Modeling, Identification and Control, Vol. 30, No. 4, 2009, pp. 203–222, ISSN 1890–1328

A Bootstrap Subspace Identification Method: Comparing Methods for Closed Loop Subspace Identification by Monte Carlo Simulations

David Di Ruscio

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

Abstract

A novel promising bootstrap subspace system identification algorithm for both open and closed loop systems is presented. An outline of the SSARX algorithm by Jansson (2003) is given and a modified SSARX algorithm is presented. Some methods which are consistent for closed loop subspace system identification presented in the literature are discussed and compared to a recently published subspace algorithm which works for both open as well as for closed loop data, i.e., the DSR_e algorithm as well as the new bootstrap subspace method presented in this paper. Experimental comparisons are performed by Monte Carlo simulations.

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

1. Introduction

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 (2007a,b). 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. In Chiuso and Picci (2005) the “whitening filter” algorithm, (in later papers denoted PBSID), is introduced, which is a variant of the SSARX algorithm by Jansson (2003). In Chiuso (2007a) the PBSID method is discussed further and an optimized version PBSID_opt presented. In Chiuso (2007b) the PBSID and PBSID_opt methods are further analyzed, and it is shown 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 (2007b) stated that the PBSID_opt algorithm is found to be asymptotically variance efficient on many examples but not in general. Based on this, it make sense to consider the PBSID_opt algorithm and investigate its behavior relative to the other algorithms.

On the other hand there are subspace methods which are based on matrix equations in which the states are eliminated and instead based on projections in order to estimate a subspace for the extended observability matrix and then to estimate the Markov parameters of the system. The PARSIM-E (Qin and Ljung (2003), Qin et al. (2005) and DSR (Di Ruscio (1996), DSR_e (Di Ruscio (2008, 2009)) methods may be viewed as subspace methods of this class which are based on matrix equations where the states are eliminated from the

doi:10.4173/mic.2009.4.2

problem.

The PARSIM-E method was compared against the DSR_e method in Di Ruscio (2000) and not found to be comparable with respect to the low variance estimates from DSR_e algorithm.

It would be of interest to compare the state based subspace algorithms, SSNEW, SSARX and PBSID_opt, with the DSR_e algorithm, in particular for systems with a low signal to noise ratio. This is one of the topics investigated in this paper. The SSNEW m-file in the original paper by Ljung and McKelvey (1995) would be a natural starting point in this work.

In the literature on closed loop subspace system identification, the methods are usually always investigated by numerical experiments with reference signals equal to a white noise signal, leading to examples of systems with a high signal to noise ratio. Jansson (2005) presented an example where the reference signal is white Gaussian noise with unit variance, the same example is used in Jansson (2003), also with the same white reference signal. In Chiuso (2007b) several examples where presented, all with white reference signals. Many methods perform almost equally well for systems with a high signal to noise ratio, and where the reference signal is white noise. In order to distinguish methods and in order to find the most superior of the published methods, one obvious option is to compare the methods on systems with a moderate to low signal to noise ratio, subject to an informative input experiment. Also most practical input and reference experiments are generated by binary or pseudo random binary experiments.

MATLAB m-file software for DSR_e and PBSID_opt are available from the internet. An m-file for the SSNEW algorithm is available in the original paper by Ljung and McKelvey (1995), and used with minor modifications. Software implementation of the SSARX algorithm has not been available so an own Matlab m-file function has been implemented, available upon request.

The contributions of this paper are itemized as follows.

- For the closed loop subspace methods where MATLAB m-file software are available, SSNEW, PBSID_opt and DSR_e, we only give a short review of the methods with the motivation to give a presentation of the methods before using them in Section 8 of numerical Monte Carlo simulations.

- Since m-file software is not available for the SSARX algorithm we give a detailed description of this algorithm with the motivation of writing an efficient m-file implementation of it.

- After presenting the SSARX algorithm, Jansson (2003), we present a new modified SSARX algorithm, denoted SSARX_mod. In this modification we exclude an intermediate linear regression step and instead use one single projection in order to compute a basis for the states. This algorithm also includes one QR and one SVD decomposition. Furthermore, observing that the residual of this projection is the future innovations, we use the first of these innovations as known and solve the Kalman filter on innovations form for the model matrices, e.g. treat the innovations form model as a deterministic system in which the predicted states, the inputs, the innovations and the outputs are known. This is the same idea as is used in the DSR_e algorithm.

- The modified SSARX algorithm denoted SSARX_mod presented in this paper has been found to produce parameter estimates with lower variance than the corresponding estimates from the recently published ARX based subspace identification algorithm PBSID_opt in Chiuso (2007b), on some hard examples with a low signal to noise ratio.

- With the modified SSARX method at hand we propose a new bootstrap subspace identification algorithm denoted SSARX_mb. The first step in this algorithm is to estimate the model matrices with the SSARX_mod algorithm and then perform a few bootstrap iterations with the SSARX_mb algorithm. This algorithm results in a superior bootstrap subspace algorithm as shown in Section 8 of numerical Monte Carlo simulations. The bootstrap subspace identification algorithm SSARX_mb seems to produce parameter estimates with approximately the same variance as the corresponding estimates from the DSR_e algorithm and our Monte Carlo simulation experiments indicate that they are “close” to asymptotic equivalent.

- Monte Carlo simulation experiments, on two examples, are presented which show that the DSR_e algorithm performs better than the SSNEW, SSARX, PBSID_opt and SSARX_mod algorithms, i.e. produce parameter estimates with lower variance, for closed loop subspace system identification, on two examples with low signal to noise ratios.

- From the Monte Carlo simulation experiments we illustrate the efficiency of the modified SSARX algorithms, e.g., SSARX_mod, and in partiucular the bootstrap method SSARX_mb, compared to the SSARX method in Jansson (2003). The bootstrap subspace algorithm, SSARX_mb produced parameter estimates close to DSR_e and PEM for closed loop data.

- Finally we present a Monte Carlo simulation experiment with the reference signal equal to a white noise signal, resulting in a system with a high signal to noise ratio. For this example the behavior of the algorithms, SSNEW, SSARX, PBSID_opt, SSARX_mod and DSR_e are about the same. Interestingly, for this example the PEM algorithm in MATLAB has convergence problems and produced the worst results.
- Interestingly, the DSR_e algorithm is on these two examples, i.e on the examples with a low signal to noise ratio, 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 rest of this paper is organized as follows. In Section 2 the problem definition is given and some preliminary notations and matrix notations used in the paper are presented in Section 3. In Section 4 the higher order ARX model identification problem as well as the SSARX algorithm by Jansson (2003) are investigated in detail. Based on the review of the SSARX algorithm a modified SSARX algorithm is presented in Section 5. A novel bootstrap subspace system identification algorithm is presented in Section 6. An outline of some other subspace algorithms for closed loop systems are presented in Section 7. Some numerical Monte Carlo simulation examples are given in Section 8 and finally some concluding remarks follow in Section 9.

2. Problem formulation

We consider linear state space dynamic models of the form

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

with x_0 as the initial predicted state and where a series of N input and output data vectors u_k and y_k $\forall\ k = 0, 1, \dots, N-1$ 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, eqs. (1-2), e_k is white with unit covariance matrix, i.e. $\mathrm{E}(e_k e_k^T) = I$.

Note that corresponding to the model defined by eqs. (1-2) 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

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

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.

One should note that from eqs. (3-4) we find the Kalman filter prediction model as

$$\begin{aligned} x_{k+1} &= (A - KD)x_k + (B - KE)u_k + Ky_k, & (5)\\ y_k &= Dx_k + Eu_k + \varepsilon_k. & (6) \end{aligned}$$

This Kalman filter prediction model is used in most of the state based subspace identification algorithms.

3. Preliminary theory and definitions

3.1. Basic matrix definitions

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.

Given a vector sequence

$$s_k \in \mathbb{R}^{nr} \ \forall\ k = 0, 1, 2, \dots, \tag{7}$$

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 is the 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, $u_k \in \mathbb{R}^r$.

Some basic matrix definitions used in connection with subspace methods are given as follows.

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

$$O_i \stackrel{\text{def}}{=} \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{i-1} \end{bmatrix} \in \mathbb{R}^{im\times n}, \tag{8}$$

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

$$C_i^d \stackrel{\text{def}}{=} \begin{bmatrix} A^{i-1}B & A^{i-2}B & \cdots & B \end{bmatrix} \in \mathbb{R}^{n\times ir}, \tag{9}$$

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 eq. (9), i.e.,

$$C_i^s \stackrel{\text{def}}{=} \begin{bmatrix} A^{i-1}C & A^{i-2}C & \cdots & C \end{bmatrix} \in \mathbb{R}^{n\times im}, \tag{10}$$

i.e., with B substituted with C in eq. (9). 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).

$$H_i^d \stackrel{\text{def}}{=} \begin{bmatrix} E & 0_{m\times r} & 0_{m\times r} & \cdots & 0_{m\times r} \\ DB & E & 0_{m\times r} & \cdots & 0_{m\times r} \\ DAB & DB & E & \cdots & 0_{m\times r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}B & DA^{i-3}B & DA^{i-4}B & \cdots & E \end{bmatrix}$$

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

$$H_i^s \stackrel{\text{def}}{=} \begin{bmatrix} F & 0_{m\times m} & 0_{m\times m} & \cdots & 0_{m\times m} \\ DC & F & 0_{m\times m} & \cdots & 0_{m\times m} \\ DAC & DC & F & \cdots & 0_{m\times m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ DA^{i-2}C & DA^{i-3}C & DA^{i-4}C & \cdots & F \end{bmatrix}$$

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

$$A/B = AB^T(BB^T)^\dagger B, \tag{11}$$

where † denotes the More-Penrose pseudo-inverse of a matrix. The following property is used

$$A/\begin{bmatrix} A \\ B \end{bmatrix} = A. \tag{12}$$

Proof of eq. (12) can be found in e.g., Di Ruscio (1997).

3.2. Basic matrix equations

The following matrix equation relates the future states, $X_{J|1}$, to the past data matrices, $U_{0|J}$ and $Y_{0|J}$, i.e.,

$$X_{J|1} = \begin{bmatrix} \tilde{C}_J^d & \tilde{C}_J^s \end{bmatrix} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} + (A-KD)^J X_{0|1}, \tag{13}$$

where the "future" states are defined as $X_{J|1} = \begin{bmatrix} x_J & x_{J+1} & \cdots & x_{N-(J+L)} \end{bmatrix}$, 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. (9-10), respectively.

Notice also the following common matrix equation in the theory of subspace system identification, i.e.

$$Y_{J|L} = O_L X_{J|1} + H_L^d U_{J|L+g-1} + H_L^s E_{J|L}. \tag{14}$$

Eq. (14) is derived from the Kalman filter model on innovations form, eqs. (1-2). A variant of eq. (14) is found from eqs. (5-6) but then with $E_{J|L}$ in eq. (14) replaced by the innovations Hankel matrix $\varepsilon_{J|L}$, and a Toeplitz matrix H_L^s defined by using $A := A - KD$, $F := 0_{m\times m}$ and $C := K$, and the corresponding Hankel matrix H_L^d defined similarly. This variant of eq. (14) is used in Jansson (2003) and presented in Section 4.2, eq. (21).

