Fast Multi-Order Computation of System Matrices in Subspace-Based System Identification

Michael Döhler, Laurent Mevel

To cite this version:

Michael Döhler, Laurent Mevel. Fast Multi-Order Computation of System Matrices in Subspace-Based System Identification. Control Engineering Practice, Elsevier, 2012, 20 (9), pp.882-894. 10.1016/j.conengprac.2012.05.005 . hal-00724068

HAL Id: hal-00724068

https://hal.inria.fr/hal-00724068

Submitted on 20 Apr 2020

Fast Multi-Order Computation of System Matrices in Subspace-Based System Identification☆

Michael Döhler, Laurent Mevel

Inria, Centre Rennes - Bretagne Atlantique, 35042 Rennes, France

Abstract

Subspace methods have proven to be efficient for the identification of linear time-invariant systems, especially applied to mechanical, civil or aeronautical structures in operation conditions. Therein, system identification results are needed at multiple (over-specified) model orders in order to distinguish the true structural modes from spurious modes using the so-called stabilization diagrams. In this paper, new efficient algorithms are derived for this multi-order system identification with subspace-based identification algorithms and the closely related Eigensystem Realization Algorithm. It is shown that the new algorithms are significantly faster than the conventional algorithms in use. They are demonstrated on the system identification of a large-scale civil structure.

Keywords: System identification, Subspace methods, State-space models, Linear systems, Multi-input/multi-output systems, Stochastic systems, Least-squares problems, System order, (Operational) modal analysis, Vibration measurement

1. Introduction

Subspace-based system identification methods have proven to be efficient for the identification of linear time-invariant systems (LTI), fitting a linear model to input/output or output only measurements taken from a system. An overview of subspace methods can be found in Benveniste and Fuchs (1985); Viberg (1995); Van Overschee and De Moor (1996); Benveniste and Mevel (2007); Akçay (2010). A broad range of applications exists in the identification of processes in automatic control, see e.g. Bastogne et al. (1998); Juricek et al. (2001); Sotomayor et al. (2003); Pan and Lee (2008). During the last decade, subspace methods found a special interest in mechanical, civil and aeronautical engineering for *modal analysis*, namely the identification of *vibration modes* (eigenvalues) and *mode shapes* (corresponding eigenvectors) of structures. Therefore, identifying an LTI system from measurements is a basic service in vibration monitoring (see e.g. Peeters and De Roeck, 1999; Hermans and Van der Auweraer, 1999; Mevel et al., 2003, 2006; Brownjohn et al., 2010). Having done this allows in particular Finite Element Model updating (Ventura et al., 2005) and Structural Health Monitoring (Carden and Fanning, 2004).

In an Operational Modal Analysis (OMA) context (Peeters and De Roeck, 1999, 2001), however, the following unusual characteristics must be taken into account:

- The number of sensors can be very large (up to hundreds, or thousands in the future); sensors can even be moved from one measurement campaign to another;
- The number of modes of interest can be quite large (up to 100 or beyond), thus calling for non-standard approaches to model reduction;
- The excitation applied to the structure is usually uncontrolled and natural, thus turbulent and non-stationary.

Because of these characteristics, usual tools from linear system identification, such as the System Identification Toolbox in Matlab, cannot be used as such. In order to retrieve the desired large number of modes, an even larger model order must be assumed while performing identification. This causes a number of spurious modes to appear in the identified models. Techniques from statistics to estimate the best model order, such as AIC, BIC or MDL (Akaike, 1974; Rissanen, 1978; Camba-Mendez and Kapetanios, 2001), or model order estimation techniques specifically for subspace methods as in Bauer (2001) lead to a model with the best prediction capacity. However, one is rather interested in a model containing only the physical modes of the investigated structure, while rejecting the spurious modes. Based on the observation that physical modes remain quite constant when estimated at different over-specified model orders, while spurious modes vary, they can be distinguished using so-called *stabilization diagrams* (Peeters and De Roeck, 1999, 2001). There, the physical modes are selected from system identification results at multiple model orders in a GUI-assisted way. Methods for an automation of this selection are e.g. found in Van der Auweraer and Peeters (2004); Scionti and Lanslots (2005); Reynders et al. (2011). As system identification is done at an over-specified model order and repeated while truncating at multiple model orders, the computational burden for this procedure is significant especially for large model orders.

A fast identification of the system parameters is of basic

☆A preliminary version of this paper was presented at the 18th IFAC World Congress, August 28 - September 2, 2011, Milano, Italy.

Email addresses: michael.dohler@inria.fr (Michael Döhler), laurent.mevel@inria.fr (Laurent Mevel)

interest, e.g. for online structural health monitoring. Existing literature on fast subspace-based system identification covers mainly four subjects:

- Convergence rates of the system or transfer matrices for a growing sample size. They are e.g. analyzed in Deistler et al. (1995); Bauer et al. (1999); Bauer and Ljung (2002); Chiuso and Picci (2004); Bauer (2005) and typically concern the theoretical properties of a subspace method.
- Reduction of the processed data by using only data of a subset of the recorded sensors, so-called *reference sensors* or *projection channels*, instead all the sensors at one point of the subspace algorithms. See e.g. Peeters and De Roeck (1999); Reynders and De Roeck (2008).
- Fast processing of the measurement data prior to estimating the observability matrix. This is considered in Cho and Kailath (1995); Mastronardi et al. (2001a) for subspace methods using a LQ decomposition of a data Hankel matrix during preprocessing, and in (Peeters, 2000, Sec. 3.2.2) for covariance-driven subspace methods.
- Recursive subspace-based identification algorithms as in Lovera et al. (2000); Oku and Kimura (2002); Mercère et al. (2008), which efficiently update an observability matrix estimate using new data.

For the complete subspace identification procedure in the engineering practice, a matrix is built from the data, from where the observability matrix is obtained by a factorization. Then, the system matrices are identified at multiple model orders from the observability matrix. This paper considers the last step, which is very taxing in practice for large model orders. A fast computation scheme is derived, where the structure of the observability matrix estimates at multiple orders is exploited when solving the least squares problem to obtain the system matrices.[1] The computational efficiency of the resulting algorithms are compared to the entire subspace algorithm. In (Döhler and Mevel, 2011a), a prior derivation was made of some of the presented algorithms, which are refined in this paper.[2] Furthermore, a fast computation of the system matrices at multiple orders is derived for the Eigensystem Realization Algorithm (ERA; Juang and Pappa, 1985; James III et al., 1995), which is closely related to covariance-driven subspace-based algorithms.

This paper is organized as follows. In Section 2, the general Stochastic Subspace Identification (SSI) algorithm is introduced and the multi-order identification problem is addressed. In Sections 3 and 4, efficient algorithms to estimate the system matrices at multiple model orders with SSI and ERA are derived, which reduce the computational burden significantly. In Section 5, the computational cost of the algorithms is compared for doing the system identification on a real test case, validating their efficiency.

2. Stochastic Subspace Identification (SSI)

2.1. The General Stochastic Subspace Identification Algorithm

Consider linear multivariable time invariant systems described by a discrete time state space model

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + v_{k+1} \\ y_k = Cx_k + Du_k + w_k \end{cases} \tag{1}$$

with the state $x \in \mathbb{R}^n$, the observed input $u \in \mathbb{R}^m$, the output $y \in \mathbb{R}^r$ and the unobserved input and output disturbances v and w. The matrices $A \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{r\times n}$ are the state transition and observation matrices, respectively. A subset of the r sensors can be used for reducing the size of the matrices in the identification process, see e.g. (Peeters and De Roeck, 1999; Reynders and De Roeck, 2008). These sensors are called projection channels or reference sensors. Let r_0 be the number of reference sensors ($r_0 \leq r$).

In this paper, the identification of the system matrices A and C is of interest. In Operational Modal Analysis, typically no observed inputs are available ($B = 0$, $D = 0$) and identification is done using the output-only data (y_k) (Peeters and De Roeck, 1999, 2001). When some inputs (u_k) are observed, combined deterministic-stochastic subspace identification algorithms can be used (Mevel et al., 2006; Reynders and De Roeck, 2008). There are many Stochastic Subspace Identification algorithms in the literature, which differ in the construction of a matrix $\mathcal{H}_{p+1,q}$ from the data, from which the observability matrix is obtained. See e.g. Benveniste and Fuchs (1985); Van Overschee and De Moor (1996); Benveniste and Mevel (2007) and the related references for an overview. They all fit in the following general framework for the identification of the system matrices A and C of system (1).

Let the parameters p and q be given such that $pr \geq qr_0 \geq n$. From the output or input/output data a matrix $\mathcal{H}_{p+1,q}$ is built according to the chosen subspace algorithm, which will be called *subspace matrix* in the following. The subspace algorithm is chosen such that the corresponding subspace matrix enjoys (asymptotically for a large number of samples) the factorization property (Benveniste and Mevel, 2007)

$$\mathcal{H}_{p+1,q} = W\mathcal{O}_{p+1}\mathcal{Z}_q \tag{2}$$

into the matrix of observability

$$\mathcal{O}_{p+1} \overset{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^p \end{bmatrix},$$

a matrix $\mathcal{Z}_q$ and an invertible weighting matrix W depending on the selected subspace algorithm. For simplicity, skip the subscripts of $\mathcal{H}_{p+1,q}$, $\mathcal{O}_{p+1}$ and $\mathcal{Z}_q$ in the following.

The observability matrix $\mathcal{O}$ is obtained from the SVD of the matrix $\mathcal{H}$ and its truncation at the desired model order n (Kung, 1978):

$$\mathcal{H} = \begin{bmatrix} U_1 & U_0 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_0 \end{bmatrix} V^T, \tag{3}$$

$$\mathcal{O} = W^{-1} U_1 \Sigma_1^{1/2}. \tag{4}$$

[1] Note that this is different from the structured total least squares problem considered in (Mastronardi et al., 2001b; Markovsky et al., 2005), which applies e.g. to maximum likelihood system identification.

[2] Algorithm 2 is revised in this paper as well as Algorithm 3, which is based on Algorithm 2.

Note that the singular values in Σ_1 must be non-zero and hence $\mathcal{O}$ is of full column rank. The observation matrix C is then found in the first block-row of the observability matrix $\mathcal{O}$. The state transition matrix A is obtained from the shift invariance property of $\mathcal{O}$, namely as the least squares solution of

$$\mathcal{O}^{\uparrow}A = \mathcal{O}^{\downarrow}, \text{ where } \mathcal{O}^{\uparrow} \overset{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix}, \mathcal{O}^{\downarrow} \overset{\text{def}}{=} \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{p} \end{bmatrix}. \tag{5}$$

Example 1. *Let $N + p + q$ be the number of available samples and $y_k^{(\text{ref})} \in \mathbb{R}^{r_0}$ the vector containing the reference sensor data, which is a subset of y_k for all samples. Then, define the data matrices*

$$\mathcal{Y}^+ = \frac{1}{\sqrt{N}} \begin{bmatrix} y_{q+1} & y_{q+2} & \dots & y_{N+q} \\ y_{q+2} & y_{q+3} & \dots & y_{N+q+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{q+p+1} & y_{q+p+2} & \dots & y_{N+p+q} \end{bmatrix},$$

$$\mathcal{Y}^- = \frac{1}{\sqrt{N}} \begin{bmatrix} y_q^{(\text{ref})} & y_{q+1}^{(\text{ref})} & \dots & y_{N+q-1}^{(\text{ref})} \\ y_{q-1}^{(\text{ref})} & y_q^{(\text{ref})} & \dots & y_{N+q-2}^{(\text{ref})} \\ \vdots & \vdots & \ddots & \vdots \\ y_1^{(\text{ref})} & y_2^{(\text{ref})} & \dots & y_N^{(\text{ref})} \end{bmatrix}.$$

