91-17

SVD-based subspace methods for multivariable continuous-time systems identification

Marc Moonen, Bart De Moor, Joos Vandewalle
ESAT Katholieke Universiteit Leuven
K.Mercierlaan 94, 3001 Heverlee, Belgium

ESAT/SISTA 1991-17

Abstract

Recently several subspace methods have appeared in the literature for multivariable discrete-time state space identification, where state space models are computed directly from input/output data. These state space identification methods are viewed as the better alternatives to polynomial model identification, owing to the better numerical conditioning associated with state space models, especially for high-order multivariable systems.

In this contribution, a similar method is described for continuous-time state space identification. Here also, the key tool is the singular value decomposition (SVD), a numerical technique known to be very robust and accurate when dealing with noisy data. The noise coloring is compensated for by using a generalization of the SVD, namely the quotient SVD. The resulting identification scheme is then shown to give consistent results under certain conditions.

This work was sponsored (partly) by the BRA 3280 project of the EC. Marc Moonen is a senior research assistant with the N.F.W.O. (Belgian National Fund for Scientific Research). Bart De Moor is a research associate with the N.F.W.O.

91-17

Moonen M., De Moor B., Vandewalle J., ``SVD-based subspace methods for multivariable continuous-time systems identification", in *Identification of Continuous-Time Systems*, (Rao G.P., and Sinha N.K., eds.), Kluwer Academic Publications, 1991, pp. 473-488., Lirias number: 180915.

91-17

1 Introduction

The greater part of the systems identification literature is concerned with computing polynomial system models, which are however known to typically give rise to numerically ill-conditioned mathematical problems. While state space models on the other hand show much more desirable numerical properties, direct state space identification is clearly a much less established area. Most commonly known are the realization algorithms of [Ho and Kalman 1966, Zeiger and Mc Ewen 1974, Kung 1978], where a discrete-time state space model is computed from a block Hankel matrix with Markov parameters. The Markov parameters might be hard to obtain though[1]*, and furthermore –as far as we know– the latter algorithms do not allow for treating the effect of inexact data. An alternative direct identification scheme is described in [Moonen et al. 1989a] and [Moonen & Vandewalle 1990], where a discrete-time state space model is computed from a block Hankel matrix with I/O data. The key tool is the singular value decomposition (SVD), a numerical technique, known to be very robust and accurate when dealing with noisy data. In the colored noise case, the noise coloring is compensated for by using a generalization of the SVD, namely the quotient SVD (QSVD) [De Moor & Golub 1989]. These algorithms are readily convertible into adaptive identification algorithms for model updating, where use is made of (Q)SVD updating techniques.*

In this contribution, a similar method is described for continuous-time state space identification. Due to the necessary use of a certain pre-filtering, the procedure itself is seen to always introduce a noise coloring, so that a QSVD is needed in any case. First, an algorithm for state space identification in the noise-free case is developed in sections 2 and 3. In section 4, we then consider the case where the data are corrupted by noise and describe the general identification set-up. The derivation of these algorithms is analogous to the derivation of the discrete-time algorithms in [Moonen et al. 1989a] and [Moonen & Vandewalle 1990]. We therefore refer to those reports for further details.

[1] for infinite impulse response systems, especially with unstable poles, computing a deconvolution directly –i.e. by solving a set of linear equations– is impossible

2 Preliminaries

For the time being, we consider linear time invariant multivariable systems with state space description as follows

$$\begin{aligned} \dot{x}(t) &= A \cdot x(t) + B \cdot u(t) \\ y(t) &= C \cdot x(t) + D \cdot u(t) \end{aligned}$$

where $u(t), y(t)$ and $x(t)$ denote the input (m-vector), output (l-vector) and state vector at time t, the dimension of $x(t)$ being the minimal system order n. A, B, C and D are the unknown system matrices to be identified.