Eq. (13) was used in Di Ruscio (2003) and may be proven from the Kalman filter state prediction eq. (5). This equation is also used in other state based subspace identification methods, e.g. Jansson (2003).

4. On ARX based subspace identification

We will in this section first discuss the relationship between a general linear state space model on innovations form as in eqs. (3-4), and the relationship between a higher order ARX regression problem and how the Markov parameters of the system, as well as the system parameters, may be extracted. These estimates are used as a pre-estimation step in e.g., Jansson (2003) and Ljung and McKelvey (1995). Furthermore we will discuss the SSARX algorithm by Jansson (2003) in some detail. Based on this a modified SSARX algorithm is presented in the next Section 5 with an efficient implementation of it, leading to a MATLAB m-file available upon request.

4.1. Higher order ARX model with model reduction

It is well known that a general linear state space model on innovations form may be identified from a higher order ARX model followed by model reduction. As a reference of this we mention a quote from Ljung (1999) on p. 336, “This means that a high order ARX model is capable of approximating any linear system arbitrarily well.”

A “large” past horizon parameter J is specified such that $(A-KD)^J \approx 0$. We then have the input and output ARX model

$$\begin{gathered} y_k = A_1 y_{k-J} + \cdots + A_J y_{k-1} \\ + B_1 u_{k-J} + \cdots + B_J u_{k-1} + \varepsilon_k = \\ \sum_{i=1}^{J} A_i y_{k-(J+1-i)} + \sum_{i=1}^{J} B_i u_{k-(J+1-i)} + \varepsilon_k, \end{gathered} \quad (15)$$

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

From the observed data and the input output ARX model eq. (15), form the higher order Ordinary Least Squares (OLS) or ARX regression problem

$$Y_{J|1} = \theta \begin{bmatrix} Y_{0|J} \\ U_{0|J} \end{bmatrix} + \varepsilon_{J|1}, \quad (16)$$

where the parameter matrix, $\theta \in \mathbb{R}^{m\times J(m+r)}$, is related to the ARX model parameters, as follows

$$\theta = \begin{bmatrix} A_1 & \cdots & A_J & B_1 & \cdots & B_J \end{bmatrix}, \quad (17)$$

and to the Markov parameters of the Kalman filter prediction model as

$$\theta = D \begin{bmatrix} \tilde{C}_J^s & \tilde{C}_J^d \end{bmatrix}, \quad (18)$$

when $J \to \infty$. This means that $A_i = D(A-KD)^{J-i}K$ and $B_i = D(A-KD)^{J-i}B\ \forall\ i = 1,\dots,J$ in this case. This regression problem is effectively solved through a LQ (transpose of QR problem) factorization.

We have some possibilities for computing the actual model matrices, if needed.

Notice that the ”multivariate” higher order ARX model eq. (16) may be formulated as a ”univariate” ARX model using that stacking each column in the product of three matrices, AXB, (of appropriate dimensions) onto each other, i.e. using the formula $\mathrm{vec}(AXB) = (B^T \otimes A)\mathrm{vec}(X)$, then eq.(16) may be formulated as a ”univariate” OLS problem, i.e.,

$$\mathrm{vec}(Y_{J|1}^T) = (I_{m\times m} \otimes W_p^T)\mathrm{vec}(\theta^T) + \mathrm{vec}(\varepsilon_{J|1}^T), \quad (19)$$

or equivalently

$$\mathrm{vec}(Y_{J|1}) = (W_p^T \otimes I_{m\times m})\mathrm{vec}(\theta) + \mathrm{vec}(\varepsilon_{J|1}), \quad (20)$$

where W_p is a concatenated matrix of the ”past” data Hankel matrices $Y_{0|J}$ and $U_{0|J}$, also defined later on in eq. (26). Here $\otimes$ is the Kronecker product. This means that the ”optimal” univariate Partial Least Squares (PLS) algorithm, see e.g., Di Ruscio (2000), may be used in order to solve the higher order ARX problem in cases when the matrix $W_p W_p^T$ is ill-conditioned.

Option 1: From the parameters in this higher order ARX model we may construct a non minimal state space model of order mJ. Following with model reduction by using Hankel matrix realizations we may find a model with the correct system order, $1 \leq n \leq mJ$. This strategy is the same as used for solving a deterministic identification problem in section 6.2 in Di Ruscio (2009), i.e. using $J := L$ in that algorithm. A method based on this approach is capable of producing correct estimates for deterministic systems even for a relatively small past horizon parameter J, and the reason for this is that the ARX model parameters are used. Using this strategy directly, based on Monte Carlo simulation experiments, it is found to give unreliable estimation results, in particular for systems with a low signal to noise ratio, and not comparable with the corresponding results from the DSR_e algorithm.

Option 2: Another option is to use the estimate of the Markov parameters of the Kalman filter prediction model, i.e., construct $D(A-KD)^{i-1}\begin{bmatrix} B & K \end{bmatrix}$ $\forall\ i = 1,\dots,J$ from the estimated parameter vector θ by using the relationship given by eq. (18). These Markov parameters (impulse responses) are used in the SSARX method by Jansson (2003) as a pre-estimation step for the Markov parameters, in order to overcome the correlation problem in closed loop systems, as we will illustrate in the next subsection 4.2.

It is also possible to use Hankel matrix realization theory, Ho and Kalman (1966), Zeiger and McEwen

(1974), with these Markov parameters and then finding a realization for the Kalman filter predictor model matrices $A-KD$, B, K and D and then finally computing A from $A-KD$ with known D and K. In this strategy the past horizon has to be chosen large in order for $(A-KD)^J$ to be negligible, even for a deterministic system, and the reason for this is that $\theta = D\left[\begin{array}{cc}\tilde{C}^s_J & \tilde{C}^d_J\end{array}\right]$ in eq. (18) only when J is large, i.e. $J \to \infty$.

In practice, many subspace methods based on the higher order ARX approach discussed in this section are believed to only work satisfactorily for systems with white noise reference signals or experiments, and for a relatively "large" past horizon parameter J. Our view on this is that such methods may not produce efficient parameter estimates in general. This is also found from numerous simulation experiments, and this approach is out of range in comparison with the results from both DSR_e and PEM, for systems with a relatively low signal to noise ratio, and therefore not considered further.

For another view and discussion of a similar approach is the Higher Order ARX (HOARX) strategy in Qin and Ljung (2006), where also two numerical examples with high past horizons $J = 10$ and $J = 20$ and white reference signals are presented.

4.2. The SSARX method

An outline of the SSARX method by Jansson (2003) is given in the following. However, one should note that a variant of SSARX denoted NEW is presented in Jansson (2005). The differences with SSARX and NEW is pointed out in Section 4.3.

The first step in the SSARX method is to identify a higher order ARX model, i.e., equivalent to the regression problem in eq. (16), and then use the Markov parameter estimates in eq. (18) to form the Toeplitz matrices $\tilde{H}^d_L$ and $\tilde{H}^s_L$ in the matrix equation

$$Y_{J|L} = \tilde{O}_L X_{J|1} + \tilde{H}^d_L U_{J|L+g-1} + \tilde{H}^s_L Y_{J|L-1} + \varepsilon_{J|L}. \quad (21)$$

Note that this matrix equation is obtained from the Kalman filter prediction model, eqs. (5-6). The extended observability matrix of the pair $(D, A-KD)$, i.e. $\tilde{O}_L = O_L(A-KD, D)$, is defined as in eq. (8). The Toeplitz matrices $\tilde{H}^d_L = H^d_L(A-KD, B-KE, D)$ and $\tilde{H}^s_L = H^s_L(A-KD, K, D)$ have the same structure as H^d_L and H^s_L defined in Section 3.1 with matrix definitions, respectively. Note that we use $F = 0$ and $E = 0$ for closed loop systems, in the definitions of the Toeplitz matrices.

One problem in case of closed loop data is that the innovations noise term, $\varepsilon_{J|L}$, is correlated with the future Hankel matrices $U_{J|L+g-1}$ and $Y_{J|L-1}$ but since the Toeplitz matrices are known one may form the regression problem

$$\tilde{Y}_{J|L} = \overbrace{\tilde{O}_L\left[\begin{array}{cc}\tilde{C}^d_J & \tilde{C}^s_J\end{array}\right]}^{\tilde{\theta}}\left[\begin{array}{c}U_{0|J}\\ Y_{0|J}\end{array}\right] + \varepsilon_{J|L}, \quad (22)$$

where the left hand side in the regression problem, $\tilde{Y}_{J|L}$, is known and given by

$$\tilde{Y}_{J|L} = Y_{J|L} - \tilde{H}^d_L U_{J|L+g-1} - \tilde{H}^s_L Y_{J|L-1}. \quad (23)$$

Eq. (22) is found by using eq. (13) in eq. (21) with $J \to \infty$, and the future states $X_{J|1}$ is related to the past data Hankel matrices as $X_{J|1} = \left[\begin{array}{cc}\tilde{C}^d_J & \tilde{C}^s_J\end{array}\right]\left[\begin{array}{c}U_{0|J}\\ Y_{0|J}\end{array}\right]$.

From this regression problem we find an estimate, $\hat{\tilde{\theta}}$, of the matrix

$$\tilde{\theta} = \tilde{O}_L\left[\begin{array}{cc}\tilde{C}^d_J & \tilde{C}^s_J\end{array}\right]. \quad (24)$$

For the sake of completeness, $\tilde{\theta}$ is computed as the OLS estimate

$$\hat{\tilde{\theta}} = \tilde{Y}_{J|L} W_p (W_p W_p^T)^{-1}, \quad (25)$$

where

$$W_p = \left[\begin{array}{c}U_{0|J}\\ Y_{0|J}\end{array}\right]. \quad (26)$$

In Jansson (2003) canonical correlation analysis is used as described in Appendix A. Here we illustrate with SVD and the difference is minor from our simulation experiments. Following by a SVD of this matrix, i.e.,

$$\hat{\tilde{\theta}} = USV^T \approx U_1 S_1 V_1^T, \quad (27)$$

where $U_1 = U(:, 1:n)$, $S_1 = S(1:n, 1:n)$ and $V_1 = V(:, 1:n)$. This gives estimates $\tilde{O}_L = U_1$ and $\left[\begin{array}{cc}\tilde{C}^d_J & \tilde{C}^s_J\end{array}\right] = S_1 V_1^T$ if an output normal realization is chosen. If not known in advance, the system order, n, may be taken as the number of non-zero singular values in S.

