Efficient Multi-Order Uncertainty Computation for Stochastic Subspace Identification

Michael Döhler, Laurent Mevel

HAL Id: hal-00824105

https://hal.inria.fr/hal-00824105

Submitted on 20 Apr 2020

Efficient Multi-Order Uncertainty Computation for Stochastic Subspace Identification

Michael Döhler*, Laurent Mevel

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

Abstract

Stochastic Subspace Identification methods have been extensively used for the modal analysis of mechanical, civil or aeronautical structures for the last ten years. So-called stabilization diagrams are used, where modal parameters are estimated at successive model orders, leading to a graphical procedure where the physical modes of the system are extracted and separated from spurious modes. Recently an uncertainty computation scheme has been derived allowing the computation of uncertainty bounds for modal parameters at some given model order. In this paper, two problems are addressed. Firstly, a fast computation scheme is proposed reducing the computational burden of the uncertainty computation scheme by an order of magnitude in the model order compared to a direct implementation. Secondly, a new algorithm is proposed to derive efficiently the uncertainty bounds for the estimated modes at all model orders in the stabilization diagram. It is shown that this new algorithm is both computationally and memory efficient, reducing the computational burden by two orders of magnitude in the model order.

Keywords: System identification, (Operational) modal analysis, Subspace methods, Uncertainty bounds, Stabilization diagram

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 [1–4]. 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. 5–8]. Having done this allows in particular Finite Element Model updating and Structural Health Monitoring.

In Operational Modal Analysis, the true model order is hardly known and moreover spurious modes appear in the estimated models. Usually, an empirical multi-order estimation procedure is used, where the system is identified at multiple (over-specified) model orders in order to distinguish the true structural modes from spurious modes using the so-called stabilization diagrams [3, 9, 10]. There, the true structural modes are assumed to stabilize when the model order increases and thus can be separated from the spurious modes.

The estimated modal parameters are afflicted with statistical uncertainty for many reasons, e.g. finite number of data samples, undefined measurement noises, non-stationary excitations, etc. Then, the system identification algorithms do not yield the exact system matrices. The statistical uncertainty of the obtained modal parameters at a chosen model order can be computed from the uncertainty of the system matrices, which depends on the uncertainty in the data due to noise and turbulence. In [11], it has been shown how uncertainty bounds for modal parameters can be obtained in such a way. An analysis of this approach and an in depth literature review on the subject is found in

Nomenclature

		λ_i	eigenvalue of A
$\mathbb{R}, \mathbb{C}$	sets of real and complex numbers	ϕ_i, χ_i	right and left eigenvector of A
$\mathfrak{R}, \mathfrak{I}$	real, imaginary part	φ_i	mode shape
$\otimes$	Kronecker product	f_i, ξ_i	frequency, damping ratio
vec	column stacking vectorization operator	τ	sampling time step
$\dagger$	Moore-Penrose pseudoinverse	$\mathcal{H}$	subspace matrix of size $(p+1)r \times qr_0$
ΔX	first order perturbation on X	$\Sigma_{\mathcal{H}}$	covariance of vec($\mathcal{H}$)
$\mathcal{J}_{Y,X}$	sensitivity of vec(Y) wrt. vec(X)	T	factor of estimate $\widehat{\Sigma}_{\mathcal{H}} = TT^T$
$O(\cdot)$	Landau notation for complexity	u_j, v_j, σ_j	left, right singular vector and value of $\mathcal{H}$
A, C	system matrices	O	observability matrix
n	system order	$O^\uparrow, O^\downarrow$	O without last/first block row
n_m	maximal system order	S_1, S_2	selection matrices with $S_1 O = O^\uparrow, S_2 O = O^\downarrow$
n_d	number of modes	I_a	identity matrix of size $a \times a$
n_b	number of data blocks	$0_{a,b}$	zero matrix of size $a \times b$
r, r_0	number of sensors, reference sensors	$\mathcal{P}_{a,b}$	permutation, vec(X^T) $= \mathcal{P}_{a,b}$vec(X), $X \in \mathbb{R}^{a,b}$

[12], where also difficulties in developing confidence intervals on modal parameters from subspace identification are pointed out.

A direct and naive implementation of the uncertainty computation method in [11] is computationally taxing, especially when dealing with large sensor sets and a high model order. It has been derived for a fixed given model order and without giving implementation details. In practice, system identification results are needed at multiple model orders for the computation of the stabilization diagram. Then, redoing the uncertainty computations at several increasing model orders yields an expensive computational burden already for moderate system orders. In this paper, efficient implementations and new algorithms are proposed to solve this problem. Firstly, the algorithm in [11] is mathematically reformulated, resulting in an efficient implementation with a computational boost in one order of magnitude in the considered model order compared to the naive implementation. Secondly, a new algorithm is proposed for the computation of uncertainty bounds at multiple model orders corresponding to all modes in a stabilization diagram. It is shown how the computation of uncertainty bounds at any lower model orders can be done at a very low cost, when computations at the maximal desired model order are already done. This results in a decrease of the computational complexity of two orders of magnitude in the maximal model order. The new schemes are derived for the computation of uncertainty bounds of natural frequencies, damping ratios and mode shapes successively. The corresponding computational cost for each part of the computations of the desired modal parameters is addressed and compared to the naive implementation of the original algorithm in [11].

The paper is organized as follows. In Section 2, some preliminary modeling and the general subspace methods are given. In Section 3, the principle of the covariance computations is explained. In Section 4, notations and results of the uncertainty computations obtained in [11] are recalled and reformulated in Section 5 for a fast implementation. The computational burden of the implementations is analyzed and compared in Section 6. In Section 7 the new multi-order uncertainty computation algorithms are derived and their merits in terms of computational cost are discussed. A numerical example is given in Section 8, where the efficiency of the new algorithms is demonstrated.

2. Stochastic Subspace Identification (SSI)

2.1. Vibration modeling

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

$$\begin{cases} \mathcal{M}\ddot{x}(t) + C\dot{x}(t) + \mathcal{K}x(t) = \upsilon(t) \\ y(t) = Lx(t) \end{cases} \tag{1}$$

2

where t denotes continuous time; $\mathcal{M}, C, \mathcal{K} \in \mathbb{R}^{d \times d}$ are mass, damping, and stiffness matrices, respectively; the (high dimensional) vector $x(t)$ is the displacement vector of the d degrees of freedom of the structure; the external force $\upsilon(t)$ is unmeasured; measurements are collected in the (low dimensional) vector $y(t)$ and matrix $L \in \mathbb{R}^{r \times d}$ indicates which degrees of freedom are actually measured, i.e. the r sensor locations.

The parameters to be identified are the eigenvalues (or modes) μ_i and mode shapes ψ_i of system (1), which comprise the *modal parameters*, and are solutions of

$$(\mu_i^2 \mathcal{M} + \mu_i C + \mathcal{K})\Psi_i = 0, \quad \psi_i = L\Psi_i. \tag{2}$$

In model (1), the measured outputs are displacements. If $y(t)$ is the output of a set of accelerometers instead of displacement sensors, it is easy to show that the mode shapes are unchanged using the fact that $\ddot{x}(t) = -\mathcal{M}^{-1}C\dot{x}(t) - \mathcal{M}^{-1}\mathcal{K}x(t) + \mathcal{M}^{-1}\upsilon(t)$. Thus, this paper applies for different kinds of sensors (displacement sensors, accelerometers) and, in general, any kind of sensor measurements fitting some linear system modeling. The following developments are made for displacement sensors without any loss of generality.

Sampling model (1) at rate $1/\tau$ yields the discrete time state space model

$$\begin{cases} x_{k+1} = Ax_k + v_k \\ y_k = Cx_k + w_k, \end{cases} \tag{3}$$

with the states $x_k = \begin{bmatrix} x(k\tau)^T & \dot{x}(k\tau)^T \end{bmatrix}^T \in \mathbb{R}^n$, the outputs $y_k = y(k\tau) \in \mathbb{R}^r$ and the process and output noise v_k and w_k. 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, with

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

The external force $\upsilon(t)$ and thus the state noise (v_k) in model (3) can be non-stationary and colored noise [4, 13]. Define r_0 as the number of so-called *projection channels* or *reference sensors* with $r_0 \leq r$, which are a subset of the r sensors and can be used for reducing the size of the matrices in the identification process [3].

The eigenstructure (λ, φ) of system (3) is defined by the eigenvalues and eigenvectors of A and by C:

$$(A - \lambda_i I)\phi_i = 0, \quad \varphi_i = C\phi_i \tag{4}$$

The desired modal parameters in (2) are equivalently found in the eigenstructure (λ, φ) of (3) and it holds

$$e^{\mu_i \tau} = \lambda_i, \quad \psi_i = \varphi_i.$$

The modal frequencies f_i and damping coefficients ξ_i are recovered directly from the eigenvalues λ_i by

$$f_i = \frac{\sqrt{a_i^2 + b_i^2}}{2\pi\tau}, \quad \xi_i = \frac{100|b_i|}{\sqrt{a_i^2 + b_i^2}}, \tag{5}$$

where $a_i = |\arctan \mathfrak{I}(\lambda_i)/\mathfrak{R}(\lambda_i)|$ and $b_i = \ln|\lambda_i|$.

Thus, vibration analysis is stated as the problem of identifying the eigenstructure of a linear dynamic system. Parameters of interest are the natural frequencies f_i, damping ratios ξ_i and mode shapes φ_i.

2.2. The general SSI algorithm

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. [1–4] 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 (3).

Let the parameters p and q be given such that $pr \geq qr_0 \geq n$. A matrix $\mathcal{H}_{p+1,q} \in \mathbb{R}^{(p+1)r \times qr_0}$ is built from the output data 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

$$\mathcal{H}_{p+1,q} = O_{p+1} \mathcal{Z}_q \tag{6}$$

into the matrix of observability

$$O_{p+1} \stackrel{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^p \end{bmatrix}$$

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

The observability matrix O is obtained from a thin SVD of the matrix $\mathcal{H}$ and its truncation at the desired model order n:

$$\begin{aligned} \mathcal{H} &= U\Sigma V^T \\ &= \begin{bmatrix} U_1 & U_0 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_0^T \end{bmatrix}, \end{aligned} \tag{7}$$

$$O = U_1 \Sigma_1^{1/2}, \tag{8}$$

where

$$U_1 = \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix} \in \mathbb{R}^{(p+1)r \times n}, \quad V_1 = \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix} \in \mathbb{R}^{qr_0 \times n}, \quad \Sigma_1 = \text{diag}\{\sigma_1, \dots, \sigma_n\} \in \mathbb{R}^{n \times n}.$$

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

$$O^\uparrow A = O^\downarrow, \quad \text{where} \quad O^\uparrow \stackrel{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{p-1} \end{bmatrix}, \quad O^\downarrow \stackrel{\text{def}}{=} \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix}. \tag{9}$$

The least squares solution can be obtained from

$$A = O^{\uparrow \dagger} O^\downarrow, \tag{10}$$

where † denotes the Moore-Penrose pseudoinverse.

2.3. Examples of SSI Algorithms

In this section, examples of subspace matrices $\mathcal{H}$ are given that are related to two widely used output-only stochastic subspace identification algorithms. 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

$$\mathcal{Y}^+ = \frac{1}{\sqrt{N}} \begin{bmatrix} y_{q+1} & y_{q+2} & \vdots & y_{N+q} \\ y_{q+2} & y_{q+3} & \vdots & y_{N+q+1} \\ \vdots & \vdots & \vdots & \vdots \\ y_{q+p+1} & y_{q+p+2} & \vdots & y_{N+p+q} \end{bmatrix}, \quad \mathcal{Y}^- = \frac{1}{\sqrt{N}} \begin{bmatrix} y_q^{(\text{ref})} & y_{q+1}^{(\text{ref})} & \vdots & y_{N+q-1}^{(\text{ref})} \\ y_{q-1}^{(\text{ref})} & y_q^{(\text{ref})} & \vdots & y_{N+q-2}^{(\text{ref})} \\ \vdots & \vdots & \vdots & \vdots \\ y_1^{(\text{ref})} & y_2^{(\text{ref})} & \vdots & y_N^{(\text{ref})} \end{bmatrix}. \tag{11}$$

For *covariance-driven* SSI [1, 3, 4], let $R_i \stackrel{\text{def}}{=} \mathbf{E}(y_k y_{k-i}^{(\text{ref})T}) \in \mathbb{R}^{r \times r_0}$ and the block Hankel matrix

$$\mathcal{H}^{\text{cov}} \stackrel{\text{def}}{=} \begin{bmatrix} R_1 & R_2 & \dots & R_q \\ R_2 & R_3 & \dots & R_{q+1} \\ \vdots & \vdots & \ddots & \vdots \\ R_{p+1} & R_{p+2} & \dots & R_{p+q} \end{bmatrix} \tag{12}$$

be the theoretical output-correlation and subspace matrices, where $\mathbf{E}$ denotes the expectation operator. It enjoys the factorization property (6), where $\mathcal{Z}$ is the controllability matrix. The empirical correlations are estimated from $\widehat{R}_i = \frac{1}{N-i} \sum_{k=i+1}^{N} y_k y_{k-i}^{(\text{ref})T}$. Another estimate of the covariance-driven subspace matrix is

$$\widehat{\mathcal{H}}^{\text{cov}} = \mathcal{Y}^+(\mathcal{Y}^-)^T. \tag{13}$$

For *data-driven* SSI with the Unweighted Principal Component (UPC) algorithm [2–4], the estimate of the subspace matrix is defined as

$$\widehat{\mathcal{H}}^{\text{dat}} = \mathcal{Y}^+(\mathcal{Y}^-)^T (\mathcal{Y}^-(\mathcal{Y}^-)^T)^\dagger \mathcal{Y}^-.$$

where † denotes the pseudoinverse. Then, factorization property (6) holds asymptotically, where $\mathcal{Z}$ is the Kalman filter state matrix. With the partitioning of the thin LQ decomposition of

$$\begin{bmatrix} \mathcal{Y}^- \\ \mathcal{Y}^+ \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{14}$$

the relation $\widehat{\mathcal{H}}^{\text{dat}} = R_{21}Q_1$ follows, where $R_{21} \in \mathbb{R}^{(p+1)r \times qr_0}$ and $Q_1 \in \mathbb{R}^{qr_0 \times N}$. As Q_1 is an orthogonal matrix, the estimate of the observability matrix $\widehat{O}$ is obtained from R_{21} in the implementation of the algorithm, and $\widehat{\mathcal{H}}^{\text{dat,R}} \stackrel{\text{def}}{=} R_{21}$ is defined as a subspace matrix.

