HAL open science

Fast Multi-Order Stochastic Subspace Identification

Michael Döhler, Laurent Mevel

To cite this version:

Michael Döhler, Laurent Mevel. Fast Multi-Order Stochastic Subspace Identification. [Research Report] RR-7429, INRIA. 2010. inria-00527484

HAL Id: inria-00527484

https://inria.hal.science/inria-00527484

Submitted on 19 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Fast Multi-Order Stochastic Subspace Identification

Michael Döhler — Laurent Mevel

N° 7429

October 2010

Stochastic Methods and Models

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7429--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

Fast Multi-Order Stochastic Subspace Identification

Michael Döhler , Laurent Mevel

Theme : Stochastic Methods and Models
Applied Mathematics, Computation and Simulation
Équipe I4S

Rapport de recherche n° 7429 — October 2010 — 15 pages

Abstract: Stochastic subspace identification methods are an efficient tool for system identification of mechanical systems in Operational Modal Analysis (OMA), where modal parameters are estimated from measured vibrational data of a structure. System identification is usually done for many successive model orders, as the true system order is unknown and identification results at different model orders need to be compared to distinguish true structural modes from spurious modes in so-called stabilization diagrams. In this paper, this *multi-order* system identification with the subspace-based identification algorithms is studied and an efficient algorithm to estimate the system matrices at multiple model orders is derived.

Key-words: System identification; Subspace methods; System order; Least-squares problems; Linear systems

This work was partially supported by the European project FP7-NMP CP-IP 213968-2 IRIS.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Identification rapide par méthodes des sous espaces à ordres multiples

Résumé : Les méthodes stochastiques d'identification des sous espaces sont un outil efficace pour l'identification des systèmes linéaires dans le cadre de l'analyse modale opérationnelle sur les structures mécaniques. Dans ce cas, les paramètres modaux sont estimés à partir des données vibration mesurées sur la structure. L'identification du système est généralement faite pour plusieurs ordres de modèles successifs, puisque l'ordre réel du modèle est inconnu. Les résultats de l'identification à différents ordres de modèle doivent être comparés pour séparer les vrais modes de la structure des modes parasites, tout ceci par l'utilisation d'un diagramme de stabilisation. Dans ce travail, l'identification par sous espaces est étudiée, puis un algorithme efficace pour estimer les matrices du système à ordres multiples est développé.

Mots-clés : Identification des systèmes; Méthodes des sous espaces; Ordre du système; Problèmes aux moindres carrés; Systèmes linéaires

1 Introduction

Subspace-based linear system identification methods have been proven efficient for the system identification of mechanical systems, fitting a linear model to (input/output or output only) measurements taken from a system. Characteristics of interest to the mechanical engineer regarding this model are its *vibration modes* (its eigenfrequencies) and its *mode shapes* (corresponding eigenvectors). Therefore, identifying this linear time-invariant system (LTI) from measurements is a basic service in vibrations monitoring. Having done this allows in particular Finite Element Models (FEM) updating and structural health monitoring.

Linear system identification is a classical and widely studied subject. In an Operational Modal Analysis (OMA) context, however, the following unusual characteristics must be taken into account:

(a) 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;

(b) The number of modes of interest can be quite large (up to 100 or beyond), thus calling for non-standard approaches to model reduction;

(c) The excitation applied to the structure can be controlled and dependent from the technology used for the shakers, or it can be uncontrolled and natural, and then turbulent and non-stationary.

Because of the features (a–c) above, usual tools from linear system identification, such as the System Identification Toolbox by Matlab, are not used as such. In particular, recommended techniques from statistics to estimate the best model order (AIC, BIC, MDL, ...) do not work at all. In order to retrieve the wanted 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. Getting rid of these is *the* main issue in this context. Basically, all methods in use estimate a number of models of different orders and build a final model by fusing them in some way or another. So-called *stabilization diagrams* are a GUI-assisted way to support the engineer while performing this.

In this paper, this *multi-order* system identification with the subspace-based identification algorithms is studied in Section 2 and efficient algorithms to estimate the system matrices at multiple model orders are derived, reducing the computational burden significantly. In Section 3, 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 SSI Algorithm

The discrete time model in state space form is:

$$\left\{ \begin{array}{rcl} X_{k+1} & = & AX_k + V_{k+1} \\ Y_k & = & CX_k \end{array} \right. \tag{1}$$

with the state $X \in \mathbb{R}^n$, the output $Y \in \mathbb{R}^r$, the state transition matrix $A \in \mathbb{R}^{n\times n}$ and the observation matrix $C \in \mathbb{R}^{r\times n}$. n is the system order, and the number of outputs r is also called number of sensors. The state noise V is unmeasured and assumed to be Gaussian, zero-mean, white.

