Modeling, Identification and Control, Vol. 30, No. 2, 2009, pp. 71–86, ISSN 1890–1328

Closed and Open Loop Subspace System Identification of the Kalman Filter

David Di Ruscio

Telemark University College, P.O. Box 203, N-3901 Porsgrunn, Norway

Abstract

Some methods for consistent closed loop subspace system identification presented in the literature are analyzed and compared to a recently published subspace algorithm for both open as well as for closed loop data, the DSR_e algorithm. Some new variants of this algorithm are presented and discussed. Simulation experiments are included in order to illustrate if the algorithms are variance efficient or not.

Keywords: Subspace, Identification, Closed loop, Linear Systems, Modeling

1 Introduction

A landmark for the development of so called subspace system identification algorithms can be said to be the algorithm for obtaining a minimal state space model realization from a series of known Markov parameters or impulse responses, i.e., as presented by Ho and Kalman (1966). Subspace system identification have been an important research field during the last 2.5 decades. Larimore (1983, 1990) presented the Canonical Variate Analysis (CVA) algorithm. The N4SID algorithm presented by Overschee and de Moor (1994) and furthermore, subspace identification of deterministic systems, stochastic systems as well as combined deterministic linear dynamic systems are developed and discussed in detail in Overschee and de Moor (1996) and where also the Robust N4SID was presented. In the standard N4SID an oblique projection is used and this method was biased for colored inputs and an extra orthogonal projection, in order to remove the effect of future inputs on future outputs, was included in the Robust N4SID. See also Katayama (2005) and the references therein for further contributions to the field. These subspace system identification algorithms may in some circumstances lead to biased estimates in case of closed loop input and output data.

A subspace system identification method, based on observed input and output data, which estimates the system order as well as the entire matrices in the Kalman filter including the Kalman filter gain matrix, K, and the square root of the innovations covariance matrix, F, was presented in Di Ruscio (1994, 1996). This algorithm is implemented in the DSR function in the D-SR Toolbox for MATLAB. The DSR estimate of the innovation square root covariance matrix F is consistent both for open loop as well as for closed loop data. The DSR method was compared with other algorithms and found to give the best model in comparison to the other methods, based on validation, and on a real world waste and water example in Sotomayor et al. (2003a,b).

The DSR_e method presented in Di Ruscio (2008) and used in the thesis by Nilsen (2005) is a simple subspace system identification method for closed and open loop systems. DSR_e is a two stage method where the innovations process is identified consistently in the first step. The second step is simply a deterministic system identification problem.

In Qin and Ljung (2003), Weilu et al. (2004) a subspace system identification algorithm denoted PARSIM_E for closed loop data is presented. Repetitive and successive computations of the innovations as well as the Markov parameters are incorporated in the algorithm in order to overcome the correlation problem

doi:10.4173/mic.2009.2.3

in traditional subspace algorithms in case of feedback in the data. An extended version is presented in Qin et al. (2005).

The main contributions of this paper can be itemized as follows:

- An extended presentation, further discussions and details of the DSR_e algorithm in Di Ruscio (2008) are given in this paper. In particular the second stage of the algorithm, the deterministic identification problem, is discussed in detail and some solution variants are discussed and presented in Sections 6.1, 6.2 and 6.3 of this paper.

- The PARSIM algorithm by Qin et al. (2005) is considered, and in particular the PARSIM-E algorithm in Qin and Ljung (2003) is discussed and presented in detail in Section 5.1. In the PARSIM-E algorithm some Markov parameters are re-estimated. A modified version of the PARSIM-E algorithm which does not re-estimate the Markov parameters is presented in Section 5.2. The modified version follows the lines in the PARSIM-S algorithm (Algorithm 2) in Qin et al. (2005). The PARSIM-S method is biased for direct closed loop data, as pointed out in Qin et al. (2005), while the modified algorithm in Section 5.2 is consistent for closed loop subspace identification.

- Due to the lack of software implementation of the PARSIM_E algorithm from the authors this algorithm is discussed and implemented along the lines in Section 5.1 and the "past-future" data matrix Eq. presented in Di Ruscio (1997) and implemented as a MATLAB m-file function. The Matlab m-file code of the PARSIM-E algorithm is enclosed in the Appendix.

  This implementation of the PARSIM_E algorithm is compared with the DSR_e function in the D-SR Toolbox for MATLAB, and a new variant of the **dsr_e** method is discussed. This new variant consists of an Ordinary Least Squares (OLS) step for solving a deterministic identification problem for computing the entire Kalman filter model matrices.

- The DSR_e algorithm is a subspace system identification algorithm which may be used to estimate the entire Kalman filter model matrices, including the covariance matrix of the innovations process and the system order, directly from known input and output data from linear MIMO systems. Simulation experiments on a linear MIMO closed loop system is given in this paper, and the algorithm is shown to give promising estimation results which are comparable to the corresponding estimates from the Prediction Error Method (PEM).

- Monte Carlo simulation experiments, on two examples (a SISO and a MIMO system), are presented which show that the DSR_e algorithm outperforms the PARSIM-E method for closed loop subspace system identification. Interestingly, the DSR_e algorithm is on these two examples shown to give as efficient parameter estimates as the Prediction Error Method (PEM).

The contributions in this paper are believed to be of interest for the search for an optimal subspace identification algorithm for open and closed loop systems. The PARSIM-E and DSR methods may be viewed as subspace methods which are based on matrix Eqs. where the states are eliminated from the problem.

Noticing that if the states of a linear dynamic system are known then the problem of finding the model matrices is a linear regression problem, see e.g., Overschee and de Moor (1996). Some methods for closed loop subspace identification with state estimation along the lines of the higher order ARX model approach presented in Ljung and McKelvey (1995) are presented in Jansson (2003, 2005) and Chiuso (2007). In this class of algorithms the states are estimated in a preliminary step and then the model matrices are estimated as a linear regression problem. In Ljung and McKelvey (1995) a method denoted, SSNEW, with MATLAB m-file code is presented. In Jansson (2003) a method denoted SSARX was presented, and in Jansson (2005) a three step method denoted, NEW, is presented. Chiuso (2007) presents the methods denoted, PBSID and PBSID_opt, and shows some asymptotic similarities with the SSARX and the PBSID algorithms. It is also shown that the PBSID_opt algorithm is a row weighted version of the SSNEW algorithm in Ljung and McKelvey (1995). Furthermore, Chiuso (2007) stated that the PBSID_opt algorithm is found to be asymptotically variance efficient on many examples but not in general.

It would be of interest to compare these methods with the DSR_e algorithm, in particular for systems with a low signal to noise ratio. The SSNEW m-file in the original paper by Ljung and McKelvey (1995) would be a natural starting point in this work, but this is outside the scope of this paper and is a topic for further research, where the results are believed to be of interests.

The rest of the paper is organized as follows. In Section 2 the problem definition and some preliminary notations used in the paper are given. In Section 3 some preliminaries and matrix notations are defined and used throughout the paper. In Section 4 the subspace identification problem with closed loop data is discussed in connection with the "past-future" data matrix presented in Di Ruscio (1997). In Section 5 the PARSIM-E method, Weilu et al. (2004), Qin et al. (2005), for closed loop system identification is discussed as well as a variant of this method. In Section 6 we discuss the **dsr_e** method with a new variant of implementation in case of a known system order. In Section 7 some examples which illustrate the behavior of the algorithms are presented. Finally some concluding remarks follow in Section 8.

2 Problem formulation

We will restrict ourselves to linearized or linear state space dynamic models of the form

$$x_{k+1} = Ax_k + Bu_k + Ce_k, \quad (1)$$
$$y_k = Dx_k + Eu_k + Fe_k, \quad (2)$$

with x_0 as the initial predicted state and where a series of N input and output data vectors u_k and y_k $\forall\ k = 0, 1, \ldots, N-1$ are known, and where there is possible feedback in the input data. In case of output feedback the feed through matrix is zero, i.e. $E = 0$. Also for open loop systems the feed through matrix may be zero. We will include a structure parameter $g = 0$ in case of feedback data or for open loop systems in which $E = 0$, and $g = 1$ for open loop systems when E is to be estimated. Furthermore, for the innovations model (1) and (2) e_k is white with unit covariance matrix, i.e. $\mathrm{E}(e_k e_k^T) = I$.

Note that corresponding to the model (1) and (2) on innovations form we may, if the system is not pure deterministic, define the more common Kalman filter on innovations form by defining the innovations as $\varepsilon_k = Fe_k$ and then $K = CF^{-1}$ is the Kalman filter gain. Hence, the Kalman filter on innovations form is defined as

$$x_{k+1} = Ax_k + Bu_k + K\varepsilon_k, \quad (3)$$
$$y_k = Dx_k + Eu_k + \varepsilon_k, \quad (4)$$

where the innovations process ε_k have covariance matrix $\mathrm{E}(\varepsilon_k \varepsilon_k^T) = FF^T$.

The quintuple system matrices (A, B, C, D, E, F) and the Kalman filter gain K are of appropriate dimensions. The problem addressed in this paper is to determine these matrices from the known data. Both closed and open loop systems are addressed.

3 Notations and definitions

Hankel matrices are frequently used in realization theory and subspace system identification. The special structure of a Hankel matrix as well as some matching notations, which are frequently used throughout the paper, are defined in the following.

Definition 3.1 (Hankel matrix) *Given a vector sequence*

$$s_k \in \mathbb{R}^{nr} \ \forall\ k = 0, 1, 2, \ldots, \quad (5)$$

where nr is the number of rows in s_k.