In the sequel, we often make use of matrices which are constructed with samples of the in- and outputs and their derivatives. We let $u^{(k)}$ and $y^{(k)}$ denote the k-th derivative of u and y, and define

$$\underbrace{U}_{2mi \times j} = \begin{bmatrix} u(t_1) & u(t_2) & \dots & u(t_j) \\ u^{(1)}(t_1) & u^{(1)}(t_2) & \dots & u^{(1)}(t_j) \\ u^{(2)}(t_1) & u^{(2)}(t_2) & \dots & u^{(2)}(t_j) \\ \vdots & \vdots & & \vdots \\ u^{(2i-1)}(t_1) & u^{(2i-1)}(t_2) & \dots & u^{(2i-1)}(t_j) \end{bmatrix}$$

$$\underbrace{Y}_{2li \times j} = \begin{bmatrix} y(t_1) & y(t_2) & \dots & y(t_j) \\ y^{(1)}(t_1) & y^{(1)}(t_2) & \dots & y^{(1)}(t_j) \\ y^{(2)}(t_1) & y^{(2)}(t_2) & \dots & y^{(2)}(t_j) \\ \vdots & \vdots & & \vdots \\ y^{(2i-1)}(t_1) & y^{(2i-1)}(t_2) & \dots & y^{(2i-1)}(t_j) \end{bmatrix}.$$

Here, the sampling intervals $t_2 - t_1$, $t_3 - t_2$, etc. need not even be equidistant. The problems associated with computing derivatives are tackled in section 4. For the time being we thus assume that these matrices can indeed be constructed. We now briefly state three important theorems which are used in the sequel for developing our identification algorithms.

Theorem 1
Time sequences $u(t), y(t), x(t)$ that satisfy the above state space equations, also satisfy the following general structured I/O-equation :

$$Y = \Gamma_{2i} \cdot X + T_{2i} \cdot U.$$

Here U and Y are the above defined data matrices, and X contains a corresponding sampling of the state vector :

$$X = \begin{bmatrix} x(t_1) & x(t_2) & \dots & x(t_j) \end{bmatrix}.$$

Finally

$$\Gamma_{2i} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{2i-1} \end{bmatrix}, \qquad T_{2i} = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & \vdots & & \vdots \\ CA^{2i-2}B & CA^{2i-3}B & \dots & D \end{bmatrix}.$$

The proof of this is straightforward by repeated substitution of the state space equations.

□

Instead of going into details, we loosely state that i and j should be chosen 'sufficiently large', so that Y and U contain enough information on the system, and in particular $j \gg \max(2mi, 2li)$ ('short fat' matrices), as this reduces the noise sensitivity (see below).

Theorem 2
Let Y, U and X be defined as in the previous theorem, and let H denote the concatenation of Y and U

$$H = \begin{bmatrix} U \\ Y \end{bmatrix}$$

then, under the conditions that
(i) $\text{rank}\{X\} = n$ *where n is the minimal system order, and*
(ii) $\text{span}_{\text{row}}\{X\} \cap \text{span}_{\text{row}}\{U\} = \emptyset$,
the following rank property holds :

$$\text{rank}\{H\} = \text{rank}\{U\} + n.$$

In other words the row space of Y adds n dimensions to the row space of U. Also, when
(iii) $\text{rank}\{U\} = 2mi =$ *number of rows in U ,*
this rank property reduces to

$$\text{rank}\{H\} = 2mi + n.$$

This theorem allows to estimate the system order, prior to further identification of the system matrices. For a proof of this, we refer to [Moonen et al. 1989], where a similar proof is given for the discrete-time case.

□

Note on condition (i) : $\text{rank}\{X\} = n$ *means that all modes are sufficiently excited (persistent excitation). When certain modes are not sufficiently excited, i.e. unobservable in the I/O-data, they cannot be identified either and application of the above rank property will reveal too low a system order. This problem is inherent in system identification.*

Note on condition (ii) : *When this condition is not satisfied, application of the rank property again reveals an underestimation of the system order. However it can be verified that rank cancellation is not generic*[2]*, and the probability that rank cancellation occurs, decreases for fixed 2i (number of block rows in U) with increasing j (number of columns in U and X).*

Note on condition (iii) : *Similar to the previous ones, this third condition is generically satisfied when the input is sufficiently rich (inherent in the identification problem).*

In the following, it is always assumed that these three conditions are satisfied.