A subset of the r sensors can be used for reducing the size of the matrices in the identification process, see e.g. [8]. These sensors are called projection channels or reference sensors. Let r_0 be the number of reference sensors ($r_0 \leq r$) and p and q chosen parameters with $(p+1)r \geq qr_0 \geq n$. From the output data a matrix $\mathcal{H}_{p+1,q} \in \mathbb{R}^{(p+1)r\times qr_0}$ is built according to a chosen SSI algorithm, see e.g. [3] for an overview. The matrix $\mathcal{H}_{p+1,q}$ will be called "subspace matrix" in the following and the SSI algorithm is chosen such that the corresponding subspace matrix enjoys (asymptotically for a large number of samples) the factorization property

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

into the matrix of observability

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

and a matrix $\mathcal{Z}_q$, with an invertible weighting matrix W depending on the selected SSI algorithm. However, W is the identity matrix for many SSI algorithms.

For simplicity, let p and q be given and skip the subscripts of $\mathcal{H}_{p+1,q}$, $\mathcal{O}_{p+1}$ and $\mathcal{Z}_q$.

Example 1. *Let N 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, the "future" and "past" data matrices are built with*

$$\mathcal{Y}^+ = \frac{1}{\sqrt{N-p-q}} \begin{pmatrix} Y_{q+1} & Y_{q+2} & \vdots & Y_{N-p} \\ Y_{q+2} & Y_{q+3} & \vdots & Y_{N-p+1} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{q+p+1} & Y_{q+p+2} & \vdots & Y_N \end{pmatrix},$$

$$\mathcal{Y}^- = \frac{1}{\sqrt{N-p-q}} \begin{pmatrix} Y_q^{(\text{ref})} & Y_{q+1}^{(\text{ref})} & \vdots & Y_{N-p-1}^{(\text{ref})} \\ Y_{q-1}^{(\text{ref})} & Y_q^{(\text{ref})} & \vdots & Y_{N-p-2}^{(\text{ref})} \\ \vdots & \vdots & \vdots & \vdots \\ Y_1^{(\text{ref})} & Y_2^{(\text{ref})} & \vdots & Y_{N-p-q}^{(\text{ref})} \end{pmatrix}.$$

For the covariance-driven SSI (see also [2], [8]), the subspace matrix $\mathcal{H}^{\text{cov}} = \mathcal{Y}^+\mathcal{Y}^{-T}$ is built, which enjoys the factorization property (2) *where $\mathcal{Z}$ is the controllability matrix.*

For the data-driven SSI with the Unweighted Principal Component (UPC) algorithm (see also [10], [8]), the matrix $\tilde{\mathcal{H}}^{\text{dat}} = \mathcal{Y}^+\mathcal{Y}^{-T}(\mathcal{Y}^-\mathcal{Y}^{-T})^{-1}\mathcal{Y}^-$ enjoys

the factorization property (2) *where* $\mathcal{Z}$ *is the Kalman filter state matrix. In practice, the respective subspace matrix* $\mathcal{H}^{\mathrm{dat}}$ *is obtained from an* RQ *decomposition of the data, such that* $\tilde{\mathcal{H}}^{\mathrm{dat}} = \mathcal{H}^{\mathrm{dat}} Q$ *with an orthogonal matrix* Q*. See the mentioned references for details on the implementations.*

Now we want to obtain the eigenstructure of the system (1) from a given matrix $\mathcal{H}$. The observability matrix $\mathcal{O}$ is obtained from a thin SVD of the matrix $\mathcal{H}$ and its truncation at the desired model order n:

$$\begin{aligned} \mathcal{H} &= U \Delta V^T \\ &= \begin{bmatrix} U_1 & U_0 \end{bmatrix} \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_0 \end{bmatrix} V^T, \qquad (3) \\ \mathcal{O} &= W^{-1} U_1 \Delta_1^{1/2}. \qquad (4) \end{aligned}$$

Note that the singular values in Δ_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

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

The eigenstructure $(\lambda, \varphi_\lambda)$ results from

$$\det(A - \lambda I) = 0, \quad A\phi_\lambda = \lambda \phi_\lambda, \quad \varphi_\lambda = C\phi_\lambda, \qquad (6)$$

where λ ranges over the set of eigenvalues of A. From λ the natural frequency and damping ratio are obtained, and φ_λ is the corresponding mode shape.