Define integer numbers j, i and nc and define the matrix $S_{j|i} \in \mathbb{R}^{inr \times nc}$ as follows

$$S_{j|i} \overset{def}{=} \begin{bmatrix} s_j & s_{j+1} & s_{j+2} & \cdots & s_{j+nc-1} \\ s_{j+1} & s_{j+2} & s_{j+3} & \cdots & s_{j+nc} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{j+i-1} & s_{j+i} & s_{j+i+1} & \cdots & s_{j+i+nc-2} \end{bmatrix},$$

which is defined as a Hankel matrix because of the special structure. A Hankel matrix is symmetric and the elements are constant across the anti-diagonals. The integer numbers j, i and nc have the following interpretations:

- *j start index or initial time in the sequence used to form $S_{j|i}$, i.e., s_j, is the upper left vector element in the Hankel matrix.*
- *i is the number of nr-block rows in $S_{j|i}$.*
- *nc is the number of columns in the Hankel matrix $S_{j|i}$.*

One should note that this definition of a Hankel matrix $S_{j|i}$ is slightly different to the notation used in Overschee and de Moor (1996) (pp. 34-35) where the subscripts denote the first and last element of the first column in the block Hankel matrix, i.e. using the notation in Overschee and de Moor (1996) a Hankel matrix $U_{0|i}$ is defined by putting u_0 in the upper left corner and u_i in the lower left corner and hence $U_{0|i}$ would have $i+1$ rows.

Examples of such vector processes, s_k, to be used in the above Hankel-matrix definition, are the measured process outputs, $y_k \in \mathbb{R}^m$, and possibly known inputs, $u_k \in \mathbb{R}^r$.

Definition 3.2 (Basic matrix definitions) *The extended observability matrix, O_i, for the pair (D, A) is defined as*

$$O_i \overset{\text{def}}{=} \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{i-1} \end{bmatrix} \in \mathbb{R}^{im \times n}, \quad (6)$$

where the subscript i denotes the number of block rows.

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

$$C_i^d \overset{\text{def}}{=} \left[\begin{array}{cccc} A^{i-1}B & A^{i-2}B & \cdots & B \end{array}\right] \in \mathbb{R}^{n \times ir}, \quad (7)$$

where the subscript i denotes the number of block columns. A reversed extended controllability matrix, C_i^s, for the pair (A, C) is defined similar to (7), i.e.,

$$C_i^s \overset{\text{def}}{=} \left[\begin{array}{cccc} A^{i-1}C & A^{i-2}C & \cdots & C \end{array}\right] \in \mathbb{R}^{n \times im}, \quad (8)$$

i.e., with B substituted with C in (7). The lower block triangular Toeplitz matrix, $H_i^d \in \mathbb{R}^{im \times (i+g-1)r}$, for the quadruple matrices (D, A, B, E).

$$H_i^d \overset{\text{def}}{=} \left[\begin{array}{ccccc} E & 0_{m\times r} & 0_{m\times r} & \cdots & 0_{m\times r} \\ DB & E & 0_{m\times r} & \cdots & 0_{m\times r} \\ DAB & DB & E & \cdots & 0_{m\times r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}B & DA^{i-3}B & DA^{i-4}B & \cdots & E \end{array}\right]$$

where the subscript i denotes the number of block rows and $i+g-1$ is the number of block columns, and where $0_{m\times r}$ denotes the $m \times r$ matrix with zeroes. A lower block triangular Toeplitz matrix $H_i^s \in \mathbb{R}^{im\times im}$ for the quadruple (D, A, C, F) is defined as

$$H_i^s \overset{\text{def}}{=} \left[\begin{array}{ccccc} F & 0_{m\times m} & 0_{m\times m} & \cdots & 0_{m\times m} \\ DC & F & 0_{m\times m} & \cdots & 0_{m\times m} \\ DAC & DC & F & \cdots & 0_{m\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}C & DA^{i-3}C & DA^{i-4}C & \cdots & F \end{array}\right]$$

Given two matrices $A \in \mathbb{R}^{i\times k}$ and $B \in \mathbb{R}^{j\times k}$, the orthogonal projection of the row space of A onto the row space of B is defined as

$$A/B = AB^T(BB^T)^\dagger B. \quad (9)$$

The following property is used

$$A/\left[\begin{array}{c} A \\ B \end{array}\right] = A. \quad (10)$$

Proof of Eq. (10) can be found in e.g., Di Ruscio (1997).

4 Background theory

Consider the “past-future” data matrix, Eq. (4) in the paper Di Ruscio (1997). This same equation is also presented in Eq. (27) in the Di Ruscio (1997) paper. We have

$$\begin{aligned} Y_{J|L} &= \left[\begin{array}{ccc} H_L^d & O_L\tilde{C}_J^d & O_L\tilde{C}_J^s \end{array}\right] \left[\begin{array}{c} U_{J|L+g-1} \\ U_{0|J} \\ Y_{0|J} \end{array}\right] \\ &+ O_L(A-KD)^J X_{0|1} + H_L^s E_{J|L}, \end{aligned} \quad (11)$$

where $\tilde{C}_J^s = C_J^s(A - KD, K)$ is the reversed extended controllability matrix of the pair $(A - KD, K)$, and $\tilde{C}_J^d = C_J^d(A - KD, B - KE)$ is the reversed extended controllability matrix of the pair $(A - KD, B - KE)$, and where C_J^d and C_J^s are defined in Eqs. (7) and (8), respectively.

One should here note that the term $O_L(A - KD)^J X_{0|1} \to 0$ exponentially as J approaches infinity. Also note Eq. (43) in Di Ruscio (2003) with proof, i.e.,

$$X_{J|1} = \left[\begin{array}{cc} \tilde{C}_J^d & \tilde{C}_J^s \end{array}\right] \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right] + (A-KD)^J X_{0|1}, \quad (12)$$

which relates the ”past” data matrices, $U_{0|J}$ and $Y_{0|J}$ to the ”future” states $X_{J|1} = \left[\begin{array}{cccc} x_J & x_{J+1} & \cdots & x_{N-(J+L)} \end{array}\right]$.

Note that Eq. (11) is obtained by putting (12) into the following common matrix equation in the theory of subspace system identification, i.e.

$$Y_{J|L} = O_L X_{J|1} + H_L^d U_{J|L+g-1} + H_L^s E_{J|L}. \quad (13)$$

Eq. (11) is of fundamental importance in connection with subspace system identification algorithms. One problem in case of closed loop subspace identification is that the future inputs, $U_{J|L+g-1}$ in the regressor matrix are correlated with the future innovations, $E_{J|L}$.

Proof 4.1 (Proof of Eq. (11))
From the innovations model, Eqs. (1) and (2), we find Eq. (13), where definitions for the Hankel matrices $Y_{J|L}$, $U_{J|L+g-1}$ and $E_{J|L}$ as well as the matrices O_L, H_L^d and H_L^s are given in Section 3.

From the Kalman filter on innovations form, Eqs. (3) and (4), we obtain the optimal Kalman filter prediction, $\bar{y}_k$, of the output, y_k, as $\bar{y}_k = Dx_k + Eu_k$, and the Kalman filter state Eq., $x_{k+1} = Ax_k + Bu_k + K(y_k - \bar{y}_k)$. From this we find Eq. (12) for the states.

Finally, putting $X_{J|1}$ given by Eq. (12) into Eq. (13) proves Eq. (11). □

The matrices H_L^d and H_L^s with Markov parameters are block lower triangular Toeplitz matrices. This lower triangular structure may be utilized in order to overcome the correlation problem, and recursively computing the innovations sequences, $\varepsilon_{J|1}, \ldots, \varepsilon_{J|L}$ in the innovations matrix $E_{J|L}$. This was the method for closed loop subspace identification proposed by Qin et al. (2005) and Weilu et al. (2004). This method denoted PARSIM is discussed in the next Section 5 and compared to the DSR_e method, Di Ruscio (2008), in an example in Section 7.1.

Another closely related subspace system identification method, i.e. the DSR_e method, Di Ruscio (2008), which works for both open as well as for closed loop identification is built as a two stage method where the innovations sequences, $\varepsilon_{J|1}$, are identified from Eq. (11) in the first step. Noticing that putting $L = 1$ and $g = 0$ (no direct feed through term in the output equation) then we see that Eq. (11) also holds for closed loop systems, i.e., we have

$$Y_{J|1} = \left[\begin{array}{cc} D\tilde{C}_J^d & D\tilde{C}_J^s \end{array} \right] \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array} \right] + FE_{J|1}. \qquad (14)$$

Hence, the DSR F matrix estimates in Di Ruscio (1996) is consistent also for closed loop data. The second step is an optimal OLS step and hence solved as a deterministic problem. The DSR_e method is further discussed in Section 6.

5 Recursive computation of innovations

5.1 The PARSIM-E Algorithm

We will in this section work through the PARSIM-E method presented by Qin and Ljung (2003), and further discussed in Weilu et al. (2004) and in the same instant indicate a drawback with this method (i.e. Markov parameters are recomputed and large matrices pseudo inverted when J large) and show how we may overcome this. This was also noticed in Qin and Ljung (2003). In order to give a simple view and understanding of the PARSIM method we present a few of the first iterations in the algorithm.

Step 1: $i = 0$

Let $L = 1$, $J \to \infty$ and $g = 0$ in (11) and we have

$$Y_{J|1} = \left[\begin{array}{cc} D\tilde{C}_J^d & D\tilde{C}_J^s \end{array} \right] \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array} \right] + \varepsilon_{J|1}. \qquad (15)$$

From Eq. (15) we find estimate of the innovations $\varepsilon_{J|1} = FE_{J|1}$ as well as the indicated matrix in the regression problem, i.e., $\left[\begin{array}{cc} D\tilde{C}_J^d & D\tilde{C}_J^s \end{array} \right]$. Hence, an ordinary Least Squares (OLS) problem is solved. This step is identical to the first step in the DSR_e method.