Suppose we would now cut the above matrices U and Y in half as follows

$$U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \qquad Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

with

$$\underbrace{U_1}_{mixj} = \begin{bmatrix} u(t_1) & u(t_2) & \dots & u(t_j) \\ u^{(1)}(t_1) & u^{(1)}(t_2) & \dots & u^{(1)}(t_j) \\ \vdots & \vdots & & \vdots \\ u^{(i-1)}(t_1) & u^{(i-1)}(t_2) & \dots & u^{(i-1)}(t_j) \end{bmatrix}$$

$$\underbrace{U_2}_{mixj} = \begin{bmatrix} u^{(i)}(t_1) & u^{(i)}(t_2) & \dots & u^{(i)}(t_j) \\ u^{(i+1)}(t_1) & u^{(i+1)}(t_2) & \dots & u^{(i+1)}(t_j) \\ \vdots & \vdots & & \vdots \\ u^{(2i-1)}(t_1) & u^{(2i-1)}(t_2) & \dots & u^{(2i-1)}(t_j) \end{bmatrix}$$

[2] unless the system is controlled by state feedback $u_k = -Fx_k$

and Y_1, Y_2 similarly constructed. Both matrix pairs satisfy an I/O-equation as follows:

$$\begin{aligned} Y_1 &= \Gamma_i \cdot X + T_i \cdot U_1 \\ Y_2 &= \Gamma_i \cdot X^{(i)} + T_i \cdot U_2 \end{aligned}$$

with

$$X^{(i)} = \begin{bmatrix} x^{(i)}(t_1) & x^{(i)}(t_2) & \dots & x^{(i)}(t_j) \end{bmatrix}.$$

The corresponding derivatives satisfy

$$\begin{aligned} Y_1^{(1)} &= \Gamma_i \cdot X^{(1)} + T_i \cdot U_1^{(1)} \\ Y_2^{(1)} &= \Gamma_i \cdot X^{(i+1)} + T_i \cdot U_2^{(1)} \end{aligned}$$

with

$$X^{(i+1)} = \begin{bmatrix} x^{(i+1)}(t_1) & x^{(i+1)}(t_2) & \dots & x^{(i+1)}(t_j) \end{bmatrix}.$$

Theorem 3
With $H_1, H_2, X^{(i)}$ defined as above

$$\text{span}_{\text{row}}\{X^{(i)}\} = \text{span}_{\text{row}}\{H_1\} \cap \text{span}_{\text{row}}\{H_2\}$$

so that any basis for this intersection constitutes a valid state vector sequence $X^{(i)}$, with the basis vectors as the consecutive row vectors. Similarly, we have

$$\text{span}_{\text{row}}\{X^{(i+1)}\} = \text{span}_{\text{row}}\{H_1^{(1)}\} \cap \text{span}_{\text{row}}\{H_2^{(1)}\}.$$

Note that different choices for a basis differ in a transformation matrix P that transforms a model A,B,C,D into an equivalent model $P^{-1}AP$, $P^{-1}B$, CP, D. For a proof of the above theorem, we again refer to [Moonen et al. 1989], where a similar proof is given for the discrete-time case.

□

The above theorem allows to calculate derivatives of the state vector sequence, by making use of the measured I/O-data and their derivatives only. Once these sequences are known, the system matrices are readily identified from a set of linear equations as follows

$$\begin{bmatrix} x^{(i+1)}(t_1) & \dots & x^{(i+1)}(t_j) \\ y^{(i)}(t_1) & \dots & y^{(i)}(t_j) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} x^{(i)}(t_1) & \dots & x^{(i)}(t_j) \\ u^{(i)}(t_1) & \dots & u^{(i)}(t_j) \end{bmatrix}.$$

The above results constitute the heart of a two-step identification scheme. First the state vector sequences are computed as the intersection of the row spaces of certain I/O data matrices. Then the system matrices are obtained all at once as the solution of a set of linear equations. In the next sections, this procedure is built into a practical computational scheme.

3 An SVD based algorithm for the noise free case

In this section, we derive a computational scheme for the case where there is no noise on the data. We also assume this in that case computing derivatives poses no problems. In the next section, the more general case with additive noise is considered.

For compact notation, it is useful to first redefine matrices H_1 and H_2 in the following way :

$$H_1 = \begin{bmatrix} u(t_1) & u(t_2) & \dots & u(t_j) \\ y(t_1) & y(t_2) & \dots & y(t_j) \\ u^{(1)}(t_1) & u^{(1)}(t_2) & \dots & u^{(1)}(t_j) \\ y^{(1)}(t_1) & y^{(1)}(t_2) & \dots & y^{(1)}(t_j) \\ \vdots & \vdots & & \vdots \\ u^{(i-1)}(t_1) & u^{(i-1)}(t_2) & \dots & u^{(i-1)}(t_j) \\ y^{(i-1)}(t_1) & y^{(i-1)}(t_2) & \dots & y^{(i-1)}(t_j) \end{bmatrix}$$