There are many papers on the used identification techniques. A complete description can be found in [2], [10], [8], [3], and the related references. A proof of non-stationary consistency of these subspace methods can be found in [3].

2.2 Multi-Order SSI

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

$$1 \le n_1 < n_2 < \ldots < n_t \le \min\{pr, qr_0\}, \qquad (7)$$

and where t is the number of models to be estimated. The choice of the model orders n_j, $j = 1, \ldots, t$, is up to the user and also depends on the problem. For example, $n_j = j + c$ or $n_j = 2j + c$ with some constant c can be chosen.

The following notation for specifying these different system orders is introduced and 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, $j \in \{1, \ldots, t\}$, 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 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. A common numerically stable way to solve it, is

$$A_j = \mathcal{O}_j^{\uparrow\dagger}\mathcal{O}_j^{\downarrow} \tag{8}$$

where † denotes the Moore-Penrose pseudoinverse. A more efficient and also numerically stable way to solve it (see also [5]), is to do the thin QR decomposition

$$\mathcal{O}_j^{\uparrow} = Q_j R_j \tag{9}$$

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

$$S_j \stackrel{\text{def}}{=} Q_j^T\mathcal{O}_j^{\downarrow}, \tag{10}$$

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

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

The observation matrix C_j is found in the first block row of $\mathcal{O}_j$.

For the computation of the system matrices A_j and C_j, $j = 1,\ldots,t$, the observability matrix $\mathcal{O}_t$ at the maximal desired model order n_t is computed first from (4). Then, the $\mathcal{O}_j$ consist of the first n_j columns of $\mathcal{O}_t$ and the matrices A_j and C_j are computed with (9) to (11). This is summarized in Algorithm 1. 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: $\mathcal{O}_t \in \mathbb{R}^{(p+1)r\times n_t}$ {observability matrix}
$n_1,\ldots,n_t$ {desired model orders satisfying (7)}

1: **for** $j = 1$ to t **do**
2: $\mathcal{O}_j^{\uparrow} \longleftarrow \mathcal{O}_{t[1:pr,1:n_j]}$, $\mathcal{O}_j^{\downarrow} \longleftarrow \mathcal{O}_{t[(pr+1):(p+1)r,1:n_j]}$
3: QR decomposition $\mathcal{O}_j^{\uparrow} = Q_jR_j$
4: $S_j \longleftarrow Q_j^T\mathcal{O}_j^{\downarrow}$
5: $A_j \longleftarrow R_j^{-1}S_j$
6: $C_j \longleftarrow \mathcal{O}_{t[1:r,1:n_j]}$
7: **end for**

Output: System matrices A_j, C_j at model orders $n_1,\ldots,n_t$

2.4 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 $n_1,\ldots,n_t$, the least squares problem for the state transition matrix A_j is solved at each model order (Equations (9) to (11) with $j = 1,\ldots,t$, see also Algorithm 1).

Now, an algorithm is presented that solves the least squares problem only once at the maximal desired model order n_t (Equations (9) to (11) with $j = t$,

leading to matrices R_t, S_t and A_t) and derives the state transition matrices A_j, $j = 1, \ldots, t-1$ directly and efficiently from R_t^{-1} and S_t, based on the following main theorem of this paper.

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

$$\mathcal{O}_t^{\uparrow} = Q_t R_t, \quad S_t = Q_t^T \mathcal{O}_t^{\downarrow}, \quad A_t = R_t^{-1} S_t, \tag{12}$$

such that A_t is the least squares solution of

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

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

$$R_t^{-1} = \begin{bmatrix} R_j^{(11)} & R_j^{(12)} \\ 0 & R_j^{(22)} \end{bmatrix}, \quad S_t = \begin{bmatrix} S_j^{(11)} & S_j^{(12)} \\ S_j^{(21)} & S_j^{(22)} \end{bmatrix}, \tag{13}$$