Step 2: $i = 1$

We may now express from Eq. (11)

$$Y_{J+1|1} = \left[\begin{array}{cccc} DB & DA\tilde{C}_J^d & DA\tilde{C}_J^s & DK \end{array} \right] \left[\begin{array}{c} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \end{array} \right] + FE_{J+1|1}. \qquad (16)$$

From Eq. (16) we find estimate of the innovations $\varepsilon_{J+1|1} = FE_{J+1|1}$. The regression problem increases in size and the matrix $\left[\begin{array}{cccc} DB & DA\tilde{C}_J^d & DA\tilde{C}_J^s & DK \end{array} \right]$ is estimated as an OLS problem.

Hence, at this stage the non-zero part of the first block rows in the Toeplitz matrices H_L^d and H_L^s, i.e. the Markov parameters DB and DK, respectively, are computed. Proceeding in this way each non-zero block row, $H_{i|1}^d$ and $H_{i|1}^s$, in the Toeplitz matrices is computed as follows.

Step 3: $i = 2$

We may now express from Eq. (11)

$$Y_{J+2|1} = \left[\begin{array}{cccc} H_{2|1}^d & DA^2\tilde{C}_J^d & DA^2\tilde{C}_J^s & H_{2|1}^s \end{array} \right] \left[\begin{array}{c} U_{J|1} \\ U_{J+1|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \\ \varepsilon_{J+1|1} \end{array} \right] + FE_{J+2|1}, \qquad (17)$$

where

$$H_{2|1}^d = \left[\begin{array}{cc} DAB & DB \end{array} \right] \in \mathbb{R}^{m \times 2r}, \qquad (18)$$

and

$$H_{2|1}^s = \left[\begin{array}{cc} DAK & DK \end{array} \right] \in \mathbb{R}^{m \times 2m}. \qquad (19)$$

From Eq. (17) we find estimate of the innovations $\varepsilon_{J+2|1} = FE_{J+2|1}$ as well as an estimate of the matrix $\left[\begin{array}{cccccc} DAB & DB & DA^2\tilde{C}_J^d & DA^2\tilde{C}_J^s & DAC & DC \end{array} \right]$. Hence, at this stage the Markov parameters DB and DK are recomputed. Note that we have used the future outputs $U_{J|2}$ in the regressor matrix to be inverted and that the regressor matrix have increased in size. Note that at iteration $i = 2$ the Markov parameters DB and DK are known from the previous iteration $i = 1$. An algorithm which does not recompute DB and DK is presented in the next section.

Step 4: $i := 3$

We may now express from Eq. (11)

$$Y_{J+3|1} =$$

$$\begin{bmatrix} H^d_{3|1} & DA^2\tilde{C}^d_J & DA^3\tilde{C}^s_J & H^s_{3|1} \end{bmatrix} \begin{bmatrix} U_{J|1} \\ U_{J+1|1} \\ U_{J+2|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \\ \varepsilon_{J+1|1} \\ \varepsilon_{J+2|1} \end{bmatrix}$$

$$+FE_{J+3|1}, \tag{20}$$

where

$$H^d_{3|1} = \begin{bmatrix} DA^2B & DAB & DB \end{bmatrix} \in \mathbb{R}^{m\times 3r}, \tag{21}$$

and

$$H^s_{3|1} = \begin{bmatrix} DA^2K & DAK & DK \end{bmatrix} \in \mathbb{R}^{m\times 3m}. \tag{22}$$

From Eq. (20) we find estimate of the innovations $\varepsilon_{J+3|1} = FE_{J+3|1}$ as well as the indicated matrix with model parameters.

Step 5: General case $i = 2, 3, \ldots, L$.

We may now express from Eq. (11)

$$Y_{J+i|1} =$$

$$\begin{bmatrix} H^d_{i|1} & DA^i\tilde{C}^d_J & DA^i\tilde{C}^s_J & H^s_{i|1} \end{bmatrix} \begin{bmatrix} U_{J|i} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|i-1} \\ \varepsilon_{J+i-1|1} \end{bmatrix}$$

$$+FE_{J+i|1} \tag{23}$$

where

$$H^d_{i|1} = \begin{bmatrix} DA^{i-1}B & \cdots & DB \end{bmatrix} \in \mathbb{R}^{m\times ir}, \tag{24}$$

and

$$H^s_{i|1} = \begin{bmatrix} DA^{i-1}K & \cdots & DK \end{bmatrix} \in \mathbb{R}^{m\times im}. \tag{25}$$

From Eq. (23) we find estimate of the innovations $\varepsilon_{J+i|1} = FE_{J+i|1}$. The process given by Eq. (23) is performed for $i = 0, 1, 2, \ldots, L$ in order to find the extended observability matrix O_{L+1} and to use the shift invariance technique in order to find A.

This means that at each new iteration (repetition) i one new last row $\varepsilon_{J+i-1|1}$ is included in the regression matrix on the right hand side of Eq. (23) and one new innovations sequence $\varepsilon_{J+i|1} = FE_{J+i|1}$ is identified. This procedure is performed for $i = 0, 1, \ldots L$ in order to identify O_{L+1} and using the shift invariance method for calculating O_L and O_LA. Here L is a user specified horizon into the future for predicting the number of states. Note also that the regression matrix to be inverted at each step increases in size for each new iteration.

We can see that at each iteration, i.e. in particular for $i \geq 2$ unnecessary Markov parameters are estimated. However, it is possible to exploit this by subtracting known terms from the left hand side of Eq. (23). Note that at iteration $i = 1$ the Markov parameters DB and DK are computed and these may be used in the next step $i = 2$. This modified strategy is discussed in the next Section 5.2.

Software implementation of the PARSIM-E algorithm has not been available so an own Matlab m-file function has been implemented. This function is available upon request.

5.2 A modified recursive algorithm

We will in this section present a modified version of the PARSIM-E method presented by Qin and Ljung (2003), and utilize the block lower triangular structure of the Toeplitz matrices H^d_L and H^s_L. In the PARSIM-E algorithm some of the Markov parameters are recomputed in the iterations for $i = 2, 3, \ldots, L$ even if they are known from the previous iterations. The algorithm will in this section be modified.

A modified version of PARSIM was also commented upon in Qin et al. (2005), i.e., in Algorithm 2 of the paper a sequential PARSIM (PARSIM-S) algorithm is presented. This algorithm utilizes the fact that some Markov parameters in the term $H^d_{i|1}U_{J|i}$ are known at each iteration $i \geq 2$. However, the term $H^s_{i|1}\varepsilon_{J|i}$ is omitted and the innovations, $\varepsilon_{J|1}$, …, $\varepsilon_{J+L|1}$ are not computed in this algorithm, and the PARSIM-S method is biased for direct closed loop data, also commented upon in Remark 2 of that paper. To make the PARSIM-S applicable to closed loop data, they refer to the innovation estimation approach as proposed in Qin and Ljung (2003).

The iterations for $i = 0$ and $i = 1$ are the same as in the previous PARSIM-E method in Eqs. (15) and (16). Hence, at this stage the Markov parameters DB, DK and the innovations sequences $\varepsilon_{J|1} = FE_{J|1}$ and $\varepsilon_{J+1|1} = FE_{J+1|1}$ are known. We will in this section modify the PARSIM-E method to exploit this and modify the left hand side regressed variable term at each new iteration $i = 2, 3, \ldots, L$.

Step 3: $i = 2$

We may now express from Eq. (11)

$$\begin{aligned} &Y_{J+2|1} - DBU_{J+1|1} - DK\varepsilon_{J+1|1} = \\ &\begin{bmatrix} DAB & DA^3\tilde{C}_J^d & DA^3\tilde{C}_J^s & DAK \end{bmatrix} \begin{bmatrix} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \end{bmatrix} \\ &+FE_{J+2|1}. \end{aligned} \tag{26}$$

From Eq. (26) we find estimate of the innovations $\varepsilon_{J+2|1} = FE_{J+2|1}$ and new Markov parameters in the indicated matrix on the right hand side are computed.

Step 4: $i := 3$

We may now express from Eq. (11)

$$\begin{aligned} &Y_{J+3|1} - DABU_{J+1|1} - DBU_{J+2|1} \\ &-DAK\varepsilon_{J+1|1} - DK\varepsilon_{J+2|1} = \\ &\begin{bmatrix} DA^2B & DA^2\tilde{C}_J^d & DA^2\tilde{C}_J^s & DA^2K \end{bmatrix} \begin{bmatrix} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \end{bmatrix} \\ &+FE_{J+3|1}. \end{aligned} \tag{27}$$

From Eq. (27) we find estimate of the innovations $\varepsilon_{J+3|1} = FE_{J+3|1}$ as well as the indicated matrix with model parameters. As we see the matrix of regression variables on the right hand side is unchanged during the iterations and one matrix inverse (projection) may be computed once.

Step 5: General case $i = 2, 3, \dots, L$.

