportant in the deterministic case, i.e. when the process as well as the observation noise are identically zero which also means that  $e_k = 0$  for all discrete time instants. However, the "optimal" choice of instruments is important in the combined deterministic and stochastic case.

Assume that  $U_{l|J}$  with  $l > 0$  is chosen as the instrumental variable matrix. This means that not only past inputs but also future inputs are used to remove future noise (first time instant in the future horizon satisfy  $J \leq k$ ). Our experiences from Monte Carlo simulations indicates that this is not an "optimal" choice. Note also that the future inputs already is used in the projection matrix  $U_{k|L+1}^\perp$ . Hence, it makes sense to use only past inputs as instruments, i.e. the choice  $U_{0|J}$ . It can also be shown (Verhagen, 1994) that by using only past inputs only the deterministic part of the model can be recovered.

Past outputs are assumed to be uncorrelated with future noise. This gives a first constraint on the discrete time instant  $k$ , i.e.  $k \geq J$ . We have

$$\lim_{K \rightarrow \infty} \frac{1}{K} E_{k|L+1} Y_{0|J}^T = 0 \quad \forall k \geq J \quad (25)$$

This statement can be proved from Equations (16) and (20). By incorporating past outputs as instruments we are also able to recover the stochastic part of the model. Note that the states which are excited from the known inputs are not necessarily the same as those which are excited from the unknown process noise variables. It is necessary that all states are excited from both known and unknown inputs and that they are observable from the output, in order to identify them.

Hence, the following past inputs and past outputs instrumental variable matrix is recommended to remove future noise from the model

$$W_1 = \begin{bmatrix} Y_{0|J} \\ U_{0|J} \end{bmatrix} \in \mathbb{R}^{J(m+r) \times K} \quad \forall J \geq 1 \quad (26)$$

A consistent equation for  $\tilde{A}$  is then given by the following autonomous matrix equation

$$Z_{k+1|L} = \tilde{A} Z_{k|L} \quad \forall k \geq J \quad (27)$$

where

$$Z_{k+1|L} \stackrel{\text{def}}{=} \frac{1}{K} Y_{k+1|L} U_{k|L+1}^\perp W_1^T \in \mathbb{R}^{mL \times ni} \quad (28)$$

$$Z_{k|L} \stackrel{\text{def}}{=} \frac{1}{K} Y_{k|L} U_{k|L+1}^\perp W_1^T \in \mathbb{R}^{mL \times ni} \quad (29)$$

Equation (27) is consistent because  $W_1$ , given by Equations (23) and (24), satisfy Equation (22). See also Equation (25).

We can now prove that the column space of the matrix  $Z_{k|L}$  coincide with the column space of the extended observability matrix  $O_L$ , when the identification (*future*) horizon parameter  $L$  is chosen great enough in order to observe all states and the *past* horizon parameter  $J$  are chosen sufficiently. Using Equations (16) and (29) with the past inputs and past outputs instrumental variable matrix gives

$$Z_{k|L} = \frac{1}{K} Y_{k|L} U_{k|L+1}^\perp W_1^T = O_L X_k U_{k|L+1}^\perp \frac{1}{K} W_1^T \in \mathbb{R}^{mL \times J(m+r)} \quad (30)$$

Assume that both the row and column dimensions of  $Z_{k|L}$  are greater or equal to the number of states, i.e.  $Lm \geq n$  and  $J(m+r) \geq n$ , and that  $L$  is chosen such that the system

is observable. The dimension of the column space of the left hand side matrix must be equal to the system order, i.e.  $\text{rank}(X_k) = n$ . Hence,

$$\text{rank}(Z_{k|L}) = \text{rank}\left(\frac{1}{K} Y_{k|L} U_{k|L+1}^T W_1^T\right) = \text{rank}(O_L X_k U_{k|L+1}^T \frac{1}{K} W_1^T) = n \quad (31)$$

The row constraints has a theoretical lower limit. From system theory we know that it is sufficiently with a number of  $L \geq n - \text{rank}(D) + 1$  observations of the output in order to observe the states of a linear system, Kalman, Falb and Arbib (1969, p. 37). However, the theoretical lower limit is the ceiling function  $L = \lceil n/m \rceil$ , defined as the integer ratio  $n/m$  rounded towards plus infinity.

From the column dimension we must ensure that the past horizon  $J$  for defining the instrumental variable matrix must satisfy  $J(m+r) \geq n$ . Hence, the theoretical lower limit is  $J = \lceil n/(m+r) \rceil$ .

The maximum system order which can be specified by the user for a specified choice of the parameter  $L$  is  $n = Lm$ . In this case the observability matrix can be estimated as the column space of  $Z_{k|L}$  only if the past horizon parameter  $J$  is chosen such that  $J(m+r) \geq Lm$ . A reasonable choice is therefore  $J = L$ .

Monte Carlo simulation experiments show a relatively constant statistical behavior of the estimates as a function of the past horizon parameter  $J$ . Hence, we simply recommend to put  $J = L$ .

We have the following algorithm for analysis and modeling of system dynamics.

*Algorithm 4.1 (System order,  $n$ , and the pair  $(D, A)$ )*

Given the matrices  $Z_{k+1|L}$  and  $Z_{k|L}$  with  $k \geq J$  which satisfy the autonomous matrix equation

$$Z_{k+1|L} = \tilde{A} Z_{k|L} \quad (32)$$

where

$$\tilde{A} = O_L A (O_L^T O_L)^{-1} O_L^T \quad (33)$$

and  $O_L$  is the extended observability matrix for the pair  $(A, D)$ .

*1. The system order  $n$*

Determine the Singular Value Decomposition (SVD)

$$Z_{k|L} = U S V^T \quad (34)$$

where  $U \in \mathbb{R}^{mL \times mL}$ ,  $S \in \mathbb{R}^{mL \times ni}$  and  $V \in \mathbb{R}^{ni \times ni}$  are given by:

$$U = [U_1 \quad U_2] \quad S = \begin{bmatrix} S_n & 0 \\ 0 & S_2 \end{bmatrix} \quad V = [V_1 \quad V_2] \quad (35)$$

where  $S_n \in \mathbb{R}^{n \times n}$  and  $n$  is the number of non-zero singular values of  $Z_{k|L}$ , which is equal to the system order.  $n$  is determined by inspection of the non-zero diagonal elements of  $S$  or  $SS^T$ . The term  $U_2 S_2 V_2^T$  represents the error by estimating the system order as the  $n$  first principal singular values.

*2. The extended observability matrix  $O_L$  for the pair  $(D, A)$*

The (extended) observability matrix can be taken directly as the first left part in  $U$ , i.e.  $U_1$ . We have



$$O_L = U(1:Lm, 1:n) = U_1 \quad (36)$$

### 3. The system matrix $A$

The system matrix  $A$  can be determined as

$$A = O_L^T Z_{k+1|L} V \begin{bmatrix} S_n^{-1} \\ 0 \end{bmatrix} = U_1^T Z_{k+1|L} V_1 S_n^{-1} \quad (37)$$

### 4. The system output matrix $D$

The matrix  $D$  can be taken as the  $m \times n$  upper sub-matrix in the observability matrix  $O_L$ , i.e.

$$D = U(1:m, 1:n) \quad (38)$$

### 5. The extended system matrix $\tilde{A}$

We have

$$\tilde{A} = O_L A (O_L^T O_L)^{-1} O_L^T = Z_{k+1|L} V_1 S_n^{-1} U_1^T \quad (39)$$

$\triangle$

We have chosen  $O_L = U_1$  in Step 2 for simplicity, because we have  $O_L^T O_L = I_{n \times n}$  in this case. This gives an output normal realization when  $L \rightarrow \infty$ . The algorithm can also be formulated with the choice  $O_L = U_1 S_n^{\frac{1}{2}}$  which gives a balanced realization when  $L \rightarrow \infty$ .  $O_L^T O_L$  is equal to the observability grammian as  $L$  tends to infinity because, in this case,  $DA^{L-1}$  tends to zero. A third choice is  $O_L = U_1 S_n$  which gives an input normal realization. These definitions are due to Moore (1981). These choices only represent different scalings of the column space and give similar state space model matrices. The scaling does not affect the statistical properties of the algorithm.

### 4.2. Realization of the deterministic sub-system

At this stage the system matrices  $A$  and  $D$  as well as the extended observability matrix  $O_L$  are known, see Section 4.1. In order to obtain a complete realization for the deterministic part of the system we need to compute the system matrices  $B$  and  $E$ .

There are many alternatives for extracting the  $B$  and  $E$  matrices. See e.g. Section 5.2 for an alternative to the method presented below.

A consistent estimate of the  $\tilde{B}$  matrix can be computed from

$$\tilde{B} \frac{1}{K} U_{k|L+1} U_{k|L+1}^T = \frac{1}{K} (Y_{k+1|L} - \tilde{A} Y_{k|L}) U_{k|L+1}^T \quad (40)$$

where  $\tilde{A}$  is determined from Algorithm 4.1, because

$$\lim_{K \rightarrow \infty} \tilde{C} \frac{1}{K} E_{k|L+1} U_{k|L+1}^T = 0 \quad \forall k \geq 0 \quad (41)$$

For known data and system matrices  $A$  and  $D$ , Equation (40) can be written as an over determined set of linear equations in the unknown system matrices  $B$  and  $E$ .

We will, in the rest of this section, discuss the simpler solution when  $U_{k|L+1} U_{k|L+1}^T$  is non-singular. The matrix  $\tilde{B}$  can be computed directly from Equation (40) in this case. We refer to Section 5.2 for the case when  $U_{k|L+1} U_{k|L+1}^T$  is singular. The system matrices

$B$  and  $E$  can be extracted from  $\tilde{B}$ .  $E$  is given directly as the lower right  $m \times r$  sub-matrix of  $\tilde{B}$ . We have

$$E = \tilde{B}(m(L-1) + 1: mL, rL + 1: r(l+1)) \quad (42)$$

$B$  is given as a function of the block columns in  $\tilde{B}$  and the matrices  $\tilde{A}$  and  $O_L$ .  $\tilde{A}$  and  $O_L$  are known from Algorithm 4.1. Define

$$\tilde{B}_i = \tilde{B}(1: mL, r(i-1) + 1: ri) \quad 1 \leq i \leq L+1 \quad (43)$$

as block column number  $i$  of  $\tilde{B}$ . We have

$$O_L B = \sum_{i=1}^{L+1} \tilde{A}^{i-1} \tilde{B}_i \quad (44)$$

and

$$B = \sum_{i=1}^{L+1} \tilde{A}^{i-1} (O_L^T O_L)^{-1} O_L^T \tilde{B}_i \quad (45)$$

One strategy for recursively extracting the  $B$ ,  $E$ ,  $H_L^d$  and  $O_L B$  matrices is as follows.

**Algorithm 4.2** Determination of  $H_L^d$ ,  $O_L B$ ,  $E$  and  $B$  from known  $\tilde{B}$ ,  $O_L$  and  $A$ .

```

 $\tilde{A} = O_L A (O_L^T O_L)^{-1} O_L^T$ 
 $E_L = \tilde{B}_{L+1}$ 
 $E = E_L(m(L-1) + 1: mL, 1: m)$ 
for  $i = 1, \dots, L$ 
   $E_{L-i} = \tilde{B}_{L-i+1} + \tilde{A} E_{L-i+1}$ 
  if  $i < L$ 
     $E_{L-i}(1: r(L-i-1), 1: r) = \text{zeros}(r(L-i-1), r)$ 
  end
end
 $O_L B = E_0$ 
 $B = (O_L^T O_L)^{-1} O_L^T O_L B$ 

```

△

The lower block triangular Toeplitz matrix  $H_L^d$  is given by the block columns  $E_1, \dots, E_L$  according to Equations (14) and (5).