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

$$\mathcal{O}_j^{\uparrow} A_j = \mathcal{O}_j^{\downarrow}, \tag{14}$$

satisfies

$$A_j = R_j^{(11)} S_j^{(11)}.$$

Proof. From (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

$$\mathcal{O}_t^{\uparrow} = \begin{bmatrix} \mathcal{O}_j^{\uparrow} & \hat{\mathcal{O}}_j^{\uparrow} \end{bmatrix}, \quad \mathcal{O}_t^{\downarrow} = \begin{bmatrix} \mathcal{O}_j^{\downarrow} & \hat{\mathcal{O}}_j^{\downarrow} \end{bmatrix}, \tag{15}$$

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

$$Q_t = \begin{bmatrix} Q_j^{(1)} & Q_j^{(2)} \end{bmatrix}, \quad R_t = \begin{bmatrix} \hat{R}_j^{(11)} & \hat{R}_j^{(12)} \\ 0 & \hat{R}_j^{(22)} \end{bmatrix}, \tag{16}$$

where $Q_j^{(1)} \in \mathbb{R}^{pr \times n_j}$ and $\hat{R}_j^{(11)} \in \mathbb{R}^{n_j \times n_j}$. Note that

$$\hat{R}_j^{(11)-1} = R_j^{(11)} \tag{17}$$

because of the upper triangular structure of R_t and the partitioning in (13). From (12) and (16) it follows

$$\mathcal{O}_t^{\uparrow} = \begin{bmatrix} Q_j^{(1)} & Q_j^{(2)} \end{bmatrix} \begin{bmatrix} \hat{R}_j^{(11)} & \hat{R}_j^{(12)} \\ 0 & \hat{R}_j^{(22)} \end{bmatrix} = \begin{bmatrix} Q_j^{(1)} \hat{R}_j^{(11)} & B \end{bmatrix} \tag{18}$$

with $B = Q_j^{(1)} \hat{R}_j^{(12)} + Q_j^{(2)} \hat{R}_j^{(22)}$. Comparing (15) and (18), it follows

$$\mathcal{O}_j^{\uparrow} = Q_j^{(1)} \hat{R}_j^{(11)}, \tag{19}$$

which obviously is a QR decomposition of $\mathcal{O}_j^{\uparrow}$.

Furthermore, with (12), (15) and (16) it follows

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

and comparing to (13) yields

$$S_j^{(11)} = {Q_j^{(1)}}^T \mathcal{O}_j^{\downarrow}. \tag{20}$$

As A_j is the least squares solution of (14) and because of QR decomposition (19), A_j yields

$$A_j = \hat{R}_j^{(11)-1} {Q_j^{(1)}}^T \mathcal{O}_j^{\downarrow}.$$

Then, the assertion follows together with (17) and (20). □

The resulting algorithm for the fast multi-order computation of the system matrices is summarized in Algorithm 2. At each model order n_j, n_j^3 flops are required to compute the state transition matrix A_j when R_t^{-1} and S_t are known.

Algorithm 2 Fast Multi-Order SSI

Input: $\mathcal{O}_t \in \mathbb{R}^{(p+1)r \times n_t}$ {observability matrix}
$n_1, \ldots, n_t$ {desired model orders satisfying (7)}

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 $\mathcal{O}_t^{\uparrow} = Q_t R_t$
4: $T \longleftarrow R_t^{-1}$, $S_t \longleftarrow Q_t^T \mathcal{O}_t^{\downarrow}$
5: **for** $j = 1$ to t **do**
6: $\quad A_j \longleftarrow T_{[1:n_j,1:n_j]} S_{t[1:n_j,1:n_j]}$
7: $\quad C_j \longleftarrow C_{t[1:r,1:n_j]}$
8: **end for**

Output: System matrices A_j, C_j at model orders $n_1, \ldots, n_t$

Remark 3. *In Algorithm 2 the fact is used that R_j^{-1} is the left upper $n_j \times n_j$ block of R_t^{-1}. As R_t is an upper triangular matrix, its inversion is done column-wise in ascending order by backward substitution, so the inversion of the matrix R_j is numerically equal to taking the left upper $n_j \times n_j$ block of the inverted matrix R_t^{-1}. Hence, Algorithms 1 and 2 give numerically identical results, where Algorithm 2 is more efficient.*

2.5 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, so that the number of numerical operations is further reduced.