For covariance-driven SSI (Benveniste and Fuchs, 1985; Peeters and De Roeck, 1999), the block Hankel matrix

$$\mathcal{H}^{\text{cov}} \overset{\text{def}}{=} \begin{bmatrix} R_1 & R_2 & \dots & R_q \\ R_2 & R_3 & \dots & R_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p+1} & R_{p+2} & \dots & R_{p+q} \end{bmatrix} \in \mathbb{R}^{(p+1)r \times qr_0}, \tag{6}$$

is built from the empirical covariances $R_i \overset{\text{def}}{=} \frac{1}{N}\sum_{k=1+i}^{N+i} y_k y_{k-i}^{(\text{ref})T}$, which is asymptotically equivalent to $\mathcal{Y}^+\mathcal{Y}^{-T}$. It enjoys the factorization property (2) *where $\mathcal{Z}$ is the controllability matrix.*

For data-driven SSI with the Unweighted Principal Component (UPC) algorithm (Van Overschee and De Moor, 1996; Peeters and De Roeck, 1999), the matrix

$$\widetilde{\mathcal{H}}^{\text{dat}} = \mathcal{Y}^+\mathcal{Y}^{-T}(\mathcal{Y}^-\mathcal{Y}^{-T})^{-1}\mathcal{Y}^- \in \mathbb{R}^{(p+1)r \times N} \tag{7}$$

enjoys the factorization property (2) *where $\mathcal{Z}$ is the Kalman filter state matrix. In practice, the respective subspace matrix $\mathcal{H}^{\text{dat}}$ is obtained from the LQ decomposition*

$$\mathcal{Y} \overset{\text{def}}{=} \begin{bmatrix} \mathcal{Y}^- \\ \mathcal{Y}^+ \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{8}$$

from $\mathcal{H}^{\text{dat}} \overset{\text{def}}{=} L_{21} \in \mathbb{R}^{(p+1)r \times qr_0}$, such that $\widetilde{\mathcal{H}}^{\text{dat}} = \mathcal{H}^{\text{dat}}Q_1$ with an orthogonal matrix Q_1. In case of large data sets, the LQ factorization (8) *can be done iteratively. Separate $\mathcal{Y}$ into n_b blocks $\mathcal{Y} = [\mathcal{Y}_1 \;\; \mathcal{Y}_2 \;\; \dots \;\; \mathcal{Y}_{n_b}]$ and do the LQ factorizations*

$$\mathcal{Y}_1 = L^{(1)}Q^{(1)}, \quad [L^{(j-1)} \;\; \mathcal{Y}_j] = L^{(j)}Q^{(j)}$$

for $j = 2, \dots, n_b$. Then it follows easily that $L^{(n_b)}$ is the L factor of $\mathcal{Y}$. Note that matrix $\mathcal{Y}$ does not need to be in memory and matrices $\mathcal{Y}_j$ can be computed from the data only when actually needed in the iteration. Alternatively, the L factor can be obtained efficiently by exploiting the displacement structure of $\mathcal{Y}$ as in (Mastronardi et al., 2001a).

There are many other subspace algorithms fulfilling factorization property (2). *Amongst them are algorithms that e.g. use input/output data, parsimonious models or additionally apply to closed-loop identification (Qin and Ljung, 2003; Benveniste and Mevel, 2007; Chiuso, 2007; Reynders and De Roeck, 2008; van der Veen et al., 2010).* □

2.2. Multi-Order SSI

The true system order n is unknown in many practical applications and it is common to do the system identification for models (1) at different model orders $n = n_j$, $j = 1, \dots, t$, with

$$1 \leq n_1 < n_2 < \dots < n_t \leq \min\{pr, qr_0\}, \tag{9}$$

and where t is the number of models to be estimated (Peeters and De Roeck, 1999, 2001; Van der Auweraer and Peeters, 2004). The choice of the model orders n_j, $j = 1, \dots, t$, is up to the user and also depends on the problem. For example, $n_j = j + d$ or $n_j = 2j + d$ with some constant d can be chosen. For example, the latter makes sense for an application as in Section 5: There, the eigenvalues of the state transition matrix are pairwise complex conjugate. Thus, two model orders are needed to recover one new mode.

The following notation for specifying these different model orders is used throughout this paper. Let $\mathcal{O}_j \in \mathbb{R}^{(p+1)r \times n_j}$, $A_j \in \mathbb{R}^{n_j \times n_j}$ and $C_j \in \mathbb{R}^{r \times n_j}$ be the observability, state transition and observation matrix at model order n_j, $j \in \{1, \dots, t\}$, respectively. Let furthermore be $\mathcal{O}_j^{\uparrow}$ and $\mathcal{O}_j^{\downarrow}$ the first respective last p block rows of $\mathcal{O}_j$, analogously to the definition in (5).

Note that in Section 2.1 model order n was used, while from now on model orders n_j will be used. The matrices A_j, C_j, $\mathcal{O}_j$, $\mathcal{O}_j^{\uparrow}$ and $\mathcal{O}_j^{\downarrow}$ fulfill the equations in Section 2.1, replacing A, C, $\mathcal{O}$, $\mathcal{O}^{\uparrow}$ and $\mathcal{O}^{\downarrow}$, as well as n_j replaces n.

2.3. Computation of the System Matrices

The system matrix A_j is the solution of the least squares problem (5) at a chosen model order n_j:

$$\mathcal{O}_j^{\uparrow}A_j = \mathcal{O}_j^{\downarrow}. \tag{10}$$

A numerically stable solution is

$$A_j = \mathcal{O}_j^{\uparrow\dagger}\mathcal{O}_j^{\downarrow}, \tag{11}$$

where † denotes the Moore-Penrose pseudoinverse. A more efficient and also numerically stable way to solve it (Golub and Van Loan, 1996), uses the thin QR decomposition

$$\mathcal{O}_j^{\uparrow} = Q_jR_j, \tag{12}$$

where $Q_j \in \mathbb{R}^{pr \times n_j}$ is a matrix with orthogonal columns and $R_j \in \mathbb{R}^{n_j \times n_j}$ is upper triangular. R_j is assumed to be of full rank, which is reasonable as O_j is of full column rank. With

$$S_j \overset{\text{def}}{=} Q_j^T O_j^{\downarrow}, \tag{13}$$

$S_j \in \mathbb{R}^{n_j \times n_j}$, the solution of the least squares problem is

$$A_j = R_j^{-1} S_j. \tag{14}$$

The observation matrix C_j is found in the first block row of O_j.

The conventional way to compute the system matrices A_j and C_j at the model orders n_j, $j = 1, \dots, t$, is the following (see e.g. Peeters and De Roeck, 2001): First, the observability matrix O_t is computed at the maximal desired model order n_t from (3)-(4). Then, the observability matrix O_j at order n_j consists of the first n_j columns of O_t. The matrices A_j and C_j are the solution of least squares problem (10) and the first block row of O_j, respectively. This approach is summarized in Algorithm 1, where the least squares solution is obtained either by using the pseudoinverse (11) or the QR decomposition with equations (12)-(14). Note that for a matrix X the matrix $X_{[a_1:a_2, b_1:b_2]}$ denotes the submatrix of matrix X containing the block from rows a_1 to a_2 and columns b_1 to b_2 of matrix X.

Algorithm 1 Multi-Order SSI

```
Input: $O_t \in \mathbb{R}^{(p+1)r \times n_t}$          {observability matrix}
       $n_1, \dots, n_t$          {desired model orders satisfying (9)}
for $j = 1$ to $t$ do
    $O_j^{\uparrow} \longleftarrow O_{t[1:pr,1:n_j]}$, $O_j^{\downarrow} \longleftarrow O_{t[(pr+1):(p+1)r,1:n_j]}$
    if method = pseudoinverse then
        $A_j \longleftarrow O_j^{\uparrow\dagger} O_j^{\downarrow}$
    else if method = QR then
        QR decomposition $O_j^{\uparrow} = Q_j R_j$
        $S_j \longleftarrow Q_j^T O_j^{\downarrow}$
        $A_j \longleftarrow R_j^{-1} S_j$
    end if
    $C_j \longleftarrow O_{t[1:r,1:n_j]}$
end for
Output: System matrices $A_j$, $C_j$ at model orders $n_1, \dots, n_t$
```

2.4. Computational Complexities

In order to compare the performance of different algorithms for the multi-order computation of the system matrices A_j and C_j, $j = 1, \dots, t$, their number of floating point operations (flops, multiplications plus summations) needs to be evaluated. Consider the observability matrix O_t at a maximal model order $n_{\max} \overset{\text{def}}{=} n_t$ be given. As C_j is always a submatrix of O_t, only the computation of the state transition matrices A_j is considered.

The parameters for counting the flops are set up and simplified as follows. While all algorithms in this paper are derived for arbitrary model orders n_j fulfilling (9), the number of flops is determined for the computation of A_j at orders $n_j = j = 1, 2, \dots, n_*$ for each algorithm, where $n_* \leq n_{\max}$. Like this, the complexity of an algorithm is indicated, where the choice of the parameters is influenced by the maximal model order $n_{\max}$ and where the computation is stopped at a (possibly smaller) model order n_*. However, $n_* = n_{\max}$ can be assumed for a first evaluation. Insignificant terms are neglected when counting the flops.

The subspace matrix $\mathcal{H}$ is often of size $(p+1)r \times qr_0$ and in practice it is set $p + 1 = q$ and $n_{\max} = qr_0$ (Basseville et al., 2001). Define the parameter $c \overset{\text{def}}{=} pr/n_{\max} \approx r/r_0$, which is independent of p, q and $n_{\max}$ and defines the ratio of the dimensions of $O_t^{\uparrow}$ and $O_t^{\downarrow}$. Hence, the total flop count is a function of c, n_* and $n_{\max}$. Furthermore, the approximations $\sum_{j=1}^{n_*} j \approx \frac{1}{2} n_*^2$, $\sum_{j=1}^{n_*} j^2 \approx \frac{1}{3} n_*^3$, $\sum_{j=1}^{n_*} j^3 \approx \frac{1}{4} n_*^4$ are used.

Table 1: Flop counts of some basic operations (Golub and Van Loan, 1996).

Operation	Matrix Sizes	Flops
$F = U\Sigma V^T$	$F, U \in \mathbb{R}^{a \times b}, \Sigma, V \in \mathbb{R}^{b \times b}$	$14ab^2 + 8b^3$
$F = QR$	$F, Q \in \mathbb{R}^{a \times b}, R \in \mathbb{R}^{b \times b}$	$4ab^2 - \frac{4}{3}b^3$
FG	$F \in \mathbb{R}^{a \times b}, G \in \mathbb{R}^{b \times c}$	$2abc$
$R^{-1}F$	$R \in \mathbb{R}^{a \times a}$ triangular, $F \in \mathbb{R}^{a \times b}$	$a^2 b$

In Table 1, the number of flops of some basic numerical operations is given. In Table 2, the flops of Algorithm 1 are counted. It can be seen that the least squares solution for the system matrix using the QR decomposition is favorable compared to using the SVD.

3. Fast Algorithms for Multi-Order SSI

The computation of the system matrices at multiple orders is a big computational burden. In the previous section it was shown that the conventional algorithm for this task (Algorithm 1) has a computational complexity of $O(n_{\max}^4)$ for the identification of the system matrices at model orders $1, 2, \dots, n_{\max}$. In this section, efficient algorithms are derived for the multi-order identification of the system matrices, having a computational complexity of only $O(n_{\max}^3)$.

3.1. A First Algorithm for Fast Multi-Order Computation of the System Matrices

Conventionally, for the computation of the system matrices A_j and C_j at the desired model orders $n_1, \dots, n_t$, the least squares problem for the state transition matrix A_j is solved at each model order.

Now, an algorithm is presented that solves the least squares problem only once at the maximal desired model order n_t (Equations (12) to (14) with $j = t$), leading to matrices R_t, S_t and A_t. Then, instead of solving the least squares problems at all the orders $n_1, \dots, n_{t-1}$, it is shown that the state transition matrices A_j at these lower orders can be computed much more efficiently from submatrices of R_t and S_t, based on the following main theorem of this paper.