#### 4.3. Realization of the stochastic sub-system

This section is mainly concerned with the problem of identifying the stochastic part of the system. However, for natural reasons most of the results in Sections 4.1 and 4.2 are extended and resolved in parallel.

The ESSM, Equation (7), gives a relation between the future data matrices and the SSM matrices. The following extension of the ESSM can be proved

$$Y_{k|L+1} = O_{L+1} A^k O_J^T Y_{0|J} + [O_{L+1} C_k^d \quad H_{L+1}^d] \begin{bmatrix} U_{0|k} \\ U_{k|L+1} \end{bmatrix}$$

$$\begin{aligned}
 & - [O_{L+1}A^kO_J^\dagger H_J^d \quad 0_{Lm \times (k+L+1-J)r}] \begin{bmatrix} U_{0|J} \\ U_{J|k+L+1-J} \end{bmatrix} \\
 & + [O_{L+1}C_k^s \quad H_{L+1}^s] \begin{bmatrix} E_{0|k} \\ E_{k|L+1} \end{bmatrix} \\
 & - [O_{L+1}A^kO_J^\dagger H_J^s \quad 0_{Lm \times (k+L+1-J)r}] \begin{bmatrix} E_{0|J} \\ E_{J|k+L+1-J} \end{bmatrix} \quad (46)
 \end{aligned}$$

Assume  $k = J$  for the sake of simplicity. Then we have

$$\begin{aligned}
 Y_{J|L+1} &= O_{L+1}A^J O_J^\dagger Y_{0|J} \\
 &+ [O_{L+1}C_J^d - O_{L+1}A^J O_J^\dagger H_J^d \quad H_{L+1}^d] \begin{bmatrix} U_{0|J} \\ U_{J|L+1} \end{bmatrix} \\
 &+ [O_{L+1}C_J^s - O_{L+1}A^J O_J^\dagger H_J^s \quad H_{L+1}^s] \begin{bmatrix} E_{0|J} \\ E_{J|L+1} \end{bmatrix} \quad (47)
 \end{aligned}$$

This last equation shows us nicely the connection between the past and future data matrices and the unknown model matrices. As we will see, it also gives us some important results. We have the following projection

$$\begin{aligned}
 & Y_{J|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp \\
 &= [O_{L+1}C_J^s - O_{L+1}A^J O_J^\dagger H_J^s \quad H_{L+1}^s] \begin{bmatrix} E_{0|J} \\ E_{J|L+1} \end{bmatrix} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp \quad (48)
 \end{aligned}$$

The projection of future inputs, past inputs and outputs onto the null-space of future noise is equal to  $E_{J|L+1}$ . We have separated the deterministic part of the system from the data by the projection in Equation (48). Hence we have the following theorem concerning the stochastic part of the system.

*Theorem 4.1 (Realization of the Toeplitz matrix  $H_{L+1}^s$ )*

The lower triangular Toeplitz matrix  $H_{L+1}^s$  with Markov parameters for the stochastic sub-system is given by the projection of past inputs, past outputs, and future inputs onto the null-space of the future outputs, i.e.

$$\begin{aligned}
 Z_{J|L+1}^s &\stackrel{\text{def}}{=} Y_{J|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp \\
 &= P_{L+1}^s E_{0|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp + H_{L+1}^s E_{J|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp \quad (49)
 \end{aligned}$$

where

$$P_{L+1}^s = O_{L+1}C_J^s - O_{L+1}A^J O_J^\dagger H_J^s \quad (50)$$

and when  $N \rightarrow \infty$ , then

$$Z_{J|L+1}^s \stackrel{\text{def}}{=} Y_{J|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp = P_{L+1}^s E_{0|L+1} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}^\perp + H_{L+1}^s E_{J|L+1} \quad (51)$$

which is a linear problem in  $H_{L+1}^s$ .

To see that this is a linear problem in the unknown matrix  $H_{L+1}^s$  one only has to write up the equations one obtains by using the structure of  $H_{L+1}^s$ , Equation (15), and a partition of the last term on the right hand side into sub-matrices.

This result with proof is to our knowledge new. However, the same solution resulting from a QR decomposition was presented in Di Ruscio (1995b).

From Theorem 4.1 and Equation (47) we immediately state the following important result. The extended observability matrix can be recovered from the column space of the matrix on the left hand side of the following matrix Equation (52).

**Theorem 4.2 (Realization of the extended observability matrix  $O_{L+1}$ )**

Given the following matrix equation

$$Z_{J|L+1} \stackrel{\text{def}}{=} (Y_{J|L+1} / \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}) U_{J|L+1}^\perp = O_{L+1} [A^J O_J^\dagger \quad C_J^d - A^J O_J^\dagger H_J^d \quad C_J^s - A^J O_J^\dagger H_J^s] \begin{bmatrix} Y_{0|J} \\ U_{0|J} \\ E_{0|L+1} / \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} \end{bmatrix} U_{J|L+1}^\perp \quad (52)$$

then the column space of the matrix  $Z_{J|L+1}$  coincide with the column space of the extended observability matrix  $O_{L+1}$  and the system order  $n$  of the SSM is given as the dimension of the column space.

The proof of Theorem 4.2 and Equation (52) is simple. From Theorem 4.1, Equation (49) we have

$$Y_{J|L+1} - Y_{J|L+1} / \tilde{Y} = P_{L+1}^s E_{0|J} - P_{L+1}^s E_{0|J} / \tilde{Y} + H_{L+1}^s E_{J|L+1} \quad (53)$$

where

$$\tilde{Y} \stackrel{\text{def}}{=} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} \quad (54)$$

and for the sake of simplicity the other matrices are defined according to Equations (49) and (50). Substituting Equation (53) into Equation (47) in order to remove the stochastic term  $H_{L+1}^s E_{J|L+1}$  gives

$$Y_{J|L+1} / \tilde{Y} = O_{L+1} A^J O_J^\dagger Y_{0|J} + P_{L+1}^d U_{0|J} + P_{L+1}^s E_{0|J} / \tilde{Y} + H_{L+1}^d U_{J|L+1} \quad (55)$$

Hence, it is necessary with an extra projection  $U_{J|L+1}^\perp$  on the right hand side in order to remove the deterministic term  $H_{L+1}^d U_{J|L+1}$  and in order to recover the extended

observability matrix. The matrix on the right hand side of Equation (52) is proportional with the extended observability matrix  $O_{L+1}$ . Hence, the column space of the matrix  $Z_{J|L+1}$  coincide with  $O_{L+1}$ . The dimension of the column space of the matrix  $Z_{J|L+1}$  is the order  $n$  of the SSM. QED

From Theorem 4.2 we immediately have the following corollary concerning the system dynamics.

*Corollary 4.1 (Identification of system dynamics)*

From  $Z_{J|L+1}$  defined in Theorem 4.2, Equation (52) we have the following relationship

$$Z_{J+1|L} = \tilde{A}Z_{J|L} \quad (56)$$

where

$$\tilde{A} \stackrel{\text{def}}{=} O_L A (O_L^T O_L)^{-1} O_L^T$$

and where  $Z_{J+1|L}$  is the  $L$  last  $(m \times K)$  block rows in  $Z_{J|L+1}$  and  $Z_{J|L}$  is the  $L$  first  $(m \times K)$  block rows in  $Z_{J|L+1}$ , i.e.

$$Z_{J+1|L} \stackrel{\text{def}}{=} (Y_{J+1|L} / \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}) U_{J|L+1}^+, \quad Z_{J|L} \stackrel{\text{def}}{=} (Y_{J+1|L} / \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}) U_{J|L+1}^+$$

Moreover, the column space of  $Z_{J|L}$  coincide with the column space of the extended observability matrix  $O_L$  and the dimension of the column space is equal to the order  $n$  of the SSM. A realization of  $n$ ,  $O_L$ ,  $A$ ,  $D$  and  $\tilde{A}$  is determined by using Equation (56) in combination with Algorithm 4.1.

The matrix on the left hand side of Equation (52) can be shown to be equivalent to

$$Z_{J|L+1} = Y_{J|L+1} W_c$$

$$W_c \stackrel{\text{def}}{=} U_{J|L+1}^+ W_1^T (W_1 U_{J|L+1}^+ W_1^T)^{-1} W_1 U_{J|L+1}^+ \quad (57)$$

Comparing Equation (57) with Equation (30) shows that these matrices are related. We have shown that the column space of both equations coincide with the column space of the extended observability matrix. The difference can be viewed as a different column weighting matrix  $W_c$ , on the right hand side of Equation (57). Equation (30) can be viewed as a special case factorization of equation (57) with  $k=J$  and  $W_c = U_{J|L+1}^+ W_1^T / K$ . We will later in Section 6 show that this last Equation (57) is extremely important and useful in order to partly compare other subspace identification methods.

From Theorem 4.1 we have the following result concerning the stochastic part of the system.

*Theorem 4.3 (Realization of  $\Delta$  and  $\tilde{C}$ )*

Assume that the number of system input and output observations,  $N \rightarrow \infty$ . Define

$$Z_{J|L+1}^s \stackrel{\text{def}}{=} Y_{J|L+1} \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}^+ \quad (58)$$

then we have

$$Z_{J+1|L}^s = \tilde{A}Z_{J|L}^s + \tilde{C}E_{J|L+1} \quad (59)$$

and

$$Z_{J+1|L}^s (Z_{J|L}^s)^+ = \tilde{C}E_{J|L+1} (Z_{J|L}^s)^+$$

where

$$\tilde{C} \stackrel{\text{def}}{=} [O_L C \quad H_L^s] - \tilde{A}[H_L^s \quad 0_{Lm \times m}]$$

The covariance matrix of the innovations  $\Delta$  is estimated directly from the column space of the left hand side matrix Equation (58), e.g.

$$Z_{J+L|1}^s (Z_{J|L}^s)^+ = FE_{J+L|1} (Z_{J|L}^s)^+ = FE_{J+L|1} \quad (60)$$

and

$$\Delta = FF^T \quad (61)$$

when

$$E_{J+L|1} \cdot E_{J+L|1}^T = I$$

$\Delta$

The importance of Theorem 4.3 is that it shows that the innovations covariance matrix can be estimated directly from the column space of the data matrix, Equation (58). The Kalman filter gain matrix  $C$  can be extracted when  $A$  and  $D$  is known.

Finally, we have the following Theorem 4.4 for the realization of the deterministic part of the system.

*Theorem 4.4 (Realization of  $\tilde{B}$ )*

Given

$$Z_{J|L+1}^d \stackrel{\text{def}}{=} Y_{J|L+1} / \begin{bmatrix} U_{J|L+1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix} \quad (62)$$

then we have

$$Z_{J+1|L}^d = \tilde{A}Z_{J|L}^d + \tilde{B}U_{J|L+1} \quad (63)$$

and

$$Z_{J+1|L}^d (Z_{J|L}^d)^+ = \tilde{B}U_{J|L+1} (Z_{J|L}^d)^+ \quad (64)$$

where

$$\tilde{B} \stackrel{\text{def}}{=} [O_L B \quad H_L^d] - \tilde{A}[H_L^d \quad 0_{Lm \times r}]$$

Theorem 4.4 is proved from Equation (55). Notice that the SSM matrix  $E$  can be estimated directly from Equation (64). This is so because of the structure of the ESSM matrix  $\tilde{B}$ .