Corollary 4. *Let R_{j+1}^{-1} and S_{j+1} (which are the upper left $n_{j+1} \times n_{j+1}$ blocks of R_t^{-1} and S_t) be partitioned into*

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

with

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

Then it holds

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

Proof. The assertion follows directly from Theorem 2 by replacing $A_j = \tilde{R}_j^{(11)} \tilde{S}_j^{(11)}$ in the product

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

□

Hence, by using (21), the computation of A_{j+1} needs less than $3(n_{j+1}-n_j)n_j^2$ flops (if $(n_{j+1} - n_j)$ is small compared to n_j), if R_t, S_t and A_j are known.

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 (21).

2.6 Computational Complexities

In the following, the complexities of the computation of the system matrices A_j and C_j, $j = 1, \ldots, t$, from an observability matrix $\mathcal{O}_t$ with the algorithms presented in Sections 2.3, 2.4 and 2.5, are evaluated. The system orders are assumed to be $n_j = j$ and the maximal model order is noted as $n_{\max} \stackrel{\text{def}}{=} n_t = t$. Furthermore, $c \stackrel{\text{def}}{=} pr/n_{\max}$ and $m \stackrel{\text{def}}{=} pr$ is defined. Note that the subspace matrix $\mathcal{H}$ is of size $(p + 1)r \times qr_0$ and in practice one would set $p + 1 = q$ (see e.g. [1]) and $n_{\max} = qr_0$. Then, $c \approx r/r_0$ and hence independent of p, q and $n_{\max}$.

According to [5], the thin SVD of $\mathcal{O}_j^{\uparrow}$ takes $14mj^2 + 8j^3$ flops and the thin Householder QR decomposition of $\mathcal{O}_j^{\uparrow}$ takes $4m^2j - \frac{4}{3}j^3$ flops. By using the

simplifications $\sum_{j=1}^{n_{\max}} j \approx \frac{1}{2} n_{\max}^2$, $\sum_{j=1}^{n_{\max}} j^2 \approx \frac{1}{3} n_{\max}^3$, $\sum_{j=1}^{n_{\max}} j^3 \approx \frac{1}{4} n_{\max}^4$, and counting the operations of the presented algorithms, the computational complexities of the computation of the system matrices from the observability matrix are obtained in Table 1.

Table 1: Computational Complexities of Multi-Order System Matrix Computation

Algorithm	Flops
SSI with pseudoinverse (Alg. 1, using (8) instead Lines 2-4)	$(5\,c + \frac{9}{4})\, n_{\max}^4$
SSI with QR (Algorithm 1)	$(\frac{5}{3}\,c - \frac{1}{12})\, n_{\max}^4$
Fast SSI (Algorithm 2)	$(5\,c - \frac{1}{2})\, n_{\max}^3 + \frac{1}{4}\, n_{\max}^4$
Iterative Fast SSI (Section 2.5)	$5\,c\, n_{\max}^3$

3 Applications

In this section, the fast multi-order computation of the system matrices is applied to practical test cases, where so-called stabilization diagrams are used that contain the system identification results at multiple model orders.

3.1 The Stabilization Diagram

In system identification, 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 Operational Modal Analysis the true system order is unknown and recommended techniques from statistics to estimate the best model order (AIC, BIC, MDL, ...) do not work at all. In order to retrieve the wanted large number of modes, an even larger model order must be assumed. Then, the subspace method yields a set of modes with both structural and spurious mathematical or noise modes, and we have to distinguish between the two types of modes. Fortunately, spurious modes tend to vary for different model orders. This is why usage suggests to plot frequencies against model order in a stability or stabilization diagram (see e.g. [9]), where the frequencies (and other modal parameters) are estimated at t increasing model orders $n_1, \ldots, n_t$. This gives results for successive different but redundant models and modes that are common to many successive models can be distinguished from the spurious modes. From the modes common to many models 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. With the new algorithms from Sections 2.4 and 2.5 this can be done much more efficiently and faster than with the conventional Algorithm 1.

3.2 Numerical Results

The system matrices A_j and C_j at model orders $n_1, \ldots, n_t$ with $n_j = j$ are computed from the observability matrix $\mathcal{O}_t$ with the different algorithms presented in this paper. This, for example, is necessary for computing a stabilization diagram containing model orders $n_1, \ldots, n_t$, see the previous section. To compare the performance of the algorithms, the system matrices are computed for stabilization diagrams with different maximal model orders n_t, like one would do in practice:

- From the data, a subspace matrix $\mathcal{H}$ of size $(p+1)r \times qr_0$ is built, where $p+1 = q$ is chosen, as e.g. recommended in [1]
- $\mathcal{O}_t$ is obtained from $\mathcal{H}$, where the maximal model order is set to $n_t = qr_0$
- A_j and C_j are computed from $\mathcal{O}_t$ at model orders $n_j = j = 1, 2, \ldots, n_t$

To evaluate the computational time for computing the set of A_j's and C_j's from order 1 until a maximal model order $n_t = qr_0$, these steps are repeated for $q = 2, \ldots, 81$ for our test case and the time is recorded for the computation of the set of A_j and C_j, $j = 1, \ldots, qr_0$, for each q.