We may now express from Eq. (11)

$$\begin{aligned} &Y_{J+i|1} - \overbrace{\begin{bmatrix} DA^{i-2}B & \cdots & DB \end{bmatrix}}^{H_{i-1|1}^d} U_{J+1|i-1} \\ &- \overbrace{\begin{bmatrix} DA^{i-2}K & \cdots & DK \end{bmatrix}}^{H_{i-1|1}^s} \varepsilon_{J+1|i-1} = \\ &\begin{bmatrix} DA^{i-1}B & DA^i\tilde{C}_J^d & DA^i\tilde{C}_J^s & DA^{i-1}K \end{bmatrix} \begin{bmatrix} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \\ \varepsilon_{J|1} \end{bmatrix} \\ &+FE_{J+i|1}, \end{aligned}$$

where for $i \geq 2$

$$\begin{aligned} H_{i|1}^d &= \begin{bmatrix} DA^{i-1}B & \cdots & DB \end{bmatrix} \\ &= \begin{bmatrix} DA^{i-1}B & H_{i-1|1}^d \end{bmatrix}, \end{aligned} \tag{28}$$

and

$$\begin{aligned} H_{i|1}^s &= \begin{bmatrix} DA^{i-1}K & \cdots & DK \end{bmatrix} \\ &= \begin{bmatrix} DA^{i-1}K & H_{i-1|1}^s \end{bmatrix}. \end{aligned} \tag{29}$$

From this general case we find estimate of the innovations $\varepsilon_{J+i|1} = FE_{J+i|1}$. The process given are performed for $i = 2, \dots, L$ in order to find the extended observability matrix O_{L+1} and to use the shift invariance technique in order to find A. Furthermore the Markov parameters in matrices $H_{L|1}^d$ and $H_{L|1}^s$ are identified. See next section for further details.

5.3 Computing the model parameters

In both the above algorithms the model matrices (A, B, K, D) may be determined as follows. The following matrices are computed, i.e.,

$$\begin{aligned} Z_{J|L+1} &= \begin{bmatrix} O_{L+1}\tilde{C}_J^d & O_{L+1}\tilde{C}_J^s \end{bmatrix} \\ &= O_{L+1} \begin{bmatrix} \tilde{C}_J^d & \tilde{C}_J^s \end{bmatrix}, \end{aligned} \tag{30}$$

and the matrices with Markov parameters

$$H_{L|1}^d = \begin{bmatrix} DA^LB & DA^{L-1}B & \cdots & DB \end{bmatrix}, \tag{31}$$

$$H_{L|1}^s = \begin{bmatrix} DA^LC & DA^{L-1}K & \cdots & DK \end{bmatrix}. \tag{32}$$

From this we find the system order, n, as the number of non-zero singular values of $Z_{J|L+1}$, i.e. from a Singular Value decomposition (SVD) of the $Z_{J|L+1}$ matrix. From this SVD we also find the extended observability matrices O_{L+1} and O_L. Furthermore A may be found from the shift invariance principle, i.e. $A = O_L^\dagger O_{L+1}(m{+}1 : (L{+}1)m, :)$ and $D = O_L(1 : m, :)$.

The B and K matrices may simply be computed by forming the matrices O_LB and O_LK from the estimated matrices $H_{L+1|1}^d$ and $H_{L+1|1}^s$ and then premultiply O_LB and O_LK with the pseudo inverse $O_L^\dagger = (O_L^TO_L)^{-1}O_L^T$.

5.4 Discussion

The two methods investigated in this section do not seem to give very promising results compared to the **dsr_e** method. It seems that the variance of the parameter estimates in general is much larger than the **dsr_e** estimates.

The reason for the high variance in the PARSIM-E method is probably that the algorithm is sensitive to data with low signal to noise ratios, and the reason for this is that “large” matrices are pseudo inverted when J is “large”. Recall that in Eq. (11) the therm $O_L(A - KD)^JX_{0|J} \to 0$ only when $J \to \infty$, because the states $X_{0|J}$ are correlated with the data matrix with regression variables. In practice the past horizon parameter J is finite and the term is negligible. However, a matrix of larger size than it could have been is inverted and this gives rise to an increased variance, compared to the DSR_e method in the next Section 6. It is also important to recognize that only the first innovations sequence, $\varepsilon_{J|1}$, really is needed, as pointed out in Di Ruscio (2008) and that it seems unnecessary to compute the innovation sequences, $\varepsilon_{J+i|1}$, $\forall$ $i = 1, \ldots L$, and then forming the innovations matrix $E_{J|L+1}$, i.e.,

$$E_{J|L+1} = \begin{bmatrix} \varepsilon_{J|1} \\ \varepsilon_{J+1|1} \\ \vdots \\ \varepsilon_{J+i|1} \\ \vdots \\ \varepsilon_{J+L|1} \end{bmatrix}. \tag{33}$$

Only the first row, $\varepsilon_{J|1}$, is actually needed in order to find the model matrices as we will show in the next Section 6. Since unnecessary future innovations rows in $E_{J|L+1}$ matrix are computed, this probably also gives rise to large variances in estimates from the PARSIM algorithm.

In the PARSIM strategy, Section 5, a matrix with increasing size, $ir+Jr+Jm+mi = (J+i)r+(J+i)m = (J+i)(m+r)$, is pseudo inverted at each new iteration $i = 0, 1, 2, \ldots, L$. The matrix to be inverted increases in size during the iterative process when $0 \leq i \leq L$. Both the pseudo inverse problem and the iterations give rise to this large variance. Updating techniques for this matrix inverse may be exploited, however.

The modified PARSIM method in Section 5.2 was also implemented with a QR factorization and the recursive algorithm implemented in terms of the R factors only. This strategy seems not to improve the variance considerably and the variances are approximately the same. The reason is probably that J should be "large" for the method to work. It is interesting to note that in this modified algorithm only a matrix of row size $r+rJ+mJ+m = (J+1)(r+m)$ is pseudo inverted and that this regressor matrix is held unchanged during the iterations and is inverted once.

We believe that the **dsr_e** method, with a consistent estimate of the innovations process in a first step, followed with an optimal estimation of a noise free deterministic subspace identification problem, or if the order is known solved by an optimal OLS (ARX) step leads us to the lower bound on the variance of the parameter estimates. Note that the first step in the **dsr_e** method is a consistent filtering of the output signal, $y_{J|1}$, into a signal part, $y^d_{J|1}$, and a noise innovations part, $\varepsilon_{J|1}$. We will in the next Section 6 discuss the **dsr_e** method.

6 The DSR_e algorithm

In Di Ruscio (2008) a very simple, efficient subspace system identification algorithm which works for both open as well as for closed loop data was presented. This algorithm was developed earlier and presented in an internal report in (2004) and used in Nilsen (2005). In this section an improved and extended presentation of the algorithm is presented.

In the DSR_e algorithm the signal content, $y^d_{J|1}$, of the future data, $y_{J|1} = \begin{bmatrix} y_J & y_{J+1} & \ldots & y_{N-1} \end{bmatrix} \in \mathbb{R}^{m\times(N-J)}$, is estimated by projecting the "past" onto the "future", and in case of closed loop data when the direct feed-through term is zero ($E = 0$), i.e. estimated by the following projection

$$y^d_{J|1} = DX_{J|1} = Y_{J|1} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}, \tag{34}$$

where the projection operator "/" is defined in Eq. (9), and where we have used that for large J or as $J \to \infty$ we have that

$$X_{J|1} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} = X_{J|1}, \tag{35}$$

which may be proved from Eq. (12) by using Eq. (10).

The past data matrices, $U_{0|J}$ and $Y_{0|J}$, are uncorrelated with the future innovations sequence, $E_{J|1}$. In the same stage the innovations sequence $\varepsilon_{J|1} = FE_{J|1}$ in Eq. (15) is then consistently estimated as

$$\varepsilon_{J|1} = FE_{J|1} = Y_{J|1} - Y_{J|1} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}. \tag{36}$$

Note that both the signal part, $y^d_{J|1}$, and the innovation part, $\varepsilon_{J|1}$, are used in the **dsr_e** algorithm.

For open loop systems we may have a direct feed through term in the output equation, i.e., $E \neq 0$ and with an output Eq. $y_k = Dx_k + Eu_k + Fe_k$ we obtain

$$Y_{J|1} = DX_{J|1} + EU_{J|1} + FE_{J|1}, \tag{37}$$

and the signal parts may in this case be computed as,

$$y^d_{J|1} = DX_{J|1} + EU_{J|1} = Y_{J|1} / \begin{bmatrix} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}, \tag{38}$$

where we have used Eq. (10). Furthermore, the innovations are determined by

$$\varepsilon_{J|1} = FE_{J|1} = Y_{J|1} - Y_{J|1} / \begin{bmatrix} U_{J|1} \\ U_{0|J} \\ Y_{0|J} \end{bmatrix}. \tag{39}$$

The algorithm is a two step algorithm where in the first step the output data are split into a signal part, $y^d_{J|1}$, and a noise (innovations) part, $\varepsilon_{J|1} = FE_{J|1}$, i.e., as

$$y_{J|1} = y^d_{J|1} + \varepsilon_{J|1}. \tag{40}$$

This step of splitting the future outputs, $y_{J|1}$, into a signal part, $DX_{J|1}$, and an innovations part, $\varepsilon_{J|1}$, is of particular importance in the algorithm. We propose the following choices for solving this first step in the algorithm:

- Using a QR (LQ) decomposition. Interestingly, the square root of the innovations covariance matrix, F, is also obtained in this first QR step, as in Di Ruscio (1996). Using the definitions or the QR decomposition gives approximately the same results in our simulation examples. One should here note that when the QR decomposition is used also the Q factors as well as the R factors are used. We have the from the QR decomposition

  $$\begin{bmatrix} U_{0|J+g} \\ Y_{0|J+1} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \tag{41}$$

  where $g = 0$ for closed loop systems and systems with no direct feed through term in the output Eq., i.e., when $E = 0$. For open loop systems and when we want to estimate the direct feed through matrix E we put $g = 1$. From, the decomposition (41) we have

  $$y^d_{J|1} = R_{21}Q_1, \tag{42}$$
  $$\varepsilon_{J|1} = R_{22}Q_2, \tag{43}$$

  and we notice that also the Q sub matrices are used. Notice also that we may take $F = R_{22}$ or compute a new QR decomposition of $\varepsilon_{J|1}$ in order to estimate F.