Then finally an estimate of the states is constructed as follows

$$\hat{X}_{J|1} = \left[\begin{array}{cc}\tilde{C}^d_J & \tilde{C}^s_J\end{array}\right]\left[\begin{array}{c}U_{0|J}\\ Y_{0|J}\end{array}\right] = S_1 V_1^T \left[\begin{array}{c}U_{0|J}\\ Y_{0|J}\end{array}\right]. \quad (28)$$

The last part of the SSARX algorithm is to find the model matrices A, B, K and D from the Kalman filter on prediction form, eqs. (3-4), as presented in Jansson (2003, 2005), as follows. The output matrix estimate

$$\hat{D} = y_{J|1} X^T_{J|1} (X_{J|1} X^T_{J|1})^{-1}, \quad (29)$$

and the innovations as the residual of this regression, i.e.,

$$\varepsilon_{J|1} = y_{J|1} - y_{J|1}/X_{J|1}, \tag{30}$$

and A, B and K from the regression problem

$$X_{J+1|1} = \left[\begin{array}{ccc} A & B & K \end{array} \right] \left[\begin{array}{c} X_{J|1} \\ u_{J|1} \\ \varepsilon_{J|1} \end{array} \right]. \tag{31}$$

An assumption in this algorithm is that the past horizon, J, is large so that the term $(A-KD)^J$ is negligible. Hence, this algorithm would be biased for finite J. However, the method is producing correct estimates in the deterministic noise free case and it is a nice example of an algorithm utilizing a pre-estimation step in order to formulate the right hand side $\tilde{Y}_{J|L}$ in eqs. (22-23), i.e. an higher order ARX regression problem is used, in order to estimate the Markov parameters of the Kalman filter on prediction form.

The SSARX algorithm is found to be asymptotic equivalent with the PBSID algorithm in Chiuso (2007b) where also an optimized version, PBSID_opt, algorithm is presented. A short view on this algorithm is given in Section 7.3.

Noticing that the regression problem, eqs. (22) and (25), for the parameter matrix $\tilde{\theta}$ is actually not needed, we propose an alternative more efficient SSARX implementation in the next Section 5.

4.3. Discussion

From the higher order ARX regression problem it is possible to find the model matrices A, B, D directly. It is therefore also possible to form the regression problem

$$Y_{J|L} - H_L^d U_{J|L-1} = O_L \left[\begin{array}{cc} \tilde{C}_J^d & \tilde{C}_J^s \end{array} \right] \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array} \right] + H_J^s E_{J|L}, \tag{32}$$

which may be an alternative to the method in Jansson (2003). However, this choice is not found to improve the estimates from Monte Carlo simulations.

One should note that a variant of SSARX denoted NEW is presented in Jansson (2005). The algorithm denoted, NEW, is a three step method. Constraints are included in the 1st step regression problem, i.e. eq. (21) with the future states taken from eq. (13), is considered as a linear regression problem in the matrices $O_L \left[\begin{array}{cc} \tilde{C}_J^d & \tilde{C}_J^s \end{array} \right]$, and the Toeplitz matrices $\tilde{H}_L^d$ and $\tilde{H}_L^s$ are constrained to be block lower triangular. In the 2nd step initial estimates of $\tilde{O}_L$ and $\left[\begin{array}{cc} \tilde{C}_J^d & \tilde{C}_J^s \end{array} \right]$, are computed through canonical correlation analysis between the future and the past, as in Appendix A. In the 3rd step one returns to the regression problem in the first step and finds updated estimates of $\tilde{O}_L$ and $\left[\begin{array}{cc} \tilde{C}_J^d & \tilde{C}_J^s \end{array} \right]$. This is described to be equivalent to taking a Newton step. Finally the states are computed. This is, as we understand it, the main difference compared to the SSARX method, in which $\tilde{H}_L^d$ and $\tilde{H}_L^s$ are estimated from the preliminary higher order ARX problem as is described in Sections 4.2 and 4.1.

5. Modified SSARX algorithms

We will in this section discuss an alternative implementation of the SSARX method which is found superior to the SSARX algorithm in Section 4.2. This results in a new modified SSARX algorithm, denoted SSARX_mod.

5.1. Modified SSARX algorithm

If a pre-estimated model or Markov parameters are not known in advance, we use the same higher order ARX/OLS regression problem as initially in the SSARX method by Jansson (2003), i.e., solving the regression problem eq. (16) with parameters related to the Markov parameters as in eq. (18). Hence, pre-estimated Toeplitz matrices $\tilde{H}_L^d$ and $\tilde{H}_L^s$ are known in advance.

From eq. (21) we have

$$\overbrace{Y_{J|L} - \tilde{H}_L^d U_{J|L+g-1} - \tilde{H}_L^s Y_{J|L-1}}^{\tilde{Y}_{J|L}} = \tilde{O}_L X_{J|1} + \varepsilon_{J|L}. \tag{33}$$

From this we find the projection equation

$$Z_{J|L} = \tilde{O}_L X_{J|1}, \tag{34}$$

where

$$Z_{J|L} = \tilde{Y}_{J|L} / \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array} \right], \tag{35}$$

and $\tilde{Y}_{J|1}$ is defined in eq. (23), i.e. the left hand side of eq. (33). In order to prove this we use that

$$X_{J|L} = X_{J|L} / \left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array} \right], \tag{36}$$

when $J \to \infty$. Notice also eq. (13) in this connection. Instead of using an SVD of the “large” basis matrix, $Z_{J|L}$, (which is possible but time consuming), for the states we propose the following LQ decomposition (transpose of QR decomposition), i.e.

$$Z_{J|L} = RQ. \tag{37}$$

Following by an SVD

$$R = USV^T \approx U_1 S_1 V_1^T, \tag{38}$$

where $U_1 = U(:, 1:n)$, $S_1 = S(1:n, 1:n)$ and $V_1 = V(:, 1:n)$. The system order may be taken as the number of non-zero singular values in S. This gives the following estimate of the states, i.e.,

$$\hat{X}_{J|1} = S_1 V_1^T Q. \tag{39}$$

One should here note that the future horizon has to be chosen such that the system order is bounded by $1 \leq n \leq mL$.

The model matrices are found similar to the SSARX algorithm in Section 4.2, i.e. as in eqs. (29) and (31). This modified SSARX algorithm is denoted SSARX_mod1 and used in the Monte Carlo simulations in Section 8.

5.2. Modified SSARX algorithm using the innovations

One should now observe that the residual of the projection problem, eqs. (33-34), gives the following estimate of the future innovations

$$\varepsilon_{J|L} = \tilde{Y}_{J|L} - \tilde{Y}_{J|L} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}. \tag{40}$$

This indicates an alternative implementation of the algorithm, i.e., in addition using the innovations as in Di Ruscio (2008, 2009). An alternative recursive computation procedure for the future innovations is presented in Qin and Ljung (2003).

Instead of computing the model matrices as in Jansson (2003, 2005) we also use the innovations, eq. (40), in addition to the state estimates, eq. (39). Hence, we propose to solve a deterministic problem, as in the DSR_e algorithm. However, here we use the state estimates and an estimate of the output matrix is taken as (when $E = 0$, i.e. no direct feed-through term)

$$\hat{D} = (y_{J|1} - \varepsilon_{J|1}) X_{J|1}^T (X_{J|1} X_{J|1}^T)^{-1}, \tag{41}$$

and the innovation, $\varepsilon_{J|1}$, is taken as the first block row in the innovations estimate, eq. (40). Furthermore A, B and K are computed from the regression problem similar as in the SSARX algorithm eq. (31), i.e.,

$$X_{J+1|1} = \begin{bmatrix} A & B & K \end{bmatrix} \begin{bmatrix} X_{J|1} \\ u_{J|1} \\ \varepsilon_{J|1} \end{bmatrix}. \tag{42}$$

This last modified SSARX algorithm is denoted SSARX_mod2 and used in the Monte Carlo simulations in Section 8. The differences in these modified SSARX algorithms and the SSARX algorithm presented in Section 4.2 are virtually small, but may be seen as major from the Monte Carlo simulations in Section 8.

5.3. Discussion

The main differences in the new modified SSARX method, denoted SSARX_mod, and the original SSARX method by Jansson (2003) are itemized as follows:

- In the first instance we skip the regression step eq. (27) and hence, we do not compute the parameter matrix given by eq. (24). Instead we use a single projection in order to compute a basis for the state, eqs. (34-35) and (37).

- In Jansson (2003) the output matrix, D, is computed from a regression problem of the output equation, as in eq. (29) and the innovations $\varepsilon_{J|1}$ (needed in order to find A, B and K from the state equation) as the residual of this regression, as in eq. (30).

  Instead we find the future innovations, $\varepsilon_{J|L}$, directly from the residual of the projection described in the first item, i.e. as in eq. (40), and use the first block row, $\varepsilon_{J|1}$ as known in the last regression problem for the model matrices as in eqs. (41-42).

Remark 5.1 *Based on the observations in this section we propose a bootstrapping subspace system identification algorithm in the next Section 6. Observing that the initial higher order ARX identification problem for the Markov parameters used to define the Toeplitz matrices $\tilde{H}_L^d$ and $\tilde{H}_L^s$ actually is not needed if the model matrices are known or estimated from other algorithms. This bootstrapping method seems to converge in a few iterations.*

6. A novel bootstrap subspace algorithm

The higher order ARX regression pre-estimation step used in many subspace identification algorithms, e.g. as in Ljung and McKelvey (1995), Jansson (2003), Chiuso (2007b), for estimating the Markov parameters (impulse responses) of the Kalman filter prediction model may be ill-conditioned, in particular for systems with a low signal to noise ratio. This may be one reason for the increased variances in the estimates for some colored reference signals used. This is observed from our Monte Carlo simulations, see e.g. the examples in Section 8 and the examples in Di Ruscio (2009) where one observed "high" variance estimates from the PARSIM-E algorithm by Qin and Ljung (2003); Qin et al. (2005).

One idea of this section is to relax the higher order ARX regression problem and instead start with another initial model at hand, e.g. the zero model $A_0 = 0$, $B_0 = 0$, $D_0 = 0$ and $K_0 = 0$. Then we obtain the projection approximate equation (for convenience with the further presentation, also define a first iterate index $i = 1$)

$$Z_{J|L}(i-1) := \tilde{O}_L X_{J|1}(i), \tag{43}$$

where the matrix $Z_{J|L}(i-1)$ is obtained by projecting the row space of the "future" Hankel matrix, $Y_{J|L}$, onto the row space of the "past" input and output Hankel matrices $U_{0|J}$ and $Y_{0|L}$, respectively, i.e.,

$$Z_{J|L}(i-1) := Y_{J|L} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}, \tag{44}$$

where the projection, "/", operator is defined in eq. (11). Also compute the innovations estimate as the residual of this projection, i.e.

$$\varepsilon_{J|L}(i-1) := Y_{J|L} - Y_{J|L} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}. \tag{45}$$

From eq. (43) we obtain the state estimate

$$X_{J|1}(i) := S_1 V_1^T Q, \tag{46}$$

where Q is computed from the QR decomposition as in eq. (37) and S_1 and V_1 from the SVD in eq. (38). From the estimated state sequence in eq. (46) we obtain a new model, A_i, B_i, D_i and K_i from the regression problems given by eqs. (41-42). With this new model at hand the above procedure may be repeated, i.e., putting $i := i + 1$ and replace $Y_{J|L}$ in eqs. (44-45) with $\tilde{Y}_{J|L}(i-1)$ given as the left hand side of eq. (33). This illustrates the first iteration for $i = 1$ in the Algorithm 6.1 to be presented below.