The test case is the Z24 bridge (see [6], [7]), 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 58m and a schematic view of the bridge is presented in Figure 1.

Because of the size of the bridge, the 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 channel with a sampling frequency of 100 Hz and the common subspace matrix of all setups was obtained with the PreGER approach described in [4] using data-driven SSI with the Unweighted Principal Component Algorithm.

As the computation time is also dependent on the constant $c \approx r/r_0$ (see Section 2.6), 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 ($c \approx 1$). The computation times for the system matrices up to a model order n_t from $\mathcal{O}_t$ on an Intel Core2 Duo CPU T8300 with 3.5 GByte in Matlab 7.10.0.499 are plotted in Figure 3. It can be seen that the solution of the least squares problem with the QR decomposition (see Algorithm 1) is more efficient than using the pseudoinverse from Equation (8), especially for the second case in Figure 3(b). However, using Algorithm 2 from Section 2.4 for multi-order system identification decreases computation time of the system matrices significantly, which can be improved further by using the iterative algorithm from Section 2.5.

An example of a stabilization diagram (see Section 3.1) containing the natural frequencies of the Z24 bridge at model orders $1, \ldots, 150$ is presented in Figure 4. 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. Going even higher than model order 150 still can improve identification results, although there only 10 modes to be identified in this case (see [7]).

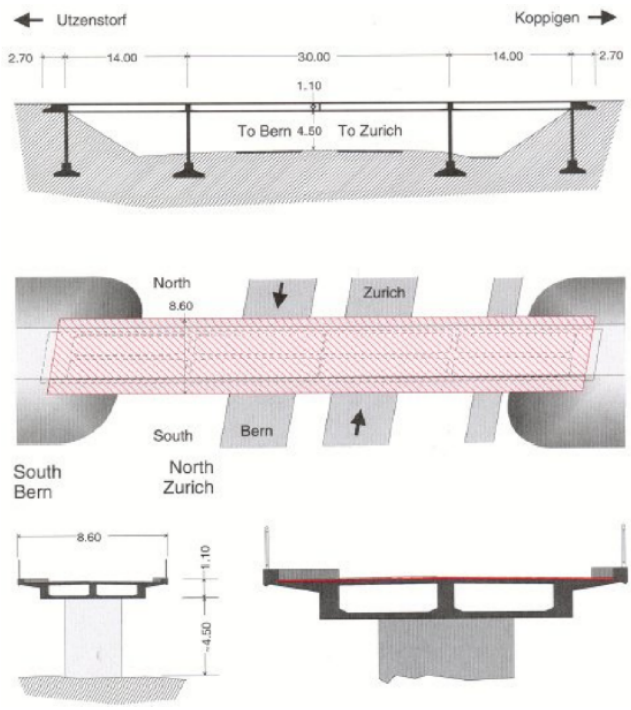
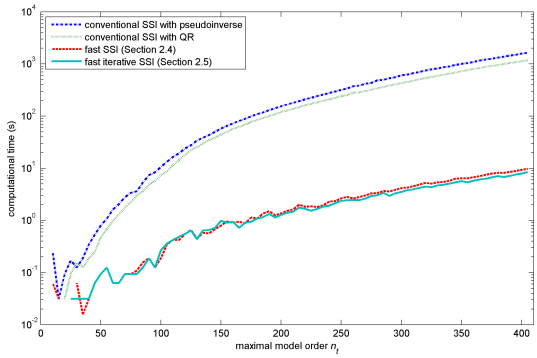


Figure 1: Schematic view of the Z24 bridge.

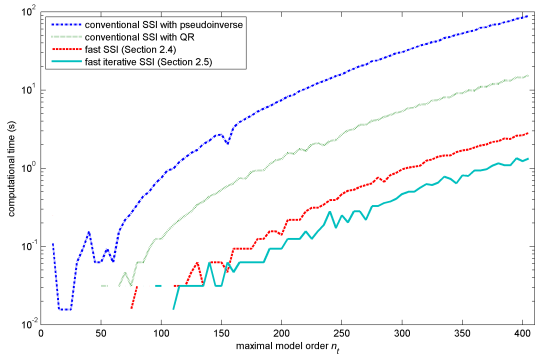
Figure 2: Photo of the Z24 bridge.