Table 2: Flop count of Algorithm 1 for $n_j = 1, 2, \dots, n_*$ and $n_t = n_{\max}$.

Line	Operation	Flops
4	$\mathcal{O}_j^{\uparrow} = U\Sigma V^T$	$14(cn_{\max})j^2 + 8j^3$
4	$A_j \leftarrow V\Sigma^{-1}(U^T\mathcal{O}_j^{\downarrow})$	$2(cn_{\max})j^2 + 2j^3$
		$\sum = \frac{16}{3}cn_{\max}n_*^3 + \frac{5}{2}n_*^4$
6	$\mathcal{O}_j^{\uparrow} = Q_jR_j$	$4(cn_{\max})j^2 - \frac{4}{3}j^3$
7	$S_j \leftarrow Q_j^T\mathcal{O}_j^{\downarrow}$	$2(cn_{\max})j^2$
8	$A_j \leftarrow R_j^{-1}S_j$	j^3
		$\sum = 2cn_{\max}n_*^3 - \frac{1}{12}n_*^4$

Theorem 2. *Let $\mathcal{O}_t$, Q_t, R_t and S_t be given at the maximal desired model order n_t with*

$$\mathcal{O}_t^{\uparrow} = Q_tR_t, \quad S_t = Q_t^T\mathcal{O}_t^{\downarrow}, \quad A_t = R_t^{-1}S_t, \tag{15}$$

such that A_t is the least squares solution of

$$\mathcal{O}_t^{\uparrow}A_t = \mathcal{O}_t^{\downarrow}.$$

Let $j \in \{1, \dots, t-1\}$, and let R_t and S_t be partitioned into blocks

$$R_t = \begin{bmatrix} R_j^{(11)} & R_j^{(12)} \\ 0 & R_j^{(22)} \end{bmatrix}, \quad S_t = \begin{bmatrix} S_j^{(11)} & S_j^{(12)} \\ S_j^{(21)} & S_j^{(22)} \end{bmatrix}, \tag{16}$$

where $R_j^{(11)}, S_j^{(11)} \in \mathbb{R}^{n_j \times n_j}$. Then, the state transition matrix A_j at model order n_j, which is the least squares solution of

$$\mathcal{O}_j^{\uparrow}A_j = \mathcal{O}_j^{\downarrow}, \tag{17}$$

satisfies

$$A_j = R_j^{(11)-1}S_j^{(11)}. \tag{18}$$

Proof. From (3) and (4) it follows that $\mathcal{O}_j$ consists of the first n_j columns of $\mathcal{O}_t$. This holds analogously for $\mathcal{O}_j^{\uparrow}$ and $\mathcal{O}_j^{\downarrow}$. Hence, $\mathcal{O}_t^{\uparrow}$ and $\mathcal{O}_t^{\downarrow}$ can be partitioned into

$$\mathcal{O}_t^{\uparrow} = \begin{bmatrix} \mathcal{O}_j^{\uparrow} & \tilde{\mathcal{O}}_j^{\uparrow} \end{bmatrix}, \quad \mathcal{O}_t^{\downarrow} = \begin{bmatrix} \mathcal{O}_j^{\downarrow} & \tilde{\mathcal{O}}_j^{\downarrow} \end{bmatrix}, \tag{19}$$

where $\tilde{\mathcal{O}}_j^{\uparrow}$ and $\tilde{\mathcal{O}}_j^{\downarrow}$ consist of the remaining columns of $\mathcal{O}_t^{\uparrow}$ and $\mathcal{O}_t^{\downarrow}$. Let Q_t be partitioned into the blocks

$$Q_t = \begin{bmatrix} Q_j^{(1)} & Q_j^{(2)} \end{bmatrix}, \tag{20}$$

where $Q_j^{(1)} \in \mathbb{R}^{pr \times n_j}$. From (15) and (20) follows

$$\mathcal{O}_t^{\uparrow} = \begin{bmatrix} Q_j^{(1)} & Q_j^{(2)} \end{bmatrix} \begin{bmatrix} R_j^{(11)} & R_j^{(12)} \\ 0 & R_j^{(22)} \end{bmatrix} = \begin{bmatrix} Q_j^{(1)}R_j^{(11)} & B \end{bmatrix} \tag{21}$$

with $B = Q_j^{(1)}R_j^{(12)} + Q_j^{(2)}R_j^{(22)}$. Comparing (19) and (21), it follows

$$\mathcal{O}_j^{\uparrow} = Q_j^{(1)}R_j^{(11)}, \tag{22}$$

which obviously is a QR decomposition of $\mathcal{O}_j^{\uparrow}$. As A_j is the least squares solution of (17) and because of QR decomposition (22), A_j satisfies

$$A_j = R_j^{(11)-1}Q_j^{(1)^T}\mathcal{O}_j^{\downarrow}. \tag{23}$$

Furthermore, from (15), (19) and (20) follows

$$S_t = \begin{bmatrix} Q_j^{(1)^T} \\ Q_j^{(2)^T} \end{bmatrix} \begin{bmatrix} \mathcal{O}_j^{\downarrow} & \tilde{\mathcal{O}}_j^{\downarrow} \end{bmatrix} = \begin{bmatrix} Q_j^{(1)^T}\mathcal{O}_j^{\downarrow} & Q_j^{(1)^T}\tilde{\mathcal{O}}_j^{\downarrow} \\ Q_j^{(2)^T}\mathcal{O}_j^{\downarrow} & Q_j^{(2)^T}\tilde{\mathcal{O}}_j^{\downarrow} \end{bmatrix},$$

and comparing to (16) yields

$$S_j^{(11)} = Q_j^{(1)^T}\mathcal{O}_j^{\downarrow}.$$

Plugging this into (23) leads to the assertion. □

Hence, steps (12) and (13) for the least squares solution of the state transition matrix A_j are not necessary anymore for $j = 1, \dots, t-1$, and (14) is replaced by (18). The resulting algorithm for this fast multi-order computation of the system matrices and its flop count are summarized in Algorithm 2 and in Table 3.

Algorithm 2 Fast Multi-Order SSI

```
Input: $\mathcal{O}_t \in \mathbb{R}^{(p+1)r \times n_t}$          {observability matrix}
       $n_1, \dots, n_t$          {desired model orders satisfying (9)}
1: $\mathcal{O}_t^{\uparrow} \longleftarrow \mathcal{O}_{t[1:pr,1:n_t]}$, $\mathcal{O}_t^{\downarrow} \longleftarrow \mathcal{O}_{t[(pr+1):(p+1)r,1:n_t]}$
2: $C_t \longleftarrow \mathcal{O}_{t[1:r,1:n_t]}$
3: QR decomposition $\mathcal{O}_t^{\uparrow} = Q_tR_t$
4: $S_t \longleftarrow Q_t^T\mathcal{O}_t^{\downarrow}$
5: for $j = 1$ to $t$ do
6:     $A_j \longleftarrow R_{t[1:n_j,1:n_j]}^{-1}S_{t[1:n_j,1:n_j]}$
7:     $C_j \longleftarrow C_{t[1:r,1:n_j]}$
8: end for
Output: System matrices $A_j$, $C_j$ at model orders $n_1, \dots, n_t$
```

Table 3: Flop Count of Algorithm 2 for $n_j = 1, 2, \dots, n_*$ and $n_t = n_{\max}$.

Line	Operation	Flops
3	$\mathcal{O}_t^{\uparrow} = Q_tR_t$	$4(cn_{\max})n_{\max}^2 - \frac{4}{3}n_{\max}^3$
4	$S_t \leftarrow Q_t^T\mathcal{O}_t^{\downarrow}$	$2(cn_{\max})n_{\max}^2$
6	$A_j \leftarrow R_{t[1:n_j,1:n_j]}^{-1}S_{t[1:n_j,1:n_j]}$	j^3
		$\sum = (6c - \frac{4}{3})n_{\max}^3 + \frac{1}{4}n_*^4$

Remark 3. *In (Döhler and Mevel, 2011a), the preliminary version of this paper, R_t^{-1} is computed explicitly. As $R_{t[1:n_j,1:n_j]}^{-1}$ is also a submatrix of R_t^{-1} due to the triangular structure, A_j could be computed as a direct matrix product with a triangular matrix, which has the same computational cost as the backward substitution for the computation in Line 6 of Algorithm 2 in this paper. Thus, the procedure of (Döhler and Mevel, 2011a) is slightly less efficient as it requires $\frac{1}{3}n_{\max}^3$ flops for the inversion of R_t in addition.*

3.2. Fast Iterative Multi-Order Computation of the System Matrices

The fast multi-order computation of the state transition matrix from the previous section can be further improved by expressing A_{j+1} with the help of A_j, which reduces further the number of numerical operations.

Corollary 4. *Let $\mathcal{O}_t$, Q_t, R_t and S_t be given at the maximal desired model order n_t. Define the submatrices*

$$\begin{aligned}
\tilde{R}_j^{(11)} &= R_{t[1:n_j,1:n_j]}, & \tilde{S}_j^{(11)} &= S_{t[1:n_j,1:n_j]},\\
\tilde{R}_j^{(12)} &= R_{t[1:n_j,(n_j+1):n_{j+1}]}, & \tilde{S}_j^{(12)} &= S_{t[1:n_j,(n_j+1):n_{j+1}]},\\
\tilde{R}_j^{(22)} &= R_{t[(n_j+1):n_{j+1},(n_j+1):n_{j+1}]}, & \tilde{S}_j^{(21)} &= S_{t[(n_j+1):n_{j+1},1:n_j]},
\end{aligned}$$
$$\tilde{S}_j^{(22)} = S_{t[(n_j+1):n_{j+1},(n_j+1):n_{j+1}]}, \tag{24}$$

and $A_1 = \tilde{R}_1^{(11)-1}\tilde{S}_1^{(11)}$. Then it holds

$$A_{j+1} = \begin{bmatrix} A_j & \tilde{R}_j^{(11)-1}\tilde{S}_j^{(12)} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -\tilde{R}_j^{(11)-1}\tilde{R}_j^{(12)}\tilde{R}_j^{(22)-1} \\ \tilde{R}_j^{(22)-1} \end{bmatrix} \begin{bmatrix} \tilde{S}_j^{(21)} & \tilde{S}_j^{(22)} \end{bmatrix} \tag{25}$$

for $j = 1, \ldots, t-1$.

Proof. According to Theorem 2, above submatrices are defined such that $R_j = \tilde{R}_j^{(11)}$, $S_j = \tilde{S}_j^{(11)}$ and

$$R_{j+1} = \begin{bmatrix} \tilde{R}_j^{(11)} & \tilde{R}_j^{(12)} \\ 0 & \tilde{R}_j^{(22)} \end{bmatrix}, \quad S_{j+1} = \begin{bmatrix} \tilde{S}_j^{(11)} & \tilde{S}_j^{(12)} \\ \tilde{S}_j^{(21)} & \tilde{S}_j^{(22)} \end{bmatrix}.$$

Then,

$$R_{j+1}^{-1} = \begin{bmatrix} \tilde{R}_j^{(11)-1} & -\tilde{R}_j^{(11)-1}\tilde{R}_j^{(12)}\tilde{R}_j^{(22)-1} \\ 0 & \tilde{R}_j^{(22)-1} \end{bmatrix}.$$

Plugging this into $A_{j+1} = R_{j+1}^{-1}S_{j+1}$, the assertion follows using Theorem 2 and replacing $A_j = \tilde{R}_j^{(11)-1}\tilde{S}_j^{(11)}$. □

The complete algorithm for this fast iterative multi-order computation of the state transition matrix is obtained from Algorithm 2 by replacing Line 6 at $j+1$ with Equation (25). Denote the resulting algorithm by **Algorithm 3**. Its flops are counted in Table 4.

Table 4: Flop count of Algorithm 3 for $n_j = 1, 2, \ldots, n_*$ and $n_t = n_{\max}$.