It is remarkable that this iterative algorithm has been found, on our Monte Carlo simulation examples in Section 8, to converge within a few iterations $1 \leq i \leq i_b$ where it seems to be sufficient with $i_b = 2$ or $i_b = 3$ iterations. Other initial models may be used, e.g. the Toeplitz matrices which are used in the SSARX algorithm by Jansson (2003), or other initial models. This is further discussed in and after the Algorithm 6.1 below.

At each iteration only an orthogonal projection for a basis for the states, and a least squares regression problem for the model matrices, are computed. We will denote this procedure for a bootstrap subspace identification method, similar as the presentation of bootstrap methods in Ljung (1999) p. 334.

Based on the above description we propose a novel bootstrap subspace identification method with outline as in Algorithm 6.1 below.

Algorithm 6.1 (Bootstrap subspace method)

Step 0: $i = 0$.

We suggest two options for initializing the bootstrap subspace algorithm. Chose one of the following:

1. *Find initial estimates of the Toeplitz matrices, $\tilde{H}_L^d(i = 0)$ and $\tilde{H}_L^s(i = 0)$, from e.g. the higher order ARX/OLS regression problem in eq. (16) where the ARX model parameter matrix, θ, is related to the Markov parameters as in eq. (18).*
2. *Chose an initial "zero" model, i.e., $\tilde{H}_L^d(i = 0) = 0$ and $\tilde{H}_L^s(i = 0) = 0$, i.e., starting from a zero model $A = 0$, $B = 0$, $D = 0$ and $K = 0$, or from some other a-priori specified model.*

Here index i is the bootstrap iteration index. Choosing a few bootstrap iterations, e.g. $i_b = 2$ or $i_b = 3$ and the following algorithm seems to converge rapidly. Put $i := 1$ and go to Step 2 in this algorithm.

Step 1: $1 \leq i \leq i_b$. From some initial known model matrices $A_{i-1}, B_{i-1}, D_{i-1}, K_{i-1}$ form the Toeplitz matrices

$$\tilde{H}_L^d(i-1) = H_L^d(A_{i-1} - K_{i-1}D_{i-1}, B_{i-1}, D_{i-1}), \tag{47}$$

$$\tilde{H}_L^s(i-1) = H_L^s(A_{i-1} - K_{i-1}D_{i-1}, K_{i-1}, D_{i-1}), \tag{48}$$

where H_L^d and H_L^s are defined as in Section (3.1). Note that the last m block column in H_L^s is skipped and that the matrix have $0_{m \times m}$ blocks on the diagonal. Similarly, the last r block column in H_L^d is skipped when $g = 0$, i.e. for closed loop system in which the direct feed-through term is zero, and $E = 0_{m \times r}$ on the block diagonal in this case.

Step 2: Compute from the modified SSARX algorithm in Sections 5.1 and 5.2 an updated model A_i, B_i, D_i, K_i, e.g., computed by the SSARX_mod2 algorithm, i.e., form the "future" corrected outputs,

$$\tilde{Y}_{J|L}(i-1) := Y_{J|L} - \tilde{H}_L^d(i-1)U_{J|L+g-1} - \tilde{H}_L^s(i-1)Y_{J|L-} \tag{49}$$

and then the projection of the "future" onto the "past"

$$Z_{J|L}(i-1) = \tilde{O}_L(i) X_{J|1}(i), \tag{50}$$

where

$$Z_{J|L}(i-1) := \tilde{Y}_{J|L}(i-1) / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}, \tag{51}$$

The rest of this step of the algorithm is as described by eqs. (37) and (42).

Step 3: if $i < i_b$ go to Step 1, else $A := A_{i_b}$, $B := B_{i_b}$, $D := D_{i_b}$ and $K := K_{i_b}$ and quit.

This bootstrap subspace identification algorithm is denoted SSARX_mb and used in Section 8 with number of bootstrap iterations $i_b = 3$ as default. $(\cdot)$_mb stands for "modified" with "bootstrap" iterations.

The main idea of the algorithm is that instead of using the pre-estimation problem in the SSARX algorithm, i.e. instead of using the first higher order regression Step 0 in the above algorithm, we instead use the model estimates $A_{i-1}, B_{i-1}, D_{i-1}, K_{i-1}$ from the previous Step 2 and perform a few iterations, $1 \leq i \leq i_b$, in order to compute the resulting model matrices $A := A_{i_b}$,$B := B_{i_b}$,$D := D_{i_b}$ and $K := K_{i_b}$.

The first Step 0 in this algorithm may be skipped and an initial model may be taken from other algorithms, or simply as the "zero" model as commented upon below. This indicates possibilities for variants of the algorithm.

On Example 8, the algorithm converged to the same model, irrespective of initial model. This is good news, because an iterative algorithm should preferably converge to the optimal solution, irrespective of initial values. This is investigated further in Section 8 of examples, see e.g., Tables 4 and 5. However, it is well known as described in Ljung (1999) on p.338 that e.g. the iterative prediction error method only is guaranteed to converge to a local solution and to find the global solution one have to start the iterations at different feasible initial values. On our MIMO Example 8.2 we observed that local solutions of the bootstrap subspace identification exists, and starting with a zero model for this example resulted in a biased model. However, starting with an initial model from the higher order ARX strategy resulted in a unbiased model with approximately as optimal parameter estimates as as the corresponding PEM estimates. This is further documented in Example 8.2.

Notice also that if $i_b = 1$, only one bootstrap iteration, this algorithm is equivalent to the modified SSARX method in Section 5, denoted SSARX_mod, if the initial model is taken from the higher order ARX regression problem.

As far as we know there is no subspace system identification algorithm in the literature which utilizes bootstrap iterations. The idea and the algorithm illustrated in this section have been found to give superior parameter estimates compared to the original SSARX method as well as the PBSID_opt and SSNEW methods as illustrated in Section 8 of examples. The method SSARX_mb illustrated in this Section has strong similarities with the DSR_e algorithm and the parameter estimates are close to the corresponding estimates from DSR_e. Monte Carlo simulation experiments indicate also that the SSARX_mb and DSR_e algorithms are close and possibly asymptotic equivalent.

The proposed bootstrap subspace algorithm should probably (and as proposed in this work) be based on the Kalman filter on prediction form, eqs. (5-6), and its corresponding Hankel matrix eq. (21) or equivalently eq. (33) since the predictor is stable, i.e., $A-KD$ eigenvalues inside the unit circle. This is probably the main reason for our observation that this bootstrap iteration subspace algorithm indeed converges within a few iterations, even for a zero initial model. Also remembering that the Kalman filter itself and its corresponding convergence properties in the sense that the covariance matrix of the state estimation error is minimized. Further convergence analysis of this bootstrap algorithm should be performed, which is a topic for further research.

Remark 6.1 *We have found that we may chose the initial model in Step 0 or Step 1 equal to zero, i.e. choosing an initial model $A_0 = 0$, $B_0 = 0$, $D_0 = 0$ and $K_0 = 0$ in Step 1 of the bootstrap Algorithm 6.1, seems to give the same model after a sufficiently large number of bootstrap iterations, i_b, for some examples but not in general.*

This indicates that the preliminary higher order ARX regression step in some subspace identification algorithms may be unnecessary. This bootstrap subspace identification algorithm is promising and should be further investigated, e.g. investigate if including a row weighting in the projection eq. (50) is influencing upon the variance, e.g., by using the CCA approach as in Appendix A.

6.1. Additional details

The aim of this subsection is to present some details and insight into the projections in the bootstrap Algorithm 6.1. Notice that the central iteration in Algorithm 6.1, i.e. eqs. (49)-(51) may be written as

$$\begin{aligned} Z_{J|L}(i-1) = \tilde{O}_L(i) X_{J|1}(i) = \tilde{Y}_{J|L}(i-1)/\left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right] \\ = Y_{J|L}/\left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right] \\ -\theta_m(i-1)\left[\begin{array}{c} U_{J|L+g-1} \\ Y_{J|L-1} \end{array}\right]/\left[\begin{array}{c} U_{0|J} \\ Y_{0|J} \end{array}\right], \end{aligned} \tag{52}$$

where the parameter matrix, $\theta_m(i-1)$,

$$\theta_m(i-1) = \left[\begin{array}{cc} \tilde{H}^d_L(i-1) & \tilde{H}^s_L(i-1) \end{array}\right]. \tag{53}$$

is defined in terms of the Markov parameters of the model matrices $A_{i-1}, B_{i-1}, D_{i-1}, K_{i-1}$ from the previous iteration.

Note that eq. (49) may be more compactly written as

$$\tilde{Y}_{J|L}(i-1) = \tilde{O}_L(i)X_{J|1}(i) + \varepsilon_{J|L} = \underbrace{\begin{bmatrix} -\tilde{H}^d_L(i-1) & \bar{H}^s_L(i-1) \end{bmatrix}}_{\bar{\theta}_m(i-1)} \begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix}. \tag{54}$$

Here, $\bar{H}^s_L$ is a modified Toeplizt matrix with the Markov parameters of the stochastic part of the system in the lower part and with identity matrices $I_{m\times m}$ on the diagonal, i.e.,

$$\bar{H}^s_L \overset{\text{def}}{=} \begin{bmatrix} I & 0 & \cdots & 0 \\ -DK & I & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ -D\bar{A}^{L-2}K & -D\bar{A}^{L-3}K & \cdots & I \end{bmatrix}, \tag{55}$$

where we have defined $\bar{A} = A - KD$ for convenience. Note also that $\bar{H}^s_L = I - \tilde{H}^s_L \in \mathbb{R}^{Lm\times Lm}$ and that the term $\bar{H}^s_L(i-1)Y_{J|L}$ in Eq. (54) is equal to $Y_{J|L} - \tilde{H}^s_L(i-1)Y_{J|L-1}$ in Eq. (49).

Eq. (54) gives insight because it nicely shows the data involved, i.e. the ”future” Hankel matrix $\begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix}$ and the ”past” Hankel matrix $\begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}$ which is used to remove the future innovations, $\varepsilon_{J|L}$. Hence, at each iteration we have that a basis for the states are computed as

$$Z_{J|L}(i-1) = \tilde{O}_L(i)X_{J|1}(i) = \bar{\theta}_m(i-1) \begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}, \tag{56}$$

where the parameter matrix $\bar{\theta}_m$ is given by

$$\bar{\theta}_m(i-1) = \begin{bmatrix} -\tilde{H}^d_L(i-1) & \bar{H}^s_L(i-1) \end{bmatrix}. \tag{57}$$

Also notice that from Eq. (40) we have that the future innovations are computed as

$$\begin{aligned} \varepsilon_{J|L} &= \tilde{Y}_{J|L}(i-1) - \tilde{Y}_{J|L}(i-1) / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} \\ &= \bar{\theta}_m(i-1) \begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}^{\perp}, \end{aligned} \tag{58}$$

where $()^{\perp}$ denotes the orthogonal compliment to the matrix.