$$H_2 = \begin{bmatrix} u^{(i)}(t_1) & u^{(i)}(t_2) & \dots & u^{(i)}(t_j) \\ y^{(i)}(t_1) & y^{(i)}(t_2) & \dots & y^{(i)}(t_j) \\ u^{(i+1)}(t_1) & u^{(i+1)}(t_2) & \dots & u^{(i+1)}(t_j) \\ y^{(i+1)}(t_1) & y^{(i+1)}(t_2) & \dots & y^{(i+1)}(t_j) \\ \vdots & \vdots & & \vdots \\ u^{(2i-1)}(t_1) & u^{(2i-1)}(t_2) & \dots & u^{(2i-1)}(t_j) \\ y^{(2i-1)}(t_1) & y^{(2i-1)}(t_2) & \dots & y^{(2i-1)}(t_j) \end{bmatrix}.$$

Notice that theorem 3 remains valid. We also introduce the following notation :

$M\{p:q, r:s\}$ is the submatrix of M at the intersection of rows $p, p+1, \dots, q$ and columns $r, r+1, \dots, s$

$M\{:, r:s\}$ is the submatrix of M containing columns $r, r+1, \dots, s$

$M\{p:q, :\}$ is the submatrix of M containing rows $p, p+1, \dots, q$.

As an example, referring to theorem 3,

$$H_1^{(1)} = H\{m+l+1 : (i+1)(m+l), :\}$$

where H is the concatenation $\begin{bmatrix} H_1 \\ H_2 \end{bmatrix}$.

The intersection of the row spaces of H_1 and H_2, can be computed from the SVD of H as follows

$$\begin{aligned} H &= \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \\ &= Q \cdot \Sigma \cdot V^T \\ &= \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & 0 \end{bmatrix} \cdot V^T \end{aligned}$$

$$\begin{aligned} \dim\{Q_{11}\} = \dim\{Q_{21}\} &= (mi+li) \times (2mi+n) \\ \dim\{Q_{12}\} = \dim\{Q_{22}\} &= (mi+li) \times (2li-n) \\ \dim\{\Sigma_{11}\} &= (2mi+n) \times (2mi+n). \end{aligned}$$

Here Q and V are orthogonal matrices, Σ is a diagonal matrix. From

$$Q_{12}^T \cdot H_1 = -Q_{22}^T \cdot H_2$$

it follows that the row space of $Q_{12}^T \cdot H_1$ equals the required intersection. However, $Q_{12}^T \cdot H_1$ contains $2li - n$ row vectors , only n of which are linearly independent (dimension of the intersection). Thus, it remains to select n suitable combinations of these row vectors. Making use of a CS-decomposition [Golub & Van Loan 1983], one easily shows that

$$Q_{12} = \begin{bmatrix} \underbrace{Q'_{12}}_{li-n} & \underbrace{Q''_{12}}_{n} & \underbrace{Q'''_{12}}_{li-n} \end{bmatrix} \cdot \begin{bmatrix} I & & \\ & C & \\ & & 0 \end{bmatrix} \cdot W^T$$

$$Q_{22} = \begin{bmatrix} \underbrace{Q'_{22}}_{li-n} & \underbrace{Q''_{22}}_{n} & \underbrace{Q'''_{22}}_{li-n} \end{bmatrix} \cdot \begin{bmatrix} O & & \\ & S & \\ & & I \end{bmatrix} \cdot W^T$$

$$
\begin{aligned}
C &= diag\{c_1, \ldots, c_n\} \\
S &= diag\{s_1, \ldots, s_n\} \\
C^2 + S^2 &= I_{n\times n}.
\end{aligned}
$$

Clearly, only Q''_{12} delivers useful combinations for the computation of the intersection, and we can take

$$
\begin{aligned}
X^{(i)} &= (Q''_{12})^T \cdot H_1 \\
&= (Q''_{12})^T \cdot H\{1 : i(m+l), :\}.
\end{aligned}
$$

The above expressions for Q_{12} and Q_{22} are in itself SVD's of these matrices, and can be computed as such. It thus suffices to compute e.g. the SVD of Q_{12}, and select left singular vectors Q''_{12} corresponding to singular values $\sigma \neq 0, 1$. The computation of the required intersection then reduces to the computation of two successive SVD's (for H and Q_{12} respectively). Finally, from the above formula, it also follows that

$$
\begin{aligned}
X^{(i+1)} &= (Q''_{12})^T \cdot H_1^{(1)} \\
&= (Q''_{12})^T \cdot H\{m+l+1 : (i+1)(m+l), :\}.
\end{aligned}
$$