4 Conclusion

In this paper, a new algorithm was derived to efficiently compute the system matrices at multiple model orders in subspace based system identification. For this computation, the computational complexity was reduced from $n_{\max}^4$ to $n_{\max}^3$, where $n_{\max}$ is the maximal desired model order. The efficiency of the new



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



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

Figure 3: Computation times for multi-order computation of system matrices (set of A_j and C_j, $n_j = j = 1, \ldots, n_t$, computed from $\mathcal{O}_t$) with the algorithms from Sections 2.3, 2.4 and 2.5.

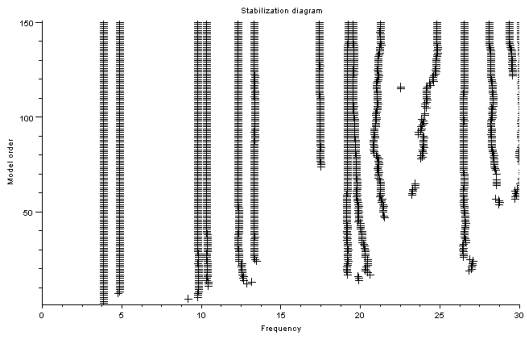


Figure 4: Stabilization diagram of Z24 bridge containing the identified natural frequencies at model orders $1, \ldots, 150$ using the fast iterative SSI from Section 2.5.

algorithm was shown on a real test case and computation time was reduced up to a factor of 100 and more. This fast algorithm can, e.g., be exploited in

online monitoring, where incoming data has to be processed quickly. Yet, the efficient computation of the eigenvalues and eigenvectors of the system matrices at different model orders remains.

Acknowledgment

The data for this research were obtained in the framework of the BRITE-EURAM Programme CT96 0277, SIMCES and provided on the SAMCO website by the SAMCO organization.

References

[1] M. Basseville, A. Benveniste, M. Goursat, L. Hermans, L. Mevel, and H. van der Auweraer. Output-only subspace-based structural identification: from theory to industrial testing practice. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):668–676, December 2001.

[2] A. Benveniste and J.-J. Fuchs. Single sample modal identification of a non-stationary stochastic process. *IEEE Transactions on Automatic Control*, AC-30(1):66–74, 1985.

[3] A. Benveniste and L. Mevel. Non-stationary consistency of subspace methods. *IEEE Transactions on Automatic Control*, AC-52(6):974–984, 2007.

[4] M. Döhler, P. Andersen, and L. Mevel. Data merging for multi-setup operational modal analysis with data-driven SSI. In *Proceedings of the 28th International Modal Analysis Conference (IMAC-XXVIII)*, Jacksonville, FL, USA, February 2010.

[5] G. Golub and C. van Loan. *Matrix computations.* Johns Hopkins University Press, 3rd edition, 1996.

[6] J. Maeck and G. De Roeck. Description of Z24 benchmark. *Mechanical Systems and Signal Processing*, 17(1):127–131, 2003.

[7] E. Parloo. *Application of frequency-domain system identification techniques in the field of operational modal analysis.* PhD thesis, Vrije Universiteit Brussel, 2003.

[8] B. Peeters and G. De Roeck. Reference-based stochastic subspace identification for output-only modal analysis. *Mechanical Systems and Signal Processing*, 13(6):855–878, November 1999.

[9] B. Peeters and G. De Roeck. Stochastic system identification for operational modal analysis: a review. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):659–667, 2001.

[10] P. van Overschee and B. De Moor. *Subspace Identification for Linear Systems: Theory, Implementation, Applications.* Kluwer, 1996.

Contents

1 Introduction **3**

2 Stochastic Subspace Identification (SSI) **3**
2.1 The General SSI Algorithm 3
2.2 Multi-Order SSI 5
2.3 Computation of the System Matrices 6
2.4 Fast Multi-Order Computation of the System Matrices 6
2.5 Fast Iterative Multi-Order Computation of the System Matrices . 8
2.6 Computational Complexities 9

3 Applications **10**
3.1 The Stabilization Diagram 10
3.2 Numerical Results 11

4 Conclusion **12**

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399