The above illustrate the central projections in the iterations for $i > 1$.

We may also implement the bootstrap method, i.e. Eqs. (56) and (58), with one QR decomposition for computing the projections.

We have

$$\begin{bmatrix} U_{0|J} \\ Y_{0|J} \\ \tilde{Y}_{J|L} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}. \tag{59}$$

from this we obtain the projections

$$Z_{J|L}(i-1) = \tilde{O}_L X_{J|1} = R_{21}Q_1, \tag{60}$$

$$\varepsilon_{J|L} = R_{22}Q_2. \tag{61}$$

An alternative which may be investigated is as follows. From the following LQ (transpose of QR decomposition we have)

$$\begin{bmatrix} U_{0|J} \\ Y_{0|J} \\ U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}. \tag{62}$$

From this we have

$$\begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix} / \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix} = R_{21}Q_1, \tag{63}$$

which is to be used in the projection for the states Eq. (56) and

$$\begin{bmatrix} U_{J|L+g-1} \\ Y_{J|L} \end{bmatrix} \begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}^{\perp} = R_{22}Q_2. \tag{64}$$

This QR decomposition approach is an alternative to the direct computations illustrated in the bootstrap subspace system identification algorithm. Notice also that this decomposition is performed once, prior to the iterations in Algorithm 6.1.

7. Outline of some other subspace methods

We will in this section give a short review of the closed loop subspace system identification algorithms DSR_e (Di Ruscio (2008, 2009)), SSNEW (Ljung and McKelvey (1995)) and PBSID_opt (Chiuso (2007b)). The aim is not to go into deep details, and for this we refer to the corresponding papers. The motivation is to give a short presentation of the methods which are to be compared by numerical Monte Carlo simulations in the next Section 8 of examples.

7.1. The DSR_e method

The DSR_e method as discussed in Di Ruscio (2008, 2009) is a simple subspace algorithm. Below we give a short outline connected to the Monte Carlo comparison experiment presented in this paper.

If the innovations process was known in addition to the input and output data then we would have found a perfect model. Having this in mind it would be of interest to find an efficient way of estimating the innovations process. An efficient estimate is as discussed below.

The DSR_e algorithm is linked to the higher order ARX model approach in the first step of the algorithm. However, instead of computing the model from the higher order ARX model we only use the residual in order to obtain an estimate of the innovations. This is indeed a great difference which gives more numerically reliable results.

Step 1 in the DSR_e algorithm is as follows: For “large” J the innovations sequence $\varepsilon_{J|1} = FE_{J|1}$ is consistently estimated as

$$\varepsilon_{J|1} = FE_{J|1} = Y_{J|1} - Y_{J|1}/\begin{bmatrix} U_{0|J} \\ Y_{0|J} \end{bmatrix}. \tag{65}$$

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

Step 2 in the DSR_e algorithm is a deterministic identification problem. In Di Ruscio (2009) two options were presented.

One option is actually to construct a higher order ARX model with future parameter L bounded as $1 \leq n \leq mL$, and then construct a higher order state space model with system order mL followed by model reduction by Hankel matrix realization, Ho and Kalman (1966), Zeiger and McEwen (1974). This option is found to work properly if L is equal to the system order (SISO) case but it will produce unreliable results for increasing L, based on the Monte Carlo simulation experiment in Section 8. Hence, this option is not considered further.

Solving the 2nd Step in the DSR_e algorithm as a deterministic identification problem as in Section 6.1 in Di Ruscio (2009) is found to produce very promising numerical results regarding variance and consistency. This feature is in this paper illustrated in Section 8 with numerical Monte Carlo simulations.

7.2. The SSNEW method

Another algorithm essentially similar to the higher order ARX model approach discussed in the above Section 4.1 is the SSNEW method presented in the original paper by Ljung and McKelvey (1995) where also an m-file code script is presented, also used in Section 8.1 with numerical examples in this paper. This method is further discussed in Ljung and McKelvey (1996).

The idea of the SSNEW method is to first estimate the parameters of a higher order ARX model, e.g. as in eq. (17), and thereafter construct future predictions from this higher order ARX estimated model, and from this form a basis for the states.

7.3. The PBSID_opt method

In Chiuso (2007b) the Predictor Based Subspace IDentification methods (PBSID) and the optimized version PBSID_opt are discussed. These methods are based on constructing a basis for the state space from the Kalman filter on prediction form. These methods belong to the class of methods which utilize the fact that once the states are known then the model matrices are found from a linear regression problem.

The PBSID algorithm was first presented in Chiuso and Picci (2005) under the name “whitening filter algorithm”. The “whitening filter algorithm” is also investigated in Qin and Ljung (2006) and where it is shown that this algorithm is a bank of high order ARX models in the regression step to extract the Markov parameters. The PBSID algorithm was found to give asymptotically the same variance estimates as the CCA (also denoted Canonical Variate Analysis (CVA)) algorithm by Larimore (1983, 1990) in case of white inputs. In case of colored non-white inputs the optimized PBSID algorithm is presented in Chiuso (2007a), and it was found that the PBSID_opt algorithm perform better or similar to the CCA algorithm in case of colored inputs.

It is in Chiuso (2007b) found that the PBSID and SSARX method by Jansson (2003) are asymptotically equivalent. The optimized version of PBSID, i.e. the PBSID_opt method, is using a weighted least squares problem in order to estimate the predictors which are used to form a basis for the state space, and this is one difference from their original PBSID method. It is in Chiuso (2007b) also found that the PBSID_opt method is a row weighted version of the SSNEW method by Ljung and McKelvey (1995).

In the PBSID algorithm a basis for the state, satisfying a system similar to eq. (34), is solved from several oblique projections (as in Overschee and de Moor (1996)), i.e. considering each block row in eq. (33) at a time. In the PBSID_opt algorithm each block row in $Y_{J|L}$ is solved for the corresponding block row in $\tilde{\theta}$, eq. (24), as a vectorized ordinary least squares problem, i.e., solving $\mathrm{vec}(Y_{J+i|1})\ \forall\ i = 0, \ldots, L-1$ for the unknown parameters. The rest of these algorithms may be performed similar as in the SSARX algorithm, Jansson (2003, 2005).

We have included the PBSID_opt algorithm as one of the methods to be compared by Monte Carlo simulation in Section 8.1 of numerical examples.

7.4. Discussion

In the SSNEW method by Ljung and McKelvey (1995) a higher order ARX model is estimated in the first step. The parameters in this higher order ARX model are used to construct future predictors which are used to form a basis for the states. Using the estimated states the model matrices are computed. In the SSNEW algorithm then both the states and the innovations are estimated and assumed known.

In the first step of the DSR_e algorithm only the residual of a higher order ARX model is used, i.e., without computing or using the parameters in the higher order ARX model. This is indeed a major difference as we will illustrate in Section 8 with numerical examples, especially when measured on the variance of the estimation results.

Since the PBSID algorithm is asymptotically equivalent with the SSARX method by Jansson (2003) we find it useful to compare the DSR_e algorithm with the PBSID_opt and SSARX algorithms, only. A Monte Carlo simulation example is presented in the next section.

8. Examples

8.1. SISO one state closed loop system

In Di Ruscio (2009) it was found that the PARSIM-E method fails to give comparable results as the PEM and DSR_e methods on a SISO system with a white innovations process and a white reference signal. In this example we compare the DSR_e method with PEM and the recently published PBSID_opt algorithm, Chiuso (2007b), as well as the SSNEW method in the original paper by Ljung and McKelvey (1995). Furthermore we also investigate the SSARX algorithm by Jansson (2003) as described in Section 4.2 and the modified SSARX algorithms, SSARX_mod1, SSARX_mod2 and the bootstrap subspace method SSARX_mb presented in Section 5.

Consider the following system with proper order of appearance

$$\begin{aligned} y_k &= x_k + e_k, && (66)\\ u_k &= -K_p y_k + r_k, && (67)\\ x_{k+1} &= Ax_k + Bu_k + Ke_k, && (68)\end{aligned}$$

We consider the following two cases:

Case 1: Same parameters as in Di Ruscio (2009), i.e., $A = 0.9$, $B = 0.5$, $K = 0.6$ and an initial value $x_1 = 0$. e_k is white noise with standard deviation 1.0. For the controller we use $K_p = 0.6$. Instead of a white noise reference signal we use r_k as a Pseudo Random Binary (PRBS) signal generated in Matlab as $R = 0.1 * idinput(N,' prbs', [0, 1/50])$ and $r_k = R(k)\ \forall\ k = 1, \dots, N$. The simulation horizon equals $N = 1000$.

Case 2: An integrator system with $A = 1$, $B = 0.5$ and $K = 1$. Innovations e_k, controller and reference r_k same as in Case 1 above.

The reference signal used in this example is a pseudo random binary signal with clock period of 50, i.e., the signal is approximately constant over intervals of length 50 samples. The reference experiment is informative enough.

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 subspace algorithms DSR_e, PBSID_opt, SSNEW, SSARX and SSARX_mb methods we used the same algorithm past horizon parameter, $J = 6$ but with varying future horizon in the range $1 \leq L \leq 5$. 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 $n = 1$.

In order to investigate if the DSR_e algorithm is capable of producing as efficient estimates as the optimal PEM method we implemented the 2nd deterministic identification problem in the DSR_e algorithm by using PEM, i.e. using PEM in order to solve the deterministic identification problem with known inputs $u_{J|1}$, the estimated innovations $\varepsilon_{J|1}$ and estimated outputs $y^d_{J|1}$. The results of this in comparison with the corresponding results from PEM are illustrated in Fig. 1. The results are important because it shows that the DSR_e algorithm is capable of producing almost similar results as PEM, on this example. It is also from this clear that it is of central importance to implement the 2nd deterministic identification step in the DSR_e algorithm as efficient as possible. This example also tells us that we have implemented the 1st important step in the DSR_e algorithm in an efficient way, i.e. that the innovations $\varepsilon_{J|1}$ and the signal part $y^d_{J|1} = DX_{J|1}$ are consistent and (close to) efficient estimates.

In Figs. 2 and 9 we present the same Monte Carlo experiment but with the actual DSR_e algorithm with the 2nd deterministic identification problem solved as a subspace identification problem as presented in Section 6.1 in Di Ruscio (2009). The results are promising and DSR_e (for $L = 3$ in this example) seems to produce estimates with lower variance than PEM. Notice however, that using $L = 1$ the DSR_e estimates are approximately similar to the PEM estimates.