In the second step, the system matrices are identified from a set of linear equations. Much as was done in [Moonen et al. 1989], it can be shown that the system matrices can be computed from a reduced set as follows (obtained after discarding the common orthogonal factor V)

$$
\begin{bmatrix} (Q''_{12})^T \cdot Q\{m+l+1 : (i+1)(m+l), 1 : 2mi+n\} \cdot \Sigma_{11} \\ Q\{mi+li+m+1 : (m+l)(i+1), 1 : 2mi+n\} \cdot \Sigma_{11} \end{bmatrix}
= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} (Q''_{12})^T \cdot Q\{1 : mi+li, 1 : 2mi+n\} \cdot \Sigma_{11} \\ Q\{mi+li+1 : mi+li+m, 1 : 2mi+n\} \cdot \Sigma_{11} \end{bmatrix}.
$$

Here, the fact that the system matrices can be computed from a reduced set of equations introduces a significant computational saving. The largest dimension in the set of equations is now $2mi+n$ instead of j, where $j \gg 2mi+n$ ('short fat' matrices). As only Q and Σ are needed from the first (largest) SVD of H, the computational effort in the first stage is cut down as well.

4 A QSVD based algorithm for the colored noise case

In the previous section we assumed noise free data, and derived a fairly simple SVD based identification scheme. In practice however, we cannot afford this luxury, so we will have to take the noise into account explicitly, when deriving a computational scheme. When dealing with noisy I/O data, one major problem is of course the computation of the derivatives. In this section, it is shown how the problem is solved by incorporating a pre-filter, and compensating for the corresponding noise coloring with a QSVD instead of an SVD.

First, it is readily seen that the dynamic relation between inputs and outputs is not changed if all in- and outputs are filtered with one and the same linear filter $\frac{F(s)}{E(s)}$. In other words, we also have

$$
\begin{aligned}
\dot{x}_f(t) &= A \cdot x_f(t) + B \cdot u_f(t) \\
y_f(t) &= C \cdot x_f(t) + D \cdot u_f(t)
\end{aligned}
$$

with unchanged A, B, C, D-matrices, and

$$
\begin{aligned}
U_f(s) &= diag\{\frac{F(s)}{E(s)}\} \cdot U(s) \\
Y_f(s) &= diag\{\frac{F(s)}{E(s)}\} \cdot Y(s) \\
X_f(s) &= diag\{\frac{F(s)}{E(s)}\} \cdot X(s).
\end{aligned}
$$

Here $\frac{F(s)}{E(s)}$ is a scalar valued rational function, which we specify as follows

$$
\frac{F(s)}{E(s)} = \frac{f_p s^p + f_{p-1} s^{p-1} + \ldots f_1 s^1 + f_0}{s^{p+2i} + e_{p+2i-1} s^{p+2i-1} + \ldots e_1 s^1 + e_0}.
$$

Typically $\frac{F(s)}{E(s)}$ serves to get rid of the differentiation. The advantage of having $2i$ zeroes at infinity is apparent from **Figure 1**, *where a realization of the pre-filter is exhibited. For an arbitrary input $z(t)$ (scalar valued), the output $z_f(t)$ is produced, as well as its derivatives up to the $(2i-1)$th. The additional flexibility in choosing the other filter coefficients can e.g. be used to construct a bandpass filter if a model in a limited frequency range is searched for. Note that the pre-filter can be implemented as a digital filter as well.*

In principle, one could now use the computational scheme of the previous sections, where however the filtered data are used throughout. For instance, one would use

$$H_{1_f} = \begin{bmatrix} u_f(t_1) & u_f(t_2) & \dots & u_f(t_j) \\ y_f(t_1) & y_f(t_2) & \dots & y_f(t_j) \\ u_f^{(1)}(t_1) & u_f^{(1)}(t_2) & \dots & u_f^{(1)}(t_j) \\ y_f^{(1)}(t_1) & y_f^{(1)}(t_2) & \dots & y_f^{(1)}(t_j) \\ \vdots & \vdots & & \vdots \\ u_f^{(i-1)}(t_1) & u_f^{(i-1)}(t_2) & \dots & u_f^{(i-1)}(t_j) \\ y_f^{(i-1)}(t_1) & y_f^{(i-1)}(t_2) & \dots & y_f^{(i-1)}(t_j) \end{bmatrix}$$