Step	Operation	Flops
Line 3 (Alg. 2)	$\mathcal{O}_t^{\uparrow} = Q_t R_t$	$(4c - \frac{4}{3})n_{\max}^3$
Line 4 (Alg. 2)	$S_t \leftarrow Q_t^T \mathcal{O}_t^{\downarrow}$	$2cn_{\max}^3$
Equ. (25)	$\tilde{R}_j^{(11)-1}\tilde{S}_j^{(12)}$, $\tilde{R}_j^{(11)-1}\tilde{R}_j^{(12)}$	$2j^2$
Equ. (25)	dyadic product, sum	$2j^2$
		$\sum = (6c - \frac{4}{3})n_{\max}^3 + \frac{4}{3}n_*^3$

3.3. Fast Iterative Computation of the System Matrices without Preprocessing at the Maximal Model Order

The maximal desired model order must be set beforehand in the algorithms for the fast algorithms in Sections 3.1 and 3.2: The QR decomposition $\mathcal{O}_t^{\uparrow} = Q_t R_t$ and $S_t = Q_t^T \mathcal{O}_t^{\downarrow}$ are computed at the maximal desired model order n_t, from which the system matrices A_j at the model orders n_j, $j = 1, \ldots, t$, are derived.

Now, an algorithm is derived that avoids a prior QR decomposition at a maximal model order n_t. Instead, the Householder reflections to obtain the Q and R factor of the QR decomposition (Golub and Van Loan, 1996) are applied only on actually required parts of the observability matrix for each j. Then, the computation of the system matrix A_j depends only on $\mathcal{O}_j$ at model order n_j in each iteration j, but not on matrices at higher model orders. This could give the advantage that the computation of A_j, $j = 1, 2, \ldots$, can be stopped at some model order depending on criteria using results that are already achieved, while avoiding additional computation needed for preprocessing at a preselected maximal model order n_t. However, note that a maximal possible model order is always given by the rank of the subspace matrix $\mathcal{H}$, thus $n_j \leq \min\{pr, qr_0\}$ for all j, cf. (9).

Recall the definition and properties of the well-known Householder reflections from (Golub and Van Loan, 1996):

Definition 5. *For a vector $x \in \mathbb{R}^m$ with the first entry $x_{[1]} \neq 0$, the Householder vector $v \in \mathbb{R}^m$ is defined as*

$$v \overset{\text{def}}{=} x + \operatorname{sign}(x_{[1]})\|x\|_2 e_1,$$

where $\|\cdot\|_2$ denotes the Euclidean norm and $e_1 \in \mathbb{R}^m$ is the unit vector $e_1^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$. The respective Householder reflection $H(v) \in \mathbb{R}^{m\times m}$ is defined as

$$H(x) \overset{\text{def}}{=} I_m - 2vv^T/(v^T v).$$

For $l \geq m$, define

$$H^{(l)}(x) \overset{\text{def}}{=} \begin{bmatrix} I_{l-m} & 0 \\ 0 & H(x) \end{bmatrix}.$$

Lemma 6. *Let $x \in \mathbb{R}^l$, $l \geq m$ and $H^{(l)}(x_{[l-m+1:l]}) \in \mathbb{R}^{l\times l}$ be the Householder reflection of Definition 5, which is an orthogonal matrix. Then,*

$$H^{(l)}(x_{[l-m+1:l]})x = \begin{bmatrix} x_{[1:l-m]} \\ \tilde{x}_{[l-m+1]} \\ 0_{m-1,1} \end{bmatrix},$$

where $\tilde{x}_{[l-m+1]}$ is an entry changed by the Householder reflection and $0_{a,b}$ is a matrix of size $a \times b$ containing zeros. The multiplication of $H^{(l)}(x_{[l-m+1:l]})$ with an arbitrary vector can be done with about $4m$ flops, without computing $H^{(l)}(x_{[l-m+1:l]})$ explicitly.

For $X \in \mathbb{R}^{l\times k}$, $l \geq m \geq k$, define the matrix $H^{(l)}(X_{[l-m+1:l,1:k]}) \in \mathbb{R}^{l\times l}$ as a product of Householder reflections, such that

$$H^{(l)}(X_{[l-m+1:l,1:k]})X = \begin{bmatrix} X_{[1:l-m,1:k]} \\ \tilde{X} \\ 0_{m-k,k} \end{bmatrix},$$

where $\tilde{X} \in \mathbb{R}^{k\times k}$ is an upper triangular matrix resulting from the Householder reflections. The multiplication of $H^{(l)}(X_{[l-m+1:l,1:k]})$ with an arbitrary vector can be done with about $2k(2m-k)$ flops, without computing $H^{(l)}(X_{[l-m+1:l,1:k]})$ explicitly.

With these definitions, the Householder reflections for the least squares solution of the system matrices can be applied stepwise on the appropriate parts of the observability matrix. Thus, the explicit computation of the factor Q of a QR decomposition is not necessary anymore.

Proposition 7. *Let the observability matrices $\mathcal{O}_1, \mathcal{O}_2, \ldots$ at model orders $n_1 < n_2 < \ldots$ be given. Define $H_1 = H^{(pr)}(\mathcal{O}_1^\uparrow)$, $\tilde{R}_1 = H_1\mathcal{O}_1^\uparrow$ and $\tilde{S}_1 = H_1\mathcal{O}_1^\downarrow$. For $j = 1, 2, \ldots$ define o_{j+1} such that*

$$\mathcal{O}_{j+1} = \begin{bmatrix} \mathcal{O}_j & o_{j+1} \end{bmatrix}.$$

and

$$\tilde{o}_{j+1}^\uparrow = H_j \cdots H_1 o_{j+1}^\uparrow, \quad \tilde{o}_{j+1}^\downarrow = H_j \cdots H_1 o_{j+1}^\downarrow. \tag{26}$$

Let $H_{j+1} \overset{\text{def}}{=} H^{(pr)}(\tilde{o}^\uparrow_{j+1[(n_j+1):pr,1:(n_{j+1}-n_j)]})$ be a collection of Householder reflections and define

$$\tilde{R}_{j+1} = \begin{bmatrix} \tilde{R}_j & H_{j+1}\tilde{o}_{j+1}^\uparrow \end{bmatrix}, \quad \tilde{S}_{j+1} = H_{j+1}\begin{bmatrix} \tilde{S}_j & \tilde{o}_{j+1}^\downarrow \end{bmatrix}. \tag{27}$$

Then, the solution of the least squares problem $\mathcal{O}_{j+1}^\uparrow A_{j+1} = \mathcal{O}_{j+1}^\downarrow$ is given by

$$A_{j+1} = (\tilde{R}_{j+1[1:n_{j+1},1:n_{j+1}]})^{-1}\tilde{S}_{j+1[1:n_{j+1},1:n_{j+1}]}.$$

Proof. As $\tilde{R}_j$ is by construction an upper triangular matrix with $\tilde{R}_{j[(n_j+1):pr,1:n_j]} = 0$, it follows $H_{j+1}\tilde{R}_j = \tilde{R}_j$ from the definition of H_{j+1}. Then, from (27) follows $\tilde{R}_{j+1} = H_{j+1}\begin{bmatrix} \tilde{R}_j & \tilde{o}_{j+1}^\uparrow \end{bmatrix}$ and together with (27), it follows

$$\begin{aligned} \tilde{R}_{j+1} &= H_{j+1}\begin{bmatrix} \tilde{R}_j & \tilde{o}_{j+1}^\uparrow \end{bmatrix} = H_{j+1}\begin{bmatrix} H_j\begin{bmatrix} \tilde{R}_{j-1} & \tilde{o}_j^\uparrow \end{bmatrix} & \tilde{o}_{j+1}^\uparrow \end{bmatrix} \\ &= H_{j+1}H_j\begin{bmatrix} \tilde{R}_{j-1} & H_{j-1}\cdots H_1\begin{bmatrix} o_j^\uparrow & o_{j+1}^\uparrow \end{bmatrix} \end{bmatrix} = \ldots \\ &= H_{j+1}H_j\cdots H_1\begin{bmatrix} \mathcal{O}_1^\uparrow & o_2^\uparrow & \ldots & o_{j+1}^\uparrow \end{bmatrix} \\ &= H_{j+1}H_j\cdots H_1\mathcal{O}_{j+1}^\uparrow. \end{aligned}$$

Analogously, it holds $\tilde{S}_{j+1} = H_{j+1}H_j\cdots H_1\mathcal{O}_{j+1}^\downarrow$. The Householder reflections were chosen such that $\tilde{R}_{j+1} \in \mathbb{R}^{pr \times n_{j+1}}$ is upper triangular. Hence, $\mathcal{O}_{j+1}^\uparrow = \tilde{Q}_{j+1}\tilde{R}_{j+1}$ with $\tilde{Q}_{j+1}^T = H_{j+1}H_j\cdots H_1 \in \mathbb{R}^{pr\times pr}$ is a full QR decomposition of $\mathcal{O}_{j+1}^\uparrow$ and $\tilde{S}_{j+1} = \tilde{Q}_{j+1}^T\mathcal{O}_{j+1}^\downarrow$. Thus, the assertion follows. □

With this proposition, the matrices $\tilde{R}_{j+1}$ and $\tilde{S}_{j+1}$ are computed iteratively. Then, Corollary 4 can be used to compute A_{j+1} at each iteration efficiently, using $\tilde{R}_{j+1}$ and $\tilde{S}_{j+1}$ instead of R_t and S_t, respectively. The complete algorithm for this fast iterative multi-order computation of the state transition matrix is obtained from Algorithm 2 by replacing Line 6 at $j+1$ with Equation (25), while replacing R_t and S_t in (24) by $\tilde{R}_{j+1}$ and $\tilde{S}_{j+1}$ from (27). Denote the resulting algorithm by **Algorithm 4**. Its flops are counted in Table 5.

Table 5: Flop count of Algorithm 4 for $n_j = 1, 2, \ldots, n_*$ and $n_t = n_{\max}$.

Step	Operation	Flops
Equ. (26)	Householder reflections	$2\sum_{k=1}^{j} 4(cn_{\max} - k)$ $\approx 8cn_{\max}j - 4j^2$
Equ. (27)	$H_{j+1}\tilde{o}_{j+1}^\uparrow, H_{j+1}\begin{bmatrix} \tilde{S}_j & \tilde{o}_{j+1}^\downarrow \end{bmatrix}$	$4(cn_{\max} - j)j$
Equ. (25)	$\tilde{R}_j^{(11)-1}\tilde{S}_j^{(12)}, \tilde{R}_j^{(11)-1}\tilde{R}_j^{(12)}$	$2j^2$
Equ. (25)	dyadic product, sum	$2j^2$
		$\sum = 6cn_{\max}n_*^2 - \frac{4}{3}n_*^3$

Table 6: Flop count comparison of multi-order system identification algorithms using SSI.

Algorithm	Flops
SSI with pseudoinverse (Alg. 1)	$\frac{16}{3}cn_{\max}n_*^3 + \frac{5}{2}n_*^4$
SSI with QR (Algorithm 1)	$2cn_{\max}n_*^3 - \frac{1}{12}n_*^4$
Fast SSI from (Döhler and Mevel, 2011a)	$(6c-1)n_{\max}^3 + \frac{1}{4}n_*^4$
Fast SSI (Algorithm 2)	$(6c - \frac{4}{3})n_{\max}^3 + \frac{1}{4}n_*^4$
Iterative Fast SSI (Algorithm 3)	$(6c - \frac{4}{3})n_{\max}^3 + \frac{4}{3}n_*^3$
Iterative Fast SSI (Algorithm 4)	$6cn_{\max}n_*^2 - \frac{4}{3}n_*^3$

3.4. Comparison of Multi-Order Algorithms

The computational complexities for the computation of the system matrices with the multi-order SSI algorithms from the last sections are summarized in Table 6. For a comparison, also the Fast SSI algorithm from (Döhler and Mevel, 2011a) is stated, cf. Remark 3. All results are given for the computation at model orders $1, 2, \ldots, n_*$ from an observability matrix of size $(cn_{\max} + r) \times n_{\max}$, where $n_* \le n_{\max}$.