In order to better compare the results we measure

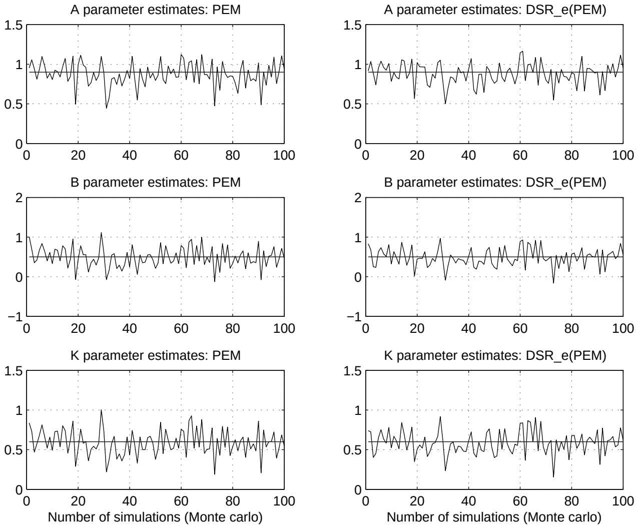

Figure 1: *A, B and K parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Past horizon $J = 6$ for the DSR_e method with the deterministic identification 2nd step solved by using the PEM method.*

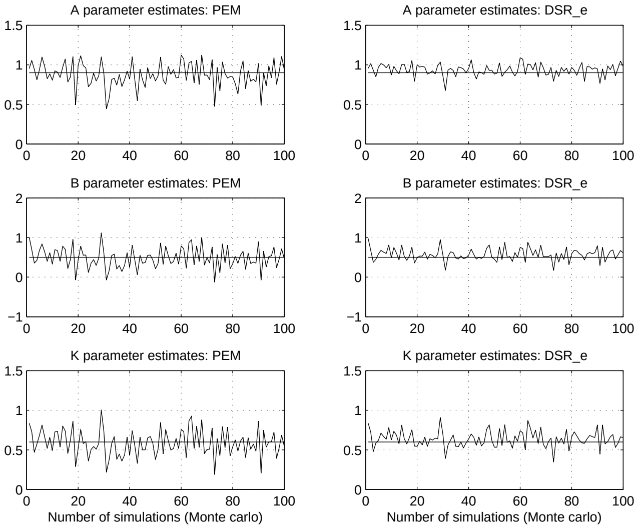

Figure 2: *A, B and K parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The DSR_e algorithm compared with corresponding PEM estimates.*

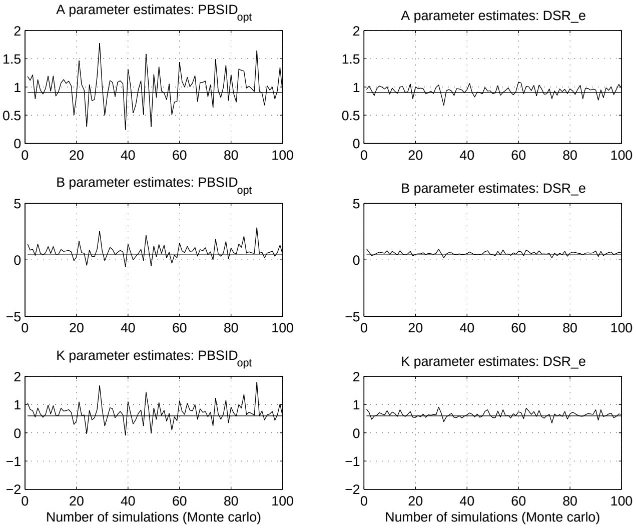

Figure 3: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The PBSID_opt algorithm as in Chiuso (2007a).*

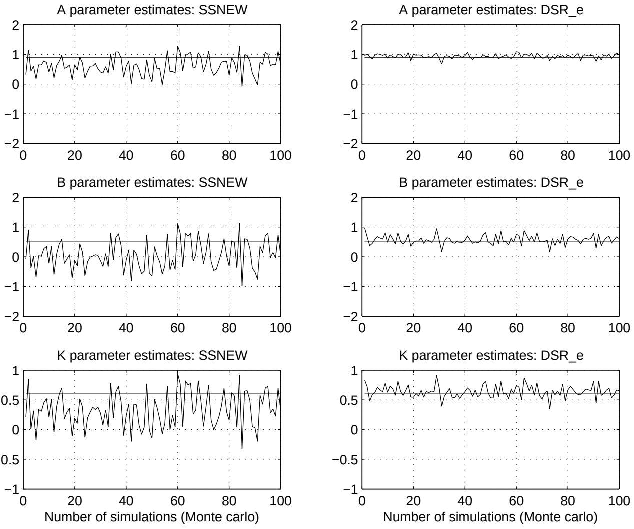

Figure 4: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The SSNEW method as in Ljung and McKelvey (1995).*

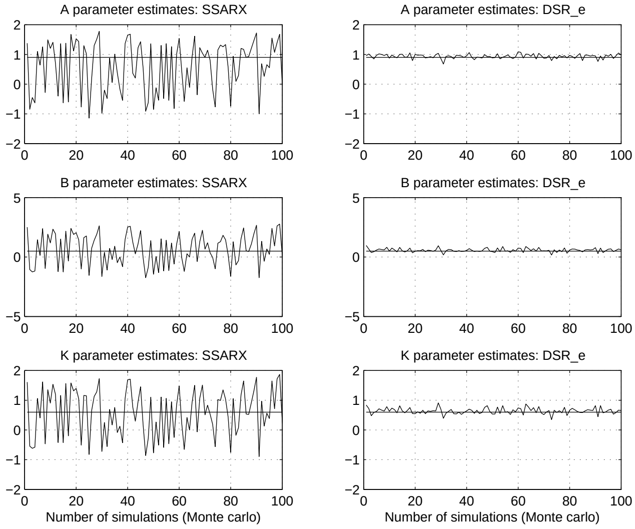

Figure 5: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The SSARX method as in Jansson (2003).*

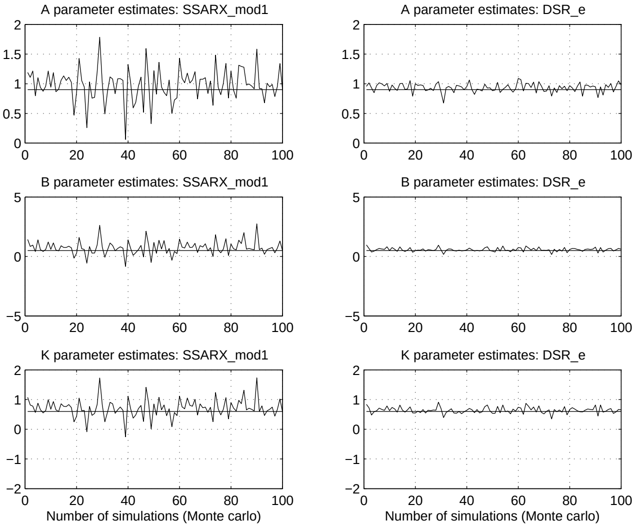

Figure 6: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The 1st modified SSARX method as presented in Section 5.1.*

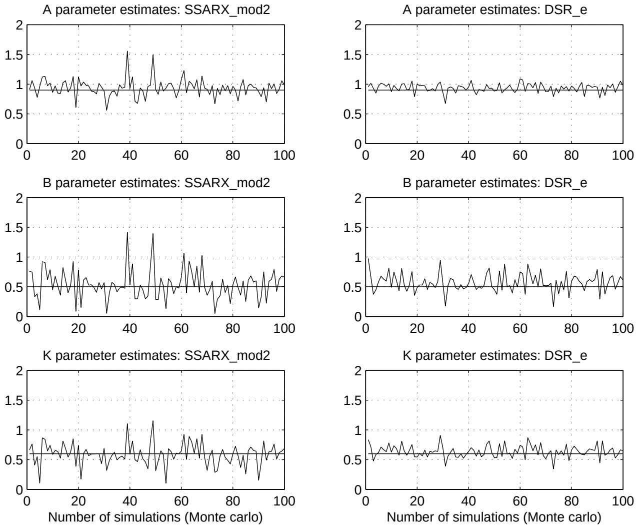

Figure 7: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The 2nd modified SSARX method as presented in Section 5.2.*

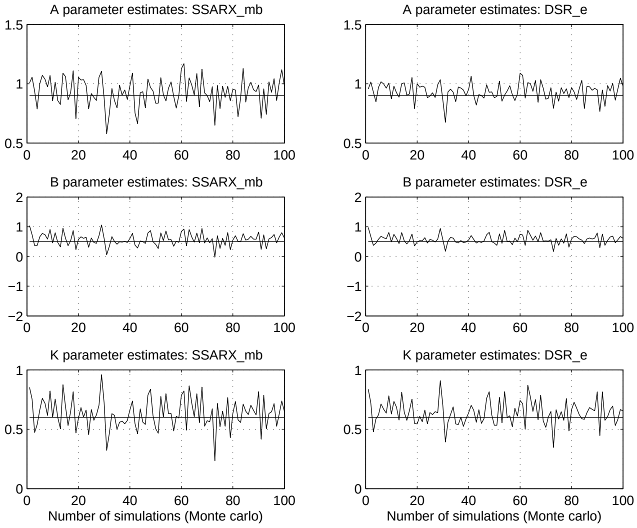

Figure 8: *Parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Future horizon $L = 3$ and past horizon $J = 6$. The bootstrap subspace method as presented in Section 6.*

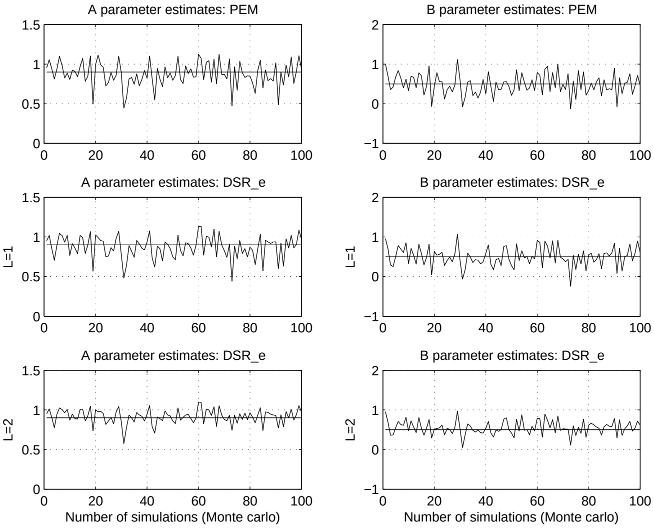

Figure 9: *A and B parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Varying future horizon parameter, $L = 1, 2$ and fixed past horizon $J = 6$ for the DSR_e method,* Di Ruscio (2009).