3. Preliminaries for covariance computations

In this section, the notations and basic principles of the covariance computations in the subsequent sections are introduced. In particular, the concept of perturbation to compute uncertainty bounds is defined. Furthermore, the computation of the covariance of the subspace matrix $\mathcal{H}$ is explained and related to uncertainty propagation to other variables.

3.1. Strategy of covariance computation and definitions

First, the notation of perturbation and uncertainties is explained and the strategy of the uncertainty computation introduced [11, 14, 15]. Starting from a covariance estimate $\widehat{\Sigma}_{\mathcal{H}} = \text{cov}(\text{vec}(\widehat{\mathcal{H}}))$ of the subspace matrix, which can be obtained easily from the measured data as explained in Section 3.2, the covariance of vector valued functions f of $\widehat{\mathcal{H}}$ can be calculated using the Taylor approximation [15]

$$f(\widehat{\mathcal{H}}) \approx f(\mathcal{H}) + \mathcal{J}_f \, \text{vec}(\widehat{\mathcal{H}} - \mathcal{H}) \quad \Rightarrow \quad \text{cov}(f(\widehat{\mathcal{H}})) \approx \mathcal{J}_f \widehat{\Sigma}_{\mathcal{H}} \mathcal{J}_f^T,$$

with the sensitivity $\mathcal{J}_f \stackrel{\text{def}}{=} \partial f(\mathcal{H})/\partial \text{vec}(\mathcal{H}) \approx \partial f(\widehat{\mathcal{H}})/\partial \text{vec}(\widehat{\mathcal{H}})$. Like this, the covariance of the subspace matrix can be propagated to the modal parameters, which are all functions of the subspace matrix. This propagation is done in several steps starting with the observability matrix, then to the system matrices, the eigenvalues and eigenvectors and finally to the modal parameters. The respective sensitivities of a quantity Y with respect to a quantity X is denoted as $\mathcal{J}_{Y,X}$. To derive the required sensitivities, perturbation theory is used [14, 15] and a first order perturbation of X is denoted by ΔX. Then, a perturbation on X and a perturbation on a function Y of X have the relation $\text{vec}(\Delta Y) = \mathcal{J}_{Y,X} \text{vec}(\Delta X)$, from where the sensitivity matrix can be obtained.

The following notation is used throughout the following sections. Let I_a be the $a \times a$ identity matrix and $0_{a,b}$ the $a \times b$ zero matrix. Define the selection matrices

$$S_1 \stackrel{\text{def}}{=} \begin{bmatrix} I_{pr} & 0_{pr,r} \end{bmatrix}, \quad S_2 \stackrel{\text{def}}{=} \begin{bmatrix} 0_{pr,r} & I_{pr} \end{bmatrix},$$

such that $S_1 O = O^\uparrow$, $S_2 O = O^\downarrow$. Furthermore, define the permutation matrix

$$\mathcal{P}_{a,b} \stackrel{\text{def}}{=} \sum_{k=1}^{a} \sum_{l=1}^{b} E_{k,l}^{a,b} \otimes E_{l,k}^{b,a}, \tag{15}$$

where $E_{k,l}^{a,b} \in \mathbb{R}^{a \times b}$ are matrices which are equal to 1 at entry (k, l) and zero elsewhere, and $\otimes$ denotes the Kronecker product [16]. In this context, the relation $\text{vec}(QRS) = (S^T \otimes R)\text{vec}(Q)$ is used for compatible matrices Q, R and S.

5

3.2. Estimating the covariance of the subspace matrix

For an estimation of the covariance $\Sigma_{\mathcal{H}}$, the data matrices $\mathcal{Y}^+$ and $\mathcal{Y}^-$ from (11) are split into n_b blocks and normalized with respect to their length, such that

$$\sqrt{N}\,\mathcal{Y}^+ = \sqrt{N_b}\begin{bmatrix} \mathcal{Y}_1^+ & \mathcal{Y}_2^+ & \dots & \mathcal{Y}_{n_b}^+ \end{bmatrix}, \quad \sqrt{N}\,\mathcal{Y}^- = \sqrt{N_b}\begin{bmatrix} \mathcal{Y}_1^- & \mathcal{Y}_2^- & \dots & \mathcal{Y}_{n_b}^- \end{bmatrix}, \tag{16}$$

where each block $\mathcal{Y}_j^+$ and $\mathcal{Y}_j^-$ may have the same length N_b, with $n_b \cdot N_b = N$ for simplicity. Each block may be long enough to assume statistical independence between the blocks. On each of theses blocks, the corresponding subspace matrix can be estimated and used for an empirical estimation of $\Sigma_{\mathcal{H}}$.

The covariance of the subspace matrix in the *covariance-driven case* follows easily from the covariance of the sample mean and was used e.g. in [11]. There, the covariance of the vectorized Hankel matrix (12) is obtained from the covariance $\Sigma_{\mathcal{R}}$ of the vectorized correlations $\mathcal{R} \stackrel{\text{def}}{=} \begin{bmatrix} R_1^T & R_2^T & \dots & R_{p+q}^T \end{bmatrix}^T \in \mathbb{R}^{(p+q)r \times r_0}$ as

$$\Sigma_{\mathcal{H}^{\text{cov}}} = S_3\,\Sigma_{\mathcal{R}}\,S_3^T \tag{17}$$

to reduce the size of the involved matrices, where $S_3 \in \mathbb{R}^{(p+1)rqr_0 \times (p+q)rr_0}$ is a selection matrix that fills the vectorized Hankel matrix $\mathcal{H}^{\text{cov}}$ with the vectorized correlations in $\mathcal{R}$ such that $\text{vec}(\mathcal{H}) = S_3\,\text{vec}(\mathcal{R})$ and is defined as

$$S_3 \stackrel{\text{def}}{=} \begin{bmatrix} S_{3,1}^T & S_{3,2}^T & \dots & S_{3,q}^T \end{bmatrix}^T \quad \text{where} \quad S_{3,k} \stackrel{\text{def}}{=} I_{r_0} \otimes \begin{bmatrix} 0_{(p+1)r,(k-1)r} & I_{(p+1)r} & 0_{(p+1)r,(q-k)r} \end{bmatrix}.$$

Let $\widehat{\mathcal{R}}_j$ be an estimate of $\mathcal{R}$ computed on the data blocks $\mathcal{Y}_j^+$ and $\mathcal{Y}_j^-$, then $\widehat{\mathcal{R}} = \frac{1}{n_b}\sum_{j=1}^{n_b}\widehat{\mathcal{R}}_j$ and a covariance estimate follows from

$$\widehat{\Sigma}_{\mathcal{H}^{\text{cov}}} = S_3\,\widehat{\Sigma}_{\mathcal{R}}\,S_3^T \quad \text{where} \quad \widehat{\Sigma}_{\mathcal{R}} = \frac{1}{n_b(n_b-1)}\sum_{j=1}^{n_b}\left(\text{vec}(\widehat{\mathcal{R}}_j) - \text{vec}(\widehat{\mathcal{R}})\right)\left(\text{vec}(\widehat{\mathcal{R}}_j) - \text{vec}(\widehat{\mathcal{R}})\right)^T. \tag{18}$$

Similarly, the estimate (13) of the covariance-driven subspace matrix can be used directly and a subspace matrix estimate $\widehat{\mathcal{H}}_j^{\text{cov}} \stackrel{\text{def}}{=} \mathcal{Y}_j^+\,(\mathcal{Y}_j^-)^T$ is computed on each data block from (16). Then, $\widehat{\mathcal{H}}^{\text{cov}} = \frac{1}{n_b}\sum_{j=1}^{n_b}\widehat{\mathcal{H}}_j^{\text{cov}}$ and a covariance estimate of the covariance-driven subspace matrix writes as

$$\widehat{\Sigma}_{\mathcal{H}^{\text{cov}}} = \frac{1}{n_b(n_b-1)}\sum_{j=1}^{n_b}\left(\text{vec}(\widehat{\mathcal{H}}_j^{\text{cov}}) - \text{vec}(\widehat{\mathcal{H}}^{\text{cov}})\right)\left(\text{vec}(\widehat{\mathcal{H}}_j^{\text{cov}}) - \text{vec}(\widehat{\mathcal{H}}^{\text{cov}})\right)^T. \tag{19}$$

For the covariance estimation in the *data-driven case*, the non-uniqueness of the LQ decomposition (14) has to be taken into account. Estimates for the UPC algorithm have been derived in [17].

4. Covariance estimates from [11]

In this section, the covariance estimation of the system matrices and the modal parameters using the covariance of the subspace matrix is recalled from [11]. Based on these results, a fast implementation is developed in the subsequent section.

4.1. Covariance estimation of the system matrices A and C

The covariance estimation of the matrices A and C is done in three steps: First, a perturbation $\Delta\mathcal{H}$ of the subspace matrix is propagated to a perturbation ΔO of the observability matrix, and second, a perturbation ΔO is propagated to perturbations ΔA and ΔC in the system matrices. Finally, the covariances of the vectorized system matrices are computed. In order to obtain ΔO, the sensitivities of the singular values and vectors in (7) are necessary. They have been derived in [15] as follows.

6

Lemma 1 ([15]). *Let σ_i, u_i and v_i be the ith singular value, left and right singular vector of the matrix $\mathcal{H} \in \mathbb{R}^{(p+1)r \times qr_0}$ in (7) and $\Delta\mathcal{H}$ a small perturbation on $\mathcal{H}$. Then it holds*

$$\Delta\sigma_i = (v_i \otimes u_i)^T \mathrm{vec}(\Delta\mathcal{H}), \ B_i \begin{bmatrix} \Delta u_i \\ \Delta v_i \end{bmatrix} = C_i \mathrm{vec}(\Delta\mathcal{H}),$$

where

$$B_i \stackrel{\text{def}}{=} \begin{bmatrix} I_{(p+1)r} & -\frac{1}{\sigma_i}\mathcal{H} \\ -\frac{1}{\sigma_i}\mathcal{H}^T & I_{qr_0} \end{bmatrix}, \quad C_i \stackrel{\text{def}}{=} \frac{1}{\sigma_i} \begin{bmatrix} v_i^T \otimes (I_{(p+1)r} - u_i u_i^T) \\ (u_i^T \otimes (I_{qr_0} - v_i v_i^T))\mathcal{P}_{(p+1)r,qr_0} \end{bmatrix}. \tag{20}$$

Using this result, the sensitivity of the observability matrix is derived in [11] based on Lemma 1 and the perturbation $\Delta O = U_1 \Delta\Sigma_1^{1/2} + \Delta U_1 \Sigma_1^{1/2}$. Note that $\mathcal{J}_{O,\mathcal{H}} = \mathcal{B} + C$ in [11].

Lemma 2 ([11]). *Let B_i and C_i be given in Lemma 1 for $i = 1, \dots, n$. Then,*

$$\mathrm{vec}(\Delta O) = \mathcal{J}_{O,\mathcal{H}} \, \mathrm{vec}(\Delta\mathcal{H})$$

where $\mathcal{J}_{O,\mathcal{H}} \in \mathbb{R}^{(p+1)rn \times (p+1)rqr_0}$ with

$$\mathcal{J}_{O,\mathcal{H}} \stackrel{\text{def}}{=} \frac{1}{2} \left(I_n \otimes U_1 \Sigma_1^{-1/2} \right) S_4 \begin{bmatrix} (v_1 \otimes u_1)^T \\ \vdots \\ (v_n \otimes u_n)^T \end{bmatrix} + \left(\Sigma_1^{1/2} \otimes \begin{bmatrix} I_{(p+1)r} & 0_{(p+1)r,qr_0} \end{bmatrix} \right) \begin{bmatrix} B_1^\dagger C_1 \\ \vdots \\ B_n^\dagger C_n \end{bmatrix}, \ S_4 \stackrel{\text{def}}{=} \sum_{k=1}^n E_{(k-1)n+k,k}^{n^2,n}, \tag{21}$$

The results of [11] on the sensitivity of the system matrices are collected in the following lemma.