The conventional Algorithm 1, either using the pseudoinverse or the QR decomposition for the solution of the least squares problem, takes $O(n_{\max}^4)$ operations for $n_* = n_{\max}$. The simplest of the derived fast algorithms, Algorithm 2, is still dependent on n_*^4, although its constant is significantly smaller than for Algorithm 1, and even $(6c - \frac{4}{3})n_{\max}^3 \ge \frac{1}{4}n_*^4$ in many cases. The iterative fast SSI algorithms from Sections 3.2 and 3.3 (Algorithms 3 and 4) take only $O(n_{\max}^3)$ operations for $n_* = n_{\max}$, finally.

3.5. Multi-Order Algorithms in the Context of the Entire Subspace Identification Algorithm

In the previous sections, fast algorithms for the multi-order computation of the state transition matrix from the observability matrix have been derived. In this section, their contribution within the entire subspace identification algorithm is analyzed, namely after the computation of the subspace matrix $\mathcal{H}$ and its SVD in (4), from where the observability matrix at the maximal order is obtained.

The computation of $\mathcal{H}$ depends strongly on the choice of the selected subspace algorithm and on the size N of the data sample. Using *covariance-driven* subspace algorithms such as in (6), the size of $\mathcal{H}$ is in the order of $cn_{\max} \times n_{\max}$ and does not depend on N. The computation of the covariances R_i, $i = 1, \ldots, p+q$, e.g. for (6) takes $2N(p+q)rr_0 \approx 4Nrn_{\max}$ flops. Also, the FFT algorithm can be used to efficiently compute the covariances (Stoica and Moses, 1997).

With *data-driven* subspace algorithms such as UPC in (7), memory problems arise more easily, as the size of $\widetilde{\mathcal{H}}$ is in the order of $cn_{\max} \times N$. In this case, a prior LQ decomposition of the data Hankel matrices as in (8) is recommended for data compression (Van Overschee and De Moor, 1996), resulting in a matrix $\mathcal{H}$ of size $cn_{\max} \times n_{\max}$ with the property

$\widetilde{\mathcal{H}} = \mathcal{H}Q$, where Q is orthogonal. Then, the observability matrix O is equivalently obtained from $\mathcal{H}$. In order to obtain the "compressed" matrix $\mathcal{H}$ from a LQ decomposition and to avoid memory problems for large N, the data Hankel matrices can be filled block-wise and the LQ decomposition is performed iteratively on the blocks. Another possibility is to make use of the displacement structure of the data Hankel matrices as in (Mastronardi et al., 2001a) to obtain the L factor efficiently, avoiding also the computation of these large matrices. The LQ decomposition of data Hankel matrices of size $((p+1)r + qr_0) \times N$ (as needed e.g. for output-only algorithms as UPC) takes around $4N((p+1)r+qr_0)^2 \approx 4Nc^2n_{\max}^2$ flops with the former approach, while the latter approach takes around $10Ncrn_{\max}$ flops. Other data-driven algorithms are based on decompositions of similar data Hankel matrices (e.g. also including inputs) and have a numerical complexity of the same magnitude (Van Overschee and De Moor, 1996; Benveniste and Mevel, 2007).

Finally, the observability matrix O_t at maximal model order is obtained from U and Σ of the SVD of $\mathcal{H}$, taking $(14c-2)n_{\max}^3$ flops Golub and Van Loan (1996).

These preprocessing costs apply for all described algorithms for the multi-order computation of the system matrices. The flop count of the conventional algorithm (Algorithm 1) is of the same order as the computation of $\mathcal{H}$ and O_t from the data for $Nr_0 \approx n_{\max}^3$. Thus, the contribution of the fast multi-order algorithms is still significant for data samples up to size $N \approx n_{\max}^3/r_0$.

4. Eigensystem Realization Algorithm (ERA)

4.1. System Identification with ERA

System identification with NExT-ERA is closely related to covariance-driven stochastic subspace identification. The Natural Excitation Technique (NExT; James III et al., 1995; Farrar and James III, 1997) states that the cross-correlation function between two responses made on unknown ambient excitation has the same form as the system's impulse response function under convenient assumptions. Then, the Eigensystem Realization Algorithm (ERA; Juang and Pappa, 1985), which was developed to analyze impulse response functions, can be applied for system identification.

For ERA, the subspace matrix $\mathcal{H}$ is built as a block Hankel matrix

$$\mathcal{H}_k \overset{\text{def}}{=} \begin{bmatrix} M_k & M_{k+1} & \dots & M_{k+q-1} \\ M_{k+1} & M_{k+2} & \dots & M_{k+q} \\ \vdots & \vdots & \ddots & \vdots \\ M_{k+p} & M_{k+p+1} & \dots & M_{k+p+q-1} \end{bmatrix},$$

where parameters p and q are used as in Section 2.1, parameter $k \in \mathbb{N}_0$ indicates a time lag and the so-called Markov parameters M_i at time lag i can be chosen as one of the following functions (Siringoringo and Fujino, 2008):

- Impulse response functions;
- Inverse Fast Fourier Transform (FFT) of frequency response functions;
- Cross-correlation functions of outputs under ambient excitation;
- Inverse FFT of cross-spectral densities under ambient excitation.

Then, $\mathcal{H}_k$ possesses the factorization property

$$\mathcal{H}_k = OA^k\mathcal{Z}$$

analogous to (2). Using the SVD decomposition

$$\mathcal{H}_k = \begin{bmatrix} U_1 & U_0 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_0^T \end{bmatrix}, \tag{28}$$

and truncation at some model order n analogous to (3), the state transition matrix A is computed as

$$A = \Sigma_1^{-1/2} U_1^T \mathcal{H}_{k+1} V_1 \Sigma_1^{-1/2}, \tag{29}$$

similar to (4) and (5). The observation matrix C is obtained as the first block row of $U_1\Sigma_1^{1/2}$ as in Section 2.1.

4.2. Fast Multi-Order Computation of the System Matrices

When the model order n in the truncation of the SVD in (28) is unknown, it is useful to do multi-order system identification at model orders $1 \leq n_1 < n_2 < \ldots < n_t$ as in Section 2. This corresponds to the partition of the SVD in (28) at model orders n_j, $j = 1, \ldots, t$, such that

$$\mathcal{H}_k = U\Sigma V^T = \begin{bmatrix} U_j & \check{U}_j \end{bmatrix} \begin{bmatrix} \Sigma_j & 0 \\ 0 & \check{\Sigma}_j \end{bmatrix} \begin{bmatrix} V_j^T \\ \check{V}_j^T \end{bmatrix}, \tag{30}$$

where U_j and V_j have n_j columns and $\Sigma_j \in \mathbb{R}^{n_j \times n_j}$. Then, the state transition matrix at model order n_j writes as

$$A_j = \Sigma_j^{-1/2} U_j^T \mathcal{H}_{k+1} V_j \Sigma_j^{-1/2}. \tag{31}$$

Proposition 8. *Let A_t from* (30) *and* (31) *at a maximal desired model order n_t be given. Then, A_j satisfying* (31) *at model order n_j is a submatrix of A_t and fulfills*

$$A_j = A_{t[1:n_j,1:n_j]}.$$

Proof. From (30) and (31), A_t is defined as

$$A_t = \Sigma_t^{-1/2} U_t^T \mathcal{H}_{k+1} V_t \Sigma_t^{-1/2}.$$

For any $j = 1, \ldots, t$, it holds by definition

$$U_t = \begin{bmatrix} U_j & \tilde{U}_j \end{bmatrix}, \ \Sigma_t^{-1/2} = \begin{bmatrix} \Sigma_j^{-1/2} & 0 \\ 0 & \tilde{\Sigma}_j^{-1/2} \end{bmatrix}, \ V_t = \begin{bmatrix} V_j & \tilde{V}_j \end{bmatrix}$$

with some matrices $\tilde{U}_j$, $\tilde{\Sigma}_j$ and $\tilde{V}_j$. Thus, plugging this in the definition of A_t it follows

$$A_t = \begin{bmatrix} \Sigma_j^{-1/2} U_j^T \mathcal{H}_{k+1} V_j \Sigma_j^{-1/2} & \Sigma_j^{-1/2} U_j^T \mathcal{H}_{k+1} \tilde{V}_j \tilde{\Sigma}_j^{-1/2} \\ \tilde{\Sigma}_j^{-1/2} \tilde{U}_j^T \mathcal{H}_{k+1} V_j \Sigma_j^{-1/2} & \tilde{\Sigma}_j^{-1/2} \tilde{U}_j^T \mathcal{H}_{k+1} \tilde{V}_j \tilde{\Sigma}_j^{-1/2} \end{bmatrix}$$

and comparing with the definition of A_j in (31), the assertion follows. □

Table 7: Flop count comparison of multi-order system identification algorithms using ERA

Algorithm	Flops
Conventional Computation	$(\frac{1}{2}c + \frac{1}{2})n_*^4$
Fast Computation (Proposition 8)	$(2c + 2)n_{\max}^3$
Iterative Fast Computation (Corollary 9)	$2cn_{\max}^2 n_* + 2n_{\max} n_*^2$

Hence, only A_t at the maximal desired model order n_t needs to be computed for multi-order system identification with ERA. Then, the matrices A_j, $j = 1, \ldots, t-1$, are simply submatrices of A_t and do not require further computations.

With the following corollary, the prior computation of A_t at the maximal desired model order n_t can be avoided and the matrices A_j are defined iteratively, analogous to Section 3.3.

Corollary 9. *Let $\mathcal{H}_k$, $\mathcal{H}_{k+1}$ and the SVD* (30) *be given. Define $T_1 = \Sigma_1^{-1/2} U_1^T \mathcal{H}_{k+1}$ and $A_1 = T_1 V_1 \Sigma_1^{-1/2}$. For $j = 1, 2, \ldots$ define u_{j+1}, σ_{j+1} and v_{j+1} such that*

$$U_{j+1} = \begin{bmatrix} U_j & u_{j+1} \end{bmatrix}, \; \Sigma_{j+1} = \begin{bmatrix} \Sigma_j & 0 \\ 0 & \sigma_{j+1} \end{bmatrix}, \; V_{j+1} = \begin{bmatrix} V_j & v_{j+1} \end{bmatrix},$$

and

$$t_{j+1} = \sigma_{j+1}^{-1/2} u_{j+1}^T \mathcal{H}_{k+1}, \quad T_{j+1} = \begin{bmatrix} T_j \\ t_{j+1} \end{bmatrix}. \tag{32}$$

Then,

$$A_{j+1} = \begin{bmatrix} A_j & T_j v_{j+1} \sigma_{j+1}^{-1/2} \\ t_{j+1} V_j \Sigma_j^{-1/2} & t_{j+1} v_{j+1} \sigma_{j+1}^{-1/2} \end{bmatrix}. \tag{33}$$

Proof. Replace A_t, U_t, Σ_t and V_t in the proof of Proposition 8 by A_{j+1}, U_{j+1}, Σ_{j+1} and V_{j+1}. □

For a comparison of the computational complexity of this algorithm with the results of Section 3.4, it is assumed that the SVD (30) is the starting point of the computation, which is analogous to assume that the observability matrix used for SSI algorithms is already known. The notation of Sections 2.4 and 3.4 is used and $\mathcal{H}_k$ is assumed to be of size $cn_{\max} \times n_{\max}$.

About $(2c + 2)n_{\max}^3$ flops are necessary to compute A_t at model order $n_t = n_{\max}$, from which the state transition matrices at inferior orders are selected in Proposition 8. In Corollary 9, the computation of A_j at some order j takes about $2cn_{\max}^2 + 4n_{\max} j$ flops, amounting to $2cn_{\max}^2 n_* + 2n_{\max} n_*^2$ flops when computing at all orders $1, 2, \ldots, n_*$. Note that the conventional computation of A_j for all $j = 1, \ldots, t$ from (31) needs about $(\frac{1}{2}c + \frac{1}{2})n_*^4$ flops.