- Another interesting solution to this step is to use a truncated Conjugate Gradient (CG) algorithm, Hestenes and Stiefel (1952), in order to compute the projection. The CG algorithm is shown to be equivalent with the Partial Least Squares algorithm in Di Ruscio (2000) for univariate (single output) systems. This will include a small bias but the variance may be small. This choice may be considered for noisy systems in which PEM and the DSR_e method have unsatisfactory variances or for problems where one has to chose a “large” past horizon parameter J. However, one have to consider a multivariate version, e.g. as the one proposed in Di Ruscio (2000).

This step of splitting the future outputs, $y_{J|1}$, into a signal part, $DX_{J|1}$, and an innovations part, $\varepsilon_{J|1}$, is consistent and believed to be close to optimal when J is “large”.

The second step in the **dsr_e** algorithm is a deterministic subspace system identification step if the system order has to be found, or an optimal deterministic Ordinary Least Squares (OLS) step if the system order is known, for finding the Kalman filter model. Using PEM for solving the second deterministic identification problem may also be an option.

Hence, the future innovations $\varepsilon_{J|1} = FE_{J|1}$ (noise part) as well as the given input and output data are given and we simply have to solve a deterministic identification problem. Note also that if the system order, n, is known this also is equivalent to a deterministic OLS or ARX problem for finding the model. This method is effectively implemented through the use of QR (LQ) factorization, see the D-SR Toolbox for Matlab function **dsr_e.m** (unknown order) and **dsr_e_ols** or PEM if the order is known.

At this stage the innovations sequence, $\varepsilon_{J|1}$, and the noise free part, $y^d_{J|1}$, of the output $y_{J|1}$ are known from the first step in the DSR_e algorithm. Hence we have to solve the following deterministic identification problem

$$x_{k+1} = Ax_k + \begin{bmatrix} B & K \end{bmatrix} \begin{bmatrix} u_k \\ \varepsilon_k \end{bmatrix}, \tag{44}$$
$$y^d_k = Dx_k, \tag{45}$$

where not only the input and output data, u_k, and, y^d_k, respectively, are known, but also the innovations, ε_k, are known. Hence, the data

$$\left.\begin{matrix} u_k \\ \varepsilon_k \\ y^d_k \end{matrix}\right\} \ \forall \ k = J, J+1, \ldots, N-1 \tag{46}$$

are known, and the model matrices (A, B, K, D) may be computed in an optimal OLS problem.

The second step in the DSR_e method is best discussed as a deterministic identification problem where we will define new input and output data satisfying a deterministic system defined as follows

$$\left.\begin{matrix} y_k := y^d_k \\ u_k := \begin{bmatrix} u_k \\ \varepsilon_k \end{bmatrix} \end{matrix}\right\} \ \forall \ k = 1, 2, \ldots, N \tag{47}$$

where $N := N - J$ is the number of samples in the time series to be used in the deterministic identification problem. Hence we have here defined new outputs, y_k, from all corresponding samples in the noise free part $y^d_{J|1} \in \mathbb{R}^{m \times K}$ where the number of columns is $K = N - J$. Similarly, new input data $u_k \in \mathbb{R}^{r+m}$, is defined from the sequence $u_{J|1} = \begin{bmatrix} u_J & u_{J+1} & \cdots & u_{N-1} \end{bmatrix}$

of the original input data, and from the the computed innovations in $\varepsilon_{J|1} = \left[\begin{array}{cccc} \varepsilon_J & \varepsilon_{J+1} & \cdots & \varepsilon_{N-1} \end{array} \right]$.

In the examples of Section 7 we illustrate the statistical properties of the method on closed loop data, and we show that DSR_e is as optimal as PEM, and far more efficient than the PARSIM method.

We believe that the first step in the DSR_e algorithm is close to optimal. Since we have some possibilities for implementing the second step in the algorithm, i.e., the deterministic identification problem, at least in the multiple output case. We discuss some possibilities for the second step separately in the following subsections. This second deterministic identification step is believed to be of interest in itself.

6.1 Deterministic subspace identification problem

Step 2 in the **dsr_e.m** implementation in the D-SR Toolbox for MATLAB is basically as presented in this subsection.

Consider an integer parameter L such that the system order, n, is bounded by $1 \leq n \leq mL$. As an example for a system with $m = 2$ outputs and $n = 3$ states it is sufficient with $L = 2$.

From the known deterministic input and output data $\left.\begin{matrix} u_k \\ y_k \end{matrix}\right\} \forall\ k = 0, 1, \ldots, N$ define the data matrix equation

$$Y_{1|L} = \tilde{A}_L Y_{0|L} + \tilde{B}_L U_{0|L+g}, \tag{48}$$

where the matrices are given by

$$\begin{aligned} \tilde{A}_L &= O_L A (O_L^T O_L)^{-1} O_L^T, & (49) \\ \tilde{B}_L &= \left[\begin{array}{cc} O_L B & H_L^d \end{array} \right] - \tilde{A}_L \left[\begin{array}{cc} H_L^d & 0_{mL\times r} \end{array} \right] & (50) \end{aligned}$$

The same data as used in Eq. (48), i.e. $Y_{0|L+1} = \left[\begin{array}{c} Y_{0|L} \\ Y_{L|1} \end{array} \right]$ and $U_{0|L+g}$ are used to form the matrix Eq.

$$Y_{0|L+1} = O_{L+1} X_{0|J} + H_{L+1}^d U_{0|L+g}. \tag{51}$$

There are some possibilities to proceed but we suggest estimating the extended observability matrix from Eq. (51) and the B, E matrices as an optimal OLS problem from Eq. (48), using the corresponding R sub matrices from the following LQ (transpose of QR) factorisation, i.e.,

$$\left[\begin{array}{c} U_{0|L+g} \\ Y_{0|L+1} \end{array} \right] = \left[\begin{array}{cc} R_{11} & 0 \\ R_{21} & R_{22} \end{array} \right] \left[\begin{array}{c} Q_1 \\ Q_2 \end{array} \right]. \tag{52}$$

Due to the orthogonal properties of the QR factorisation we have

$$R_{22} = O_{L+1} X_{0|J} Q_2^T, \tag{53}$$

and the system order, n, the extended observability matrix O_{L+1}, and thereafter A and D, may be estimated from an SVD of R_{22}, and using the shift invariance technique. An alternative to this is to form the matrix Eq.

$$\bar{R}_{22} = \tilde{A}_L \underline{R}_{22}, \tag{54}$$

and estimate the system order as the n largest non-zero singular values of the mL singular values of the matrix $\underline{R}_{22}$. Here $\bar{R}_{22}$ is obtained as R_{22} with the first m block row deleted and $\underline{\mathrm{R}}_{22}$ as R_{22} with the last m block row deleted. Using only the n largest singular values we have from the SVD that $\underline{\mathrm{R}}_{22} = U_1 S_1 V_1^T$ and we may chose $O_L = U_1$ and find A from Eq. (54), i.e., as

$$A = U_1^T \bar{R}_{22} V_1 S_1^{-1}. \tag{55}$$

Note that there are common block rows in $\underline{R}_{22}$ and $\bar{R}_{22}$. This may be utilized and we may use

$$A = U_1^T \left[\begin{array}{c} \bar{U}_1 \\ R_{22}(Lm+1:(L+1)m,:) V_1 S_1^{-1} \end{array} \right], \tag{56}$$

which is used in the DSR algorithms. This means that we, for the sake of effectiveness, only use the truncated SVD of $\underline{\mathrm{R}}_{22}$ and the last block row in R_{22} in order to compute an estimate of the A matrix. The D matrix is taken as the first m block row in $O_L = U_1$. This way of computing the A and D matrices is a result from Di Ruscio (1994).

Finally, the parameters in the B and E matrices are estimated as an optimal OLS step, using the structure of matrix $\tilde{B}_L$ and the equation

$$\bar{R}_{21} = \tilde{A}_L \underline{\mathrm{R}}_{21} + \tilde{B}_L R_{11}, \tag{57}$$

where $\bar{R}_{21}$ is obtained as R_{21} with the first m block row deleted and $\underline{\mathrm{R}}_{21}$ as R_{21} with the last m block row deleted. Since $\tilde{A}_L$ now is known the problem $\tilde{B}_L R_{11} = \bar{R}_{21} - \tilde{A}_L \underline{\mathrm{R}}_{21}$ may be solved for the parameters in B (and E for open loop systems) as an optimal OLS problem in the unknown $(n+m)r$ parameters in B and E, Di Ruscio (2003). We also mention in this connection that it is possible to include constraints in this OLS step, in order to e.g. solve structural problems, e.g. with $K = 0$ as in output error models.

6.2 Deterministic OLS/ARX step 2

The content in this section is meant as an option for solving the second (deterministic identification) step in the DSR_e algorithm. However, the theory may also be used to identify a higher order ARX model followed by model reduction to a model with the adequate system order.

Specify a prediction horizon parameter, L, as specified above. We have the input and output ARX model

$$\begin{aligned} y_k &= A_1 y_{k-L} + \cdots + A_L y_{k-1} \\ &+ B_1 u_{k-L} + \cdots + B_L u_{k-1}, \end{aligned} \tag{58}$$

where $A_i \in \mathbb{R}^{m\times m}$ and $B_i \in \mathbb{R}^{m\times r}$ $\forall$ $i = 1, \ldots, L$ are matrices with system parameters to be estimated. We only consider the case with a delay of one sample, i.e. without direct feed through term ($E = 0$) in Eq. (2).