Lemma 3 ([11]). *Let the system matrix A be obtained from O in (10) and C from the first block row of O. Then, a perturbation in O is propagated to A and C by*

$$\mathrm{vec}(\Delta A) = \mathcal{J}_{A,O} \, \mathrm{vec}(\Delta O), \quad \mathrm{vec}(\Delta C) = \mathcal{J}_{C,O} \, \mathrm{vec}(\Delta O),$$

where $\mathcal{J}_{A,O} \in \mathbb{R}^{n^2 \times (p+1)rn}$, $\mathcal{J}_{C,O} \in \mathbb{R}^{rn \times (p+1)rn}$, with

$$
\begin{aligned}
\mathcal{J}_{A,O} &\stackrel{\text{def}}{=} (I_n \otimes O^{\uparrow\dagger} S_2) - (A^T \otimes O^{\uparrow\dagger} S_1) + ((O^{\downarrow T} S_2 - A^T O^{\uparrow T} S_2) \otimes (O^{\uparrow T} O^{\uparrow})^{-1})\mathcal{P}_{(p+1)r,n}, \\
\mathcal{J}_{C,O} &\stackrel{\text{def}}{=} I_n \otimes \begin{bmatrix} I_r & 0_{r,pr} \end{bmatrix}.
\end{aligned} \tag{22}
$$

Proof. Using the product rule for the sensitivity of $A = O^{\uparrow\dagger} O^{\downarrow} = (O^{\uparrow T} O^{\uparrow})^{-1} O^{\uparrow T} O^{\downarrow}$ and Kronecker algebra leads to the assertion. Note that $\mathcal{J}_{A,O} = \mathcal{A}_1$ and $\mathcal{J}_{C,O} = \mathcal{A}_2$ in [11]. $\qquad\square$

Finally, the covariances of the vectorized system matrices A and C are obtained from

$$\Sigma_{A,C} \stackrel{\text{def}}{=} \mathrm{cov}\left(\begin{bmatrix} \mathrm{vec}(A) \\ \mathrm{vec}(C) \end{bmatrix} \right) = \begin{bmatrix} \mathcal{J}_{A,O} \\ \mathcal{J}_{C,O} \end{bmatrix} \mathcal{J}_{O,\mathcal{H}} \Sigma_{\mathcal{H}} \mathcal{J}_{O,\mathcal{H}}^T \begin{bmatrix} \mathcal{J}_{A,O}^T & \mathcal{J}_{C,O}^T \end{bmatrix}, \tag{23}$$

where $\Sigma_{\mathcal{H}} = S_3 \Sigma_{\mathcal{R}} S_3^T$ for covariance-driven subspace identification. In this case, an efficient computation of $\Sigma_{A,C}$ is obtained from

$$\Sigma_{A,C} = \mathcal{A} \Sigma_{\mathcal{R}} \mathcal{A}^T, \quad \text{where} \quad \mathcal{A} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{J}_{A,O} \\ \mathcal{J}_{C,O} \end{bmatrix} \mathcal{J}_{O,\mathcal{H}} S_3. \tag{24}$$

As the product with S_3 reduces the size of the involved matrices significantly, $(\mathcal{J}_{O,\mathcal{H}} S_3)$ is computed first.

4.2. Covariance estimation of the modal parameters

In [11], the sensitivity derivations for the eigenvalues and eigenvectors of a matrix and subsequently for the modal parameters are stated, based on derivations in [14, 15]. They are summarized in the following.

Lemma 4. *Let λ_i, ϕ_i and χ_i be the i-th eigenvalue, left eigenvector and right eigenvector of A with*

$$A\phi_i = \lambda_i\phi_i, \quad \chi_i^* A = \lambda_i\chi_i^*, \tag{25}$$

where * denotes the complex conjugate transpose. Then,

$$\Delta\lambda_i = \mathcal{J}_{\lambda_i,A}\mathrm{vec}(\Delta A), \quad \Delta\phi_i = \mathcal{J}_{\phi_i,A}\mathrm{vec}(\Delta A),$$

where $\mathcal{J}_{\lambda_i,A} \in \mathbb{C}^{1\times n^2}$, $\mathcal{J}_{\phi_i,A} \in \mathbb{C}^{n\times n^2}$, with

$$\mathcal{J}_{\lambda_i,A} \stackrel{\mathrm{def}}{=} \frac{1}{\chi_i^*\phi_i}(\phi_i^T \otimes \chi_i^*), \quad \mathcal{J}_{\phi_i,A} \stackrel{\mathrm{def}}{=} (\lambda_i I_n - A)^\dagger \left(\phi_i^T \otimes \left(I_n - \frac{\phi_i\chi_i^*}{\chi_i^*\phi_i}\right)\right). \tag{26}$$

Lemma 5. *Let λ_i and ϕ_i be the i-th eigenvalue and left eigenvector of A and $\tilde{\lambda}_i \stackrel{\mathrm{def}}{=} \ln(\lambda_i)/\tau = (b_i+a_i\mathrm{i})/\tau$ the eigenvalue of the corresponding continuous-time state transition matrix as in (5). Let furthermore the natural frequency f_i and the damping ratio ξ_i be given in (5), and suppose that the element k of the mode shape φ_i is scaled to unity, i.e. $\varphi_i = C\phi_i/(C\phi_i)_k$. Then,*

$$\Delta f_i = \mathcal{J}_{f_i,A}\mathrm{vec}(\Delta A), \quad \Delta\xi_i = \mathcal{J}_{\xi_i,A}\mathrm{vec}(\Delta A), \quad \Delta\varphi_i = \mathcal{J}_{\varphi_i,A,C}\begin{bmatrix}\mathrm{vec}(\Delta A)\\\mathrm{vec}(\Delta C)\end{bmatrix},$$

where $\mathcal{J}_{f_i,A}, \mathcal{J}_{\xi_i,A} \in \mathbb{R}^{1\times n^2}$, $\mathcal{J}_{\varphi_i,A,C} \in \mathbb{C}^{r\times(n^2+rn)}$, with

$$\begin{bmatrix}\mathcal{J}_{f_i,A}\\\mathcal{J}_{\xi_i,A}\end{bmatrix} \stackrel{\mathrm{def}}{=} \begin{bmatrix}\mathcal{J}_{f_i,\lambda_i}\\\mathcal{J}_{\xi_i,\lambda_i}\end{bmatrix}\begin{bmatrix}\mathfrak{R}(\mathcal{J}_{\lambda_i,A})\\\mathfrak{I}(\mathcal{J}_{\lambda_i,A})\end{bmatrix}, \quad \mathcal{J}_{\varphi_i,A,C} \stackrel{\mathrm{def}}{=} \frac{1}{(C\phi_i)_k}\left(I_r - \begin{bmatrix}0_{r,k-1} & \varphi_i & 0_{r,r-k}\end{bmatrix}\right)\begin{bmatrix}C\mathcal{J}_{\phi_i,A} & \phi_i^T \otimes I_r\end{bmatrix},$$

and

$$\begin{bmatrix}\mathcal{J}_{f_i,\lambda_i}\\\mathcal{J}_{\xi_i,\lambda_i}\end{bmatrix} \stackrel{\mathrm{def}}{=} \frac{1}{\tau|\lambda_i|^2|\tilde{\lambda}_i|}\begin{bmatrix}1/(2\pi) & 0\\0 & 100/|\tilde{\lambda}_i|^2\end{bmatrix}\begin{bmatrix}\mathfrak{R}(\tilde{\lambda}_i) & \mathfrak{I}(\tilde{\lambda}_i)\\-\mathfrak{I}(\tilde{\lambda}_i)^2 & \mathfrak{R}(\tilde{\lambda}_i)\mathfrak{I}(\tilde{\lambda}_i)\end{bmatrix}\begin{bmatrix}\mathfrak{R}(\lambda_i) & \mathfrak{I}(\lambda_i)\\-\mathfrak{I}(\lambda_i) & \mathfrak{R}(\lambda_i)\end{bmatrix} \in \mathbb{R}^{2\times2},$$

where the real and the imaginary part of a variable are denoted by $\mathfrak{R}(\cdot)$ and $\mathfrak{I}(\cdot)$, respectively.

Finally, the covariances of the modal parameters can be computed from

$$\mathrm{cov}\left(\begin{bmatrix}f_i\\\xi_i\end{bmatrix},\begin{bmatrix}f_j\\\xi_j\end{bmatrix}\right) = \begin{bmatrix}\mathcal{J}_{f_i,A} & 0_{1,rn}\\\mathcal{J}_{\xi_i,A} & 0_{1,rn}\end{bmatrix}\Sigma_{A,C}\begin{bmatrix}\mathcal{J}_{f_j,A} & 0_{1,rn}\\\mathcal{J}_{\xi_j,A} & 0_{1,rn}\end{bmatrix}^T,$$

$$\mathrm{cov}\left(\begin{bmatrix}\mathfrak{R}(\varphi_i)\\\mathfrak{I}(\varphi_i)\end{bmatrix},\begin{bmatrix}\mathfrak{R}(\varphi_j)\\\mathfrak{I}(\varphi_j)\end{bmatrix}\right) = \begin{bmatrix}\mathfrak{R}(\mathcal{J}_{\varphi_i,A,C})\\\mathfrak{I}(\mathcal{J}_{\varphi_i,A,C})\end{bmatrix}\Sigma_{A,C}\begin{bmatrix}\mathfrak{R}(\mathcal{J}_{\varphi_j,A,C})\\\mathfrak{I}(\mathcal{J}_{\varphi_j,A,C})\end{bmatrix}^T. \tag{27}$$

Let n_d be the number of modes that are selected for a covariance computation, where $n_d \leq n/2$. In order to obtain the uncertainty bounds of the modal parameters, their (co-)variances with themselves are of interest, i.e. the computation of

$$\mathrm{cov}\left(\begin{bmatrix}f_i\\\xi_i\end{bmatrix},\begin{bmatrix}f_i\\\xi_i\end{bmatrix}\right), \quad \mathrm{cov}\left(\begin{bmatrix}\mathfrak{R}(\varphi_i)\\\mathfrak{I}(\varphi_i)\end{bmatrix},\begin{bmatrix}\mathfrak{R}(\varphi_i)\\\mathfrak{I}(\varphi_i)\end{bmatrix}\right)$$

for $i = 1,\ldots,n_d$ in (27). The necessary steps for the covariance computation of the modal parameters are summarized in Algorithm 1.

5. A fast implementation for the computation of covariance estimates

In this section, a fast implementation for the covariance computation of the modal parameters is derived by a mathematical reformulation of the algorithm from the previous section. A detailed analysis of its computational complexity will follow in Section 6.

Algorithm 1 Covariance computation of modal parameters – naive implementation

---

Input: $\mathcal{H} = \mathcal{H}^{\text{cov}}$ in (12), $\Sigma_{\mathcal{R}}$ in (18) such that $\Sigma_{\mathcal{H}} = S_3 \Sigma_{\mathcal{R}} S_3^T$

    O, A, C from SSI in (7)–(10)

    λ_i, ϕ_i, χ_i and modal parameters f_i, ξ_i, φ_i for $i = 1, \ldots, n_d$ in (4)–(5), (25)

1: Computation of $\mathcal{J}_{O,\mathcal{H}} S_3$ using Lemma 2 by stacking $B_i^\dagger (C_i S_3)$
2: Computation of $\Sigma_{A,C}$ using Lemma 3 and (24): Compute $\mathcal{J}_{A,O}$, $\mathcal{J}_{C,O}$, $\mathcal{A}$ and finally $\Sigma_{A,C} = \mathcal{A} \Sigma_{\mathcal{R}} \mathcal{A}^T$
3: **for** $i = 1$ to n_d **do**
4:     Compute $\mathcal{J}_{\lambda_i,A}$ in Lemma 4, $\mathcal{J}_{f_i,A}$ and $\mathcal{J}_{\xi_i,A}$ in Lemma 5 and the covariance of f_i and ξ_i in (27)
5:     Compute $\mathcal{J}_{\phi_i,A}$ in Lemma 4, $\mathcal{J}_{\varphi_i,A,C}$ in Lemma 5 and the covariance of φ_i in (27)
6: **end for**

Output: Covariances of the modal parameters

5.1. Factorization of $\Sigma_{\mathcal{H}}$

In the covariance computations an estimate $\widehat{\Sigma}_{\mathcal{H}}$ of the covariance of the subspace matrix is used, which is a symmetric matrix. Consider a decomposition $\widehat{\Sigma}_{\mathcal{H}} = TT^T$, where the matrix T is a matrix square root of $\widehat{\Sigma}_{\mathcal{H}}$. This decomposition will be useful in the following sections for an efficient computation of the covariances of the system matrices or modal parameters. The matrix T can be obtained as follows. The estimate $\widehat{\Sigma}_{\mathcal{H}}$ has the shape

$$\widehat{\Sigma}_{\mathcal{H}} = \frac{1}{n_b(n_b - 1)} \sum_{j=1}^{n_b} (h_j - \bar{h})(h_j - \bar{h})^T,$$

using some samples h_j, $j = 1, \ldots, n_b$, where $\bar{h} \stackrel{\text{def}}{=} \frac{1}{n_b} \sum_{j=1}^{n_b} h_j$ (cf. (18), (19)). Then, defining

$$T \stackrel{\text{def}}{=} \frac{1}{\sqrt{n_b(n_b - 1)}} \begin{bmatrix} \tilde{h}_1 & \tilde{h}_2 & \ldots & \tilde{h}_{n_b} \end{bmatrix} \quad \text{with} \quad \tilde{h}_k \stackrel{\text{def}}{=} h_k - \bar{h},$$

it follows $\widehat{\Sigma}_{\mathcal{H}} = TT^T$, where $T \in \mathbb{R}^{(p+1)rqr_0 \times n_b}$. Thus, the matrix T is directly obtained from the samples of the subspace matrix on the data blocks without any additional computational cost. Moreover, the number of blocks n_b is often limited in practice due to available data length, with usually $n_b < (p + 1)rqr_0$, which means that matrix T has much less columns than $\widehat{\Sigma}_{\mathcal{H}}$.

5.2. Sensitivity derivation for the observability matrix

First, further properties of the permutation matrix $\mathcal{P}_{a,b}$ are stated, which will be useful for a simplification of the sensitivity computations.

Lemma 6. *The permutation matrix $\mathcal{P}_{a,b}$ defined in (15) writes in a simplified way as*

$$\mathcal{P}_{a,b} = \begin{bmatrix} I_a \otimes e_1 & I_a \otimes e_2 & \ldots & I_a \otimes e_b \end{bmatrix},$$

where $e_i \in \mathbb{R}^b$ are unit vectors with the entry 1 at position i and zero elsewhere. Let $G \in \mathbb{R}^{a \times b}$ and $H \in \mathbb{R}^{c \times d}$. Then,

(a) $\mathcal{P}_{a,b} \operatorname{vec} G = \operatorname{vec} G^T$,
(b) $(G \otimes H)\mathcal{P}_{b,d} = \mathcal{P}_{a,c}(H \otimes G)$.