All the projections in this section, Equations (49), (52) and (62), can be effectively computed from a QR decomposition, either directly from the projections defined in this section, or as will be shown in Section (5).

We will conclude this section by point out the relationship between the data matrices and the projection matrices. The data matrix with future system outputs and the projection matrices, (49) and (62) are related as

$$Y_{J|L+1} = Z_{J|L+1}^d + Z_{J|L+1}^s$$

$Z^d$  represents the outputs from the deterministic part of the system.  $Z^s$  represents the outputs from the stochastic part of the system.  $Y_{j|L+1}$  is the data matrix with future outputs from the combined deterministic and stochastic system.

### 5. Implementation with QR decomposition

We will here use the QR decomposition in order to compute the column space of the projection matrices derived in Section 4. The QR decomposition is also used in the subspace identification methods by Verhagen (1994) and Van Overschee and De Moor (1994).

Define the following QR decomposition

$$\frac{1}{\sqrt{K}} \tilde{Y} = RQ = \frac{1}{\sqrt{K}} \begin{bmatrix} U_{k|L+1} \\ W_i \\ Y_{k|L} \\ Y_{k+1|L} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & R_{42} & R_{43} & R_{44} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix} \quad (65)$$

where

$$R \in \mathbb{R}^{(r(L+1)+ni+2mL) \times (r(L+1)+ni+2mL)} \quad (66)$$

$$Q \in \mathbb{R}^{(r(L+1)+ni+2mL) \times K} \quad (67)$$

Note that this decomposition perhaps more precisely could have been defined as a lower Left,  $Q$ -orthogonal (LQ) decomposition. See Golub and Van Loan (1983) for the computation.

The QR decomposition can be viewed as a data compression step. The data matrix  $\tilde{Y}$  which usually have a large number of columns is compressed to a usually much smaller lower triangular matrix  $R$  which contain all relevant information of the system for which the data was generated. As we will show, the orthogonal  $Q$  matrix is not needed in the algorithm.

Note that the first  $(L-1)m$  rows in  $Y_{k+1|L}$  are common with the last  $(L-1)m$  rows in  $Y_{k|L}$ . This means that  $Y_{k+1|L}$  can be substituted with  $Y_{k+L|1}$  in the QR decomposition, Equation (65). This is utilized in the efficient implementation of the DSR algorithm. However, for the sake of simplicity we will present the results according to Equation (65).

By definition, the instrumental variable matrix  $W_i$ , is uncorrelated with  $E_{k|L+1}$ . We can therefore remove the noise matrix  $E_{k|L+1}$  from Equation (7) by post-multiplying with  $\frac{1}{K} W_i^T$ . We have from (65) that

$$\lim_{K \rightarrow \infty} \frac{1}{K} E_{k|L+1} W_i^T = \left( \lim_{K \rightarrow \infty} \frac{1}{K} E_{k|L+1} [Q_1^T \quad Q_2^T] \right) \begin{bmatrix} R_{21}^T \\ R_{22}^T \end{bmatrix} = 0 \quad (68)$$

Post-multiplying Equation (7) with  $[Q_1^T \quad Q_2^T]$ , using (68) and substituting for the corresponding  $R_{ij}$  sub-matrices from (65) gives

$$[R_{41} \quad R_{42}] = \tilde{A} [R_{31} \quad R_{32}] + \tilde{B} [R_{11} \quad 0] \quad (69)$$

which gives one matrix equation for  $\tilde{A}$  and one for both  $\tilde{A}$  and  $\tilde{B}$ . We will in the next Sections 5.1 and 5.2 show how the order  $n$  and the system quadruple  $(A, B, D, E)$  are computed from (69).

The stochastic part of the system, defined by the matrices  $C$  and  $\Delta$ , is computed from

$$R_{43} - \tilde{A}R_{33} = \tilde{C}E_{k|L+1}Q_3^T \quad (70)$$

$$R_{44} = \tilde{C}E_{k|L+1}Q_4^T \quad (71)$$

This will be shown in Section 5.3.

Note that the QR decomposition compresses the possible large data matrices into a number of (smaller) matrices which contain the information of the system. It is also interesting that the matrices (information) which define the deterministic part ( $A, B, D, E$ ) and the stochastic part ( $\Delta, C$ ) are separated by the QR decomposition. The user must specify the parameter  $k \geq J$  in Equation 65. See Figure 1 for a definition of the horizons involved. We recommend putting  $k = J$ . The matrix  $W_l$  with  $l = 0$ , Equation (24), is recommended for  $W_i$  in 65.

### 5.1. Realization of $A$ and $D$

We have from Equation (69) that

$$R_{42} = \tilde{A}R_{32} \quad (72)$$

and we choose

$$Z_{k+1|L} = R_{42} \quad (73)$$

$$Z_{k|L} = R_{32} = USV^T \quad (74)$$

in Algorithm (4.1) in order to determine  $A, D$  and the extended observability matrix  $O_L$ . The system order is determined by inspection of the dominant singular values of  $S$  or  $SS^T$ .

Note that the first  $(L-1)m$  rows in  $R_{42}$  is equal to the last  $(L-1)m$  rows in  $R_{32}$ . This is utilized in the efficient implementation of the DSR algorithm, in order to reduce the computational work.

Note also that if  $\tilde{A}$  is computed as the projection of  $R_{32}$  onto  $R_{42}$  then  $\tilde{A}$  takes a special canonical form. This is due to the common rows.

The  $A$  matrix may also be determined as follows

$$O_1 = O_L(1:(L-1)m, 1:n) \quad (75)$$

$$O_2 = O_L(m+1:Lm, 1:n) \quad (76)$$

$$A = (O_1^T O_1)^{-1} O_1^T O_2 \quad (77)$$

However, we must put  $L = L+1$  in this case if the extended observability matrix is estimated as the left singular vectors in (74). This will increase the computational work. This last method is the so called shift invariance method for computing the transition matrix  $A$  from the extended observability matrix, Kung (1978). The shift invariance method is used in the subspace algorithms (N4SID) by Van Overschee and De Moor (1994) and (MOESP) by Verhagen (1994). The parameter which defines the number of block rows in the N4SID and MOESP algorithms is denoted  $I$ . This parameter is related to the DSR parameter  $L$  as  $I = L+1$ . This is one of the differences between the DSR algorithm and the N4SID and MOESP algorithms.

This means that N4SID and MOESP computes a number  $Im = (L+1)m$  of singular values. However, the system order can only be chosen according to  $Lm$  of these singular values, i.e. the maximum system order which can be chosen for a specified parameter



$I = L + 1$  is  $n = Lm$ . For comparison, the DSR algorithm computes only a number  $Lm$  singular values and the maximum system order which can be chosen for a specified parameter  $L$  is  $n = Lm$ . Hence, the DSR algorithm seems to be more consistent with respect to choosing the system order as the number of non-zero singular values.

The shift invariance method can be included in the DSR algorithm, but to a higher computational expense. This strategy can be described as follows. The extended observability matrix can be estimated from the column space of the matrix formed from  $R_{32}$  and  $R_{42}$ . Compute the SVD

$$\begin{bmatrix} R_{32} \\ R_{42} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} S_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = \begin{bmatrix} U_{11} S_n V_1^T \\ U_{21} S_n V_1^T \end{bmatrix}$$

We then have that  $A$  and  $D$  are determined from the shift invariance method, e.g. from the left singular vectors as follows

$$O_L = U_{11} \quad (78)$$

$$O_L A = U_{21} \quad (79)$$

$$A = (U_{11}^T U_{11})^{-1} U_{11}^T U_{21} \quad (80)$$

$$D = U_{11}(1:m, 1:n) \quad (81)$$

## 5.2. Realization of $B$ and $E$

We have from Equation (69) that

$$\tilde{B}R_{11} = R_{41} - \tilde{A}R_{31} \quad (82)$$

$\tilde{B}$  can be determined directly from (82) if the input  $u$  is persistently exciting of order  $L + 1$ .  $R_{11}$  is non singular in this case. We have

$$\tilde{B} = (R_{41} - \tilde{A}R_{31})R_{11}^T(R_{11}R_{11}^T)^{-1} \quad (83)$$

The  $B$  and  $E$  matrices are then extracted from  $\tilde{B}$  as pointed out in Section 4.2

At this stage, the system order is identified (Algorithm 4.1). It is possible to determine  $B$  and  $E$  if the input is only persistently exciting of order  $p + 1$  where  $L_{\min} \leq p \leq L$ , directly without recomputing the algorithm with  $L = L_{\min}$  or  $L = p$ . The minimal observability index, for a given system order, is  $L_{\min} = n - \text{rank}(D) + 1$  when  $n \geq \text{rank}(D)$  and  $L_{\min} = 1$  when  $n \leq \text{rank}(D)$ .

Define

$$O_p = O_L(1:mp, 1:n) \quad (84)$$

$$\tilde{A}_p = O_p A (O_p^T O_p)^{-1} O_p^T \quad (85)$$

$$R_{11}^p = R_{11}(1:r(p+1), 1:r(p+1)) \quad (86)$$

$$R_{31}^p = R_{31}(1:mp, 1:r(p+1)) \quad (87)$$

$$R_{41}^p = R_{41}(1:mp, 1:r(p+1)) \quad (88)$$

We then have

$$\tilde{B}_p R_{11}^p = R_{41}^p - \tilde{A}_p R_{31}^p \quad (89)$$

This result is a consequence of Equation (7) with  $L$  substituted with  $p$ . Note that the minimal observability index, for a given system order, is  $p = n - \text{rank}(D) + 1$  when

$n \geq \text{rank}(D)$ . The  $B$  and  $E$  matrices are then extracted from  $\tilde{B}_p$  as shown in Section 4.2 and Algorithm 4.2 with  $L$  substituted with  $p$ .

### 5.3. Realization of $C$ and $\Delta$

#### Corollary 5.1 (Realization of $C$ and $\Delta$ )

Given the lower left triangular matrix  $R$ , determined by the QR decomposition in Equation (65). An estimate of the square root of the innovations covariance matrix is given by the  $m \times m$  lower right sub-matrix of  $R$ , i.e.

$$F = R_{44}(m(L-1)+1:mL, m(L-1)+1:mL) \quad (90)$$

and the estimate of the innovations covariance matrix is

$$\Delta = FF^T \quad (91)$$

Furthermore, when  $J > 1$ , then an estimate of the Kalman filter gain matrix  $C$  can be computed from

$$O_L CF = R_{43}(1:mL, 1:m) \quad (92)$$

If  $F$  is non-singular, then we have

$$C = (O_L^T O_L)^{-1} O_L^T R_{43}(1:mL, 1:m) F^{-1} \quad (93)$$

An estimate of the lower left block triangular Toeplitz matrix  $H_L^s$  for the stochastic subsystem  $(D, A, CF, F)$  is given by

$$H_L^s = R_{43}(1:mL, m+1:m(L+1)) \quad (94)$$

The estimate of the lower left block triangular Toeplitz matrix for the stochastic subsystem  $(D, A, C, I)$ , according to Equation (15), can be formed from the block columns  $F_1, \dots, F_L$  which can be computed from

$$[F_1 F \quad F_2 F \quad \dots \quad F_L F] = R_{43}(1:mL, m+1:m(L+1)) \quad (95)$$

$\Delta$

The stochastic subsystem is identified separately from the deterministic subsystem. The necessary separation into deterministic and stochastic subsystems are implicitly done by the QR decomposition.