The computational complexities of the derived multi-order ERA algorithms are summarized in Table 7.

5. Application: Structural Vibration Analysis

In this section, the fast multi-order computation of the system matrices is applied to a practical test case from vibration analysis. So-called stabilization diagrams are used that contain the system identification results at multiple model orders. In the following, the underlying models for modal analysis are recalled (Prevosto et al., 1991).

5.1. Modeling and Eigenstructure Identification

The behavior of a mechanical structure is described by a continuous-time, time-invariant, linear dynamical system, modeled by the vector differential system

$$\left\{ \begin{array}{rcl} M\ddot{x}(t) + C_1\dot{x}(t) + Kx(t) & = & \upsilon(t) \\ y(t) & = & Lx(t) \end{array} \right. \tag{34}$$

where t denotes continuous time; M, C_1, and K are mass, damping, and stiffness matrices, respectively; the (high dimensional) state vector $x(t)$ is the displacement vector of the degrees of freedom of the structure; the external unmeasured force $\upsilon(t)$ is unmeasured noise; measurements are collected in the (low dimensional) vector $y(t)$ and matrix L indicates which degrees of freedom are actually measured, i.e. the sensor locations.

The parameters to be identified are the eigenvalues (or modes) μ and mode shapes ψ_μ of system (34), which comprise the *modal parameters*, and are solutions of

$$(\mu^2 M + \mu C_1 + K)\Psi_\mu = 0, \quad \psi_\mu = L\Psi_\mu. \tag{35}$$

Sampling model (34) at rate $1/\tau$ yields the discrete time state space model (1), where the state and the output are

$$x_k = \begin{bmatrix} x(k\tau) \\ \dot{x}(k\tau) \end{bmatrix}, \quad y_k = y(k\tau),$$

the state transition and observation matrices are

$$A = \mathrm{e}^{\mathcal{L}\tau}, \text{ where } \mathcal{L} = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}C_1 \end{bmatrix}, \quad C = \begin{bmatrix} L & 0 \end{bmatrix}.$$

The external force $\upsilon(t)$ and thus the state noise (v_k) in model (1) can be non-stationary and colored noise (Benveniste and Mevel, 2007; Basseville et al., 2007). The eigenstructure $(\lambda, \varphi_\lambda)$ of system (1) is defined by the eigenvalues and eigenvectors of A and by C:

$$(A - \lambda I)\phi_\lambda = 0, \quad \varphi_\lambda = C\phi_\lambda. \tag{36}$$

The desired modal parameters in (35) are equivalently found in the eigenstructure $(\lambda, \varphi_\lambda)$ of (1) and it holds

$$\mathrm{e}^{\mu\tau} = \lambda, \quad \psi_\mu = \varphi_\lambda.$$

The modal frequencies f and damping coefficients ρ are recovered directly from the eigenvalues λ by

$$f = \frac{a}{2\pi\tau}, \quad \rho = \frac{100|b|}{\sqrt{a^2 + b^2}}, \tag{37}$$

where $a = |\arctan \Re(\lambda)/\Im(\lambda)|$ and $b = \ln|\lambda|$.

Thus, vibration analysis is stated as the problem of identifying the eigenstructure of a linear dynamic system. Parameters of interest are modes (modal frequencies f, damping ratios ρ) and mode shapes φ_λ.

5.2. The Stabilization Diagram

In Operational Modal Analysis (OMA), the eigenstructure of mechanical, civil and aeronautical structures is identified from output-only data under ambient excitation. With forced excitation e.g. by shakers (exogenous inputs, OMAX), some of the inputs are available. In both cases, the selection of the model order in (3), and thus the parameters p and q of the subspace matrix $\mathcal{H}$ on one hand, and the handling of excitation and measurement noises on the other hand, are two major practical issues.

In order to retrieve a desired number of modes, an even larger model order must be assumed while performing identification. A number of spurious modes appears in the identified model due to this over-specification, as well as due to colored noise or non-linearities that appear in practice. Techniques from statistics to estimate the best model order may lead to a model with the best prediction capacity, but one is rather interested in a model containing only the physical modes of the investigated structured, while rejecting the spurious modes. Based on the observation that physical modes remain quite constant when estimated at different over-specified model orders, while spurious modes vary, they can be distinguished using so-called *stabilization diagrams* (Peeters and De Roeck, 1999, 2001; Van der Auweraer and Peeters, 2004). There, frequencies estimated from multi-order system identification are plotted against the model order. From the modes common to many models and using further stabilization criteria, such as threshold on damping values, low variation between modes and mode shapes of successive orders etc., the final estimated model is obtained.

At each of these model orders, the system matrices have to be computed first in order to get the eigenstructure of the respective systems. For system identification with SSI algorithms or with ERA, the system matrices can be estimated efficiently and fast with the new algorithms derived in this paper.

5.3. Numerical Results of Multi-Order System Identification

All system identification algorithms of this paper are applied to the system identification of the Z24 Bridge (Maeck and De Roeck, 2003; Parloo, 2003). It was a prestressed concrete bridge with three spans, supported by two intermediate piers and a set of three columns at each end. Both types of supports are rotated with respect to the longitudinal axis which results in a skew bridge. The overall length is 58 m and a schematic view of the bridge is presented in Figure 1.

Because of the size of the bridge, the system response was measured in nine setups of up to 33 sensors each, with five reference sensors common to all setups. Altogether, the structure was measured at $r = 251$ sensor positions, of which are $r_0 = 5$ reference sensors. In each setup, 65,536 samples were collected for each sensor with a sampling frequency of 100 Hz. The common subspace matrix of all setups was obtained with the merging approach described in Döhler and Mevel (2011b).

The different algorithms presented in this paper are tested on an Intel Xeon CPU 3.40 GHz with 16 GByte in Matlab 7.10.0.499 using one processor kernel. With these algorithms, the system matrices A_j and C_j are computed at model orders $n_j = 1, 2, \ldots, n_{\max}$. From these results, the modal parameters of the system can be identified using the stabilization diagram. To compare the performance of the algorithms, the system matrices are computed for stabilization diagrams with different maximal model orders $n_{\max}$.

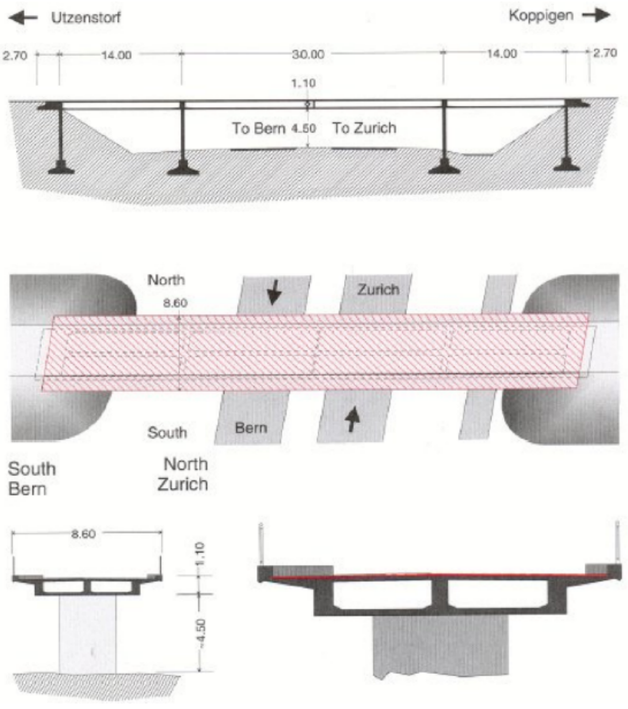


Figure 1: Schematic view of the Z24 Bridge.

- For SSI algorithms, a subspace matrix $\mathcal{H}$ of size $(p+1)r \times qr_0$ is built from the data, where $p + 1 = q$ is chosen, as recommended in Basseville et al. (2001). $\mathcal{O}_t$ is obtained from $\mathcal{H}$, where the maximal model order is $n_{\max} = n_t = qr_0$. Then, the time is recorded for the computation of A_j and C_j from $\mathcal{O}_t$ at model orders $n_j = j = 1, 2, \ldots, n_{\max}$.
- For ERA, the matrix $\mathcal{H}$ from the subspace algorithm is used for simplicity. Set $n_{\max} = qr_0$, $\mathcal{H}_k = \mathcal{H}_{[1:pr,1:qr_0]}$ and $\mathcal{H}_{k+1} = \mathcal{H}_{[(r+1):(p+1)r,1:qr_0]}$. Then, the SVD of $\mathcal{H}_k$ is performed and the time is recorded for the computation of A_j and C_j from U, Σ, V and $\mathcal{H}_{k+1}$ at model orders $n_j = j = 1, 2, \ldots, n_{\max}$.

These steps are repeated for $q = 2, \ldots, 100$ in order to evaluate the computational time for obtaining the set of A_j's and C_j's from order 1 until a maximal model order $n_{\max} = qr_0$. As the computation time is also dependent on the constant $c \approx r/r_0$ (see Section 2.4), first a computation is done with all $r = 251$ sensors ($c \approx 50$), and second a computation with only a subset of $r = 5$ sensors ($c \approx 1$).

5.3.1. Computation Times of Multi-Order SSI Algorithms at Different Maximal Model Orders

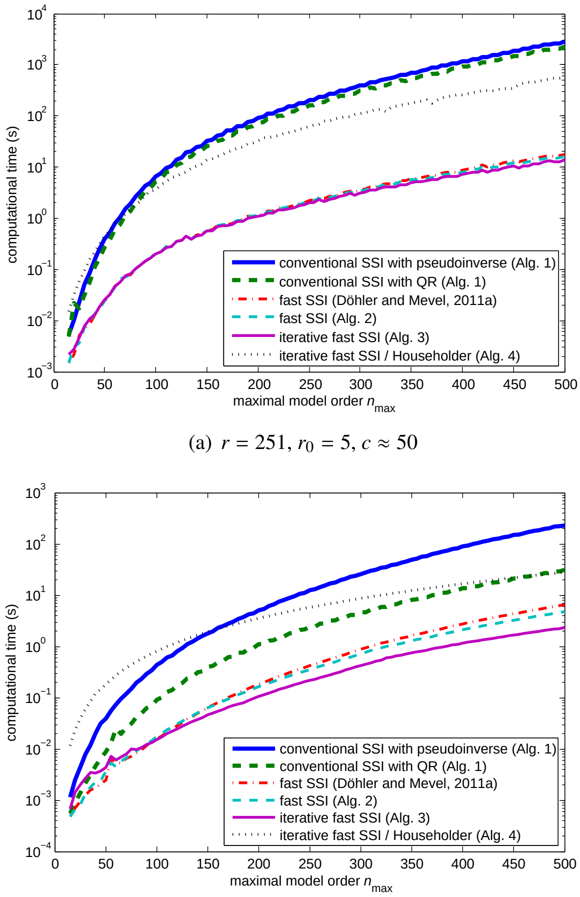
The computation times for the computation of the system matrices at model orders $1, 2, \ldots, n_{\max}$ from an observability matrix $O_{n_{\max}}$ with the SSI algorithms are presented in Figure 2 for different maximal model orders $n_{\max}$. It can be seen that the solution of the least squares problem with the QR decomposition is more efficient than using the pseudoinverse in the conventional Algorithm 1, as expected. The Algorithms 2 and 3 derived in this paper yield a significant reduction in the computation times, as well as the earlier version of Algorithm 2 from (Döhler and Mevel, 2011a). Although Algorithm 4 theoretically needs the least number of operations, it does not perform well in practice. This might be due to the fact that manually handling many simple matrix operations in this algorithm (Householder reflections) is less efficient in Matlab than performing a QR decomposition with the inherent `qr` function. Besides Algorithm 4, all other algorithms seem to be consistent with their theoretical performance.