and a similar H_{2_f}, etcetera. Although the problem of computing derivatives is apparently solved, the SVD scheme still fails to give consistent results, the reason for this being the noise coloring introduced by the pre-filter. Suppose the data matrix H_f is corrupted by additive noise

$$H_f = H_{f_{exact}} + H_{f_{noise}}$$

then it is well known that the SVD scheme delivers consistent results in the 'white noise case', where

$$E\{H_{f_{noise}} \cdot H_{f_{noise}}^T\} = \sigma \cdot I.$$

Here $E\{\cdot\}$ is the expectation operator and I is the identity matrix. In the general case, we have

$$E\{H_{f_{noise}} \cdot H_{f_{noise}}^T\} = \sigma \cdot L_{noise} \cdot L_{noise}^T$$

where L_{noise} is an (assumed known) lower triangular factor. The noise coloring can then be compensated for, by using a QSVD for the matrix pair $\{H, L_{noise}\}$ instead of an SVD of H [Moonen & Vandewalle 1990, De Moor & Golub 1989].

So the first thing is to compute the L_{noise} for our case. Therefore, we

convert the filter model to state space form, which reads as follows

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & I & & & & & \\ & \ddots & \ddots & & & & \\ & & \ddots & \ddots & & & \\ & & & 0 & I & & \\ & & & & -E_{2i+p-1} & I & \\ & & & & \vdots & & \ddots \\ & & & & -E_{2i} & & I \\ -E_0 & -E_1 & \dots & -E_{2i-2} & -E_{2i-1} & & 0 \end{bmatrix}}_{A_{filter}} \cdot x(t) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \\ F_p \\ \vdots \\ F_1 \\ F_0 \end{bmatrix}}_{B_{filter}} \cdot \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}$$

$$\begin{bmatrix} u_f(t) \\ y_f(t) \end{bmatrix} = \begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix} \cdot x(t)$$

where

$$E_i \stackrel{def}{=} e_i \cdot I_{(m+l)\times(m+l)}$$

$$F_i \stackrel{def}{=} f_i \cdot I_{(m+l)\times(m+l)}.$$

Here, for compact notation, we consider the pre-filtering of all the in- and outputs at once. The first $2i(m+l)$ components of the state vector constitute the filter outputs and their derivatives. Suppose now that the filter inputs are subject to additive wide-sense stationary noise with covariance V, i.e.

$$E\{\begin{bmatrix} u^T(t)_{noise} & y^T(t)_{noise} \end{bmatrix} \cdot \begin{bmatrix} u(t)_{noise} \\ y(t)_{noise} \end{bmatrix}\} = V.$$

The covariance matrix for the noise on the state vector (for $t \to \infty$) is then known to be proportional to $L \cdot L^T$, the solution of the Lyapunov equation

$$A_{filter} \cdot (L \cdot L^T) + (L \cdot L^T) \cdot A_{filter}^T + B_{filter} \cdot V \cdot B_{filter}^T = 0.$$

This equation can be solved for the lower triangular factor L directly, as described in e.g. [Hammarling 1982]. The upper left block in L, then equals L_{noise}

$$L_{noise} = L\{1 : 2i(m+l), 1 : 2i(m+l)\}.$$

The required matrix L_{noise} can thus be computed from the knowledge of V and the filter coefficients. Once L_{noise} is known, this matrix can be used in a QSVD based scheme, which is similar to the original SVD scheme. First the QSVD of the matrix pair $\{H_f, L_{noise}\}$ is computed

$$\begin{aligned} H_f &= Q^{-T} \cdot \Sigma \cdot V^T \\ L_{noise} &= Q^{-T} \cdot \Sigma_L \cdot V_L^T \end{aligned}$$

where now Q is a square non-singular matrix (not necessarily orthogonal), V and V_L are orthogonal, and Σ is a diagonal matrix with non-increasing entries along the diagonal. With the same partitioning for Q as in the SVD scheme, we now use a second QSVD for $\{Q_{12}, Q_{22}\}$ (instead of a CS-decomposition), to give

$$Q_{12} = \begin{bmatrix} Q'_{12} & Q''_{12} & Q'''_{12} \end{bmatrix} \cdot \begin{bmatrix} I & & \\ & C & \\ & & 0 \end{bmatrix} \cdot W^T$$

$$Q_{22} = \begin{bmatrix} Q'_{22} & Q''_{22} & Q'''_{22} \end{bmatrix} \cdot \begin{bmatrix} O & & \\ & S & \\ & & I \end{bmatrix} \cdot W^T$$

$$\begin{aligned} C &= diag\{c_1, \ldots, c_n\} \\ S &= diag\{s_1, \ldots, s_n\} \\ C^2 + S^2 &= I_{n \times n}. \end{aligned}$$