The first  $(L-1)m$  rows in  $Y_{k+1|L}$  are common with the  $(L-1)m$  last rows in  $Y_{k|L}$ .  $Q_4$  is uncorrelated with  $U_{k|L+1}$ ,  $W_i$ ,  $Y_{k|L}$  and with the  $(L-1)m$  first rows in  $Y_{k+1|L}$ . The first  $(L-1)m$  rows in  $Y_{k+1|L} Q_4^T$  and  $E_{k|L+1} Q_4^T$  are therefore zero. We then have from (65) and the structure of  $\tilde{C}$ , given by (13) and (15), that

$$Y_{k+1|L} Q_4^T = \tilde{C} E_{k|L+1} Q_4^T = R_{44} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & F \end{bmatrix} \quad (96)$$

where  $R_{44} \in \mathbb{R}^{Lm \times Lm}$ . Hence, the square root of the innovations noise process covariance matrix is estimated directly as the  $m \times m$  lower left matrix, denoted  $F$ , in the lower triangular matrix  $R$  from the QR decomposition, Equation (65). Note that  $F$  also is lower left triangular and can be compared to a Cholesky factorization of  $\Delta$ . This result is believed to be of some importance. The result (91) is then clarified.

The matrices  $Q_i$ ,  $i = 1, 2, 3, 4$ , are orthogonal matrices and we have

$$\frac{1}{K} E_{k|L+1} Q_3^T = \begin{bmatrix} F & 0 & \dots & 0 \\ 0 & F & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & F \\ 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{(L+1)m \times Lm} \quad (97)$$

Another strategy is then to compute  $OCF$  from Equations (70) and (97), and an algorithm similar to Algorithm 4.2 This is formulated in the following Corollary 5.2.

*Corollary 5.2 (Realization of C)*

Given the sub-matrices  $R_{43}$  and  $R_{33}$  from the QR decomposition in Equation (67) and the ESSM transition matrix  $\tilde{A}$ . Define according to Equation (70)

$$\tilde{C}F \stackrel{\text{def}}{=} R_{43} - \tilde{A}R_{33} \quad (98)$$

The matrix  $O_L CF$  can then be computed from  $\tilde{C}F$ , e.g. by a procedure similar to Algorithm 4.2.

△

This strategy is found from Monte Carlo simulations to be the best one when the past horizon parameter is  $J = 1$ , but no significant difference when  $J > 1$ .

*5.4. Special remarks*

One advantage of the QR implementation of the algorithm is that potential ill-conditioning of the covariance matrices are concentrated in a certain triangular matrix. This ill-conditioning usually results from ill-conditioned noise processes (process noise and measurements noise) and due to rounding-off errors. Note that the triangular matrix  $R$  is the square root of the covariance matrix ( $H = \frac{1}{K} \tilde{Y} \tilde{Y}^T$  where  $\tilde{Y}$  is defined in (65)) and that the triangular matrix is computed without ever computing the covariance matrix. The method can therefore be defined as numerically stable.

The QR decomposition is not unique. The  $R$  matrix is post-multiplied by a diagonal permutation matrix  $E$  such that  $R := RE$  have positive diagonal elements. The diagonal elements of  $E$  are equal to the sign of the corresponding diagonal elements of  $R$  which was the result from the QR decomposition. Note also that  $Q := EQ$  and  $EE = I$ . This gives a more unique coordinate system for the estimated  $(A, B, D, E)$  quadruple. This scaling is also one of the reasons for the simple solutions for the  $C$  and  $\Delta$  matrices in Section 5.3. The scaling ensures that the diagonal blocks of (97) gets the same sign.

Note that common rows in the data matrices  $Y_{k|L}$  and  $Y_{k+1|L}$  can be removed prior to obtaining the QR decomposition in equation (67). It is also clear from the above that the orthogonal  $Q$  matrix ( $QQ^T = I$ ) is not needed in the algorithm. This will reduce the computational effort considerably. In fact, the QR factorization works on a data matrix of size only  $r(2L+1) + m(2L+1) \times K$  and not of size  $r(2L+1) + 3mL \times K$  as indicated in (65).

Another strategy for determining  $R$  is to first compute  $H = \frac{1}{K} \tilde{Y} \tilde{Y}^T$  where  $\tilde{Y}$  is defined in (65) and then the SVD,  $H = USV^T$ , followed by a QR decomposition of  $US^{\frac{1}{2}}$  in order to obtain the lower triangular matrix,  $R$ . This strategy reduced the number of flops and the accuracy of the  $R$  matrix when MATLAB was used for the computations. However,

no significant difference in the estimated models were observed. This strategy can be numerically ill-conditioned due to possible rounding errors when forming the product of rows in  $\tilde{Y}$  with columns in  $\tilde{Y}^T$  in order to compute the correlation matrix  $\tilde{Y}\tilde{Y}^T$ . This strategy is therefore not recommended.

Equation (82) is defined by the triangular factors for Equation (40) with  $k \geq J$ .

$$\tilde{B}U_{k|L+1}U_{k|L+1}^T = Y_{k+1|L}U_{k|L+1}^T - \tilde{A}Y_{k|L}U_{k|L+1} \quad (99)$$

It is also possible to extract the triangular factors for

$$\tilde{B}U_{0|L+1}U_{0|L+1}^T = Y_{1|L}U_{0|L+1}^T - \tilde{A}Y_{0|L}U_{0|L+1} \quad (100)$$

directly from the QR decomposition (65) when  $k = J = L$ . The first block row in  $U_{L|L+1}$  is equal to the last block row in  $U_{0|L+1}$ . Hence, the matrices in Equation (100) can be defined from the lower triangular  $R$  matrix, equation (65). It is therefore natural to choose

$$\tilde{B}[R_{11} \ R_{11}^0] = [R_{41} \ R_{41}^0] - \tilde{A}[R_{31} \ R_{31}^0] \quad (101)$$

for defining  $\tilde{B}$  and an equation for computing the  $B$  and  $E$  system matrices. Equation (101) consists of the triangular factors for both  $\frac{1}{K}U_{k|L+1}U_{k|L+1}^T$  and  $\frac{1}{K}U_{0|L+1}U_{0|L+1}^T$ . Equation (101) have effect for systems where the input signal is poor with frequencies, but gives no additional effect compared to (82) for e.g. white noise inputs.

Note that the stochastic part of the model is determined from QR and SV Decompositions only. The Markov parameters and the square root of the innovations covariance matrix are determined from a QR decomposition only. The Kalman filter gain matrix is determined from the Markov parameters and the extended observability matrix. No matrix Lyapunov or non-linear matrix Riccati equations has to be solved.

The method have in this work been illustrated for systems which is not strictly proper (only proper), i.e. the case when  $E \neq 0$  in the underlying model (2). The method can also be implemented to handle proper systems, i.e. systems where  $E$  is known to be zero. This can be done by deleting the last block row in  $U_{k|L+1}$  and the last block column in  $\tilde{B}$ , see Equation (7).

## 6. Comparison with existing algorithms

A comparison with the DSR algorithm and three different subspace algorithms will be given, i.e. N4SID, Van Overschee and De Moor (1994) and CVA, Larimore (1983), (1990), PO-MOESP, Verhagen (1994).

The first and common step in subspace identification algorithms is to estimate the extended observability matrix from the column space of a known data matrix. We will therefore concentrate our discussion about the similarities and differences in the way those methods estimate the extended observability matrix. We will only briefly discuss how the system matrices are estimated by the different methods.

It is shown in Van Overschee and De Moor (1996) that the different methods are related through certain row and column weighting with the N4SID data matrix as the key matrix.

We will below present a different approach, with the matrix  $Z_{J|L+1}$  defined in Section 4.3, Theorem 4.2 and Equation (57), as the key matrix.

Multiplying Equation (57) from left with an extra row weighting matrix  $W_r$  and using the SVD as discussed in Algorithm 4.1 gives

$$W_r Z_{J|L+g} = W_r Y_{J|L+g} W_c = U_1 S_n V_1^T + U_2 S_2 V_2^T \quad (102)$$

where  $g$  is an integer parameter and

$$W_c \stackrel{\text{def}}{=} U_{j|L+1}^+ W_1^T (W_1 U_{j|L+1}^+ W_1^T)^{-1} W_1 U_{j|L+1}^+$$

The extended observability matrix can be estimated as, e.g.

$$O_{L+g} = W_r^{-1} U_1 \quad (103)$$

The matrix  $W_c$  is symmetric and can be written as

$$W_c \stackrel{\text{def}}{=} \underbrace{U_{j|L+1}^+ W_1^T (W_1 U_{j|L+1}^+ W_1^T)^{-1}}_{w_c^2} \underbrace{(W_1 U_{j|L+1}^+ W_1^T)^{-1} W_1 U_{j|L+1}^+}_{(w_c^2)^T} \quad (104)$$

From the above factorization of the matrix  $W_c$  we have at least four matrices ( $W_c^i$ ,  $i = 1, \dots, 4$ ) which all have essentially the same column space as  $W_c$ , i.e.

$$W_c^1 = U_{j|L+1}^+ W_1^T (W_1 U_{j|L+1}^+ W_1^T)^{-1} W_1 U_{j|L+1}^+$$

$$W_c^2 = U_{j|L+1}^+ W_1^T (W_1 U_{j|L+1}^+ W_1^T)^{-1}$$

$$W_c^3 = U_{j|L+1}^+ W_1^T (W_1 U_{j|L+1}^+ W_1^T)^{-1} K$$

$$W_c^4 = U_{j|L+1}^+ W_1^T \frac{1}{K}$$

$$W_c^5 = U_{j|L+1}^+$$

The matrix  $W_c^5$  is only sufficient for purely deterministic systems and is shown for the sake of completeness.

These column weighting matrices are used in the DSR algorithm which are presented in this work. The past horizon parameter  $J$  is usually chosen as  $J \geq L$ . The parameter  $g = 0$  in Equation (102) and the row weighting matrix is the  $Lm \times Lm$  identity matrix, denoted  $W_r = I_{Lm}$ . Algorithm 4.1 is used in order to identify the extended observability matrix  $O_L$  from the column space of the matrix  $Z_{j|L}$ .

We will now illustrate the similarity and difference with two published algorithms, CVA by Larimore (1990) and PO-MOESP by Verhagen (1994).

### 6.1. PO-MOESP

The PO-MOESP algorithm in Verhagen (1994) estimates the extended observability matrix  $O_{L+1}$  from Equations (102) and (103) with the following matrices

$$\left. \begin{aligned} W_c &= U_{L+1|L+1}^+ W_1^T (W_1 U_{L+1|L+1}^+ W_1^T)^{-1} W_1 U_{L+1|L+1}^+ \\ W_r &= I_{(L+1)m} \\ g &= 1 \end{aligned} \right\} \text{PO-MOESP} \quad (105)$$

From Theorem 4.2 and the factorization in Equation (104) we conclude that the two algorithms PO-MOESP and DSR estimate the extended observability matrix from a known data matrix which essentially have the same column space. The only difference is that PO-MOESP estimate the extended observability matrix  $O_{L+1}$ , of larger size than DSR, in order to use the shift invariance method for extracting the system matrix  $A$  as explained in Section 5.1.