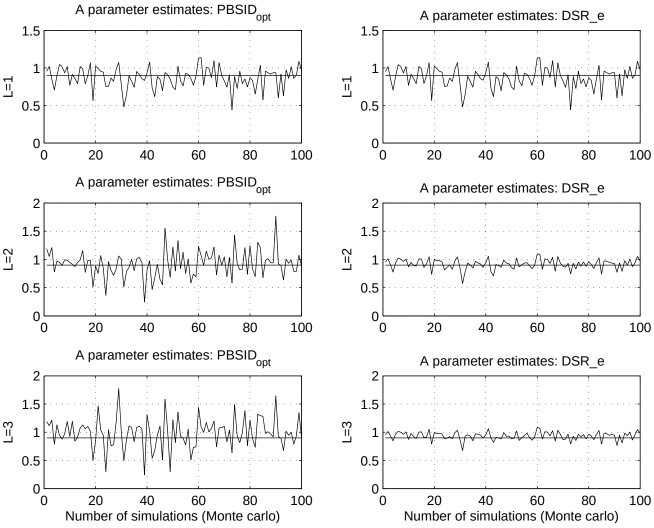

Figure 10: *A parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Varying future horizon parameter, $L = 1, 2, 3$ and fixed past horizon $J = 6$. Results from PBSID_opt,* Chiuso (2007b), *and DSR_e,* Di Ruscio (2009)

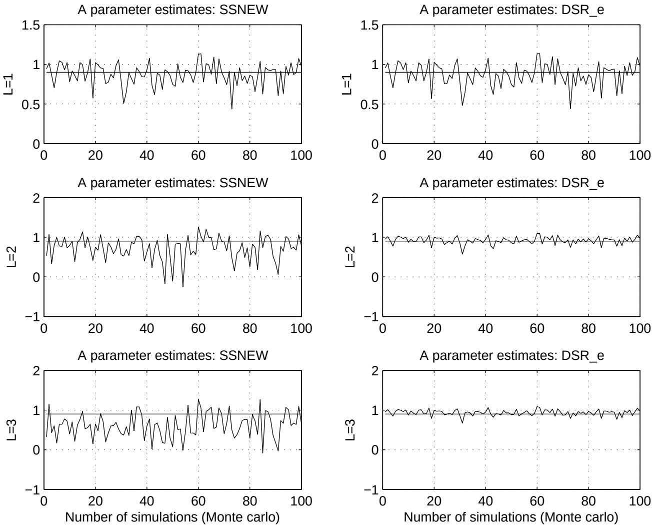

Figure 11: *A parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 1. Varying future horizon parameter, $L = 1, 2, 3$ and fixed past horizon $J = 6$. Results from SSNEW,* Ljung and McKelvey (1995)*, and DSR_e,* Di Ruscio (2009).

the size of the covariance matrix of the error between the estimates and the true parameters, i.e.,

$$P_{\text{alg}} = \frac{N}{M-1}\sum_{i=1}^{M}(\hat{\theta}_i - \theta_0)(\hat{\theta}_i - \theta_0)^T, \tag{69}$$

as

$$V_{\text{alg}} = \text{trace}(P_{\text{alg}}), \tag{70}$$

where subscript "alg" means the different algorithms, i.e. PEM, DSR_e, PBSID_opt, SSNEW SSARX, SSARX_mod and SSARX_mod2. For Case 1 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 as in Table 1 and for Case 2 as in Table 2.

As we see the variance is close and approximately the same for $L = 1$. However, when the future horizon increases the variance of the PBSID_opt, SSARX and SSNEW methods seems to diverge and produce bad estimates. Interestingly, the DSR_e method behaves completely different and produce parameter estimates with a smaller variance as the future horizon increases. This is also the case for the bootstrap subspace method SSARX_mb which produces estimates close to the corresponding estimates from DSR_e, see Tables 1 and 2.

In order to investigate the convergence properties of the bootstrap method, SSARX_mb, we implemented

Table 1: Results from Monte Carlo simulations, with parameters as in Case 1. The trace of the covariance matrix, P_{alg}, of the error between estimates and true parameters. A pseudo random ref. signal, $R = 0.1 * \mathit{idinput}(N,'\mathit{prbs}',[0,1/50])$ with $N = 1000$ and past horizon $J = 6$. SISO example with $A = 0.9$, $B = 0.5$, $K = 0.6$ and $K_p = 0.6$.

V_{alg}	$L = 1$	$L = 2$	$L = 3$	$L = 5$
V_{PEM}	115.47	115.47	115.47	115.47
$V_{\text{DSR\_e}}$	99.93	53.31	44.70	44.70
$V_{\text{PBSID\_opt}}$	99.96	350.79	569.38	1128.8
V_{SSNEW}	95.19	486.69	761.08	1459.1
V_{SSARX}	99.94	2287.4	3189.9	4063.7
$V_{\text{SSARX\_mod1}}$	99.94	355.14	569.90	1110.9
$V_{\text{SSARX\_mod2}}$	99.79	115.94	119.12	144.96
$V_{\text{SSARX\_mb}}$	99.93	107.15	109.04	110.65

Table 2: Results from Monte Carlo simulations, with parameters as in Case 2. The trace of the covariance matrix, P_{alg}, of the error between estimates and true parameters. A pseudo random ref. signal, $R = 0.1 * \mathit{idinput}(N,'\mathit{prbs}',[0,1/50])$ with $N = 1000$ and past horizon $J = 6$. Integrator example with $A = 1$, $B = 0.5$, $K = 1$ and $K_p = 0.6$.

V_{alg}	$L = 1$	$L = 2$	$L = 3$	$L = 5$
V_{PEM}	665.44	665.44	665.44	665.44
$V_{\text{DSR\_e}}$	564.25	144.67	57.98	26.65
$V_{\text{PBSID\_opt}}$	564.29	677.17	828.28	1256.8
V_{SSNEW}	553.95	2539.1	3590.0	5596.8
V_{SSARX}	564.28	18921.5	31483.2	53966.3
$V_{\text{SSARX\_mod1}}$	564.28	677.91	830.97	1291.1
$V_{\text{SSARX\_mod2}}$	564.04	568.76	572.15	981.58
$V_{\text{SSARX\_mb}}$	564.04	343.38	216.74	154.09

Table 3: Results from Monte Carlo simulations, with parameters as in Case 1. The trace of the covariance matrix, P_{alg}, of the error between estimates and true parameters. A Gaussian white noise ref. signal, $R = \mathit{randn}(N,1))$, i.e. with unit variance and $N = 1000$. Past horizon $J = 6$. SISO example with $A = 0.9$, $B = 0.5$, $K = 0.6$ and $K_p = 0.6$.

V_{alg}	$L = 1$	$L = 2$	$L = 3$	$L = 5$
V_{PEM}	24.36	24.36	24.36	24.36
$V_{\text{DSR\_e}}$	5.84	4.13	4.18	5.38
$V_{\text{PBSID\_opt}}$	4.49	3.87	3.86	3.88
V_{SSNEW}	4.25	5.60	5.82	6.45
V_{SSARX}	4.40	4.64	4.72	4.62
$V_{\text{SSARX\_mod1}}$	4.40	4.58	4.67	4.54
$V_{\text{SSARX\_mod2}}$	4.42	4.52	4.69	4.63
$V_{\text{SSARX\_mb}}$	4.42	5.08	4.77	4.99

Table 4: Parameters as in Case 1. The trace of the covariance matrix, P_{alg}, of the error between estimates and true parameters. Varying future horizon, L, vs the number of bootstrap iterations, i_b, of the algorithm SSARX_mb. Result from the higher order ARX regression problem as initial model.

$L \vert i_b$	$i_b = 1$	$i_b = 2$	$i_b = 3$	$i_b = 4$
1	99.79	99.79	99.79	99.79
2	115.94	87.12	87.24	87.23
3	119.12	80.00	79.45	79.59

Table 5: Parameters as in Case 1. The trace of the covariance matrix, P_{alg}, of the error between estimates and true parameters. Varying future horizon, L, vs the number of bootstrap iterations, i_b, in the SSARX_mb algorithm. A zero initial model, i.e., $A_0 = 0$, $B_0 = 0$, $D_0 = 0$ and $K_0 = 0$.

$L \vert i_b$	$i_b = 1$	$i_b = 2$	$i_b = 3$	$i_b = 4$
1	99.79	99.79	99.79	99.79
2	115.94	87.43	87.23	87.23
3	119.12	80.89	79.44	79.59

the algorithm with different number of bootstrap iterations, and different initial models. See Tables 4 and 5. As we see from this example $i_b = 2$ or $i_b = 3$ bootstrap iterations is enough. And even more impressive, the bootstrap method seems to converge to the same model, irrespective of initial model. This is good news for which statistically efficiency possibly may be proven.

As we see from Figs. 2 to 12 presented in this paper, the estimates from the DSR_e algorithms seem to be superior compared to the corresponding results from the other subspace algorithms, SSNEW, SSARX and PBSID_opt.

Also the DSR_e algorithm with future horizon parameter $L = 3$ seems to produce parameter estimates with smaller variance compared to the corresponding estimates from PEM, see Figs. 2 and 9. This may also be seen from Tables 1 and 2. The reason for this is not clear, because PEM is capable of producing optimal estimates. However, this may be due to a “trade off between bias and variance”, but this is not further discussed. Notice as mentioned that the parameter estimates from DSR_e are unbiased and approximately similar to the PEM estimates for $L = 1$ on this example, but the variance seems to be smaller but biased for increasing future horizon and in particular $L = 3$. Noticing that the behavior of the PBSID_opt, SSNEW and SSARX methods seems to diverge for increasing future horizon.

This same behavior is observed with the parameters in Case 2 above. The results are approximately the same for future horizon $L = 1$. In Fig. 12 we illustrate the A parameter estimates from PBSID_opt and DSR_e for varying future horizon parameter L. The performance of the DSR_e algorithm seems to be superior compared to the PBSID_opt algorithm.

Finally, we also used a white reference signal in the Case 1 example and the results are presented in Table 3. In this case all methods produced approximately the same model but with PBSID_opt producing the smallest criterion V_{alg}. However, the differences between the methods are minor.

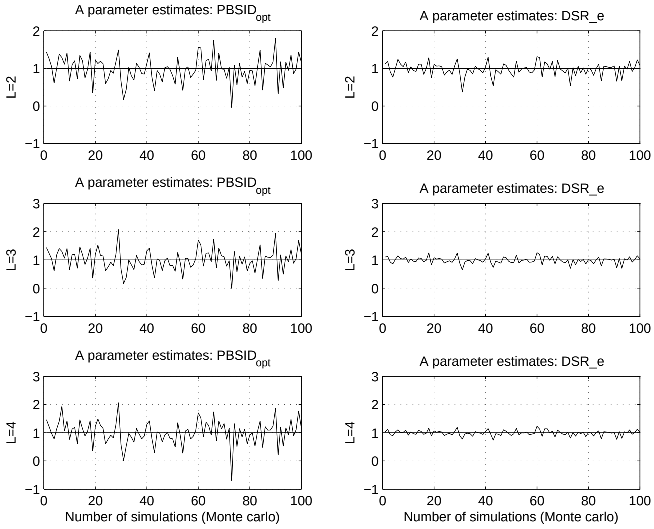

Figure 12: *A parameter estimates of the closed loop system in Example 8.1 with parameters as in Case 2. Varying future horizon parameter, $L = 2, 3, 4$ and fixed past horizon $J = 6$.*

The estimation results from the SSARX_mb algorithm in Table 1 converged to the same $V_{\text{SSARX\_mb}}$ values irrespective of the initial models used, e.g. for a zero initial model, an initial model from a higher order ARX strategy or using the results from the SSARX algorithm as initial values.