Proof. The proof follows directly from the properties of the Kronecker product and can be found in [18, 19]. $\qquad\square$

The sensitivity computation on O is based on Lemma 2, where the sensitivities of the ith left and right singular vectors are obtained simultaneously as $B_i^\dagger C_i$. In the following, an alternative computation is proposed to compute these sensitivities separately, as only the *left* singular vectors are needed for O. The new computation makes use of the block structure of B_i. It avoids the costly computation of the pseudoinverse of B_i and uses the inversion of a smaller matrix instead. The merits of this result will be further discussed in Section 6.

Proposition 7. *Define*

$$K_i \stackrel{\text{def}}{=} \left(I_{qr_0} + \begin{bmatrix} 0_{qr_0-1,qr_0} \\ 2v_i^T \end{bmatrix} - \frac{\mathcal{H}^T \mathcal{H}}{\sigma_i^2} \right)^{-1}, \tag{28}$$

$$\widetilde{B}_{i,1} \stackrel{\text{def}}{=} \left[I_{(p+1)r} + \frac{\mathcal{H}}{\sigma_i} K_i \left(\frac{\mathcal{H}^T}{\sigma_i} - \begin{bmatrix} 0_{qr_0-1,(p+1)r} \\ u_i^T \end{bmatrix} \right) \quad \frac{\mathcal{H}}{\sigma_i} K_i \right], \tag{29}$$

$$\widetilde{B}_{i,2} \stackrel{\text{def}}{=} \left[K_i \left(\frac{\mathcal{H}^T}{\sigma_i} - \begin{bmatrix} 0_{qr_0-1,(p+1)r} \\ u_i^T \end{bmatrix} \right) \quad K_i \right], \tag{30}$$

$$\widetilde{C}_i \stackrel{\text{def}}{=} \frac{1}{\sigma_i} \begin{bmatrix} (I_{(p+1)r} - u_i u_i^T)(v_i^T \otimes I_{(p+1)r}) \\ (I_{qr_0} - v_i v_i^T)(I_{qr_0} \otimes u_i^T) \end{bmatrix}. \tag{31}$$

Then, a perturbation on $\mathcal{H}$ is propagated to the left and right singular vectors by

$$\text{vec}(\Delta U_1) = \begin{bmatrix} \widetilde{B}_{1,1}\widetilde{C}_1 \\ \vdots \\ \widetilde{B}_{n,1}\widetilde{C}_n \end{bmatrix} \text{vec}(\Delta\mathcal{H}), \quad \text{vec}(\Delta V_1) = \begin{bmatrix} \widetilde{B}_{1,2}\widetilde{C}_1 \\ \vdots \\ \widetilde{B}_{n,2}\widetilde{C}_n \end{bmatrix} \text{vec}(\Delta\mathcal{H}). \tag{32}$$

Proof. See Appendix A. □

With the sensitivities of the singular vectors, the sensitivity of the observability matrix is obtained efficiently in the following corollary.

Corollary 8. *Let $\widetilde{B}_{i,1}$ and $\widetilde{C}_i$ be given in Lemma 7 for $i = 1, \dots, n$. Then, a first order perturbation on $\mathcal{H}$ is propagated to the observability matrix by*

$$\text{vec}(\Delta O) = \widetilde{\mathcal{J}}_{O,\mathcal{H}} \, \text{vec}(\Delta\mathcal{H}),$$

where

$$\widetilde{\mathcal{J}}_{O,\mathcal{H}} \stackrel{\text{def}}{=} \frac{1}{2} \begin{bmatrix} \sigma_1^{-1/2} u_1 (v_1 \otimes u_1)^T \\ \vdots \\ \sigma_n^{-1/2} u_n (v_n \otimes u_n)^T \end{bmatrix} + \begin{bmatrix} \sigma_1^{1/2} \widetilde{B}_{1,1}\widetilde{C}_1 \\ \vdots \\ \sigma_n^{1/2} \widetilde{B}_{n,1}\widetilde{C}_n \end{bmatrix}.$$

Proof. As in Lemma 2, the first order perturbation on O is obtained from

$$\Delta O = \frac{1}{2} U_1 \Sigma_1^{-1/2} \Delta\Sigma_1 + \Delta U_1 \Sigma_1^{1/2}$$

$$= \frac{1}{2} \begin{bmatrix} u_1 \sigma_1^{-1/2} \Delta\sigma_1 & \dots & u_n \sigma_n^{-1/2} \Delta\sigma_n \end{bmatrix} + \begin{bmatrix} \sigma_1^{1/2} \Delta u_1 & \dots & \sigma_n^{1/2} \Delta u_n \end{bmatrix}.$$

Vectorizing this equation and plugging in $\Delta\sigma_i = (v_i \otimes u_i)^T \text{vec}(\Delta\mathcal{H})$ from Lemma 1 as well as $\text{vec}(\Delta U_1)$ from Proposition 7 leads to the assertion. □

Remark 9. *The product of $\widetilde{\mathcal{J}}_{O,\mathcal{H}}$ with matrix T, where $\Sigma_{\mathcal{H}} = TT^T$ (see Section 5.1), can be computed as follows. Each block line of the product $\widetilde{\mathcal{J}}_{O,\mathcal{H}} T$ writes as*

$$(\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_i = \frac{1}{2} \sigma_i^{-1/2} u_i (v_i \otimes u_i)^T T + \sigma_i^{1/2} \widetilde{B}_{i,1}\widetilde{C}_i T$$

$$= \frac{1}{2} \sigma_i^{-1/2} u_i v_i^T (I_{qr_0} \otimes u_i^T) T + \sigma_i^{1/2} \widetilde{B}_{i,1} \frac{1}{\sigma_i} \begin{bmatrix} (I_{(p+1)r} - u_i u_i^T)(v_i^T \otimes I_{(p+1)r}) T \\ (I_{qr_0} - v_i v_i^T)(I_{qr_0} \otimes u_i^T) T \end{bmatrix}.$$

Compute

$$T_{i,1} \stackrel{\text{def}}{=} (I_{qr_0} \otimes u_i^T) T, \quad T_{i,2} \stackrel{\text{def}}{=} (v_i^T \otimes I_{(p+1)r}) T, \tag{33}$$

and finally

$$(\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_i = \frac{1}{2} \sigma_i^{-1/2} u_i (v_i^T T_{i,1}) + \sigma_i^{-1/2} \widetilde{B}_{i,1} \begin{bmatrix} T_{i,2} - u_i(u_i^T T_{i,2}) \\ T_{i,1} - v_i(v_i^T T_{i,1}) \end{bmatrix}. \tag{34}$$

5.3. Sensitivity derivation for A

The sensitivity computation in Lemma 3, yielding $\text{vec}(\Delta A) = \mathcal{J}_{A,O} \text{vec}(\Delta O)$ for $A = O^{\uparrow\dagger} O^{\downarrow}$, can be written in the following way, which proves to be useful for computing the product $\mathcal{J}_{A,O} \widetilde{\mathcal{J}}_{O,\mathcal{H}}$ subsequently.

Proposition 10. *The sensitivity $\mathcal{J}_{A,O}$ can be written as*

$$\mathcal{J}_{A,O} = (I_n \otimes (O^{\uparrow T} O^{\uparrow})^{-1}) \left(-(A^T \otimes I_n)(\mathcal{P}_{n,n} + I_{n^2})(I_n \otimes O^{\uparrow T} S_1) + \mathcal{P}_{n,n}(I_n \otimes O^{\downarrow T} S_1) + (I_n \otimes O^{\uparrow T} S_2) \right).$$

Proof. See Appendix A. □

Suppose a decomposition $\Sigma_{\mathcal{H}} = TT^T$ of the covariance of the subspace matrix (see Section 5.1), where T has n_b columns. Then, the covariance of the vectorized system matrices can be obtained from

$$\Sigma_{A,C} = U_{A,C} U_{A,C}^T \quad \text{where} \quad U_{A,C} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{J}_{A,O} \\ \mathcal{J}_{C,O} \end{bmatrix} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T. \tag{35}$$

In order to compute $U_{A,C}$, the blockwise computation of $\widetilde{\mathcal{J}}_{O,\mathcal{H}} T$ in Remark 9 can be used for a memory-efficient implementation: Define $Q^{(1)}, Q^{(2)}, Q^{(3)} \in \mathbb{R}^{n^2 \times n_b}$ and $Q^{(4)} \in \mathbb{R}^{rn \times n_b}$ with

$$\begin{aligned} Q^{(1)} &\stackrel{\text{def}}{=} (I_n \otimes O^{\uparrow T} S_1) \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \quad Q^{(2)} \stackrel{\text{def}}{=} (I_n \otimes O^{\downarrow T} S_1) \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \\ Q^{(3)} &\stackrel{\text{def}}{=} (I_n \otimes O^{\uparrow T} S_2) \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \quad Q^{(4)} \stackrel{\text{def}}{=} (I_n \otimes [I_r \ 0_{r,pr}]) \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \end{aligned} \tag{36}$$

which thus can be computed as

$$Q^{(1)} = \begin{bmatrix} O^{\uparrow T} S_1 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_1 \\ \vdots \\ O^{\uparrow T} S_1 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_n \end{bmatrix}, \quad Q^{(2)} = \begin{bmatrix} O^{\downarrow T} S_1 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_1 \\ \vdots \\ O^{\downarrow T} S_1 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_n \end{bmatrix},$$
$$Q^{(3)} = \begin{bmatrix} O^{\uparrow T} S_2 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_1 \\ \vdots \\ O^{\uparrow T} S_2 (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_n \end{bmatrix}, \quad Q^{(4)} = \begin{bmatrix} [I_r \ 0_{r,pr}] (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_1 \\ \vdots \\ [I_r \ 0_{r,pr}] (\widetilde{\mathcal{J}}_{O,\mathcal{H}} T)_n \end{bmatrix}. \tag{37}$$

Then, from Proposition 10 follows

$$\mathcal{J}_{A,O} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T = (I_n \otimes (O^{\uparrow T} O^{\uparrow})^{-1}) \left(-(A^T \otimes I_n)(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + \mathcal{P}_{n,n}Q^{(2)} + Q^{(3)} \right), \tag{38}$$

and from the definition of $\mathcal{J}_{C,O}$ in (22) follows

$$\mathcal{J}_{C,O} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T = Q^{(4)}, \tag{39}$$

from which the covariance in (35) can be obtained.

5.4. Sensitivity derivation for modal parameters

Suppose the decomposition $\Sigma_{\mathcal{H}} = TT^T$ of the covariance of the subspace matrix (see Section 5.1). Instead of computing $\Sigma_{A,C}$ explicitly before computing the covariance of the modal parameters as in Section 4, it is suggested to compute the covariances based on the development

$$\text{cov}\left(\begin{bmatrix} f_i \\ \xi_i \end{bmatrix}, \begin{bmatrix} f_j \\ \xi_j \end{bmatrix} \right) = U_{f_i,\xi_i} U_{f_j,\xi_j}^T, \quad \text{cov}\left(\begin{bmatrix} \Re(\varphi_i) \\ \Im(\varphi_i) \end{bmatrix}, \begin{bmatrix} \Re(\varphi_j) \\ \Im(\varphi_j) \end{bmatrix} \right) = U_{\varphi_i} U_{\varphi_j}^T, \tag{40}$$

where

$$U_{f_i,\xi_i} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{J}_{f_i,A} \\ \mathcal{J}_{\xi_i,A} \end{bmatrix} \mathcal{J}_{A,O} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \quad U_{\varphi_i} \stackrel{\text{def}}{=} \begin{bmatrix} \Re(\mathcal{J}_{\varphi_i,A,C}) \\ \Im(\mathcal{J}_{\varphi_i,A,C}) \end{bmatrix} \begin{bmatrix} \mathcal{J}_{A,O} \\ \mathcal{J}_{C,O} \end{bmatrix} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T, \tag{41}$$

11

following from (27). First, consider the computation of U_{f_i,ξ_i}. From Lemma 5 follows

$$U_{f_i,\xi_i} = \begin{bmatrix} \mathcal{J}_{f_i,\lambda_i} \\ \mathcal{J}_{\xi_i,\lambda_i} \end{bmatrix} \begin{bmatrix} \mathfrak{R}(\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T) \\ \mathfrak{I}(\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T) \end{bmatrix},$$ (42)

where, using Lemma 4 and (38),

$$
\begin{aligned}
\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T &= \frac{1}{\chi_i^*\phi_i}(\phi_i^T \otimes \chi_i^*)(I_n \otimes (O^{\uparrow T}O^{\uparrow})^{-1})\left(-(A^T \otimes I_n)(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + \mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}\right) \\
&= -\frac{1}{\chi_i^*\phi_i}(\phi_i^T A^T \otimes \chi_i^*(O^{\uparrow T}O^{\uparrow})^{-1})(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} \\
&\quad + \frac{1}{\chi_i^*\phi_i}(\phi_i^T \otimes \chi_i^*(O^{\uparrow T}O^{\uparrow})^{-1})(\mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}) \\
&= \frac{1}{\chi_i^*\phi_i}(\phi_i^T \otimes \chi_i^*(O^{\uparrow T}O^{\uparrow})^{-1})\left(-\lambda_i(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + \mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}\right) \\
&= \frac{1}{\chi_i^*\phi_i}\chi_i^*(O^{\uparrow T}O^{\uparrow})^{-1}Q_i,
\end{aligned}
$$ (43)

where

$$Q_i \stackrel{\text{def}}{=} (\phi_i^T \otimes I_n)\left(-\lambda_i(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + \mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}\right).$$ (44)

Thus, once $Q^{(1)}$, $Q^{(2)}$ and $Q^{(3)}$ are computed in a preprocessing step, the product $\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T$ and thus U_{f_i,ξ_i} can be obtained by using only permutations ($\mathcal{P}_{n,n}$) or matrix products with small or sparse matrices.