Using the triangular factors from the QR decomposition in Section 5 then we have

$$R_{32} R_{22}^T (R_{22} R_{22}^T)^{-1} R_{22} Q_2 = (U_1 S_n V_1^T + U_2 S_2 V_2^T) Q_2 \quad (106)$$

where the orthogonal matrix  $Q_2$  is not needed because the column space can be estimated as the matrix  $U_1$ .

A major difference is that the PO-MOESP algorithm does not estimate the stochastic part of the model. DSR estimate the Kalman gain and innovations covariance matrix directly from the data as shown in Sections 4.3 and 5.3.

### 6.2. Canonical Variate Analysis

The CVA algorithm in Larimore (1990) estimates the extended observability matrix  $O_{L+1}$  from Equations (102) and (103) with the following matrices

$$\left. \begin{aligned} W_c &= U_{L+1|L+1}^+ W_1^T (W_1 U_{L+1|L+1}^+ W_1^T)^{-1} \\ W_r &= (Y_{L+1|L+1} U_{L+1|L+1}^+ Y_{L+1|L+1}^T)^{-1} \\ g &= 1 \end{aligned} \right\} \text{CVA} \quad (107)$$

As we can see, the column weighting matrix  $W_c$  used by the CVA algorithm fit into the factorization in Equation (104). A difference is that the CVA algorithm uses a row weighting matrix  $W_r$ .

The only difference is that the DSR algorithm takes the SVD of a matrix of size only  $Lm \times J(r+m)$  where usually  $J=L$ , in order to identify  $O_L$ . The other methods takes the SVD of a matrix of size  $(L+1)m \times J(r+m)$ , in order to identify  $O_{L+1}$ . The reason is to separate out the sub-matrices  $O_L$  and  $O_L A$  from  $O_{L+1}$ . See also Section 5.1 for a discussion.

From Theorem 4.2 and the factorization in Equation (104) we conclude that the two algorithms, CVA, and DSR essentially have the same column space.

An interpretation of the CVA algorithm is that the system order is estimated as the number of principal angles between the matrix  $Y_{L+1|L+1} U_{L+1|L+1}^+$  and  $W_1 U_{L+1|L+1}^+$  different from  $\pi/2$ . The principal angles can be effectively computed using the SVD, see e.g., Van Overschee (1995), p. 29 and Golub and Van Loan (1989), p. 428.

By using the triangular factors as shown in Section 5 we get the following method for computing the principal angles

$$(R_{32} R_{32}^T + R_{33} R_{33}^T) - \frac{1}{2} R_{32} R_{22}^T (R_{22} R_{22}^T)^{-1} = U_1 S_n V_1^T + U_2 S_2 V_2^T \quad (108)$$

The system order is here identified as the number of singular values equal to one.

The next step in the CVA algorithm is to define a memory which defines a valid sequence of system states. The system matrices can then be estimated from a least squares problem.

### 6.3. N4SID

The N4SID algorithm is different. The following weighting matrices are used

$$\left. \begin{aligned} W_c &= (W_1 U_{L+1|L+1}^+ W_1^T)^{-1} W_1 \\ W_r &= I_{(L+1)m} \\ g &= 1 \end{aligned} \right\} \text{N4SID} \quad (109)$$

The column weighting matrix used in the N4SID algorithm do generally not have the same column space as  $W_c$  or any of the column weighting matrices which results from Equation (104) and Theorem 4.2. This is possibly the reason why N4SID gives bad results for deterministic input signals.

In Viberg (1995) it is pointed out that the difference in the PO-MOESP and N4SID algorithms is only the extra projection  $U_{L+1|L+1}^+$ . However, it is also claimed that the resulting subspace estimates should therefore have very similar properties. From

Theorem 4.2 and the above discussion we conclude that this conclusion in Viberg (1995) is wrong. This is illustrated in example 2, Section 7.2.

From the above discussion we have the following relationship between the column weighting matrix  $W_c$  in Equation (104) and the matrix  $W_c$  in (109) used by N4sid.

$$W_c = W_c^{N4sid} U_{L+1|L+1}^+ \quad (110)$$

In Theorem 4.2, Equations (52) and (55), it is proved that the extra projection  $U_{L+1|L+1}$  is necessary in order to remove the deterministic term  $H_{L+1}^d U_{L+1|L+1}$  and establish the data matrix  $Z_{L+1|L+1}$  which have the same column space as the extended observability matrix. See also Example 2, Section 7.2, for an illustration.

The N4SID method computes the SVD of the data matrix defined in Equation (102) with the above matrices  $W_r$  and  $W_c$ , Equation (109). If the triangular factors as shown in Section 5 are used then we have

$$R_{32} R_{22}^T (R_{22} R_{22}^T)^{-1} [R_{21} R_{22}] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = (U_1 S_n V_1^T + U_2 S_2 V_2^T) \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \quad (111)$$

The orthogonal matrices  $Q_1$  and  $Q_2$  are not used. The system order is identified as the number of non-zero singular values and the extended observability matrix  $O_{L+1}$  is estimated from the column space. The rest of the N4SID algorithm can briefly be described as follows. From the extended observability matrix  $O_{L+1}$  and the system data, then a valid sequence of system states are estimated and a least squares problem is solved in order to construct the system matrices  $A, B, D, E$ . The covariance matrices for the process and measurements noise are then identified from a residual and a Riccati equation is solved in order to construct the Kalman filter gain matrix  $C$  and the innovations covariance matrix  $\Delta$ .

The DSR method does not use state sequences and the Kalman gain matrix  $C$  and the innovations covariance matrix  $\Delta$  are constructed directly from the data, without recursions of non-linear matrix Riccati equations.

#### 6.4. The main differences and similarities

Both algorithms, N4SID and CVA, estimates in the first instance a sequence of states. When the states are known, the state space model matrices can be determined by simple linear regression. Both methods must solve a matrix Riccati equation in order to identify the Kalman filter gain and the innovations covariance matrices.

The DSR algorithm is based on first writing up an extended state space model (ESSM) where the unknown states are eliminated from the problem. The ESSM shows us the relationship between the known data matrices and the SSM matrices. Hence, the DSR algorithm does not have any problems with unknown states; unknown initial values, etc. The state space model matrices are then extracted from the ESSM. DSR estimates the Kalman gain and innovations covariance matrices directly from the data, without recursions of non-linear matrix equations, e.g. the Riccati equation.

The PO-MOESP algorithm does not estimate the stochastic part of the model. We have shown that the CVA, PO-MOESP and DSR algorithms gives consistent estimates of the extended observability matrix. The algorithms fit into the same Theorem 4.2. We have shown that the N4SID algorithm in general does not give consistent estimates of the extended observability matrix. However, it will give consistent results if an extra projection of future inputs is included. From the above discussion we have the following

relationship between the column weighting matrix  $W_c$  in Equation (104) and the matrices used by PO-MOESP, CVA and N4SID. See (105), (107) and (109).

$$W_c = W_c^{\text{PO-MOESP}} = W_c^{\text{CVA}} (W_c^{\text{CVA}})^T = W_c^{\text{N4SID}} U_{L+1|L+1}^+$$

This is the most important similarities and differences between the method presented in this work and the existing methods which are published.

## 7. Numerical examples

### 7.1. Example 1: Monte Carlo Simulation

A single input single output (SISO) system with one state is chosen to compare the algorithm presented in this paper, which is entitled DSR (Deterministic and Stochastic system identification and Realization), with two other algorithms, CVA (which stands for Canonical Variate Analysis, Larimore (1983), (1990)) and the prediction error method implemented in the MATLAB function ARMAX (i.e., in the system identification toolbox, Ljung (1991)).

$$x_{k+1} = 0.9x_k + 0.5u_k + 0.6e_k \quad (112)$$

$$y_k = 1.0x_k - 1.0u_k + e_k \quad (113)$$

Three types of input signals were used. One input equal to a sum of four sinusoid signals,  $u^1$ , one input equal to a white noise signal with unit covariance,  $u^2$ , and one equal to a sine,  $u^3$ .

$$u^1 \quad u_k = 0.2(\sin(\frac{k}{25}) + \sin(\frac{k}{10}) + \sin(\frac{k}{5}) + \sin(k))$$

$$u^2 \quad \text{White noise, unit covariance}$$

$$u^3 \quad u_k = \sin(k)$$

For each input the time series  $(y_k, u_k)$  was generated by simulating the model with 100 different white noise sequences  $e_k$  also with unit variance.

The DSR algorithm parameter  $L$  was changed from 1 to 5 and the CVA parameter  $I$  from 2 to 6. For each  $L$  and  $I$ , the mean and standard deviation of the parameters of the 100 different estimated models are presented in Tables 1 to 6. The results obtained by the ARMAX algorithm are also shown in the tables. See Ljung (1991) for the description of the parameters  $nn = [1, 2, 1, 0]$  which is used as arguments to ARMAX.

The true deterministic system quadruple is denoted  $(a, b, d, e) = (0.9, 0.5, 1, -1)$  and the deterministic steady state gain and deterministic zero are denoted  $H^d(1) = 4.0$  and  $p_d(1) = 1.4$ , respectively. The parameters in the stochastic part of the model are  $(c, \Delta) = (0.6, 1)$ . The stochastic steady state gain and stochastic zero are denoted  $H^s(1) = 7.0$  and  $p_s(1) = 0.3$ , respectively. The signal to noise ratio is approximately 0.4, hence, the identification problem is not simple.

The CVA algorithm sometimes estimated systems with negative (b, d) parameters, i.e. sometimes an estimated quadruple  $(a, b, d, e)$  and sometimes  $(a, -b, -d, e)$ . This happened with the algorithm parameter  $I = 3$  and with a random input signal. It is believed that this can be avoided by using a scaling similar to that presented in Section 5.4.