At each maximal model order, the computation time in Figure 2 corresponds to the total time that is needed to compute the system matrices A_j and C_j for a stabilization diagram having this maximal model order. It is clearly shown that the new fast iterative multi-order SSI with Algorithm 3 outperforms the other algorithms. For $n_{\max} = 500$ it takes a total of 13.7 s, while the conventional multi-order Algorithm 1 using the pseudoinverse takes 2873 s, thus being more than 200 times faster.

5.3.2. Computation Times of Multi-Order SSI Algorithms for Maximal Model Order 500

The accumulated computation times of the system matrices at model orders $1, 2, \ldots, n_*$ for the maximal model order $n_{\max} = 500$ are presented in Figure 3 for $n_* = 1, \ldots, 500$. This corresponds to the time that is needed to compute the system matrices up to each order n_* for a stabilization diagram of maximal model order $n_{\max} = 500$.

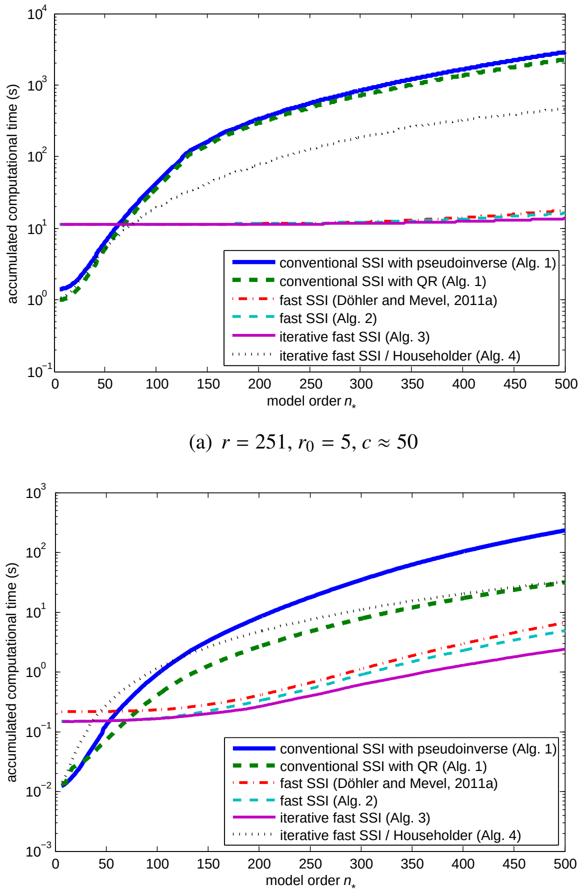
Due to a preprocessing step at the maximal model order in Algorithm 2, its prior version from (Döhler and Mevel, 2011a) and Algorithm 3, the accumulated computation time for the first model orders n_* is higher in these fast algorithms than for the conventional algorithms. However, this changes quickly at higher model orders and the new algorithms outperform the conventional ones, with Algorithm 3 being the fastest. It is also noted that the preprocessing step takes the main part of the computation, while after that, the computation of the system matrices at all orders $1, 2, \ldots, 500$ take only about 2.4 seconds for Algorithm 3 in Figures 3(a) and 3(b).



(a) $r = 251$, $r_0 = 5$, $c \approx 50$

(b) $r = r_0 = 5$, $c \approx 1$

Figure 2: Computation times for multi-order SSI system identification of system matrices A_j and C_j, $n_j = j = 1, \ldots, n_{\max}$ from $O_{n_{\max}}$ at different maximal model orders $n_{\max}$.



(a) $r = 251$, $r_0 = 5$, $c \approx 50$

(b) $r = r_0 = 5$, $c \approx 1$

Figure 3: Accumulated computation times for multi-order SSI system identification up to different model orders n_* with maximal model order $n_{\max} = 500$.

5.3.3. Computation Times of Multi-Order ERA Algorithms

The computation times for the computation of the system matrices at model orders $1, 2, \ldots, n_{\max}$ from an observability matrix $O_{n_{\max}}$ with the ERA algorithms are presented in Figure 4 for different maximal model orders $n_{\max}$. The accumulated computation times of the system matrices at model orders $1, 2, \ldots, n_*$ for the maximal model order $n_{\max} = 500$ are presented in Figure 5 for $n_* = 1, \ldots, 500$.

From both figures it can be seen that the new multi-order algorithms for ERA developed in this paper are significantly faster than the conventional ERA algorithm. Due to its implementation, the fast ERA algorithm from Proposition 8 outperforms its iterative variant from Corollary 9, except when stopping the computation at a low model order. For the computation of the system matrices at orders $1, 2, \ldots, 500$ with $n_{\max} = 500$, the fast multi-order ERA algorithm takes a total of 5 s, while the conventional multi-order ERA algorithm takes 980 s, thus being about 200 times faster.

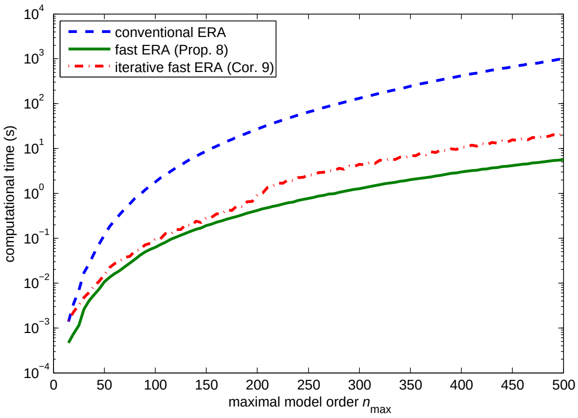


Figure 4: Computation times for multi-order ERA system identification of system matrices A_j and C_j, $n_j = j = 1, \ldots, n_{\max}$ from $O_{n_{\max}}$ at different maximal model orders $n_{\max}$ with $r = 251$, $r_0 = 5$, $c \approx 50$.

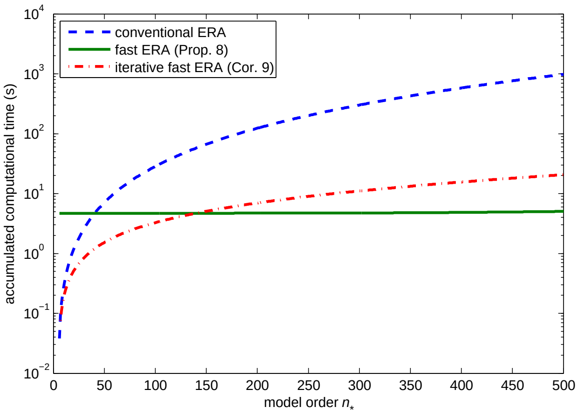


Figure 5: Accumulated computation times for multi-order ERA system identification up to different model orders n_* with maximal model order $n_{\max} = 500$ and $r = 251$, $r_0 = 5$, $c \approx 50$.

5.3.4. Discussion of the Results

In Figure 2, computation times are obtained for multi-order system identification with SSI for different maximal model orders, where for each maximal order $n_{\max}$ the system matrices are computed at orders $1, 2, \ldots, n_{\max}$. In Figure 3, the accumulated computation times at each of these orders $1, 2, \ldots, n_{\max}$ are obtained for $n_{\max} = 500$. With these results, the performance of the algorithms proposed in this paper can be evaluated for the computation of the system matrices up to a maximal model order $n_{\max}$ as well as up to a lower order $n_* \leq n_{\max}$. From both figures it can be seen that the conventional algorithm (Algorithm 1 using the pseudoinverse), which is widely used, is the slowest except when stopping the computation at a very small model order. Using the QR decomposition in Algorithm 1 yields already faster results, while the new Algorithms 2 and 3 yield significant improvements in the computation time, as well as an earlier version of Algorithm 2 from (Döhler and Mevel, 2011a). The latter algorithm yields slightly slower results than Algorithm 2. Algorithm 3 is in general the fastest of these algorithms, except at very low model orders, where computation times are lower than 0.01 s. Algorithm 4 theoretically needs even less operations (cf. Table 6), but is in practice slower due to its implementation. As the theoretical performances of Algorithms 3 and 4 are very close, Algorithm 3 is favorable.

In Figures 4 and 5 corresponding results for the ERA algorithms from Section 4 are obtained. Although the fast multi-order computation (Proposition 8) and its iterative variant (Corollary 9) theoretically need the same number of operations for $n_* = n_{\max}$ in Figure 4, the former algorithm is faster due to implementation reasons. When stopping the computation at lower model orders in Figure 5, the iterative algorithm is faster only at very low model orders. In both cases, the conventional multi-order algorithm is the slowest.

5.3.5. Computation Times of Multi-Order Algorithms within the Entire Subspace Identification Algorithm

The actual computation of the subspace matrix $\mathcal{H}$ and the observability matrix O_t from the data depends on the selected subspace algorithm, for which a large variety is found in literature, and also depends strongly on the size N of the data sample (see Section 3.5). In the following, an example of this computation is given, which is the starting point of the multi-order algorithms, and both computation times are added.

Using covariance-driven SSI, the computation time for the matrices $\mathcal{H}$ and O_t is compared to the conventional algorithm (Algorithm 1) and the fastest algorithm (Algorithm 3) for the multi-order computation of the system matrices in Table 8 for some chosen maximal model orders. Note that for $n_{\max} = 70$, the relation $N \approx n_{\max}^3/r_0$ holds. At this order, the decrease of the entire computation time is 15 % with the new algorithms. For higher model orders, the contribution of the new algorithms is increasing compared to the computation time of $\mathcal{H}$ and O_t.

Moreover, a simulation was performed with the same sensor configuration as in the example above ($r = 251$, $r_0 = 5$), but using a very long data set of length $N = 1{,}600{,}000$ and model order $n_{\max} = 200$, where $N = n_{\max}^3/r_0$. Computation time for $\mathcal{H}$ is then 428 s using the FFT for the correlation computations.

Table 8: Computation times of the parts of the entire subspace algorithm (r = 251, r_0 = 5, $c \approx 50$).

$n_{\max}$	$\mathcal{H}$	O_t	Alg. 1	Alg. 3	total decrease
70	8.16 s	0.06 s	1.5 s	0.07 s	15 %
200	8.96 s	0.75 s	92 s	1.1 s	89 %
500	17.1 s	8.4 s	2873 s	13.7 s	99 %

A decrease of 18 % of the total computation time is obtained using Algorithm 3 compared to Algorithm 1. Note that further improvements of the computation time of $\mathcal{H}$ for the covariance-driven subspace algorithms are possible in multi-processor environments, as the FFT computation can be easily parallelized (Tan et al., 2008).

Thus, the new algorithms presented in this paper indeed have a significant contribution in the context of the entire subspace algorithm for moderately sized data samples in relation to the maximal model order.

5.4. Modal Parameter Estimation and Stabilization Diagram

In order to obtain the modal parameters of the vibration analysis example (cf. Sections 5.1 and 5.2), the multi-order system identification performed in the previous section is one part of the task. From these results, the eigenstructure of the investigated structure is obtained from the system matrices at the multiple orders using (36) and (37) in the next step. Note that the eigenstructure computation has a complexity of $O(n_j^3)$ at each model order n_j and thus $O(n_{\max}^4)$ for the entire computation. However, this step took a total time of 168 s for $n_{\max} = 500$, thus being a small part compared to the conventional multi-order system identification algorithms. It is beyond the scope of this paper to optimize this process.

A stabilization diagram containing the natural frequencies of the Z24 Bridge at model orders $1, \ldots, 250$ is presented in Figure 6. Note that some of the modes – the ones that might not be very well excited – stabilize late in the diagram, making it necessary to use high model orders for system identification. Using even higher model orders than 250 can still improve the identification results, although there are only 10 modes present in this case (Parloo, 2003).