8.2. 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

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

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

The feedback is obtained with

$$G = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}. \quad (76)$$

and r_k with a binary sequence as a reference for each of the two outputs in y_k, generated by using the MATLAB IDENT Toolbox function idinput, i.e., $R = \begin{bmatrix} R_1 & R_2 \end{bmatrix}$ with $R_1 = idinput(N,'prbs', [0, 1/50])$ and $R_2 = idinput(N,'prbs', [0, 1/75])$.

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 three eigenvalues of the system, one real and a complex conjugate pair, i.e.

$$\begin{aligned} \lambda_1 &= 0.85, && (77)\\ \lambda_{2,3} &= 0.75 \pm j0.37. && (78) \end{aligned}$$

A Monte Carlo simulation with $M = 100$ different experiments, $1 \leq i \leq M$, is performed. The simulation results for the eigenvalue estimates from PEM, PBSID_opt, DSR_e and the SSARX_mb methods are illustrated in Fig. 13 for number of samples $N = 1000$ and in Fig. 14 for $N = 10000$ samples. For the DSR_e and SSARX_mb methods we used the algorithm horizon parameters, $J = 6$ and $L = 2$. For the PBSID_opt method we had to chose $L = 3$ in order to obtain results, probably du to the definition of the future horizon parameter in this method.

The initial model for the SARX_mb method was taken from a higher order ARX model with past horizon $J = 6$. For this example the SSARX_mb method converged to a biased model when a zero initial model was used, which indicates that the initial model for the SSARX_mb method should be reasonable.

For this MIMO system with three states example we observed sensitivity for the initial model chosen in the new bootstrap subspace identification algorithm SSARX_mb. Using a zero initial model resulted in a biased model, however using an initial model from the higher order ARX strategy resulted in a unbiased model with approximately the same variance as the corresponding PEM estimates.

9. Conclusion

We have in this paper shown by numerical Monte Carlo simulations, based on one example presented, that the

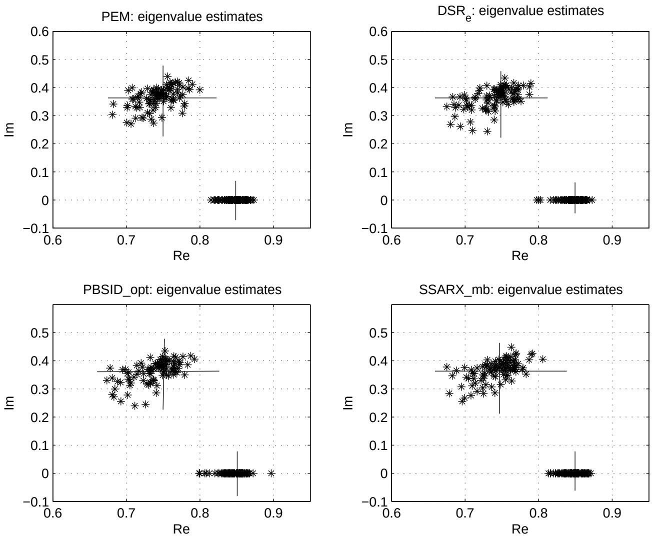

Figure 13: *Eigenvalue estimates from the PEM, PBSID_opt, DSR_e and SSARX_mb methods. The correct eigenvalues are* $\lambda_1 = 0.85$ *and a complex conjugate pair,* $\lambda_{2,3} = 0.75 \pm j0.3708$. *Complex eigenvalues are in complex conjugate pairs. Hence, the negative imaginary part is not presented. Number of samples,* $N = 1000$.

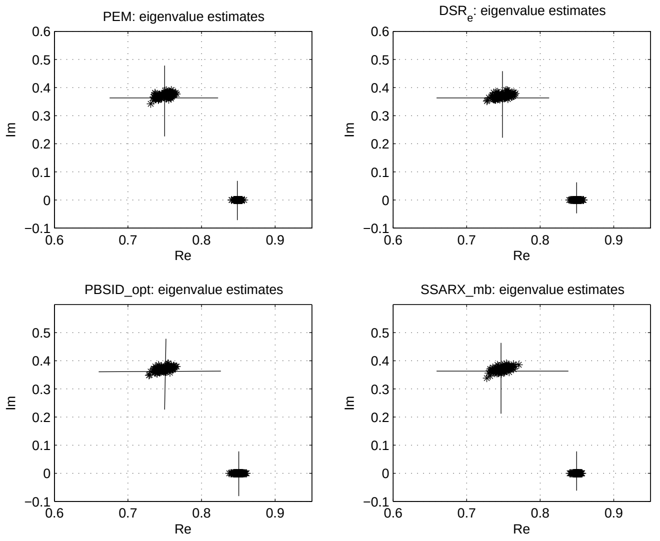

Figure 14: *Eigenvalue estimates from the PEM, PBSID_opt, DSR_e and SSARX_mb methods. The correct eigenvalues are* $\lambda_1 = 0.85$ *and a complex conjugate pair,* $\lambda_{2,3} = 0.75 \pm j0.3708$. *Complex eigenvalues are in complex conjugate pairs. Hence, the negative imaginary part is not presented. Number of samples,* $N = 10000$.

DSR_e algorithm outperforms the SSNEW, SSARX and PBSID_opt methods for closed loop subspace system identification, in particular for systems with colored (non-white) reference signals and when the future horizon is larger than the system order, i.e., $L > n$, i.e. on the examples presented in this paper. One other examples the performance are more similar.

On the SISO one state example presented the variance is approximately the same when the future horizon, L, is equal to the system order ($L = 1$). However, the algorithms behave completely differently for increasing future horizon. The DSR_e algorithm produced estimation results with smaller variance which seems to converge, while the corresponding variance from the SSNEW, SSARX and PBSID_opt algorithms produced estimation results with increasing variance as the future horizon increases.

The original SSARX algorithm is discussed and a new modified version is presented. We have presented a modified implementation of the SSARX algorithm, Jansson (2003), which produces estimates with variances which are close to the variance on the corresponding PBSID_opt algorithm, Chiuso (2007b), even for a finite data example and for colored reference signal.

Based on the modified SSARX algorithm we present a new bootstrap subspace identification algorithm which is found promising. This bootstrap algorithm seems to converge after a few iterations and seems to produce parameter estimates close to the corresponding estimates from PEM and the DSR_e algorithm. Monte Carlo simulation experiments indicate some asymptotic equivalent properties with the bootstrap subspace method and the DSR_e method. This work is believed to be important for the search for a variance efficient subspace identification algorithm.

A. Canonical correlation analysis

In Jansson (2003, 2005) the Canonical Correlation Analysis (CCA) also known as Canonical Variate Analysis (CVA) is used in order to obtain the estimate of $\tilde{O}_L$ and the states $X_{J|L}$. See also Overschee and de Moor (1996). Consider the LS problem, eq. (22), with parameter matrix, $\tilde{\theta}$, defined in eq. (24), i.e.,

$$\tilde{Y}_{J|L} = \tilde{\theta} W_p, \tag{79}$$

where W_p contains the past data Hankel matrices as defined in eq. (26). The OLS estimate of eq. (79) is

$$\hat{\tilde{\theta}} = \tilde{Y}_{J|L} W_p (W_p W_p^T)^{-1}. \tag{80}$$

Define a matrix, M, which is a weighting of the parameter matrix estimate, eq. (80), as follows

$$\begin{aligned} M &= (\tilde{Y}_{J|L} \tilde{Y}_{J|L}^T)^{-\frac{1}{2}} \hat{\tilde{\theta}} (W_p W_p^T)^{\frac{1}{2}} \\ &= (\tilde{Y}_{J|L} \tilde{Y}_{J|L}^T)^{-\frac{1}{2}} \tilde{Y}_{J|L} W_p^T (W_p W_p^T)^{-\frac{1}{2}}. \end{aligned} \tag{81}$$

Perform the SVD of M

$$M = USV^T \approx U_1 S_1 V_1^T, \tag{82}$$

where $S_1 = S(1:n, 1:n)$ contains the n largest singular values different from “one”, not “zero”, since the “canonical angles” are considered, and $U_1 = U(:, 1:n)$, $V_1(:, 1:n)$. Putting eq. (81) equal to the approximation in eq. (82), i.e.,

$$M = (\tilde{Y}_{J|L}\tilde{Y}_{J|L}^T)^{-\frac{1}{2}}\hat{\tilde{\theta}}(W_p W_p^T)^{\frac{1}{2}} = U_1 S_1 V_1^T, \tag{83}$$

which gives

$$\hat{\tilde{\theta}} = (\tilde{Y}_{J|L}\tilde{Y}_{J|L}^T)^{\frac{1}{2}} U_1 S_1 V_1^T (W_p W_p^T)^{-\frac{1}{2}}. \tag{84}$$

In Jansson (2003, 2005) an input normal realization is used and hence,

$$\hat{\tilde{O}}_L = (\tilde{Y}_{J|L}\tilde{Y}_{J|L}^T)^{\frac{1}{2}} U_1 S_1, \tag{85}$$

$$\hat{X}_{J|1} = V_1^T (W_p W_p^T)^{-\frac{1}{2}}. \tag{86}$$

References

Chiuso, A. On the relation between CCA and predictor based subspace identification. *IEEE Transaction on Automatic Control*, 2007a. 52(10):1795–1812. doi:10.1109/TAC.2007.906159.

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

Chiuso, A. and Picci, G. Consistency analysis of some closed-loop subspace identification methods. *Automatica*, 2005. 41(3):377–391. doi:10.1016/j.automatica.2004.10.015.

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.

Di Ruscio, D. Closed and Open Loop Subspace System Identification of the Kalman Filter. *Modeling, Identification and Control*, 2009. 30(2):71–86. doi:10.4173/mic.2009.2.3.

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.

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

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

Ljung, L. *System Identification: Theory for the User*. Prentice Hall PTR, 1999.

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

Ljung, L. and McKelvey, T. Subspace identification from closed loop data. *Signal Processing*, 1996. 52(12):209–215. doi:10.1016/0165-1684(96)00054-0.

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*. pages 887–892, 2003.

Qin, S. J. and Ljung, L. On the role of future horizon in closed-loop subspace identification. In *Proc. 14th IFAC SYSID Symposium*. pages 1080–1084, 2006.

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.

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.