Second, consider the computation of U_{φ_i}. From Lemma 5 follows

$$\mathcal{J}_{\varphi_i,A,C}\begin{bmatrix} \mathcal{J}_{A,O} \\ \mathcal{J}_{C,O} \end{bmatrix}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T = \frac{1}{(C\phi_i)_k}\left(I_r - \begin{bmatrix} 0_{r,k-1} & \varphi_i & 0_{r,r-k} \end{bmatrix}\right)\left(C\mathcal{J}_{\phi_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T + (\phi_i^T \otimes I_r)\mathcal{J}_{C,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T\right),$$ (45)

where, using Lemma 4, (38), (39) and (44),

$$
\begin{aligned}
\mathcal{J}_{\phi_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T &= (\lambda_i I_n - A)^\dagger\left(\phi_i^T \otimes \left(I_n - \frac{\phi_i\chi_i^*}{\chi_i^*\phi_i}\right)\right)(I_n \otimes (O^{\uparrow T}O^{\uparrow})^{-1}) \\
&\quad \cdot \left(-(A^T \otimes I_n)(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + \mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}\right) \\
&= (\lambda_i I_n - A)^\dagger\left(I_n - \frac{\phi_i\chi_i^*}{\chi_i^*\phi_i}\right)(O^{\uparrow T}O^{\uparrow})^{-1} \\
&\quad \cdot \left(-(\phi_i^T A^T \otimes I_n)(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)} + (\phi_i^T \otimes I_n)(\mathcal{P}_{n,n}Q^{(2)} + Q^{(3)})\right) \\
&= (\lambda_i I_n - A)^\dagger\left(I_n - \frac{\phi_i\chi_i^*}{\chi_i^*\phi_i}\right)(O^{\uparrow T}O^{\uparrow})^{-1}Q_i,
\end{aligned}
$$ (46)

and $\mathcal{J}_{C,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T = Q^{(4)}$. Then, U_{φ_i} from (41) is obtained by stacking the real and imaginary part of (45). This provides a computation for U_{φ_i}, where no large Kronecker products are used anymore.

The necessary steps for the covariance computation of the modal parameters with the modified steps from this section are summarized in Algorithm 2.

6. Computational efficiency of Algorithms 1 and 2

In order to compare the performance of the two implementations for covariance computations, their computational complexity is evaluated. The following assumptions are made for a comparison:

- The subspace matrix $\mathcal{H}$ is of size $(p + 1)r \times qr_0$ and in practice it is set $p + 1 = q$ [20].

12

Algorithm 2 Covariance computation of modal parameters – fast implementation
___
Input: $\mathcal{H}, T$ such that $\Sigma_{\mathcal{H}} = TT^T$ in Section 5.1
  O, A, C from SSI in (7)–(10)
  $\lambda_i, \phi_i, \chi_i$ and modal parameters f_i, ξ_i, φ_i for $i = 1, \ldots, n_d$ in (4)–(5), (25)

1: Computation of $Q^{(1)}, \ldots, Q^{(4)}$ in (36): For $i = 1, \ldots, n$ compute $\widetilde{B}_{i,1}$ in (29), $T_{i,1}, T_{i,2}, (\widetilde{\mathcal{J}}_{O,\mathcal{H}}T)_i$ in Remark 9 and
   the i-th block line of $Q^{(1)}, \ldots, Q^{(4)}$ in (37)
2: Computation of $(O^{\uparrow T}O^{\uparrow})^{-1}$, $(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)}$ and the sum $\mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}$
3: **for** $i = 1$ to n_d **do**
4:   Compute Q_i in (44), $\mathcal{J}_{f_i,\lambda_i}, \mathcal{J}_{\xi_i,\lambda_i}$ in Lemma 5, $\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T$ in (43), U_{f_i,ξ_i} in (42) and the covariance of f_i
     and ξ_i in (40)
5:   Compute $\mathcal{J}_{\phi_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T$ in (46), U_{φ_i} from (41) and (45), and the covariance of φ_i in (40)
6: **end for**
___
Output: Covariances of the modal parameters
___

- The maximal possible model order that can be selected is then $n_m \overset{\text{def}}{=} qr_0$ and the model order $n = n_m$ is chosen
  for the computations.

- A linear relation is assumed between the model order n_m and the number of data blocks n_b for the covariance
  computation. Moreover, n_b is bounded in practice due to limited length of the available data.

Furthermore, define the parameter $c \overset{\text{def}}{=} pr/n_m \approx r/r_0 \geq 1$, which is independent of p, q and n_m.

For both the naive implementation in Algorithm 1 as well as the fast implementation in Algorithm 2, the uncertainty quantification of the modal parameters is divided into four steps: Two preprocessing steps, one step for the covariance computation of frequencies and damping ratios and one step for the mode shapes. A detailed analysis of the computational cost of each of these step is made in Appendix B in Table B.8. These results are summarized in Table 1.

Table 1: Comparison of complexities for Algorithms 1 and 2.

Step in Algorithm	#	Complexity in Algorithm 1	Complexity in Algorithm 2
Preprocessing	Step 1	$O(c^3 n_m^4)$	$O(c n_m^4)$
	Step 2	$O(c^2 n_m^5)$	$O(c n_m^3)$
Covariance of f_i, ξ_i's	Step 4	$O(n_d n_m^4)$	$O(n_d n_m^3)$
Covariance of φ_i's	Step 5	$O(n_d n_m^4)$	$O(n_d n_m^3)$

Comparing the computation of Algorithms 1 and 2, the following can be observed:

- In Algorithm 2, no explicit computation of $\widehat{\Sigma}_{\mathcal{H}}$ or $\widehat{\Sigma}_{\mathcal{R}}$ is necessary, as the vectorized subspace matrix estimates
  on each data block are directly used in matrix T, which yields $\widehat{\Sigma}_{\mathcal{H}} = TT^T$. This factorization is an advantage for
  all subspace algorithms, while the factorization $\Sigma_{\mathcal{H}} = S_3\Sigma_{\mathcal{R}}S_3^T$ to reduce the size of the matrices in Algorithm
  1 works only for covariance-driven subspace identification.

- Algorithm 2 is linear in c, compared to c^3 in Algorithm 1. This is especially an advantage when using reference
  sensors, as $c \approx r/r_0$, and significantly increases speed of the first preprocessing step.

- The second preprocessing step is two orders lower in n_m in Algorithm 2 compared to the same step in Algorithm 1.

- The covariance computation of frequencies and damping ratios as well of the mode shapes is one order lower
  in n_m in Algorithm 2 compared to Algorithm 1.

- The overall complexity with respect to n_m of Algorithm 2 is $O(n_m^4)$, compared to $O(n_m^5)$ for Algorithm 1.

Furthermore, memory requirement is lower for Algorithm 2: While matrices of size $n_m^2 \times c n_m^2$ (matrix $\mathcal{J}_{A,O}$) are processed in Algorithm 1, the maximal size of involved matrices in the Algorithm 2 is $c n_m^2 \times n_b$ (matrix T). After the first preprocessing step, the maximal size of the involved matrices is further reduced to $n_m^2 \times n_b$.

Note that the computational complexity of the modal parameter estimation without their covariance computation is $O(n_m^3)$ [21] and the maximal size of the involved matrices is $c n_m \times n_m$ once $\mathcal{H}$ is computed.

7. Multi-order SSI and covariance computation

In Operational Modal Analysis (OMA), the eigenstructure of mechanical, civil and aeronautical structures is identified from output-only data under ambient excitation. Usually, system identification results are obtained at multiple model orders and are plotted in the well established *stabilization diagrams* [3, 9, 10]. In this section it is shown how the covariance computation at multiple model orders can be performed efficiently based on Algorithm 2 in $O(n_m^4)$ operations, while an approach based on Algorithm 1 needs $O(n_m^6)$ operations.

7.1. The stabilization diagram and multi-order SSI

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 appear 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 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 stabilization diagrams. 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. Another criterion for the selection of the modes in a stabilization diagram can be the uncertainty bounds on the estimated modal parameters, as those with a low uncertainty are in general more likely to be the desired physical modes.

In order to obtain the modes at multiple model orders in a stabilization diagram, system identification is done for models (3) at different model orders n by truncating the SVD (7) at the respective model orders. For simplicity, all the model orders $n = 1, 2, 3, \ldots, n_m$ are considered in the following.

The following notation for specifying these different model orders is used. Let $O_n \in \mathbb{R}^{(p+1)r \times n}$, $A_n \in \mathbb{R}^{n \times n}$ and $C_n \in \mathbb{R}^{r \times n}$ be the observability, state transition and observation matrix at model order n, respectively. Let furthermore be $O_n^\uparrow$ and $O_n^\downarrow$ the first respective last p block rows of O_n, analogously to the definition in (9).

Then, system identification at these multiple model orders consists in the following steps: First, the observability matrix O_{n_m} is computed at the maximal desired model order n_m from (7)–(8). Then, the observability matrix O_n at order n consists of the first n columns of O_{n_m} due to the truncation in (7), i.e.

$$O_n = O_{n_m} \begin{bmatrix} I_n \\ 0_{n_m-n,n} \end{bmatrix}. \tag{47}$$

The matrix A_n is the solution of least squares problem (9) for O_n and C_n is the first block row of O_n, from which the modal parameters are computed at model order n in (4)–(5).

7.2. Multi-order covariance computation

7.2.1. Multi-order covariance computation based on Algorithm 1

In order to compute the covariance of the modal parameters at multiple model orders, the relation (47) can be used. Instead of computing the sensitivity $\mathcal{J}_{O_n,\mathcal{H}}$ at each model order n, it can be computed once at the maximal model order n_m and from (47) it follows

$$\mathcal{J}_{O_n,\mathcal{H}} = \left([I_n \ \ 0_{n,n_m-n}] \otimes I_{(p+1)r} \right) \mathcal{J}_{O_{n_m},\mathcal{H}}. \tag{48}$$

14

Thus, the first preprocessing step, namely the computation of $\mathcal{J}_{O_n,\mathcal{H}}S_3$ in Step 1 in Algorithm 1, is done at the maximum model order $n = n_m$. At each model order $n = 1, 2, \ldots, n_m$, the matrix $\mathcal{J}_{O_n,\mathcal{H}}S_3$ is obtained by selecting the first $(p+1)rn$ rows of $\mathcal{J}_{O_{n_m},\mathcal{H}}S_3$ according to (48). Then, the covariance Σ_{A_n,C_n} needs to be computed in order to obtain the covariance of the modal parameters at these model orders. Finally, the multi-order covariance computation using Algorithm 1 is now summarized in Algorithm 3.

Algorithm 3 Multi-order covariance computation of modal parameters based on Algorithm 1

Input: $\mathcal{H} = \mathcal{H}^{\text{cov}}$ in (12), $\Sigma_{\mathcal{R}}$ in (18) such that $\Sigma_{\mathcal{H}} = S_3 \Sigma_{\mathcal{R}} S_3^T$

  O_{n_m} from SSI at maximal model order n_m in (7)–(10)

  System matrices A_n, C_n and values $\lambda_i^{(n)}$, $\phi_i^{(n)}$, $\chi_i^{(n)}$, $f_i^{(n)}$, $\xi_i^{(n)}$, $\varphi_i^{(n)}$ for $i = 1, \ldots, n_d^{(n)}$ identified at model orders $n = 1, \ldots, n_m$

  1: Computation of $\mathcal{J}_{O_{n_m},\mathcal{H}}S_3$ in Step 1 of Algorithm 1 at model order n_m
  2: **for** $n = 1$ to n_m **do**
  3:   Select O_n from first n columns of O_{n_m} and $\mathcal{J}_{O_n,\mathcal{H}}S_3$ from first $(p+1)rn$ rows of $\mathcal{J}_{O_{n_m},\mathcal{H}}S_3$
  4:   Execute Steps 2–6 of Algorithm 1 at model order n
  5: **end for**

Output: Covariances of the modal parameters at model orders $n = 1, \ldots, n_m$

Assume that $n_t \approx 2crn_m$ and $\sum_{n=1}^{n_m} n^3 \approx \frac{1}{4}n_m^4$. Then, considering that for each model order $n = 1, 2, \ldots, n_m$, the computation of Σ_{A_n,C_n} takes more than $2cn_t n_m n^3$ operations (see Table B.8), the total complexity of Algorithm 3 amounts to $O(c^2 n_m^6)$.

7.2.2. Multi-order covariance computation based on Algorithm 2

In Algorithm 2, the matrices $Q^{(1)}$, $Q^{(2)}$, $Q^{(3)}$ and $Q^{(4)}$ are computed in Step 1. They are defined at the model order n as

$$Q_n^{(1)} \stackrel{\text{def}}{=} (I_n \otimes O_n^{\uparrow T} S_1)\widetilde{\mathcal{J}}_{O_n,\mathcal{H}}\, T, \qquad Q_n^{(2)} \stackrel{\text{def}}{=} (I_n \otimes O_n^{\downarrow T} S_1)\widetilde{\mathcal{J}}_{O_n,\mathcal{H}}\, T,$$
$$Q_n^{(3)} \stackrel{\text{def}}{=} (I_n \otimes O_n^{\uparrow T} S_2)\widetilde{\mathcal{J}}_{O_n,\mathcal{H}}\, T, \qquad Q_n^{(4)} \stackrel{\text{def}}{=} (I_n \otimes [I_r\ \ 0_{r,pr}])\widetilde{\mathcal{J}}_{O_n,\mathcal{H}}\, T, \tag{49}$$

as in (36). Then, they can be easily related to their counterparts at the maximum model order n_m.