Again it can be shown that Q''_{12} delivers the useful combinations for computing the required intersections (see [Moonen & Vandewalle 1990] for details). The system matrices are finally computed from

$$\begin{bmatrix} (Q''_{12})^T \cdot Q^{-T}\{m+l+1 : (i+1)(m+l), 1 : 2mi+n\} \cdot \Sigma_{11} \\ Q^{-T}\{mi+li+m+1 : (m+l)(i+1), 1 : 2mi+n\} \cdot \Sigma_{11} \end{bmatrix}$$
$$= \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} (Q''_{12})^T \cdot Q^{-T}\{1 : mi+li, 1 : 2mi+n\} \cdot \Sigma_{11} \\ Q^{-T}\{mi+li+1 : mi+li+m, 1 : 2mi+n\} \cdot \Sigma_{11} \end{bmatrix}.$$

Similar to what we had in the SVD case, only the Q and Σ matrices are needed from the first (largest) QSVD. This not only cuts down the computational effort in the first stage, but furthermore it allows for constructing

efficient adaptive algorithms, where use is made of QSVD updating techniques. We refer to [Moonen & Vandewalle 1990] for details.

Finally, it can be shown that the above double QSVD scheme delivers consistent results for infinite data sequences, if we let $j \to \infty$. For details we refer to [Moonen & Vandewalle 1990], were similar results for the discrete-time case are derived, and [De Moor 1990].

5 Conclusions

It has been demonstrated how state space identification techniques for multivariable discrete-time systems are translated into identification techniques for continuous-time systems. The key tool is the singular value decomposition (SVD), a numerical technique known to be very robust and accurate when dealing with noisy data. The noise coloring is compensated for by using a generalization of the SVD, namely the quotient SVD. The resulting identification scheme is shown to give consistent results under certain conditions.

References

[1] B. De Moor and G.H. Golub 1989. Generalized Singular Value Decompositions: A proposal for a standardized nomenclature. *Numerical Analysis Project.* Manuscript NA-89-04. Dept. of Comp. Sc., Stanford Univ.

[2] B. De Moor 1990. The SVD and long and short spaces of noisy matrices. *ESAT-SISTA report 1990-38.* Dept. of E.E., Katholieke Universiteit Leuven, Belgium.

[3] G.H. Golub and C.F. Van Loan 1988. *Matrix computations.* Second Edition. North Oxford academic Publishing Co., Johns Hopkins Press.

[4] S.J. Hammarling 1982. Numerical solution of the stable non-negative definite Lyapunov equation. *IMA Journal of Numerical Analysis* **2**, pp 303-323.

[5] B.L. Ho and R.E. Kalman 1966. Effective construction of linear state variable models from input output functions. *Regelungstechnik* **14**, pp 545-548.

[6] *S.Y. Kung 1978. A new identification and model reduction algorithm via singular value decomposition. Proc. 12th Asilomar Conf. on Circuits, Systems and Computers, Pacific Grove, pp 705-714.*

[7] *M. Moonen, B. De Moor, L. Vandenberghe, J. Vandewalle 1989a. On- and off-line identification of linear state space models. Int. Journal of Control* **49**, *No. 1, pp 219-232.*

[8] *M. Moonen, J. Vandewalle 1990b. A QSVD approach to on- and off-line state space identification. Int. Journal of Control* **51**, *No. 5, pp 1133-1146.*

[9] *C.C. Paige and M. Saunders 1981. Towards a generalized singular value decomposition. SIAM J. Numer. Anal.,* **18**, *No. 3, pp 398-405.*

[10] *C.F. Van Loan 1976. Generalizing the singular value decomposition. SIAM J. Numer. Anal.* **13**, *No. 1, pp 76-83.*

[11] *H.P. Zeiger and A.J. Mc Ewen 1974. Approximate linear realizations of given dimensions via Ho's algorithm. IEEE Trans.* **AC-19**, *p 153.*