6. Conclusion

In this paper, new algorithms were derived to efficiently compute the system matrices A and C at multiple model orders from the observability matrix in stochastic subspace-based system identification (SSI) and for the closely related Eigensystem Realization Algorithm (ERA). The computational complexity for this task was reduced from $O(n_{\max}^4)$ to $O(n_{\max}^3)$, where $n_{\max}$ is the maximal desired model order, due to a mathematical reformulation of the underlying least squares problems. The new algorithms can be used with a very general class of SSI algorithms, in which A and C are estimated from the observability matrix. Considering also the computation of the observability matrix from the data, it was shown that the contribution of the new multi-order computation of the system matrices is significant for moderately sized data samples up to a size of $N \approx n_{\max}^3 / r_0$.

These algorithms are especially applied for (operational) modal analysis of mechanical, civil or aeronautical structures, where eigenstructure identification results at large multiple model orders are used to distinguish physical from spurious modes using stabilization diagrams. Their efficiency was shown on a real test case and computation time was reduced up to a factor of 200 and more for the computation of the system matrices from the observability matrix. These fast algorithms can e.g. be exploited in online monitoring, where incoming data has to be processed quickly. Among the presented algorithms, Algorithm 3 has been shown to be the fastest for multi-order SSI.

Future work contains an improved implementation of Algorithm 4, which needs slightly less flops in theory. Further extensions of this work could include the fast estimation of the system matrices B and D and the adaption to multi-order recursive system identification.

Acknowledgment

The authors would like to thank the reviewers who significantly helped in improving this paper.

The support from the European projects FP7-PEOPLE-2009-IAPP 251515 ISMS and FP7-NMP CP-IP 213968-2 IRIS is acknowledged. The data for this research were obtained in the framework of the BRITE-EURAM Programme CT96 0277, SIMCES and provided by the SAMCO organization.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Autom. Control 19 (6), 716–723.

Akçay, H., 2010. An insight into instrumental variable frequency-domain subspace identification. Automatica 46 (2), 375–382.

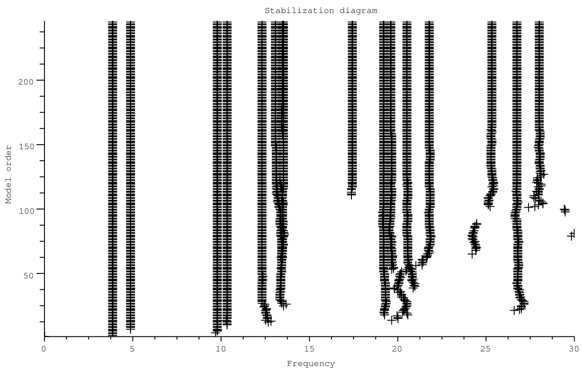


Figure 6: Stabilization diagram of Z24 Bridge containing the identified natural frequencies at model orders $1, \ldots, 250$ using the fast iterative SSI (Algorithm 3).

Basseville, M., Benveniste, A., Goursat, M., Hermans, L., Mevel, L., Van der Auweraer, H., 2001. Output-only subspace-based structural identification: from theory to industrial testing practice. J. Dyn. Syst. Meas. Contr. 123 (4), 668–676.

Basseville, M., Benveniste, A., Goursat, M., Mevel, L., 2007. In-flight monitoring of aeronautic structures: vibration-based on-line automated identification versus detection. IEEE Contr. Syst. Mag. 27 (5), 27–42.

Bastogne, T., Noura, H., Sibille, P., Richard, A., 1998. Multivariable identification of a winding process by subspace methods for tension control. Control Eng. Pract. 6 (9), 1077–1088.

Bauer, D., 2001. Order estimation for subspace methods. Automatica 37 (10), 1561–1573.

Bauer, D., 2005. Asymptotic properties of subspace estimators. Automatica 41 (3), 359–376.

Bauer, D., Deistler, M., Scherrer, W., 1999. Consistency and asymptotic normality of some subspace algorithms for systems without observed inputs. Automatica 35 (7), 1243–1254.

Bauer, D., Ljung, L., 2002. Some facts about the choice of the weighting matrices in Larimore type of subspace algorithms. Automatica 38 (5), 763–773.

Benveniste, A., Fuchs, J.-J., 1985. Single sample modal identification of a nonstationary stochastic process. IEEE Trans. Autom. Control AC-30 (1), 66–74.

Benveniste, A., Mevel, L., 2007. Non-stationary consistency of subspace methods. IEEE Trans. Autom. Control AC-52 (6), 974–984.

Brownjohn, J. M. W., Magalhães, F., Caetano, E., Cunha, A., 2010. Ambient vibration re-testing and operational modal analysis of the Humber Bridge. Eng. Struct. 32 (8), 2003–2018.

Camba-Mendez, G., Kapetanios, G., 2001. Testing the rank of the Hankel covariance matrix: A statistical approach. IEEE Trans. Autom. Control 46 (2), 331–336.

Carden, E., Fanning, P., 2004. Vibration based condition monitoring: a review. Struct. Health Monit. 3 (4), 355–377.

Chiuso, A., 2007. On the relation between CCA and predictor-based subspace identification. IEEE Trans. Autom. Control 52 (10), 1795–1812.

Chiuso, A., Picci, G., 2004. Asymptotic variance of subspace methods by data orthogonalization and model decoupling: a comparative analysis. Automatica 40 (10), 1705–1717.

Cho, Y. M., Kailath, T., 1995. Fast subspace-based system identification: An instrumental variable approach. Automatica 31 (6), 903–905.

Deistler, M., Peternell, K., Scherrer, W., 1995. Consistency and relative efficiency of subspace methods. Automatica 31 (12), 1865–1875.

Döhler, M., Mevel, L., 2011a. Fast multi-order stochastic subspace identification. In: Proc. 18th IFAC World Congress. Milan, Italy.

Döhler, M., Mevel, L., 2011b. Modular subspace-based system identification from multi-setup measurements. IEEE Trans. Autom. Control, to appear.

Farrar, C. R., James III, G. H., 1997. System identification from ambient vibration measurements on a bridge. J. Sound Vib. 205 (1), 1–18.

Golub, G., Van Loan, C., 1996. Matrix computations, 3rd Edition. Johns Hopkins University Press.

Hermans, L., Van der Auweraer, H., 1999. Modal testing and analysis of structures under operational conditions: industrial application. Mech. Syst. Signal Pr. 13 (2), 193–216.

James III, G., Carne, T., Lauffer, J., 1995. The natural excitation technique (NExT) for modal parameter extraction from operating structures. Modal Analysis 10, 260–277.

Juang, J., Pappa, R., 1985. Eigensystem realization algorithm for modal parameter identification and model reduction. J. Guid. Control Dynam. 8 (5), 620–627.

Juricek, B. C., Seborg, D. E., Larimore, W. E., 2001. Identification of the Tennessee Eastman challenge process with subspace methods. Control Eng. Pract. 9 (12), 1337–1351.

Kung, S., 1978. A new identification and model reduction algorithm via singular value decomposition. In: Proc. 12th Asilomar Conf. Circuits, Systems, and Computers. pp. 705–714.

Lovera, M., Gustafsson, T., Verhaegen, M., 2000. Recursive subspace identification of linear and non-linear Wiener state-space models. Automatica 36 (11), 1639–1650.

Maeck, J., De Roeck, G., 2003. Description of Z24 benchmark. Mech. Syst. Signal Pr. 17 (1), 127–131.

Markovsky, I., Willems, J., Van Huffel, S., De Moor, B., Pintelon, R., 2005. Application of structured total least squares for system identification and model reduction. IEEE Trans. Autom. Control 50 (10), 1490–1500.

Mastronardi, N., Kressner, D., Sima, V., Van Dooren, P., Van Huffel, S., 2001a. A fast algorithm for subspace state-space system identification via exploitation of the displacement structure. J. Comput. Appl. Math. 132 (1), 71–81.

Mastronardi, N., Lemmerling, P., Van Huffel, S., 2001b. The structured total least squares problem. Contemp. Math. 280 (12), 157–176.

Mercère, G., Bako, L., Lecœuche, S., 2008. Propagator-based methods for recursive subspace model identification. Signal Process. 88 (3), 468–491.

Mevel, L., Basseville, M., Goursat, M., 2003. Stochastic subspace-based structural identification and damage detection - application to the steel-quake benchmark. Mech. Syst. Signal Pr. 17 (1), 91–101.

Mevel, L., Benveniste, A., Basseville, M., Goursat, M., Peeters, B., Van der Auweraer, H., Vecchio, A., 2006. Input/output versus output-only data processing for structural identification - application to in-flight data analysis. J. Sound Vib. 295 (3), 531–552.

Oku, H., Kimura, H., 2002. Recursive 4SID algorithms using gradient type subspace tracking. Automatica 38 (6), 1035–1043.

Pan, Y., Lee, J. H., 2008. Modified subspace identification for long-range prediction model for inferential control. Control Eng. Pract. 16 (12), 1487–1500.

Parloo, E., 2003. Application of frequency-domain system identification techniques in the field of operational modal analysis. Ph.D. thesis, Vrije Universiteit Brussel.

Peeters, B., 2000. System identification and damage detection in civil engineering. Ph.D. thesis, Katholieke Universiteit Leuven.

Peeters, B., De Roeck, G., 1999. Reference-based stochastic subspace identification for output-only modal analysis. Mech. Syst. Signal Pr. 13 (6), 855–878.

Peeters, B., De Roeck, G., 2001. Stochastic system identification for operational modal analysis: a review. J. Dyn. Syst. Meas. Contr. 123 (4), 659–667.

Prevosto, M., Olagnon, M., Benveniste, A., Basseville, M., Le Vey, G., 1991. State-space formulation, a solution to modal parameter estimation. J. Sound Vib. 148 (2), 329–342.

Qin, S., Ljung, L., 2003. Parallel QR implementation of subspace identification with parsimonious models. In: Proc. 13th IFAC SYSID Symp. pp. 1631–1636.

Reynders, E., De Roeck, G., 2008. Reference-based combined deterministic-stochastic subspace identification for experimental and operational modal analysis. Mech. Syst. Signal Pr. 22 (3), 617–637.

Reynders, E., Houbrechts, J., De Roeck, G., 2011. Automated interpretation of stability diagrams. In: Proc. 29th Int. Modal Anal. Conf. Jacksonville, USA.

Rissanen, J., 1978. Modeling by shortest data description. Automatica 14 (5), 465–471.

Scionti, M., Lanslots, J., 2005. Stabilisation diagrams: Pole identification using fuzzy clustering techniques. Adv. Eng. Softw. 36 (11-12), 768–779.

Siringoringo, D. M., Fujino, Y., 2008. System identification of suspension bridge from ambient vibration response. Eng. Struct. 30 (2), 462–477.

Sotomayor, O. A. Z., Park, S. W., Garcia, C., 2003. Multivariable identification of an activated sludge process with subspace-based algorithms. Control Eng. Pract. 11 (8), 961–969.

Stoica, P., Moses, R., 1997. Introduction to spectral analysis. Prentice Hall, Englewood Cliffs.

Tan, J., Chen, X., Xiao, L., 2008. An optimized parallel fft algorithm on multiprocessors with cache technology in linux. In: Int. Symp. Computer Science and Computational Technology. IEEE, pp. 105–109.

Van der Auweraer, H., Peeters, B., 2004. Discriminating physical poles from mathematical poles in high order systems: use and automation of the stabilization diagram. In: Proc. 21st IEEE Instr. Meas. Techn. Conf. pp. 2193–2198.

van der Veen, G., van Wingerden, J., Verhaegen, M., 2010. Closed-loop MOESP subspace model identification with parametrisable disturbances. In: Proc. 49th IEEE Conf. Decision Control. IEEE, pp. 2813–2818.

Van Overschee, P., De Moor, B., 1996. Subspace Identification for Linear Systems: Theory, Implementation, Applications. Kluwer.

Ventura, C. E., Lord, J. F., Turek, M., Brincker, R., Andersen, P., Dascotte, E., 2005. FEM updating of tall buildings using ambient vibration data. In: Proc. 6th Int. Conf. Struct. Dyn. (EURODYN), Paris, France.

Viberg, M., 1995. Subspace-based methods for the identification of linear time-invariant systems. Automatica 31 (12), 1835–1851.

List of Figures