Proposition 11. *The matrices $Q_n^{(1)}$, $Q_n^{(2)}$, $Q_n^{(3)}$ and $Q_n^{(4)}$ can be selected from the respective matrices at maximum model order n_m as*

$$Q_n^{(1)} = S_{4,n}Q_{n_m}^{(1)}, \quad Q_n^{(2)} = S_{4,n}Q_{n_m}^{(2)}, \quad Q_n^{(3)} = S_{4,n}Q_{n_m}^{(3)}, \quad Q_n^{(4)} = [I_{rn}\ \ 0_{rn,r(n_m-n)}]Q_{n_m}^{(4)},$$

where $S_{4,n}$ is the selection matrix

$$S_{4,n} \stackrel{\text{def}}{=} [I_n\ \ 0_{n,n_m-n}] \otimes [I_n\ \ 0_{n,n_m-n}].$$

Proof. It holds

$$\widetilde{\mathcal{J}}_{O_n,\mathcal{H}} = \left([I_n\ \ 0_{n,n_m-n}] \otimes I_{(p+1)r}\right)\widetilde{\mathcal{J}}_{O_{n_m},\mathcal{H}}.$$

analogous to (48). Plugging this and (47) into the definition of $Q_n^{(1)}$ in (49) leads to

$$Q_n^{(1)} = \left(I_n \otimes \left(O_{n_m}^{\uparrow} \begin{bmatrix} I_n \\ 0_{n_m-n,n} \end{bmatrix}\right)^T S_1\right)\left([I_n\ \ 0_{n,n_m-n}] \otimes I_{(p+1)r}\right)\widetilde{\mathcal{J}}_{O_{n_m},\mathcal{H}}\, T$$
$$= \left([I_n\ \ 0_{n,n_m-n}] \otimes [I_n\ \ 0_{n,n_m-n}]O_{n_m}^{\uparrow T} S_1\right)\widetilde{\mathcal{J}}_{O_{n_m},\mathcal{H}}\, T$$
$$= S_{4,n}Q_{n_m}^{(1)}.$$

The relations for $Q_n^{(2)}$ and $Q_n^{(3)}$ follow analogously. The relation for $Q_n^{(4)}$ follows directly from (37). $\qquad\square$

Algorithm 4 Fast multi-order covariance computation of modal parameters based on Algorithm 2

Input: $\mathcal{H}$, T such that $\Sigma_{\mathcal{H}} = TT^T$ in Section 5.1

    O_{n_m} from SSI at maximal model order n_m in (7)–(10)

    System matrices A_n, C_n and values $\lambda_i^{(n)}$, $\phi_i^{(n)}$, $\chi_i^{(n)}$, $f_i^{(n)}$, $\xi_i^{(n)}$, $\varphi_i^{(n)}$ for $i = 1, \ldots, n_d^{(n)}$ identified at model orders $n = 1, \ldots, n_m$

  1: Computation of $Q_{n_m}^{(1)}, \ldots, Q_{n_m}^{(4)}$ in Step 1 of Algorithm 2 at model order n_m

  2: **for** $n = 1$ to n_m **do**

  3:     Select O_n from first n columns of O_{n_m} and $Q_n^{(1)}, \ldots, Q_n^{(4)}$ from the rows of $Q_{n_m}^{(1)}, \ldots, Q_{n_m}^{(4)}$ in Proposition 11

  4:     Execute Steps 2–6 of Algorithm 2 at model order n

  5: **end for**

Output: Covariances of the modal parameters at model orders $n = 1, \ldots, n_m$

Thus, the matrices $Q_{n_m}^{(1)}$, $Q_{n_m}^{(2)}$, $Q_{n_m}^{(3)}$ and $Q_{n_m}^{(4)}$ can be computed once at the maximum model order n_m. The respective matrices at inferior model orders n are selected from their rows. From these matrices $Q_n^{(1)}$, $Q_n^{(2)}$, $Q_n^{(3)}$ and $Q_n^{(4)}$ at each model order n, the covariances of the modal parameters are computed as outlined in Algorithm 2. Finally, the resulting multi-order covariance computation is now summarized in Algorithm 4.

Assume again a linear relation between n_m and n_b and $\sum_{n=1}^{n_m} n^3 \approx 1/4\, n_m^4$, $\sum_{n=1}^{n_m} n^2 \approx 1/3\, n_m^3$. The computation of $Q_{n_m}^{(1)}$, $Q_{n_m}^{(2)}$, $Q_{n_m}^{(3)}$ and $Q_{n_m}^{(4)}$ at the maximum model order n_m in Algorithm 4 takes $O(cn_m^4)$ operations (see Table 1). The remaining computations in Algorithm 4 correspond to Steps 2–6 of Algorithm 2 at model orders $n = 1, \ldots, n_m$, which take a total of $O((c + n_d)n_m^4)$ operations (see Table B.8). Considering only n_m as a parameter and the number of selected modes n_d for the covariance computation as a constant (being usually small with respect to n_m), the entire covariance computation of the modal parameters at model orders $n = 1, \ldots, n_m$ with Algorithm 4 thus takes $O(n_m^4)$ operations. Note that this is of the same order as the underlying Algorithm 2, computing only at model order $n = n_m$. Also, it is of the same order as the modal parameter estimation only without the covariance computation at multiple model orders [21]. The computational complexities of the derived algorithms are summarized in Table 2.

Table 2: Computational complexities of Algorithms 1–4 in model order n_m.

	Direct computation	Fast computation
Algorithms 1 and 2	$O(n_m^5)$	$O(n_m^4)$
Multi-order Algorithms 3 and 4	$O(n_m^6)$	$O(n_m^4)$

8. Application: Modal analysis of Z24 Bridge

In this section, the fast covariance computation of the modal parameters is applied to a practical test case from vibration analysis, the Z24 Bridge [22]. The computation times are compared between Algorithms 1 and 2 for the covariance computation at one model order, as well as between Algorithms 3 and 4 for the covariance computation for a whole stabilization diagram. A detailed performance analysis of the algorithms of this paper is discussed.

8.1. The test case

The proposed algorithms have been applied on vibrational data of the Z24 Bridge [22], a benchmark of the COST F3 European network. The analyzed data is the response of the bridge to ambient excitation (traffic under the bridge) measured in 154 points, mainly in the vertical and at some points also the transverse and lateral directions, and sampled at 100 Hz. Altogether, nine data sets have been recorded, each covering a part of the whole structure. In this study, only the first data set is used to demonstrate the computations. It contains data from 33 sensors and each signal contains 65,535 samples.

8.2. Comparison of the computation times

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. The parameters are set as follows:

- Three reference sensors are used ($r_0 = 3$) from $r = 33$ available sensors, leading to $c \approx 11$.

- The covariance-driven 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 [20].

- The maximal model order is set to $n_m = qr_0$.

- The number of data blocks used for the covariance computation is $n_b = 200$.

To compare the performance of the algorithms, the modal analysis and covariance computation is done for different maximal model orders n_m by choosing $q = 2, \ldots, 70$ for the subspace matrix.

The computation times of the different algorithms from this paper for the covariance computations of the modal parameters are presented in Figure 1 for different maximal model orders n_m together with the computation times of the point estimates of the modal parameters only based on [21]. In Figure 1(a), the modal parameters and their covariance were computed at model order $n = n_m$ with Algorithms 1 and 2. The accumulated computation time of the modal parameters and their covariance at model orders $n = 1, 2, \ldots, n_m$ in a stabilization diagram is presented in Figure 1(b), using Algorithms 3 and 4.
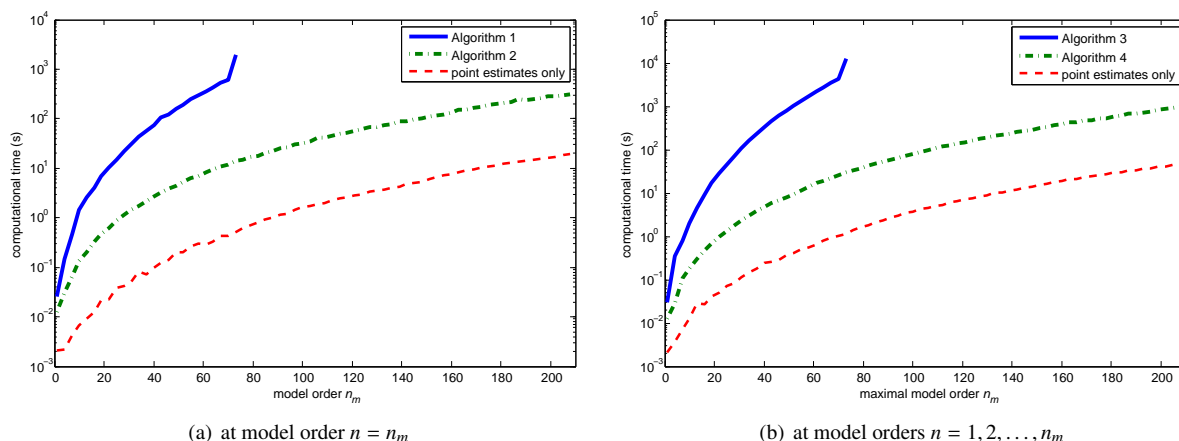

(a) at model order $n = n_m$ (b) at model orders $n = 1, 2, \ldots, n_m$

Figure 1: Computation times for covariance computation of modal parameters and their point estimates only for different maximal model orders n_m (log scale).

In both cases it can be seen that the new fast algorithms (Algorithm 2 at one model order and Algorithm 4 at multiple model orders) outperform clearly their direct counterparts (Algorithms 1 and 3). A comparison of the computation time at a selected model order is made in Table 3.

Table 3: Computation times for covariance computation of modal parameters and their point estimates only at model order $n_m = 72$.

	Direct computation	Fast computation	Point estimates only
Algorithms 1 and 2	597 s	11.9 s	0.5 s
Multi-order Algorithms 3 and 4	4342 s	25.4 s	1.1 s

In order to compare the extra cost that is necessary for the multi-order computation (model orders $n = 1, 2, \ldots, n_m$) compared to the computation at model order $n = n_m$ only, the ratio between the computation times of the respective

algorithms is shown in Figure 2(a). The theoretical ratio between Algorithms 3 and 1 is $O(n_m^6)/O(n_m^5)$ and thus linear, which is confirmed in Figure 2(a). The theoretical ratio between the fast Algorithms 4 and 2 is $O(n_m^4)/O(n_m^4)$ and thus constant. In Figure 2(a) it can indeed be seen that the latter ratio agrees with the theory for large n_m, where a slope can be seen in the ratio that seems to converge to a constant around 3. This means that the additional cost for computing uncertainty bounds at all orders $n = 1, 2, \ldots, n_m - 1$ is only around twice the cost that is necessary for the computation at $n = n_m$. Note that the computational cost for Algorithm 4 depends also on the parameter n_d, which might slightly increase when n_m increases and which could be a reason for the observed slope in the ratio. Still, the observed ratio appears to be less than linear.

In Figure 2(b), the ratio between the computation times of the original and the fast algorithms is shown, both for Algorithms 1 and 2 as well as their multi-order variants (Algorithms 3 and 4). Apparently, the former ratio grows quadratically and the latter linearly, confirming their theoretical computational complexities.
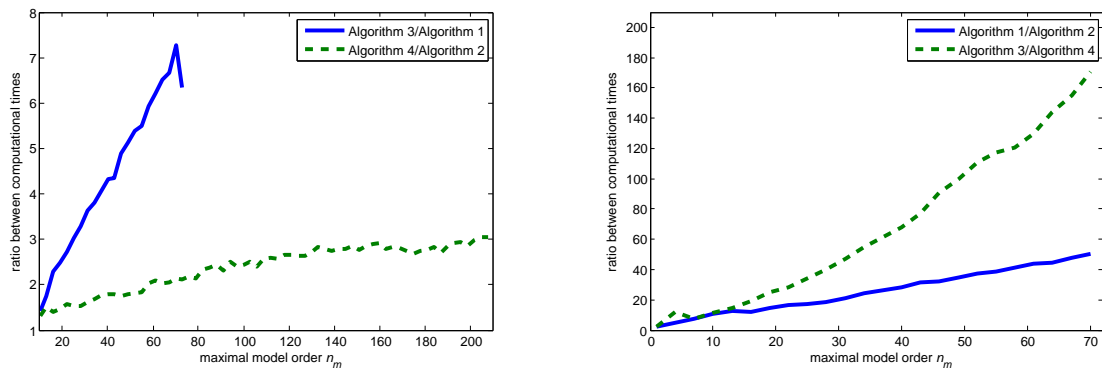
(a) multi-order algorithms compared to their underlying algorithms at one model order

(b) original algorithms compared to their fast versions

Figure 2: Ratio between computation times of derived algorithms at different maximal model orders n_m.

Algorithms 1 and 3 ran into memory problems at model order $n_m = 75$ in the Matlab implementation of the algorithm, which slowed down the computation, and memory was not sufficient for $n_m > 75$, although 16 GByte of memory were used. Algorithms 2 and 4 did not experience memory problems until model order 200 and beyond.

8.3. Results of multi-order covariance computation of modal parameters

With Algorithms 3 and 4, the covariances of the modal parameters in a stabilization diagram can be computed. As both algorithms are mathematically equivalent, the computed covariances are identical, where Algorithm 4 is faster and can go higher in model order. Figure 3 shows the stabilization diagram, where the standard deviations $\pm\sigma$ of the natural frequencies are plotted as horizontal bars. Figure 3(a) shows the diagram, where only modes with damping values in the range $[0.1\%, 6\%]$ are plotted, and in Figure 3(b) additionally a threshold on the standard deviations of 1.5% of the frequency value was used. Such a threshold on the standard deviations significantly clears up the diagram, which hence can be used as a stabilization criterion of the identified modes. Note that other stabilization criteria (e.g. thresholds on damping values, variability between the modes, etc.) should be used before the uncertainty computation to speed up the computation.