From the observed data form the OLS/ARX regression problem

$$Y_{L|1} = \theta \begin{bmatrix} Y_{0|L} \\ U_{0|L} \end{bmatrix} \tag{59}$$

where the parameter matrix, θ, $\in \mathbb{R}^{m\times L(m+r)}$, is given by

$$\theta = \begin{bmatrix} A_1 \cdots A_L B_1 \cdots B_L \end{bmatrix}. \tag{60}$$

This regression problem is effectively solved through a LQ (transpose of QR problem) factorization.

Note that for single output systems then we simply may chose $L = n$ and the state space model matrices A, B, K and D are determined simply from the correspondence with θ and the observer canonical form.

For multiple output systems we may chose L smaller than the system order according to the inequality $1 \leq n \leq mL$. Hence, a non-minimal state space model of order mL on block observer form is constructed directly from the OLS estimate, θ, i.e.,

$$A = \begin{bmatrix} A_L & I_m & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & 0 \\ A_i & 0 & & \ddots & 0 \\ \vdots & \vdots & & & I_m \\ A_1 & 0 & \ldots & & 0 \end{bmatrix}, B = \begin{bmatrix} B_L \\ \vdots \\ B_i \\ \vdots \\ B_1 \end{bmatrix},$$
$$D = \begin{bmatrix} I_m & 0 & \cdots & 0 \end{bmatrix}. \tag{61}$$

Hence, the A matrix in the non-minimal state space model has the $A_i \in \mathbb{R}^{m\times m}$ matrices in the left block column. The B matrix in the non-minimal state space model of order $n \leq mL$ is constructed from the $B_i \in \mathbb{R}^{m\times r}$ matrices. Noticing that the model has the same impulse response matrices as the underlying system, a state space realization with order n is constructed by Hankel matrix realization theory through a Singular Value Decomposition (SVD).

Using the realization algorithm in Ho and Kalman (1966), Hankel matrices $H_{1|L} = O_L C_J$ and $H_{2|L} = O_L A C_J$ are constructed from impulse response matrices $h_i = CA^{i-1}B$ $\forall$ $i = 1, \ldots, L+J$, where $L + J$ is the number of impulse response matrices. Using the SVD realization algorithm in Zeiger and McEwen (1974) gives the model matrices A, B and D and the system order, n.

The above method with outline of implementation is included in the D-SR Toolbox for MATLAB function **dsr_e_ols.m**.

6.3 Discussion

We have tried with different implementations of the deterministic identification problem in the second step of the DSR_e algorithms. In Section 6.1 we presented a deterministic subspace identification problem for identifying both the system order, n, as well as the model matrices A, B, K, D, E. Interestingly, this choice was found to be the best one for all simulation experiments. To our knowledge this subspace system identification method is both consistent and close to efficient for closed as well as for open loop data. The efficiency of the algorithm is illustrated by simulation experiments in the section of examples.

In Section 6.2 we implemented the deterministic identification problem as an OLS/ARX step. However, due to the identification of the block observer form non minimal state space model of order mL and the need for model reduction, this did not match the low variance estimates from the **dsr_e** algorithm with the deterministic subspace solution as presented in Section 6.1.

An optimal PEM step for direct identification of the parameters of an n-th order canonical form state space model of the deterministic identification second step in the DSR_e algorithm is also considered. This option does not in general improve the estimates and in some examples gave larger variance than the **dsr_e** implementation. However, this option should be investigated further.

Finally one should note that there are two horizon integer parameters involved in the DSR_e algorithm. The first parameter J defines “past” input and output data which is used to remove noise from the “future” outputs. The parameter J should be chosen “large” such that the error term $D(A-KF)^J X_{0|J}$ in the first projection is negligible. Note that the size of this term may be analyzed after one run of the algorithm. However, on our simulation experiments J in the range $6 \leq J \leq 10$ gave parameter estimates with similar variances as the PEM estimates, see the examples in Section 7. The horizon J is only used in the first step so that the second step of the algorithm is independent of this parameter.

In the second step a prediction horizon parameter L for the system order is specified such that the system order is bounded by, $1 \leq n \leq mL$. In the single output case low variance estimates are obtained with $L = n$

and in the multiple output case L should be chosen as small as possible, as a rule of thumb. Note however that we have observed that it often is better to chose $L = 2$ than $L = 1$ even for a first order system in which $n = 1$. The horizon parameters may also with advantage be chosen based on model validation.

7 Numerical examples

7.1 SISO one state closed loop system

We have found that the PARSIM-E method works well on the system in Weilu et al. (2004), Case 2 below. However, we will here study a slightly modified system in which the PARSIM-E method fails to give comparable results as the PEM and DSR_e methods, Case 1 below. Hence, this example is a counter example which shows that a method based on recursive innovations computations may give poor estimates.

Consider the following system with proper order of appearance

$$\begin{aligned} y_k &= x_k + e_k, && (62)\\ u_k &= -K_p y_k + r_k, && (63)\\ x_{k+1} &= A x_k + B u_k + K e_k, && (64)\end{aligned}$$

We consider the following two cases:

Case 1 $A = 0.9$, $B = 0.5$, $K = 0.6$ and an initial value $x_1 = 0$. e_k is white noise with standard deviation 0.1. For the controller we use $K_p = 0.6$ and r_k as a white noise signal with $\mathrm{E}(r_k^2) = 1$ generated in Matlab as $R = randn(N, 1)$ and $r_k = R(k)$. Simulation horizon $N = 1000$.

Case 2 Example in Weilu et al. (2004). The same parameters as in Case 1 above but with $B = 1$, $K = 0.5$, e_k is white noise with standard deviation 1 and r_k as a white noise signal with $\mathrm{E}(r_k^2) = 2$. Simulation horizon $N = 4000$.

The system was simulated with discrete time instants $1 \leq k \leq N$. This was done $M = 100$ times with different noise realizations on the innovations e_k but with the same reference signal, r_k, i.e. a Monte Carlo simulation is performed for both the above cases. For the DSR_e, DSR_e_ols and the PARSIM-E methods we used the same algorithm horizon parameters, $J = 6$ and $L = 3$. The system identification Toolbox Ident function **pem** was used with the calls $dat = iddata(Y, U, 1)$ and $m = pem(dat, n)$ with system order $n = 1$.

This example shows clearly that the variance of the PARSIM-E method is larger than the corresponding estimates from the DSR_e algorithm which are comparable and as efficient as the optimal PEM estimates, in

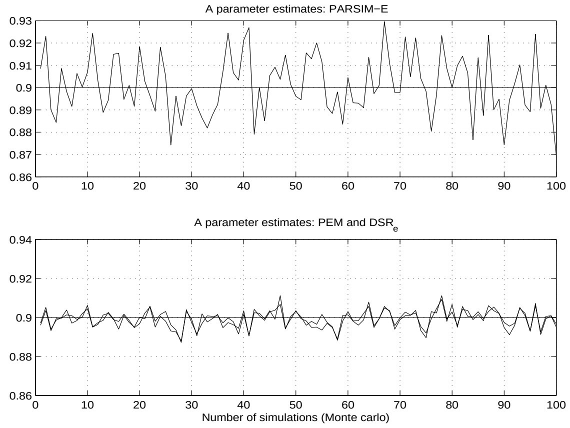

Figure 1: *Pole estimates of the closed loop system in Example 7.1 with parameters as in Case 1.*

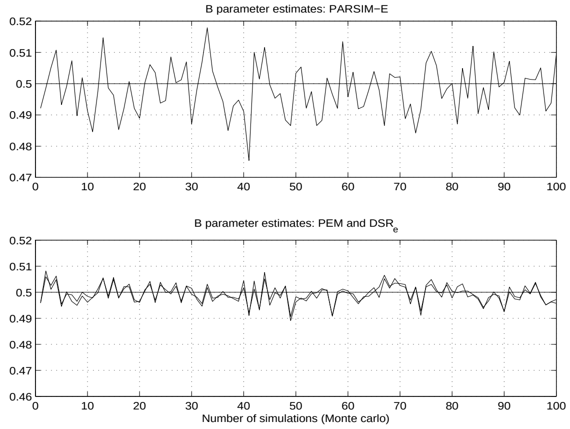

Figure 2: *B parameter estimates of the closed loop system in Example 7.1 with parameters as in Case 1. In Figs. 1 and 2 the PARSIM-E algorithm fails to give acceptable estimates compared to the corresponding DSR_e and PEM estimates.*

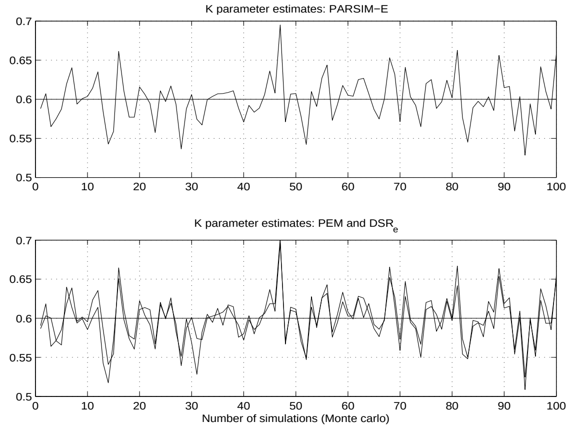

Figure 3: *K parameter estimates of the closed loop system in Example 7.1 with parameters as in Case 1. Here the PARSIM-E algorithm gives acceptable estimates compared to the corresponding DSR_e and PEM estimates.*

particular for the A and B estimates as illustrated in Figures 1 and 2. The variance of the K estimates from PARSIM are comparable with the DSR_e and PEM estimates, see Figure 3. To our belief we have coded the PARSIM-E method in an efficient manner. An m-file implementation of the basic PARSIM-E algorithm is enclosed this paper. We also see that the DSR_e estimates are as optimal as the corresponding PEM estimates. We have here used the standard **dsr_e** method as presented in Sections 6 and 6.1, as well as a version of DSR_e with an OLS/ARX second step as presented in Section 6.2, denoted DSR_e_ols.