The results are very good for both the DSR and the CVA algorithms, see Tables (1) and (2). There are small differences in the estimated models for both methods when  $N$  is large, see Tables (1) and (2). This indicates that the asymptotic statistical



Table 1. Mean of the parameters of 100 different estimated models: input type  $u^1$ ,  $N = 10000$

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.8998	0.4965	1.0000	-1.0021	1.3975	3.9598	0.6088	1.0045	0.2910	7.0910
DSR	L = 2	0.8995	0.4971	1.0000	-1.0021	1.3979	3.9516	0.6003	0.9989	0.2992	6.9867
DSR	L = 3	0.8998	0.4965	1.0000	-1.0021	1.3975	3.9604	0.5992	0.9985	0.3006	6.9940
DSR	L = 4	0.8996	0.4967	1.0000	-1.0019	1.3977	3.9572	0.5992	0.9981	0.3004	6.9875
DSR	L = 5	0.8998	0.4963	1.0000	-1.0020	1.3975	3.9644	0.5989	0.9977	0.3009	6.9976
CVA	L = 2	0.8996	0.4967	1.0000	-1.0021	1.3977	3.9545	0.6008	1.0046	0.2988	7.0015
CVA	L = 3	0.8996	0.4972	1.0000	-1.0026	1.3980	3.9572	0.5995	1.0000	0.3001	6.9849
CVA	L = 4	0.8998	0.4952	1.0000	-1.0034	1.3961	3.9471	0.5990	0.9994	0.3008	6.9930
CVA	L = 5	0.8996	0.4984	1.0000	-1.0012	1.4003	3.9714	0.5992	0.9993	0.3003	6.9830
CVA	L = 6	0.8996	0.4975	1.0000	-1.0016	1.3992	3.9659	0.5993	0.9993	0.3003	6.9872
ARMAX	nn	0.8995	0.4982	1	-1.0024	1.3987	3.9604	0.5995		0.3000	6.9769

Table 2. Standard deviation of the parameters of 100 different estimated models: input type  $u^1$ ,  $N = 10000$ 

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.0054	0.0269	0	0.0677	0.0436	0.3020	0.0101	0.0153	0.0130	0.3170
DSR	L = 2	0.0052	0.0261	0	0.0678	0.0434	0.3063	0.0099	0.0152	0.0121	0.3021
DSR	L = 3	0.0054	0.0257	0	0.0678	0.0434	0.3190	0.0102	0.0152	0.0128	0.3211
DSR	L = 4	0.0056	0.0259	0	0.0678	0.0435	0.3236	0.0105	0.0151	0.0135	0.3309
DSR	L = 5	0.0059	0.0261	0	0.0677	0.0433	0.3328	0.0111	0.0151	0.0145	0.3447
CVA	I = 2	0.0052	0.0266	0	0.0678	0.0437	0.2959	0.0101	0.0153	0.0124	0.2994
CVA	I = 3	0.0051	0.0294	0	0.0691	0.0471	0.3423	0.0097	0.0152	0.0118	0.3009
CVA	I = 4	0.0053	0.0333	0	0.0702	0.0517	0.3920	0.0096	0.0152	0.0118	0.3155
CVA	I = 5	0.0055	0.0381	0	0.0704	0.0572	0.4456	0.0096	0.0151	0.0119	0.3236
CVA	I = 6	0.0058	0.0376	0	0.0699	0.0563	0.4554	0.0096	0.0152	0.0122	0.3391
ARMAX	nn	0.0051	0.0269	0	0.0662	0.0420	0.2925	0.0095		0.0115	0.2983

Table 3. Mean of the parameters of 100 different estimated models: input type  $u^2$ ,  $N = 200$

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.8861	0.5074	1.0000	-0.9922	1.4047	4.0874	0.6093	0.9961	0.2768	7.0385
DSR	L = 2	0.8884	0.5023	1.0000	-0.9945	1.3991	3.9833	0.6045	0.9699	0.2839	6.9509
DSR	L = 3	0.8886	0.5016	1.0000	-0.9971	1.3980	3.9781	0.6094	0.9476	0.2792	7.0266
DSR	L = 4	0.8891	0.4998	1.0000	-0.9932	1.3985	4.0110	0.6128	0.9218	0.2763	7.1207
DSR	L = 5	0.8902	0.4999	1.0000	-0.9940	1.3992	4.0741	0.6139	0.8991	0.2763	7.2495
CVA	I = 2	0.8865	0.5077	1.0000	-0.9920	1.4056	3.9629	0.6027	1.0102	0.2838	6.8462
CVA	I = 3	0.8865	0.5031	1.0000	-0.9954	1.3996	3.8955	0.6072	0.9953	0.2793	6.8840
CVA	I = 4	0.8859	0.5004	1.0000	-0.9974	1.3957	3.8534	0.6111	0.9916	0.2749	6.9022
CVA	I = 5	0.8852	0.4994	1.0000	-0.9973	1.3938	3.8399	0.6122	0.9844	0.2730	6.9141
CVA	I = 6	0.8840	0.5007	1.0000	-0.9307	1.3960	3.7969	0.6145	0.9801	0.2695	6.8643
ARMAX	nn	0.8864	0.5036	1.0000	-0.9935	1.3993	3.8848	0.5987	0.9971	0.2877	6.7799

Table 4. Standard deviation of the parameters of 100 different estimated models: input type  $u^2$ ,  $N = 200$ 

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.0379	0.0957	0	0.0725	0.1320	2.3261	0.0101	0.0907	0.0884	2.4984
DSR	L = 2	0.0351	0.0907	0	0.0716	0.1222	1.8766	0.0131	0.0889	0.0915	1.9967
DSR	L = 3	0.0357	0.0931	0	0.0793	0.1234	1.8585	0.0158	0.0887	0.0964	2.0583
DSR	L = 4	0.0363	0.0945	0	0.0844	0.1227	1.9096	0.0183	0.0872	0.1046	2.1884
DSR	L = 5	0.0361	0.0977	0	0.0895	0.1212	1.9666	0.0210	0.0865	0.1095	2.4102
CVA	I = 2	0.0354	0.0956	0	0.0726	0.1325	1.9081	0.0104	0.0936	0.0844	1.9693
CVA	I = 3	0.0351	0.0987	0	0.0737	0.1374	1.8680	0.0106	0.0959	0.0839	2.0161
CVA	I = 4	0.0357	0.0992	0	0.0768	0.1375	1.8757	0.0115	0.0931	0.0871	2.0303
CVA	I = 5	0.0367	0.0994	0	0.0778	0.1347	1.9511	0.0119	0.0951	0.0904	2.1405
CVA	I = 6	0.0376	0.0983	0	0.0785	0.1331	1.8513	0.0123	0.0955	0.0974	2.0864
ARMAX	nn	0.0348	0.0921	0	0.0675	0.1225	1.7760	0.0723	0.0933	0.0867	1.8969

Table 5. Mean of the parameters of 100 different estimated models: input type  $u^3$ ,  $N = 500$

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.8952	0.4927	1.0000	-1.0011	1.3923	4.0591	0.6013	1.0006	0.2939	7.0975
DSR	L = 2	0.8938	0.4925	1.0000	-1.0021	1.3900	3.9881	0.5961	0.9879	0.2977	6.9587
DSR	L = 3	0.8934	0.4923	1.0000	-1.0026	1.3893	3.9382	0.6000	0.9784	0.2934	6.9510
DSR	L = 4	0.8931	0.4924	1.0000	-1.0022	1.3893	3.9376	0.6018	0.9695	0.2914	6.9572
DSR	L = 5	0.8936	0.4930	1.0000	-1.0024	1.3903	3.9738	0.6030	0.9612	0.2906	7.0060
CVA	I = 2	0.8944	0.4923	1.0000	-1.0017	1.3906	3.9908	0.5944	1.0047	0.2999	6.9579
CVA	I = 3	0.8933	0.4921	1.0000	-1.0013	1.3894	3.9558	0.5959	0.9998	0.2974	6.9273
CVA	I = 4	0.8931	0.4915	1.0000	-1.0000	1.3893	3.9105	0.5966	0.9992	0.2965	6.8970
CVA	I = 5	0.8930	0.4916	1.0000	-1.0003	1.3892	3.9064	0.5967	0.9986	0.2963	6.8926
CVA	I = 6	0.8931	0.4905	1.0000	-0.9997	1.3886	3.8941	0.5972	0.9977	0.2960	6.8930
ARMAX	nn	0.8936	0.4927	1.0000	-1.0024	1.3898	3.9354	0.5952	0.9980	0.2983	6.8954

Table 6. Standard deviation of the parameters of 100 different estimated models: input type  $u^3$ ,  $N = 500$ 

Alg.	Par.	a	b	d	e	$p_d(1)$	$H^d(1)$	c	$\Delta$	$p_s(1)$	$H^s(1)$
DSR	L = 1	0.0277	0.0658	0	0.0768	0.1019	1.6168	0.0409	0.0644	0.0542	1.4790
	L = 2	0.0274	0.0660	0	0.0752	0.0996	1.6257	0.0442	0.0637	0.0580	1.4551
	L = 3	0.0268	0.0660	0	0.0765	0.0993	1.5491	0.0456	0.0638	0.0583	1.4129
	L = 4	0.0272	0.0661	0	0.0757	0.1000	1.5699	0.0470	0.0626	0.0600	1.4067
	L = 5	0.0275	0.0665	0	0.0760	0.1000	1.6054	0.0479	0.0640	0.0609	1.4341
CVA	I = 2	0.0269	0.0659	0	0.0759	0.1002	1.5547	0.0417	0.0653	0.0542	1.4149
	I = 3	0.0275	0.0658	0	0.0755	0.0991	1.5903	0.0431	0.0642	0.0560	1.4462
	I = 4	0.0268	0.0666	0	0.0748	0.0995	1.5164	0.0415	0.0645	0.0532	1.3767
	I = 5	0.0267	0.0664	0	0.0752	0.0990	1.5091	0.0420	0.0645	0.0539	1.3780
	I = 6	0.0262	0.0667	0	0.0760	0.1007	1.5003	0.0433	0.0660	0.0539	1.3668
ARMAX	nn	0.0260	0.0663	0	0.0747	0.0995	1.5207	0.0421	0.0643	0.0535	1.3581

distribution of the parameter estimates is the same. The example indicate that both the DSR and CVA algorithms are insensitive to variation in the number of block rows.

The DSR algorithm is found to be approximately better than the CVA when the number of samples are small (for this example and with  $N = 200$  and  $N = 500$  samples) see Tables 3 to 6.

It is also interesting to observe that the results from DSR are as good as the results from the ARMAX function, even for a simple SISO system. This indicates that DSR gives asymptotically statistical optimal results for this example, both for purely deterministic inputs ( $u^1$  and  $u^3$ ) and stochastic input sequences ( $u^2$ ). Note that the prediction error method (ARMAX function) is based on iterative optimization but that the DSR algorithm only is based on SVD and QR decompositions. Prediction error methods are rather complicated for MIMO systems while the DSR algorithm is very simple.

Figures 2 to 8 are included in order to illustrate the asymptotic mean and variance properties of the DSR algorithm for varying horizon parameters  $L$  and  $J$ . The figures illustrate that the algorithm is numerically robust and that the estimates are consistent and fairly insensitive for the parameters  $L$  and  $J$ .

### 7.2. Example 2

We will in this example investigate the problem with colored input signals and the N4SID algorithm.

Consider the same SISO one state example as in Example 1, Equations (112) and (113). The two input signals were chosen. One equal to a pure sinusoid signal, input type  $u^3$ , and one equal to a white noise sequence with unit variance, input type  $u^2$ . The inputs are the same as defined in Example 1. The number of samples was fixed to  $N = 500$ . The standard deviation of the innovation was varied from  $\Delta^{0.5} = 0$  to  $\Delta^{0.5} = 0.01$  in order to investigate the sensitivity for noise. The number of block rows in the data matrices was chosen as  $L = 2$ .

The extended observability matrix  $O_3$  was estimated from the column space of the matrix  $Z_{L+1|L+1}$ , Equation (102). The dimension of the column space is estimated as the number of "non zero" singular values, see Figure 9. We have also introduced the normalized singular value  $(s_1 - s_2)/s_1$  as shown in Figure 10. This means that when  $(s_1 - s_2)/s_1 = 1$  then the number of states is  $n = 1$ .