Figure 8.3 shows the zoom on the first mode in the diagram, where it can be seen that the mode stabilizes in the diagram after reaching a certain model order ($n = 53$ in this case). The standard deviations of the frequencies beyond this order are the smallest and slightly larger between orders 18–36 (where the frequency is also stable), while they are considerably larger for the remaining model orders. Note however, that the mode obviously stabilizes only after model order 53 and a low standard deviation of a mode does not necessarily mean a good estimation quality as the mode can be biased due to a wrong model order selection as in this case between orders 18–36.

18

(a) full diagram

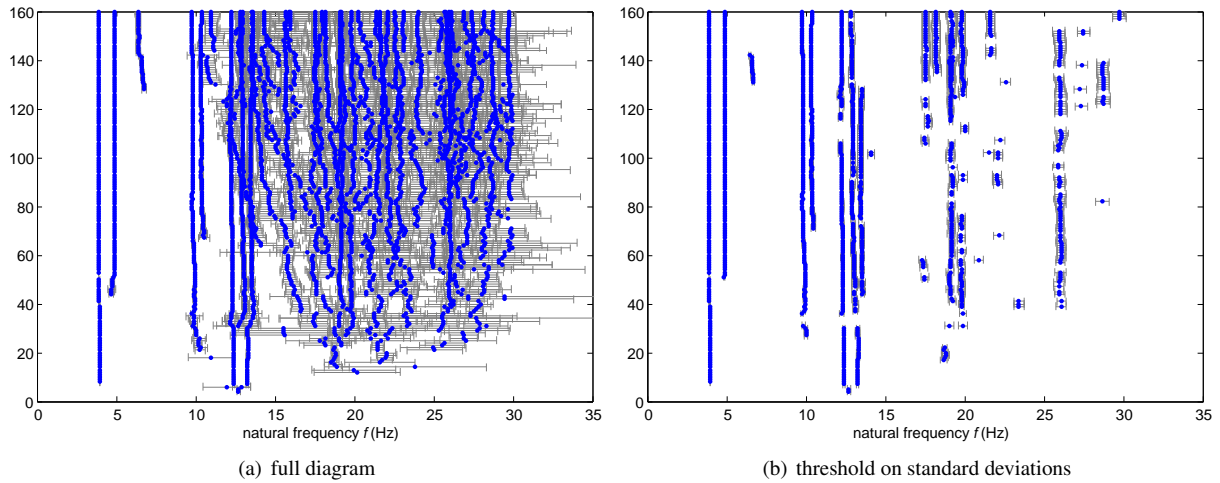(b) threshold on standard deviations

Figure 3: Stabilization diagram with standard deviations of frequencies (horizontal bars) for maximal model order $n_m = 160$.
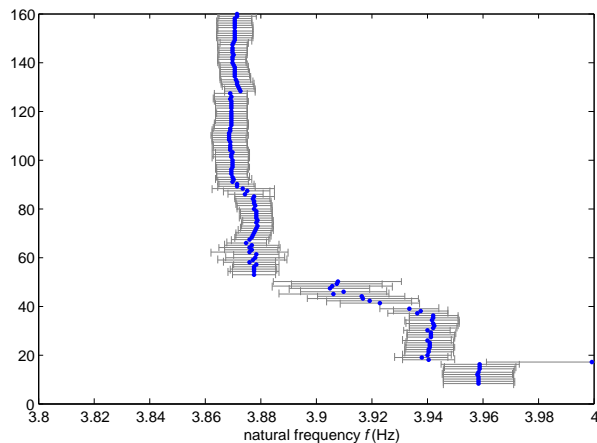

Figure 4: Zoom at the first mode in the stabilization diagram.

9. Discussion and conclusion

In this paper, a fast implementation of the algorithm of [11] and an extension of this algorithm were derived to efficiently compute the uncertainty bounds for system matrices A and C and associated modal parameters at multiple model orders in stochastic subspace-based system identification (SSI). The validity of the uncertainty computation procedure has been proved in [11], so the objective was not to discuss the quality and bias related to the covariance estimates but to focus on achieving best computational and algorithmic efficiency to be able to apply the procedure to realistic large scale problems.

With the fast implementation (Algorithm 2) a significant increase in computational efficiency could be achieved compared to a direct and naive implementation of the uncertainty computation algorithm of [11] (Algorithm 1), which we still optimized in a sense that sparse matrices are used whenever possible as well as efficient matrix products leading to smaller matrices, in order to allow a fair comparison. Furthermore, the new implementation lead to less memory requirement, allowing the uncertainty computation at higher model orders. An analysis of the computational complexity showed a decrease from $O(n_m^5)$ to $O(n_m^4)$, where n_m is the model order, going along with a decrease of the complexity in other parameters.

As the stabilization diagram is a standard tool in Operational Modal Analysis, the uncertainty computation for all its elements at multiple model orders is of basic interest and was not considered in [11]. In this paper, a new algorithm (Algorithm 4) was derived for this task and compared to Algorithm 3, which is the multi-order extension of the direct implementation in Algorithm 1. The computational complexity was reduced from $O(n_m^6)$ to $O(n_m^4)$, where n_m is the maximal desired model order in the diagram. This was possible due to a mathematical reformulation of the computation, which takes advantage of the multi-order structure of the problem. Also, the efficiency of the algorithms was shown with a numerical example. The multi-order uncertainty computation has moreover the practical advantage that spurious modes can be neglected by setting thresholds on the obtained uncertainty bounds.

Note that the comparison of the runtime of the algorithms is independent from the actually used test case, but depends only of the size of the involved matrices. Therefore, the achieved numerical results are generic and repeatable.

These algorithms can especially be applied for Operational Modal Analysis of mechanical, civil or aeronautical structures. Their efficiency was shown on a real test case, where the computation time was reduced up to a factor of over 100 and computations at higher model orders were possible due to lower memory requirement. Indeed, the computation time lies only between seconds and a few minutes even at high model orders, which makes it possible to use these fast algorithms e.g. in online Structural Health Monitoring, where incoming data has to be processed quickly.

Future work contains an in depth evaluation of the computed standard deviations as a tool to remove spurious modes and to use the information for the computation of a best mode fit in the stabilization diagram.

Acknowledgment

Appendix A. Proofs of Section 5

Proof of Proposition 7. Lemma 1 states that Δu_i and Δv_i satisfy the condition

$$B_i \begin{bmatrix} \Delta u_i \\ \Delta v_i \end{bmatrix} = C_i \, \text{vec}(\Delta \mathcal{H}),$$

where B_i and C_i are defined in (20) and where B_i is not of full rank. As the singular vectors u_i and v_i, $i = 1, \ldots, n$, are orthonormal, they satisfy $u_i^T u_i = 1$ and $v_i^T v_i = 1$. It follows $u_i^T \Delta u_i = 0$ and $v_i^T \Delta v_i = 0$ and thus the condition $u_i^T \Delta u_i + v_i^T \Delta v_i = 0$ can be added to the system of equations for Δu_i and Δv_i, which was also suggested in [15]. If $\mathcal{H}$ is full column rank, this leads to a system of full column rank. Without loss of generality, this condition can be added to the last row of the matrices and a solution writes as

$$\begin{bmatrix} \Delta u_i \\ \Delta v_i \end{bmatrix} = \left(B_i + \begin{bmatrix} 0_{c,(p+1)r} & 0_{c,qr_0} \\ u_i^T & v_i^T \end{bmatrix} \right)^{-1} C_i \, \text{vec}(\Delta \mathcal{H}) = \begin{bmatrix} O & P \\ Q & R \end{bmatrix}^{-1} C_i \, \text{vec}(\Delta \mathcal{H}), \tag{A.1}$$

where $c = (p+1)r + qr_0 - 1$ and

$$O \stackrel{\text{def}}{=} I_{(p+1)r}, \quad P \stackrel{\text{def}}{=} -\frac{\mathcal{H}}{\sigma_i}, \quad Q \stackrel{\text{def}}{=} -\frac{\mathcal{H}^T}{\sigma_i} + \begin{bmatrix} 0_{qr_0-1,(p+1)r} \\ u_i^T \end{bmatrix}, \quad R \stackrel{\text{def}}{=} I_{qr_0} + \begin{bmatrix} 0_{qr_0-1,qr_0} \\ v_i^T \end{bmatrix}.$$

The block matrix inversion formula is used, which writes as

$$\begin{bmatrix} O & P \\ Q & R \end{bmatrix}^{-1} = \begin{bmatrix} O^{-1} + O^{-1}PS_O^{-1}QO^{-1} & -O^{-1}PS_O^{-1} \\ -S_O^{-1}QO^{-1} & S_O^{-1} \end{bmatrix},$$

where $S_O \stackrel{\text{def}}{=} R - QO^{-1}P$ and thus

$$S_O = I_{qr_0} + \begin{bmatrix} 0_{qr_0-1,qr_0} \\ v_i^T \end{bmatrix} - \left(-\frac{\mathcal{H}^T}{\sigma_i} + \begin{bmatrix} 0_{qr_0-1,(p+1)r} \\ u_i^T \end{bmatrix} \right) \left(-\frac{\mathcal{H}}{\sigma_i} \right) = I_{qr_0} + \begin{bmatrix} 0_{qr_0-1,qr_0} \\ v_i^T \end{bmatrix} + \begin{bmatrix} 0_{qr_0-1,qr_0} \\ u_i^T \mathcal{H}/\sigma_i \end{bmatrix} - \frac{\mathcal{H}^T \mathcal{H}}{\sigma_i^2}.$$

20

From $u_i^T \mathcal{H}/\sigma_i = v_i^T$ and with K_i defined in (28) follows $K_i = S_O^{-1}$. Then, the block matrix inversion yields with (29) and (30)

$$\left(B_i + \begin{bmatrix} 0_{c,(p+1)r} & 0_{c,qr_0} \\ u_i^T & v_i^T \end{bmatrix} \right)^{-1} = \begin{bmatrix} \widetilde{B}_{i,1} \\ \widetilde{B}_{i,2} \end{bmatrix}. \tag{A.2}$$

Finally, $\widetilde{C}_i = C_i$ with C_i in (20) and $\widetilde{C}_i$ in (31) is shown. From Lemma 6(b) follows

$$(u_i^T \otimes (I_{qr_0} - v_i v_i^T))\mathcal{P}_{(p+1)r,qr_0} = \mathcal{P}_{1,qr_0}((I_{qr_0} - v_i v_i^T) \otimes u_i^T),$$

where $\mathcal{P}_{1,qr_0} = I_{qr_0}$ using again Lemma 6. Then, $\widetilde{C}_i = C_i$ follows from

$$(I_{qr_0} - v_i v_i^T) \otimes u_i^T = (I_{qr_0} - v_i v_i^T) \cdot I_{qr_0} \otimes 1 \cdot u_i^T = ((I_{qr_0} - v_i v_i^T) \otimes 1)(I_{qr_0} \otimes u_i^T),$$
$$v_i^T \otimes (I_{(p+1)r} - u_i u_i^T) = 1 \cdot v_i^T \otimes (I_{(p+1)r} - u_i u_i^T) \cdot I_{(p+1)r} = (1 \otimes (I_{(p+1)r} - u_i u_i^T))(v_i^T \otimes I_{(p+1)r}),$$

and the assertion follows together with (A.1) and (A.2). $\qquad\square$

Proof of Proposition 10. With the relation $\Delta X^{-1} = -X^{-1} \Delta X X^{-1}$, a perturbation of $A = O^{\uparrow\dagger} O^{\downarrow} = (O^{\uparrow T} O^{\uparrow})^{-1} O^{\uparrow T} O^{\downarrow}$ writes as

$$\begin{aligned}
\Delta A &= \Delta (O^{\uparrow T} O^{\uparrow})^{-1} O^{\uparrow T} O^{\downarrow} + (O^{\uparrow T} O^{\uparrow})^{-1} \Delta (O^{\uparrow T} O^{\downarrow}) \\
&= -(O^{\uparrow T} O^{\uparrow})^{-1} \Delta (O^{\uparrow T} O^{\uparrow})(O^{\uparrow T} O^{\uparrow})^{-1} O^{\uparrow T} O^{\downarrow} + (O^{\uparrow T} O^{\uparrow})^{-1} \Delta (O^{\uparrow T} O^{\downarrow}) \\
&= (O^{\uparrow T} O^{\uparrow})^{-1} \left(-\Delta (O^{\uparrow T} O^{\uparrow}) A + \Delta (O^{\uparrow T} O^{\downarrow}) \right).
\end{aligned} \tag{A.3}$$

Developing $\Delta(O^{\uparrow T} O^{\uparrow})$ and $\Delta(O^{\uparrow T} O^{\downarrow})$ separately leads to

$$\begin{aligned}
\Delta(O^{\uparrow T} O^{\uparrow}) &= \Delta O^{\uparrow T} O^{\uparrow} + O^{\uparrow T} \Delta O^{\uparrow} \\
&= \Delta O^T S_1^T O^{\uparrow} + O^{\uparrow T} S_1 \Delta O, \\
\mathrm{vec}(\Delta(O^{\uparrow T} O^{\uparrow})) &= (O^{\uparrow T} S_1 \otimes I_n)\mathrm{vec}(\Delta O^T) + (I_n \otimes O^{\uparrow T} S_1)\mathrm{vec}(\Delta O),
\end{aligned}$$

where $\mathrm{vec}(\Delta O^T) = \mathcal{P}_{(p+1)r,n}\mathrm{vec}(\Delta O)$ and $(O^{\uparrow T} S_1 \otimes I_n)\mathcal{P}_{(p+1)r,n} = \mathcal{P}_{n,n}(I_n \otimes O^{\uparrow T} S_1)$ according to Lemma 6. Then,

$$\mathrm{vec}(\Delta(O^{\uparrow T} O^{\uparrow})) = \mathcal{P}_{n,n}(I_n \otimes O^{\uparrow T} S_1)\mathrm{vec}(\Delta O) + (I_n \otimes O^{\uparrow T} S_1)\mathrm{vec}(\Delta O) = (\mathcal{P}_{n,n} + I_{n^2})(I_n \otimes O^{\uparrow T} S_1)\mathrm{vec}(\Delta O)$$

and with a similar development, it follows

$$\mathrm{vec}(\Delta(O^{\uparrow T} O^{\downarrow})) = \mathcal{P}_{n,n}(I_n \otimes O^{\downarrow T} S_1)\mathrm{vec}(\Delta O) + (I_n \otimes O^{\uparrow T} S_2)\mathrm{vec}(\Delta O).$$

Then, the assertion follows from vectorizing (A.3). $\qquad\square$

Appendix B. Computational evaluation of Algorithms 1 and 2

Appendix B.1. Preliminaries

In order to compare the performance of different algorithms for covariance computations, their number of floating point operations (flops, multiplications plus summations) needs to be evaluated. For simplicity, no difference is made between real-valued and complex-valued operations. Insignificant terms are neglected when counting the flops.