For this example the PEM, DSR_e and DSR_e_ols algorithms give approximately the same parameter estimates. Results from DSR_e and PEM are illustrated in the lower part of Figures 1, 2 and 3.

In order to better compare the results we measure the size of the covariance matrix of the error between the estimates and the true parameters, i.e.,

$$P_{\text{alg}} = \frac{N}{M-1}\sum_{i=1}^{M}(\hat{\theta}_i - \theta_0)(\hat{\theta}_i - \theta_0)^T, \tag{65}$$

as

$$V_{\text{alg}} = \text{trace}(P_{\text{alg}}), \tag{66}$$

where subscript “alg” means the different algorithms, i.e. PEM, DSR_e, DSR_e_ols and PARSIM-E. Here the true parameter vector is $\theta_0 = \begin{bmatrix} 0.9 & 0.5 & 0.6 \end{bmatrix}^T$ and the corresponding estimates $\hat{\theta}_i = \begin{bmatrix} \hat{A} & \hat{B} & \hat{K} \end{bmatrix}_i^T$ for each estimate $i = 1, \ldots, M$ in the Monte carlo simulation. From the simulation results of Case 1 we obtain the results

$$\begin{aligned} V_{\text{DSR\_e}} &= 0.8712, & (67)\\ V_{\text{PEM}} &= 0.9926, & (68)\\ V_{\text{DSR\_e\_ols}} &= 1.0565, & (69)\\ V_{\text{PARSIM-E}} &= 1.0774. & (70) \end{aligned}$$

It is interesting to note that the DSR_e algorithm for this example gives a lower value than PEM for the V_{alg} measure of the parameter error covariance matrix in Eq. (65). The reason for this is believed to be due to some default settings of the accuracy in PEM and this is not considered further.

Finally we illustrate the estimated A parameters for the same system as in Weilu et al. (2004), i.e., with parameters in Case 2 above, see Figure 4. For this example the PARSIM-E method gives parameter estimates with approximately the same variance as DSR_e and PEM. However, this is generally not the case as we have demonstrated in Figures 1, 2 and 3.

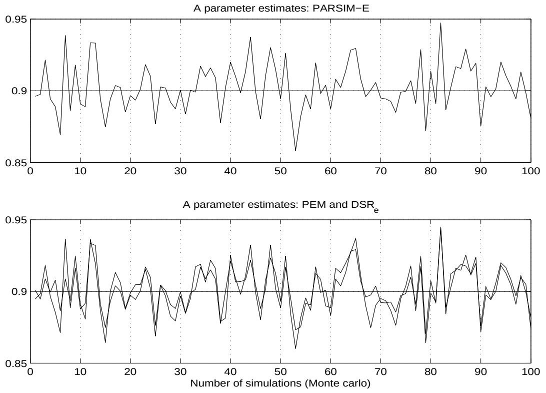

Figure 4: *Pole estimates of the closed loop system in Example 7.1 with parameters as in Case 2, and using PARSIM_e, PEM and DSR_e with, $B = 1.0$, $K = 0.5$ and $E(e_k^2) = 1$ and r_k white noise with variance $E(r_k^2) = 2$. See Figure 1 for a counter example.*

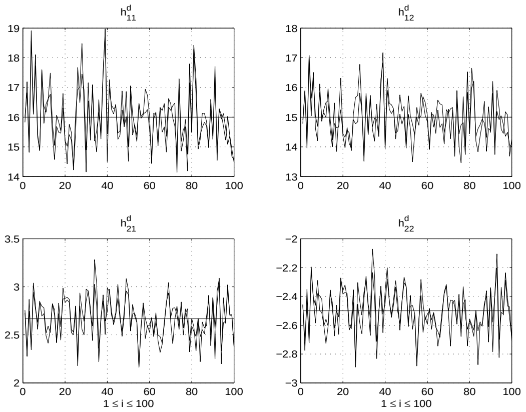

Figure 5: *Elements in the deterministic gain matrix, $h^d = D(I_3 - A)B$, estimates of the closed loop system in Example 7.2. Both PEM and DSR_e_ols estimates are illustrated and the correct gains indicated with solid lines.*

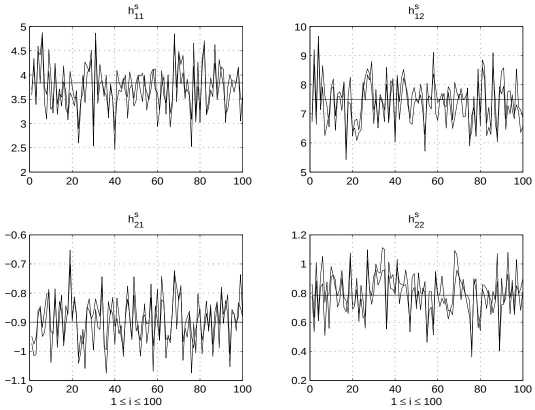

Figure 6: *Elements in the Kalman filter stochastic gain matrix, $h^s = D(I_3 - A)^{-1}K + I_2$, estimates of the closed loop system in Example 7.2. Both PEM and DSR_e_ols estimates are illustrated and the correct gain solid line.*

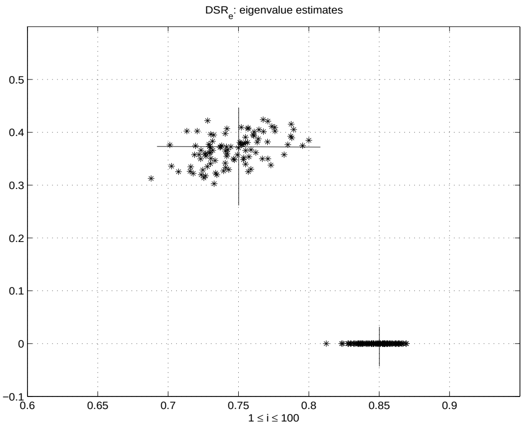

Figure 7: *Eigenvalue estimates for from the DSR_e method. The correct eigenvalues are $\lambda_1 = 0.85$ and a complex conjugate pair, $\lambda_{2,3} = 0.75 \pm j0.3708$. Complex eigenvalues are in complex conjugate pairs. Hence, the negative imaginary part is not presented.*

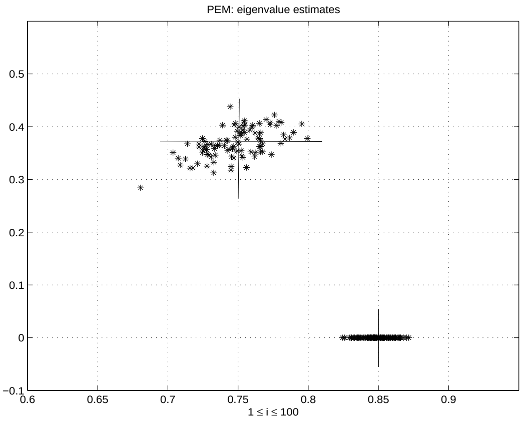

Figure 8: *Eigenvalue estimates from PEM. The correct eigenvalues are $\lambda_1 = 0.85$ and a complex conjugate pair, $\lambda_{2,3} = 0.75 \pm j0.3708$. Complex eigenvalues are in complex conjugate pairs. Hence, the negative imaginary part is not presented.*

7.2 Closed loop MIMO 2×2 system with $n = 3$ states

Consider the MIMO output feedback system

$$\begin{aligned} y_k &= Dx_k + w_k, && (71)\\ u_k &= G(r_k - y_k), && (72)\\ x_{k+1} &= Ax_k + Bu_k + v_k. && (73) \end{aligned}$$

where

$$A = \begin{bmatrix} 1.5 & 1 & 0.1 \\ -0.7 & 0 & 0.1 \\ 0 & 0 & 0.85 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{74}$$

$$D = \begin{bmatrix} 3 & 0 & -0.6 \\ 0 & 1 & 1 \end{bmatrix}. \tag{75}$$

The feedback is obtained with

$$G = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}. \tag{76}$$

and r_k with a binary sequence as a reference for each of the two outputs in y_k. The process noise v_k and measurements noise, w_k are white with covariance matrices $\mathrm{E}(v_k v_k^T) = 0.05^2 I_3$ and $\mathrm{E}(w_k w_k^T) = 0.1^2 I_2$, respectively. For comparison purpose we present the deterministic gain from u_k to y_k, i.e.,

$$h^d = D(I_3 - A)^{-1}B = \begin{bmatrix} 16 & 15 \\ 2.6667 & -2.5 \end{bmatrix}, \tag{77}$$

and the stochastic gain of the Kalman filter from innovations, e_k to output y_k as

$$h^s = D(I_3 - A)^{-1}K + I_2 = \begin{bmatrix} 2.8368 & 7.4925 \\ -0.8998 & -0.2143 \end{bmatrix} \tag{78}$$

A Monte Carlo simulation with $M = 100$ different experiments, $1 \leq i \leq M$, is performed. The simulation results for the deterministic gain matrix is illustrated in Figure 5 and for the stochastic gain matrix, h^s, is illustrated in Figure 6. Furthermore we have presented eigenvalue estimates for both DSR_e and the PEM method in Figures 7 and 8, respectively. For the DSR_e method we used the algorithm horizon parameters, $J = 6$ and $L = 3$. The PARSIM-E estimates are out of range and are not presented.