The conclusions from Figures 9 and 10 are that the DSR algorithm gives reasonable estimates for both the system order and the actual pole (the CVA and PO-MOESP gives essentially the same results for this example) and that the N4SID algorithm does not work at all for this system with a pure deterministic sinusoid input signal ( $u_k = \sin(k)$ ). However, note that when the input was changed to a white noise sequence (input type  $u^3$ ) then the two algorithms gave essentially the same singular values as well as pole estimates.

### 7.3. Example 3

A two input two output system with the following model matrices is considered.

$$A = \begin{bmatrix} 1.5 & 1 & 0.1 \\ -0.7 & 0 & 0.1 \\ 0 & 0 & 0.85 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (114)$$

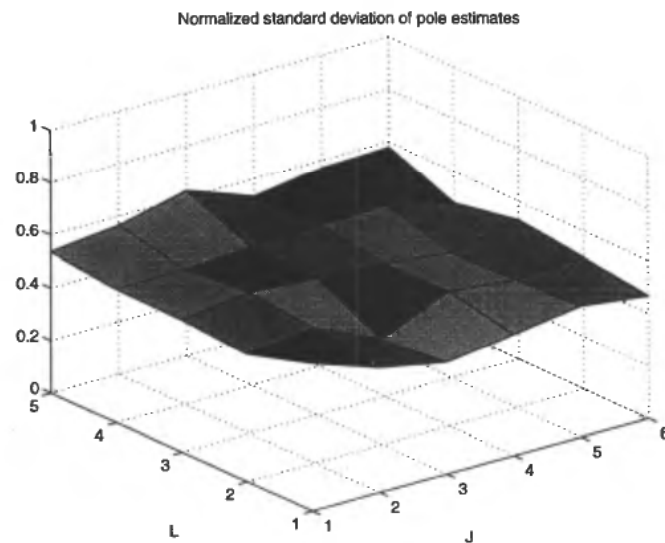


Figure 3. The standard deviation multiplied (normalized) with  $N^{1/2}$  of the pole estimates for a Monte Carlo simulation with varying past horizon parameter  $J$  and varying identification parameter  $L$ . The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sum of 5 sinusoid signals ( $u^1$ ). The maximum standard deviation of the pole estimates was  $0.6265/N^{1/2}$  at  $L = 5$  and  $J = 3$ . The minimum standard deviation was  $0.4005/N^{1/2}$  at  $L = 2$  and  $J = 3$ .

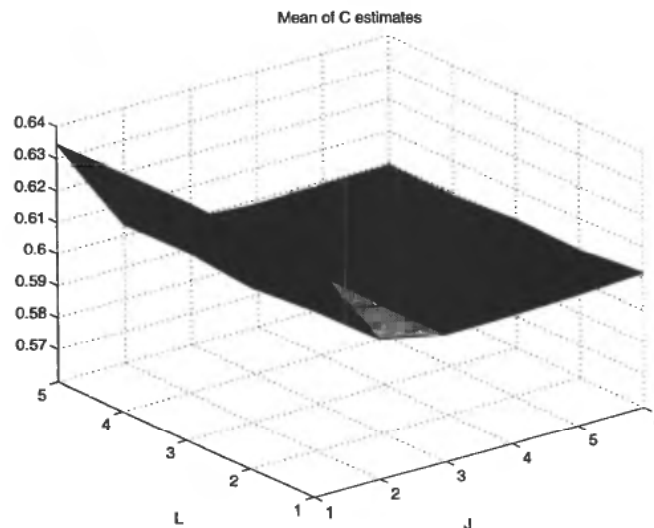


Figure 4. The mean value of the Kalman filter gain estimates for a Monte Carlo simulation with varying past horizon parameter  $J$  and identification parameter  $L$ . The estimates is computed by the algorithm in Corollary 5.1. The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sinusoid signal ( $u^3$ ). The actual value is  $C = 0.6$ . The mean of the estimates at the flat region described by  $3 \leq J \leq 6$  and  $1 \leq L \leq 5$  was 0.6000. This indicates that the estimates are consistent for past horizon parameters  $J > 2$  independently of the choice of  $L$ , but clearly biased for  $J = 1$ . The estimates for  $J = 2$  should be further investigated.



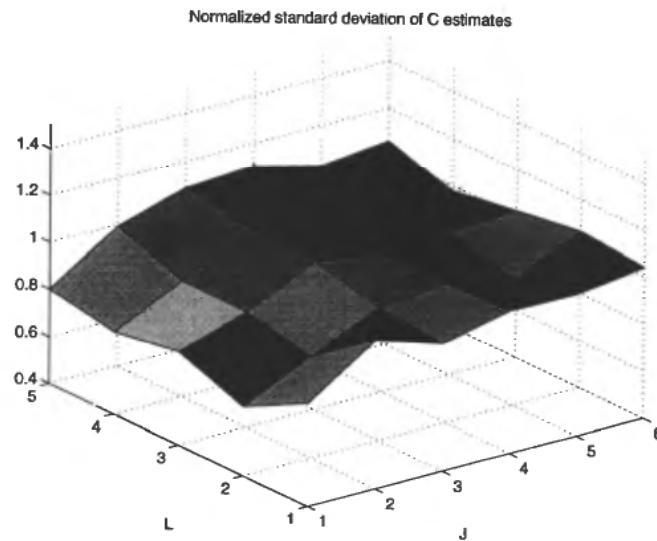


Figure 5. The standard deviation multiplied (normalized) with  $N^{1/2}$  of the Kalman filter gain estimates for a Monte Carlo simulation with varying past horizon parameter  $J$  and varying identification parameter  $L$ . The estimates is computed by the algorithm in Corollary 5.1. The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sinusoid signal ( $u^3$ ). The following parameters is found from the region with consistent estimates. The maximum standard deviation of the pole estimates was  $1.2135/N^{1/2}$  at  $L = 4$  and  $J = 5$ . The minimum standard deviation was  $0.9344/N^{1/2}$  at  $L = 2$  and  $J = 5$ . The mean of all standard deviations was  $1.0315/N^{1/2}$ .

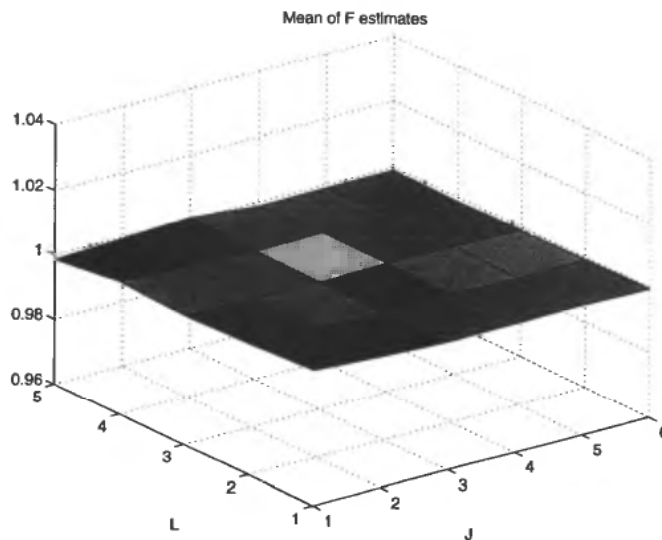


Figure 6. The mean value of the square root of the innovations variance estimates  $F$  for a Monte Carlo simulation with varying past horizon parameter  $J$  and identification parameter  $L$ . The estimates is computed by the algorithm in Corollary 5.1. The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sinusoid signal ( $u^3$ ). The actual parameter value is  $F = 1$ . The mean of all the estimates in the figure is  $0.9996$  with a standard deviation of  $7.4 \times 10^{-4}$ . This indicates that the estimates are consistent.

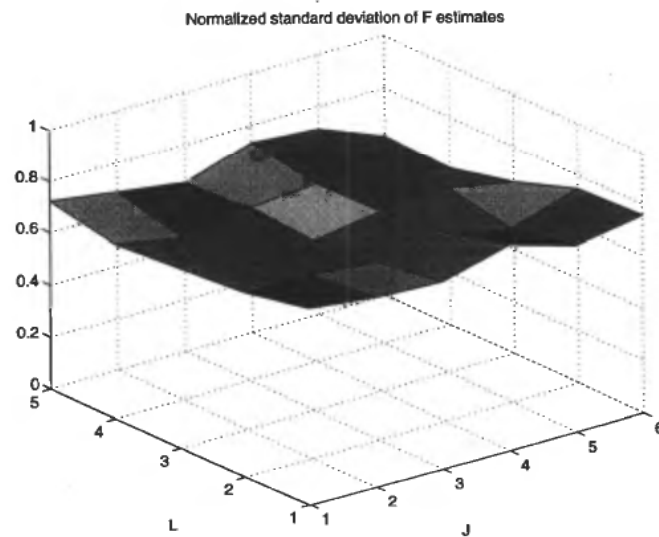


Figure 7. The standard deviation multiplied (normalized) with  $N^{1/2}$  of the innovations variance estimates  $F$  for a Monte Carlo simulation with varying past horizon parameter  $J$  and varying identification parameter  $L$ . The estimates is computed by the algorithm in Corollary 5.1. The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sinusoid signal ( $u^1$ ). The minimum standard deviation was  $0.6016/N^{1/2}$  at  $L = 4$  and  $J = 6$ .

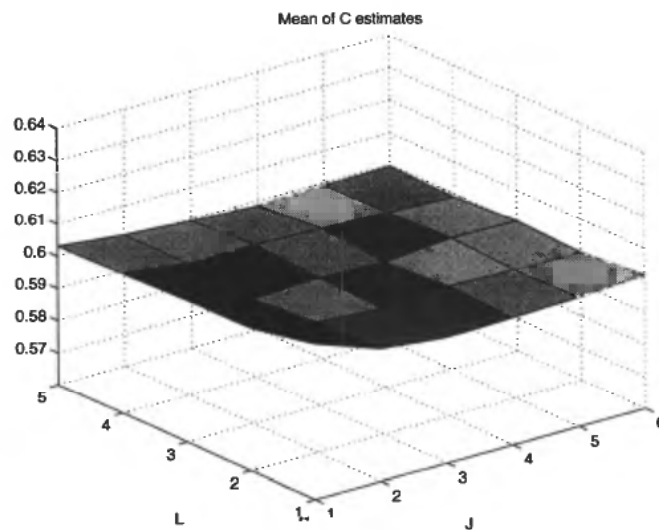


Figure 8. The mean value of the Kalman filter gain estimates for a Monte Carlo simulation with varying past horizon parameter  $J$  and identification parameter  $L$ . The estimates is computed by the algorithm in Corollary 5.2. The number of samples in each simulation was  $N = 15000$  and the number of simulations for each pair  $L, J$  was 100. The input was a sinusoid signal ( $u^3$ ). The actual value is  $C = 0.6$ . The bias for  $J = 1$  as shown in Figure 4 when the estimates was computed as in Corollary 5.2 is approximately eliminated.