The following conventions are used. The subspace matrix $\mathcal{H}$ is often of size $(p + 1)r \times qr_0$ and in practice it is set $p + 1 = q$ [20]. A maximal possible model order is then $n_m \overset{\text{def}}{=} qr_0$. Define the parameter $c \overset{\text{def}}{=} pr/n_m \approx r/r_0$, which is independent of p, q and n_m and defines the ratio of the dimensions of $O^{\uparrow}$ and $O^{\downarrow}$. Consider the SVD of $\mathcal{H}$ in (7) and the observability matrix O in (8) at a model order $n \le n_m$ be given. For a first evaluation, $n = n_m$ can be assumed.

The number of flops of some basic numerical operations is given in Table B.4.

Table B.4: Flop count of some basic numerical operations [14].

Operation	Matrix Sizes	Flops
$F = U\Sigma V^T$	$F, U \in \mathbb{R}^{a \times b}, \Sigma, V \in \mathbb{R}^{b \times b}$	$14ab^2 + 8b^3$
FG	$F \in \mathbb{R}^{a \times b}, G \in \mathbb{R}^{b \times c}$	$2abc$
$F \otimes G$	$F \in \mathbb{R}^{a \times b}, G \in \mathbb{R}^{c \times d}$	$abcd$
$(I_c \otimes F)G$	$F \in \mathbb{R}^{a \times b}, G \in \mathbb{R}^{bc \times d}$	$2abcd$

Appendix B.2. Algorithm 1

Appendix B.2.1. Covariance computation of the system matrices A and C

The computation of $\Sigma_{A,C}$ is done in Steps 1 and 2 in Algorithm 1. In Step 1, the product $\mathcal{J}_{O,\mathcal{H}} S_3$ is computed, where S_3 is of size $(p+1)rqr_0 \times (p+q)rr_0$. Defining $n_t \stackrel{\text{def}}{=} (p+q)rr_0$ and using the notation of Section Appendix B.1, its size is approximated by $cn_m^2 \times n_t$. The significant operations for the computation of $\mathcal{J}_{O,\mathcal{H}} S_3$ are summarized in Table B.5.

Table B.5: Flop count for computation of $\mathcal{J}_{O,\mathcal{H}} S_3$ for Step 1 of Algorithm 1.

Reference	Operation	Flops
Equ. (20)	$C_i S_3$	$(c^2 + c)n_m^3$
Equ. (20), (21)	SVD of B_i	$22(c+1)^3 n_m^3$
Equ. (21)	$B_i^\dagger (C_i S_3)$	$4(c+1)^2 n_t n_m^2$
	$\sum_{i=1}^n \approx 22c^3 n_m^3 n + 4c^2 n_t n_m^2 n$	

In Step 2, the matrices $\mathcal{J}_{A,O}$, $\mathcal{J}_{A,O}$ and their product with $\mathcal{J}_{O,\mathcal{H}} S_3$ are computed in order to obtain $\mathcal{A}$ in (24). Note that $\mathcal{J}_{A,O}$ is of size $n^2 \times cn_m n$. In this computation, the significant operation is the computation of the product of $\mathcal{J}_{A,O}$ with $(\mathcal{J}_{O,\mathcal{H}} S_3)$, which takes $2cn_t n_m n^3$ flops. Finally, the product $\Sigma_{\mathcal{H}} = \mathcal{A}\Sigma_{\mathcal{R}}\mathcal{A}^T$ in (24) is computed with around $2n_t^2 n^2 + 2n_t n^4$ flops, amounting to a total of

$$22c^3 n_m^3 n + 4c^2 n_t n_m^2 n + 2cn_t n_m n^3 + 2n_t^2 n^2 + 2n_t n^4 \text{ flops}$$

for the computation of the covariance of the vectorized system matrices A and C.

Appendix B.2.2. Covariance computation of modal parameters

Let $\Sigma_{A,C}$ be given in (24). Then, the covariance computation of the modal parameters corresponds to Steps 4 and 5 in Algorithm 1. For each mode i, the sensitivities $\mathcal{J}_{f_i,A}$, $\mathcal{J}_{\xi_i,A}$ and $\mathcal{J}_{\varphi_i,A,C}$ need to be computed to get the covariances of the modal parameters in (27). The significant operations are the computation of the product in (27), which takes around $4n_d n^4$ flops for the n_d natural frequencies and damping ratios, and $4rn_d n^4$ flops for n_d mode shapes, when $\Sigma_{A,C}$ is known.

Appendix B.3. Algorithm 2

Appendix B.3.1. Covariance computation of the system matrices A and C

The new computation of the covariance of the vectorized system matrices $\Sigma_{A,C}$ is analyzed in this section, although it is not recommended to compute $\Sigma_{A,C}$ explicitly in order to obtain the covariances of the modal parameters. The matrix T for a covariance estimate $\widehat{\Sigma}_{\mathcal{H}} = TT^T$ is supposed to be given (see Section 5.1), which is of size $cn_m^2 \times n_b$. In the first step, the matrices $Q^{(1)}, Q^{(2)}, Q^{(3)} \in \mathbb{R}^{n^2 \times n_b}$ and $Q^{(4)} \in \mathbb{R}^{rn \times n_b}$ in Step 1 of Algorithm 2 are computed, whose significant operations are summarized in Table B.6. Then, the product $\mathcal{J}_{A,O} \widetilde{\mathcal{J}}_{O,\mathcal{H}} T$ in (35) is computed, taking

around $4n_bn^3$ flops, before computing the covariance $\Sigma_{A,C}$ in (35) using also (39), which takes around $2n_bn^4$ flops. This amounts to a total of

$$4cn_m^3n + 8cn_bn_m^2n + 6cn_bn_mn^2 + 4n_bn^3 + 2n_bn^4 \text{ flops}$$

for the computation of the covariance of the vectorized system matrices A and C.

Table B.6: Flop count for computation of $Q^{(1)}, Q^{(2)}, Q^{(3)}$ and $Q^{(4)}$ from $\Sigma_{\mathcal{H}} = TT^T$.

Reference	Operation	Flops
Remark 9	$T_{i,1}, T_{i,2}$	$4cn_bn_m^2$
Prop. 7	$\mathcal{H}K_i$	$4cn_m^3$
Remark 9	$(\widetilde{\mathcal{J}}_{O,\mathcal{H}}\,T)_i$	$4cn_bn_m^2$
Equ. (37)	block i of $Q^{(1)}, Q^{(2)}, Q^{(3)}, Q^{(4)}$	$6cn_bn_mn$
	$\sum_{i=1}^{n} \approx 4cn_m^3n + 8cn_bn_m^2n + 6cn_bn_mn^2$	

Appendix B.3.2. Covariance computation of modal parameters

The new covariance computation of the n_d modal parameters ($i = 1, \ldots, n_d$) is summarized in Algorithm 2. First, the matrices $Q^{(1)}, Q^{(2)}, Q^{(3)}$ and $Q^{(4)}$ (see Step 1) need to be known, whose significant computations are summarized in Table B.6. In Step 2, the matrix $(O^{\uparrow T}O^{\uparrow})^{-1}$ as well as $(\mathcal{P}_{n,n} + I_{n^2})Q^{(1)}$ and the sum $\mathcal{P}_{n,n}Q^{(2)} + Q^{(3)}$ are computed in a further preprocessing step, amounting to around

$$2cn_mn^2 + 3n_bn^2 \text{ flops.}$$

Then, the covariances of the modal parameters are computed in Steps 4–5 for each mode i. Their flop count is summarized in Table B.7. The relevant operation to obtain $\text{cov}([f_i \ \xi_i]^T)$ is then the computation of Q_i in (44), taking $4n_bn^2$ flops for each mode and thus

$$4n_bn_dn^2 \text{ flops}$$

for all modes. The relevant operations for the computation of the covariance of the mode shape take $22n^3 + 10n_bn^2 + 4rn_bn + 2n_br^2$ flops for each mode and thus

$$22n_dn^3 + 6n_bn_dn^2 + 4n_brn_dn + 2n_br^2n_d \text{ flops}$$

for all modes, once Q_i is computed.

Table B.7: Flop count for covariance computation of a frequency and damping ratio (first part) and of the mode shapes (second part) for each mode in Section 5.4.

Reference	Operation	Flops
Equ. (44)	Q_i	$4n_bn^2$
Equ. (43)	$\mathcal{J}_{\lambda_i,A}\mathcal{J}_{A,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T$	$2n_bn$
Equ. (46)	SVD of $(\lambda_i I_n - A)$	$22n^3$
Equ. (45)–(46)	$C(\lambda_i I_n - A)^\dagger \left(I_n - \frac{\phi_i\chi_i^*}{\chi_i^*\phi_i}\right)(O^{\uparrow T}O^{\uparrow})^{-1}Q$	$6n_bn^2 + 2n_brn$
Equ. (45)	$(\phi_i^T \otimes I_r)\mathcal{J}_{C,O}\widetilde{\mathcal{J}}_{O,\mathcal{H}}T$	$2n_brn$
Equ. (41)	$\text{cov}([\Re(\varphi_i^T) \ \Im(\varphi_i^T)]^T)$	$2n_br^2$
	$\sum \approx 22n^3 + 10n_bn^2 + 4n_brn + 2n_br^2$	

Appendix B.4. Comparison of covariance computation of modal parameters

The computational cost of the covariance computation of the modal parameters with Algorithms 1 and 2 from the previous sections is summarized in Table B.8. Note that $n_t = (p + q)rr_0 \approx 2crn_m$.

Table B.8: Comparison of flop counts for Algorithms 1 and 2.

Step in Algorithm		Flops of Algorithm 1	Flops of Algorithm 2
Preprocessing	Step 1	$22c^3 n_m^3 n + 4c^2 n_t n_m^2 n$	$4c n_m^3 n + 8c n_b n_m^2 n + 6c n_b n_m n^2$
	Step 2	$2c n_t n_m n^3 + 2n_t^2 n^2 + 2n_t n^4$	$2c n_m n^2 + 3n_b n^2$
Covariance of f_i, ξ_i's	Step 4	$4n_d n^4$	$4n_b n_d n^2$
Covariance of φ_i's	Step 5	$4r n_d n^4$	$(22n^3 + 10n_b n^2 + 4n_b rn + 2n_b r^2)n_d$

References

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

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

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

[4] A. Benveniste, L. Mevel, Nonstationary consistency of subspace methods, IEEE Transactions on Automatic Control AC-52 (6) (2007) 974–984.

[5] L. Hermans, H. Van der Auweraer, Modal testing and analysis of structures under operational conditions: industrial application, Mechanical Systems and Signal Processing 13 (2) (1999) 193–216.

[6] L. Mevel, M. Basseville, M. Goursat, Stochastic subspace-based structural identification and damage detection - Application to the steel-quake benchmark, Mechanical Systems and Signal Processing 17 (1) (2003) 91–101.

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

[8] J. Brownjohn, F. Magalhães, E. Caetano, A. Cunha, Ambient vibration re-testing and operational modal analysis of the Humber Bridge, Engineering Structures 32 (8) (2010) 2003–2018.

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

[10] P. Bakir, Automation of the stabilization diagrams for subspace based system identification, Expert Systems with Applications 38 (12) (2011) 14390–14397.

[11] E. Reynders, R. Pintelon, G. De Roeck, Uncertainty bounds on modal parameters obtained from stochastic subspace identification, Mechanical Systems and Signal Processing 22 (4) (2008) 948–969.

[12] E. Carden, A. Mita, Challenges in developing confidence intervals on modal parameters estimated for large civil infrastructure with stochastic subspace identification, Structural Control and Health Monitoring 18 (1) (2011) 53–78.

[13] M. Basseville, A. Benveniste, M. Goursat, L. Mevel, In-flight monitoring of aeronautic structures: vibration-based on-line automated identification versus detection, IEEE Control Systems Magazine 27 (5) (2007) 27–42.

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

[15] R. Pintelon, P. Guillaume, J. Schoukens, Uncertainty calculation in (operational) modal analysis, Mechanical Systems and Signal Processing 21 (6) (2007) 2359–2373.

[16] J. Brewer, Kronecker products and matrix calculus in system theory, IEEE Transactions on Circuits and Systems 25 (9) (1978) 772–781.

[17] M. Döhler, L. Mevel, Robust subspace based fault detection, in: Proc. 18th IFAC World Congress, Milan, Italy, 2011.

[18] P. Fackler, Notes on matrix calculus, Tech. rep., North Carolina State University (2005).

[19] M. Döhler, Subspace-based system identification and fault detection: Algorithms for large systems and application to structural vibration analysis, Ph.D. thesis, Université de Rennes 1, France (2011).

[20] M. Basseville, A. Benveniste, M. Goursat, L. Hermans, L. Mevel, 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) (2001) 668–676.

[21] M. Döhler, L. Mevel, Fast multi-order computation of system matrices in subspace-based system identification, Control Engineering Practice 20 (9) (2012) 882–894.

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