8 Conclusion

We have in this paper investigated the subspace based method for closed loop identification proposed by Weilu et al. (2004). Alternative versions of this method are discussed and implemented in parallel and compared with the subspace identification method DSR_e which works for both open and closed loop system. We have found by simulation experiments that the variance of the PARSIM-E method is much larger and in general not comparable with the DSR_e and PEM algorithms.

The first step in the PARSIM-E and the DSR_e algorithms is identical. However, the methods differ in the second step. In the PARSIM-E algorithm iterations are included in order to estimate the future innovations over the "future" horizon $0 \leq i \leq L$. In the DSR_e second step a simple deterministic subspace system identification problem is solved.

The difference in the second step/stage of the PARSIM-E and the DSR_e methods is believed to be the reason for the high variance in the PARSIM-

E estimates. In the PARSIM-E algorithm a matrix, $P = O_{L+1}[\tilde{C}_J^d \tilde{C}_J^s] \in \mathbb{R}^{(L+1)m \times J(r+m)}$, is computed in order to compute the extended observability matrix O_{L+1}. The matrix P will be of much larger size than necessary since J is “large” in order for the term $O_{L+1}(A - KD)^J X_{0|J}$ to be negligible. In the DSR_e algorithm the horizon J is skipped from the second step of the algorithm and the extended observability matrix O_{L+1} is identified from a matrix of size $(L+1)m \times (L+1)(m+r)$ and where L may be chosen according to $1 \leq n \leq mL$.

We have tested the DSR_e algorithm against some implementation variants using Monte Carlo simulations. The DSR_e algorithm implemented in the D-SR Toolbox for MATLAB is found to be close to optimal by simulation experiments.

References

Chiuso, A. The role of vector autoregressive modeling in predictor-based subspace identification. *Automatica*, 2007. 43(6):1034–1048. doi:10.1016/j.automatica.2006.12.009.

Di Ruscio, D. Methods for the identification of state space models from input and output measurements. In *10th IFAC Symp. on System Identif.* 1994 .

Di Ruscio, D. Combined Deterministic and Stochastic System Identification and Realization: DSR-a subspace approach based on observations. *Modeling, Identification and Control*, 1996. 17(3):193–230. doi:10.4173/mic.1996.3.3.

Di Ruscio, D. On subspace identification of the extended observability matrix. In *36th Conf. on Decision and Control.* 1997 .

Di Ruscio, D. A weighted view of the partial least squares algorithm. *Automatica*, 2000. 36(6):831–850. doi:10.1016/S0005-1098(99)00210-1.

Di Ruscio, D. Subspace System Identification of the Kalman Filter. *Modeling, Identification and Control*, 2003. 24(3):125–157. doi:10.4173/mic.2003.3.1.

Di Ruscio, D. Subspace system identification of the Kalman filter: open and closed loop systems. In *Proc. Intl. Multi-Conf. on Engineering and Technological Innovation.* 2008 .

Hestenes, M. R. and Stiefel, E. Methods for Conjugate gradients for Solving Linear Systems. *J. Res. National Bureau of Standards*, 1952. 49(6):409–436.

Ho, B. L. and Kalman, R. E. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 1966. 14(12):545–592.

Jansson, M. Subspace identification and arx modeling. In *13th IFAC Symp. on System Identif.* 2003 .

Jansson, M. A new subspace identification method for open and closed loop data. In *IFAC World Congress.* 2005 .

Katayama, T. *Subspace Methods for System Identification.* Springer, 2005.

Larimore, W. E. System identification, reduced order filtering and modeling via canonical variate analysis. In *Proc. Am. Control Conf.* 1983 pages 445–451.

Larimore, W. E. Canonical variate analysis in identification, filtering and adaptive control. In *Proc. 29th Conf. on Decision and Control.* 1990 pages 596–604.

Ljung, L. and McKelvey, T. Subspace identification from closed loop data. 1995. Technical Report LiTH-ISY-R-1752, Linköping University, Sweden.

Nilsen, G. W. *Topics in open and closed loop subspace system identification: finite data based methods.* Ph.D. thesis, NTNU-HiT, 2005. ISBN 82-471-7357-3.

Overschee, P. V. and de Moor, B. N4SID: Subspace Algorithms for the Identification of Combined Deterministic Stochastic Systems. *Automatica*, 1994. 30(1):75–93. doi:10.1016/0005-1098(94)90230-5.

Overschee, P. V. and de Moor, B. *Subspace identification for linear systems.* Kluwer Acad. Publ., 1996.

Qin, S. J. and Ljung, L. Closed-loop subspace identification with innovation estimation. In *Proc. 13th IFAC SYSID Symposium.* 2003 pages 887–892.

Qin, S. J., Weilu, L., and Ljung, L. A novel subspace identification approach with enforced causal models. *Automatica*, 2005. 41(12):2043–2053. doi:10.1016/j.automatica.2005.06.010.

Sotomayor, O. A. Z., Park, S. W., and Garcia, C. Model reduction and identification of wastewater treatment plants - a subspace approach. 2003a. Latin American Applied Research, Bahia Blanca, v.33, p. 135-140, Mais Informacoes.

Sotomayor, O. A. Z., Park, S. W., and Garcia, C. Multivariable identification of an activated sludge process with subspace-based algorithms. *Control Engineering Practice*, 2003b. 11(8):961–969. doi:10.1016/S0967-0661(02)00210-1.

Weilu, L., Qin, J., and Ljung, L. A Framework for Closed-loop Subspace Identification with Innovations Estimation. 2004. Techn. report no. 2004-07, Texas-Wisconsin Modeling and Control Consortium.

Zeiger, H. and McEwen, A. Approximate linear realizations of given dimensions via Ho's algorithm. *IEEE Trans. on Automatic Control*, 1974. 19(2):153. doi:10.1109/TAC.1974.1100525.

9 Appendix: Software

```
function [a,d,b,k,HLd,HLs]=parsim_e(Y,U,L,g,J,n)
% [A,C,B,K]=parsim_e(Y,U,L,g,J,n)
%Purpose
% Implementation of the PARSIM-E algorithm
% On input
% Y,U -The output and input data matrices
% L - Prediction horizon for the states
% g - g=0 for closed loop systems
% J - Past horizon so that (A-KD)^J small
% n - System order

if nargin == 5; n = 1; end
if nargin == 4; n = 1; J = L; end
if nargin == 3; n = 1; J = L; g = 0; end
if nargin == 2; n = 1; J = 1; g = 0; L=1; end

[Ny,ny] = size(Y);
if isempty(U)==1; U=zeros(Ny,1); end

[Nu,nu] = size(U);

N  = min(Ny,Nu);
K  = N - L - J;

% 1. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if nargin < 6
disp('Ordering the given input output data')
end
YL = zeros((L+J+1)*ny,K);
UL = zeros((L+J+g)*nu,K);
for i=1:L+J+1
    YL(1+(i-1)*ny:i*ny,:) = Y(i:K+i-1,:)';
end
for i=1:L+J+g
    UL(1+(i-1)*nu:i*nu,:) = U(i:K+i-1,:)';
end

%
Yf=YL(J*ny+1:(J+L+1)*ny,:);
Uf=UL(J*nu+1:(J+L+g)*nu,:);

%
Up=UL(1:J*nu,:);
Yp=YL(1:J*ny,:);
Wp=[Up;Yp];

% general case, PARSIM-E algorithm
Uf_J=[]; Ef_J=[]; OCds=[];
for i=1:L+1
    Wf=[Uf_J;Wp;Ef_J];
    y_Ji=YL((J+i-1)*ny+1:(J+i)*ny,:);

    Hi=y_Ji*Wf'*pinv(Wf*Wf');
    Zd=Hi*Wf;
    e_Ji=y_Ji-Zd;

    Ef_J=[Ef_J;e_Ji];

    OCds=[OCds; ...
    Hi(:,(i-1)*nu+1:(i-1)*nu+J*nu+J*ny)];

    if i < L+1
        Uf_J=UL(J*nu+1:(J+i)*nu,:);
    end
end

% Compute A and D/C
[U,S,V]=svd(OCds);
U1=U(:,1:n); S1=S(1:n,1:n);
OL=U1(1:L*ny,:);
OLA=U1(ny+1:(L+1)*ny,:);
a=pinv(OL)*OLA;
d=U1(1:ny,:);

% Markov parameters
HLd=Hi(:,1:L*nu);
HLs=Hi(:,L*nu+J*nu+J*ny+1:end);

% Form OL*B matrix from impulse responses in Hi
ni=(L-1)*nu;
for i=1:L
    %ni+1,2*ni
   OLB((i-1)*ny+1:i*ny,1:nu)=Hi(:,ni+1:ni+nu);
   ni=ni-nu;
end
b=pinv(U1(1:L*ny,:))*OLB;

% Form OL*C matrix from impulse responses in Hi
Hs=Hi(:,L*nu+J*nu+J*ny+1:end);
ni=(L-1)*ny;
for i=1:L
   OLC((i-1)*ny+1:i*ny,1:ny)=Hs(:,ni+1:ni+ny);
   ni=ni-ny;
end
k=pinv(U1(1:L*ny,:))*OLC;
% END PARSIM_E
```