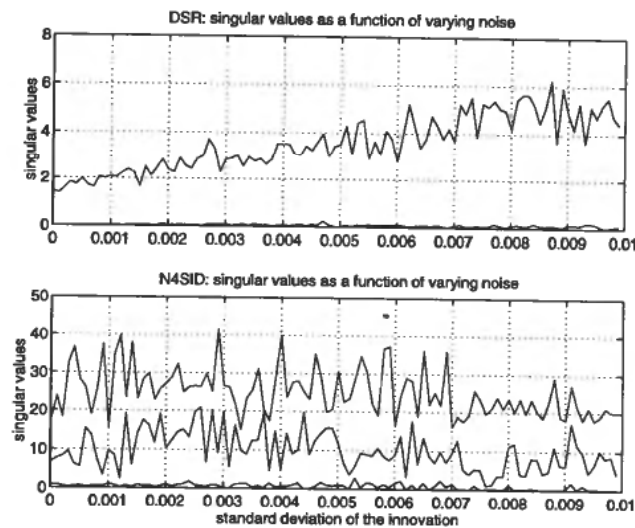


Figure 9. The singular values for model order selection as a function of varying innovations noise level. The input to the system was a pure sinusoid signal (input type  $u^3$ ). The other parameters is as described in Example 2. The singular values from the DSR algorithm is shown in the upper figure and for the N4SID algorithm in the lower. The actual system order is  $n = 1$ . As we can see, the system order is fairly well detected by the DSR algorithm and that the N4SID algorithm does not work at all for this system with a sinusoid input signal.

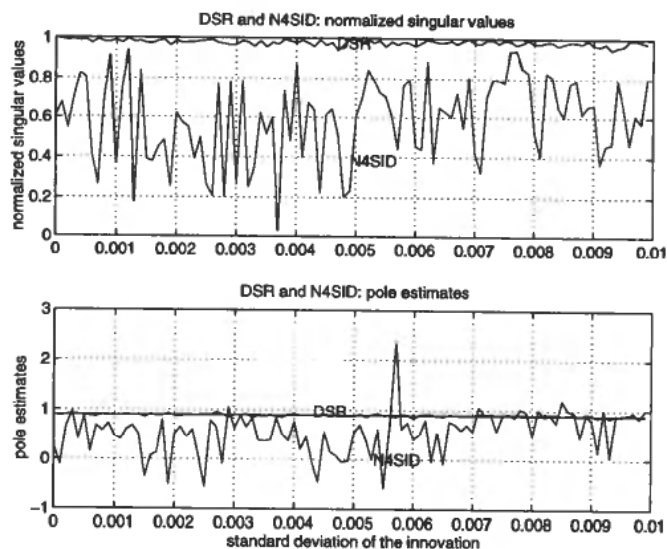


Figure 10. This figure shows the normalized singular values  $(s_1 - s_2)/s_1$  and the pole estimates as a function of varying innovations noise level for the system in Example 2. The input to the system was a pure sinusoid signal (input type  $u^3$ ). The other parameters is as described in Example 2. The normalized singular values from both the DSR and N4SID algorithms are shown in the upper figure. The pole estimates is shown in the lower figure. The actual system order is  $n = 1$  and the actual pole is 0.9. As we can see, the system order and the pole is fairly well estimated by the DSR algorithm and that the N4SID algorithm does not work at all for this system with a sinusoid input signal.

$$D = \begin{bmatrix} 3 & 0 & -0.6 \\ 0 & 1 & 1 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (115)$$

$$C = \begin{bmatrix} 0 & 0.1 \\ 0.1 & 0 \\ 0 & 0.2 \end{bmatrix} \quad \Delta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (116)$$

The systems have a deterministic zero at  $-3$ , i.e. the system is non-minimum phase. The eigenvalues of  $A$  are  $0.85, 0.75 \pm 0.3708i$ .

The deterministic and stochastic gain matrices are given by

$$H^d(1) = \begin{bmatrix} 16 & 15 \\ 2.6667 & -2.5 \end{bmatrix} \quad H^s(1) = \begin{bmatrix} 2.5 & 4.7 \\ -0.25 & 1.1833 \end{bmatrix}$$

The algorithm gives exact results when  $\Delta = 0$ . Hence, this result is not presented. The time series  $y_k, u_k$  was generated by simulating the model with one particular random noise process  $e_k$  with covariance  $\Delta$ . The input was  $u = [u^2 u^1]^T$ . The DSR parameter was fixed to  $L = 6$ . The following estimates are obtained by DSR.

$$\hat{H}^d(1) = \begin{bmatrix} 15.6575 & 14.6168 \\ 2.5576 & -2.4188 \end{bmatrix} \quad \hat{H}^s(1) = \begin{bmatrix} 2.3625 & 4.5583 \\ -0.2436 & 1.1275 \end{bmatrix}$$

$$\hat{\Delta} = \begin{bmatrix} 1.0531 & -0.0244 \\ -0.0244 & 0.9859 \end{bmatrix}$$

In order to analyze the accuracy of the innovations covariance matrix estimates from Theorem 4.3 a Monte Carlo simulation with 100 experiments and varying number of observations was performed. In Van Overschee (1995) a robust expansion of the N4SID algorithm is presented. This version of N4SID is denoted ROBUST. The extra projection  $U_{L+1|L+1}^+$  onto the original N4SID projection is included in the ROBUST algorithm. See Section 6.

The results from DSR, PEM and ROBUST are shown in Figure 12. This result indicated convergence problems for the PEM algorithm.

The estimates from the Monte Carlo simulation shown in Figure 4.

$L/J$	1	2	3	4	5	6	
1	0.6329	0.6038	0.5997	0.5999	0.6005	0.6018	
2	0.6333	0.6033	0.5995	0.6007	0.6000	0.5997	
$\bar{c}(L, J) = 3$	0.6346	0.6016	0.6002	0.5991	0.6008	0.6004	(117)
4	0.6346	0.6037	0.6014	0.6004	0.5998	0.5990	
5	0.6345	0.6035	0.5997	0.5986	0.5992	0.5995	

## 8. Concluding remarks

A method for subspace identification and realization of state space models on innovations formed directly from given input output data is presented. The method determines both the deterministic part and the stochastic part of the model. The algorithm gives exact results in the deterministic case and consistent results when the system is influenced by noise.

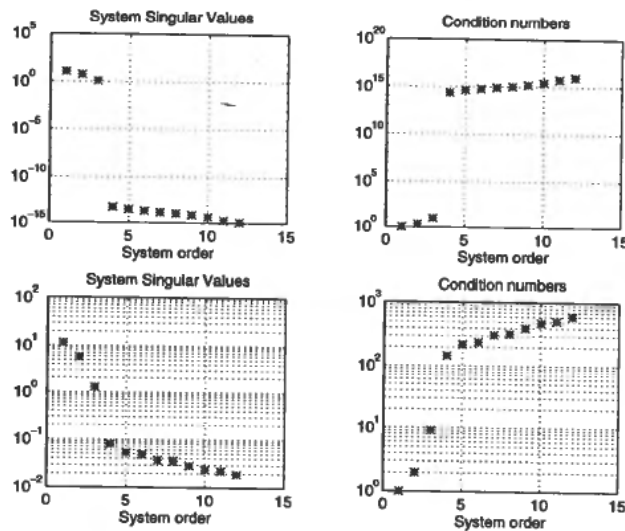


Figure 11. Singular values and conditions numbers for system order estimation. The noise free case with  $\Delta = 0$ , upper. The case with  $\Delta = I$ , lower.

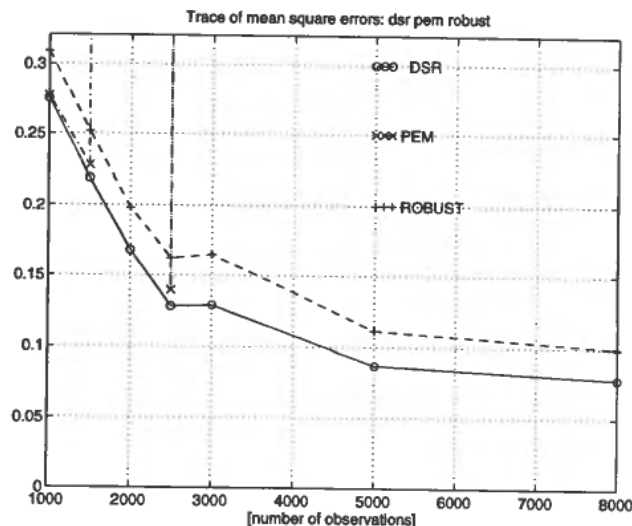


Figure 12. Results from a Monte Carlo simulation in order to investigate the accuracy of the innovation estimates. The model in Example 3 is simulated 100 times with varying number of samples. The DSR estimates is computed from Theorem 4.3. The PEM algorithm converged only for number of samples  $N < 3000$ .

#### REFERENCES

- DI RUSCIO, D. (1994). Methods for the identification of state space models from input and output measurements. *SYSID 94, The 10th IFAC Symposium on System Identification*, Copenhagen, July 4–6.
- DI RUSCIO, D. (1995). A method for the identification of state space models from input and output measurements. *Modeling, Identification and Control*, vol. 16, no. 3. Program commercial available by Fantoft Process AS, Box 306, N-1301 Sandvika.
- DI RUSCIO, D. (1995b). A method for identification of combined deterministic and stochastic systems. *Proceedings of the third European Control Conference, ECC95*, roma, September 5–8, pp. 429–434.

- DI RUSCIO, D., and HOLMBERG, A. (1996). Subspace identification for dynamic process analysis and modeling. *Control Systems 96*, Halifax, Nova Scotia, May 1996.
- GOLUB, G. H., and VAN LOAN, C. F. (1983). *Matrix Computations*. North Oxford Academic Publishers Ltd.
- LARIMORE, W. E. (1983). System identification, reduced order filtering and modeling via canonical variate analysis. *Proc. of the American Control Conference*, San Francisco, USA, pp. 445–451.
- LARIMORE, W. E. (1990). Canonical Variate Analysis in Identification, Filtering and Adaptive Control. *Proc. of the 29th Conference on Decision and Control*, Honolulu, Hawaii, December 1990, pp. 596–604.
- LJUNG, L. (1991). *System Identification Toolbox*. The Mathworks, Inc.
- FAURRE, P. L. (1976). Stochastic realization algorithms, in R. K. Mehra and D. G. Lainiotis (eds), *System Identification: Advances and Case Studies*, Academic Press.
- KALMAN, R. E., FALB, P. L., and ARBIB, M. A. (1969). *Topics in mathematical system theory*, McGraw-Hill Book Company.
- KUNG, S. Y. (1978). A new identification and Model Reduction Algorithm via Singular Value Decomposition. *Conf. on Circuits, Systems and Computers*, Pacific Grove, CA, November 1978, pp. 705–714.
- MOORE, B. C. (1981). Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Trans. on Automatic Control*, Vol. AC-26, pp. 17–31.
- VAN OVERSCHEE, P., and DE MOOR, B. (1994). N4SID: Subspace Algorithms for the Identification of Combined Deterministic Stochastic Systems. *Automatica*, vol. 30, no. 1, pp. 75–94.
- VAN OVERSCHEE, P. (1995). *Subspace Identification: theory-implementation-application*. PhD thesis, Katholieke Universiteit Leuven, Belgium.
- VAN OVERSCHEE, P., and DE MOOR, B. (1996). A Unifying Theorem for Three Subspace System Identification Algorithms. *Automatica*, vol. 31, no. 12, pp. 1853–1864.
- VERHAGEN, M. (1994). Identification of the deterministic part of MIMO state space models given on innovations form from input output data. *Automatica*, vol. 30, no. 1, pp. 61–74.
- VIBERG, M. (1995). Subspace-Based Methods for the Identification of Linear Time-invariant Systems. *Automatica*, vol. 31, no. 12, pp. 1835–1851.