Variance estimation of modal parameters from output-only and input/output subspace-based system identification

Philippe Mellinger, Michael Döhler, Laurent Mevel

▶ **To cite this version:**

HAL Id: hal-01328435

https://hal.inria.fr/hal-01328435

Submitted on 8 Jun 2016

Variance estimation of modal parameters from output-only and input/output subspace-based system identification

Philippe Mellinger, Michael Döhler*, Laurent Mevel

Inria / IFSTTAR, I4S, Campus de Beaulieu, 35042 Rennes, France

Abstract

An important step in the operational modal analysis of a structure is to infer on its dynamic behavior through its modal parameters. They can be estimated by various modal identification algorithms that fit a theoretical model to measured data. When output-only data is available, i.e. measured responses of the structure, frequencies, damping ratios and mode shapes can be identified assuming that ambient sources like wind or traffic excite the system sufficiently. When also input data is available, i.e. signals used to excite the structure, input/output identification algorithms are used. The use of input information usually provides better modal estimates in a desired frequency range. While the identification of the modal mass is not considered in this paper, we focus on the estimation of the frequencies, damping ratios and mode shapes, relevant for example for modal analysis during in-flight monitoring of aircrafts. When identifying the modal parameters from noisy measurement data, the information on their uncertainty is most relevant. In this paper, new variance computation schemes for modal parameters are developed for four subspace algorithms, including output-only and input/output methods, as well as data-driven and covariance-driven methods. For the input/output methods, the known inputs are considered as realizations of a stochastic process. Based on Monte Carlo validations, the quality of identification, accuracy of variance estimations and sensor noise robustness are discussed. Finally these algorithms are applied on real measured data obtained during vibrations tests of an aircraft.

Keywords: (Operational) modal analysis, Subspace algorithms, Variance estimation, Aircraft application

1. Introduction

The design and maintenance of mechanical, aeronautical or civil structures subject to noise and vibrations are an important topic in structural engineering. Laboratory and in-operation vibration tests are performed on structures for a modal analysis [1]. The desired modal parameters (frequencies, damping ratios, mode shapes) are related to the poles and observed eigenvectors of a linear time-invariant system. Subspace-based linear system identification algorithms have been proven efficient for the identification of such systems, fitting a linear model to either output-only measurements or input and output measurements taken from a system [2]. Usually, ambient excitation sources

---

*Corresponding author. Tel: +33 2 99 84 22 25.
Email addresses: `philippe.mellinger@epfedu.fr` (Philippe Mellinger), `michael.doehler@inria.fr` (Michael Döhler), `laurent.mevel@inria.fr` (Laurent Mevel)

(e.g. wind, traffic) act on a structure during real tests. Besides these non-measurable excitation sources, in addition some measurable excitation sources (e.g. artificial excitation from hammer impact or shaker, ground motion) may be available. Thus, inputs to the considered system are in general unknown, but in some cases, some known inputs may be available in addition to the unknown inputs.

Operational Modal Analysis (OMA) refers to the identification of modal parameters from output-only measurements under unknown ambient excitation. In Operational Modal Analysis with eXogenous inputs (OMAX), some known or measured excitation is applied to the structure besides the ambient noise. Throughout this paper, the known input sequence is assumed to be a realization of a stochastic process. For example, control surface excitations can be used during either flight or ground vibration tests in the particular case of an aircraft. A stochastic framework can model the excitation of flaps actioned by a pilot during in-flight vibration tests, which are the known inputs in this case. Other unknown ambient excitation has to be considered, which is commonly assumed to be white noise. The measured outputs are due to both known and unknown inputs, and may be corrupted by noise (i.e. sensor noise uncorrelated with known inputs).

Many methods for OMA and OMAX are available in the literature [3]. In this work, we focus on subspace methods. By appropriate projections of the measurement data [2], the state space representation of the corresponding linear system is properly estimated. Then, the modal parameters are retrieved from the estimated state space model. There are a variety of subspace algorithms, which are defined by the way the measurements are projected. Data-driven subspace algorithms are characterized by large Hankel matrices filled with raw data. Orthogonal or oblique projections are carried out on their row space [2, 4], for which numerical schemes with the QR decomposition [2] and iterative and fast approaches [5–7] have been studied. In the stochastic framework, those projectors are essentially conditional expectations of the future output data with respect to some instrument depending on the algorithm [8]. Connections with Prediction Error Methods [9] and closed loop identification have been investigated [10–12]. It is well understood that the projections in the data-driven methods can be partly expressed in terms of auto- and cross-covariances of the respective time series [2, 13]. The class of covariance-driven subspace algorithms is purely based on auto- and cross-covariance matrices of the measurements and has been used for modal identification [14] or damage detection [15, 16] using vibration data for more than a decade. Like with data-driven methods, covariance-driven methods can be based on output-only data or input/output data [13, 17–20].

The purpose of this paper is the variance analysis of modal parameters obtained from subspace methods. Using noisy measurement data, subspace algorithms provide parameter estimates that are afflicted with statistical uncertainty due to finite data, unknown inputs and sensor noise properties. In-depth asymptotic variance analyses for subspace algorithms have been carried out in the automatic control literature [21–24], which are of important theoretical value. In these works, analytical expressions for the asymptotic covariance of the system matrices are derived for several output-only and input/output subspace algorithms. However, explicit expressions for the actual covariance estimation from measurements are not available therein.

In the field of vibration analysis, such explicit expressions have been proposed for some subspace methods. For

example, an application of the variance analysis in [21] for input/output subspace identification was tempted in [25] in the context of modal analysis, however without entirely satisfying results according to the authors of the study. A different and successful approach was developed in [26], where an estimated covariance on the measurements is propagated to the desired parameters based on a sensitivity analysis. The required sensitivities are derived analytically through the propagation of a first-order perturbation from the data to the identified parameters. This approach has the advantage of computational convenience: the sample covariance as the starting point is directly linked to the measurements and therefore easy to compute, and the sensitivities are computed using the system identification estimates. In [27], details of this scheme are developed for the covariance computation for output-only covariance-driven subspace identification. This approach offers advantages beyond technical correctness. The exploitation of the stabilization diagram, which is a basic tool for OMA(X), is a crucial step in modal analysis, but computationally costly for the covariance computation. A well-adapted fast and memory efficient implementation in an operational modal analysis context has been proposed in the covariance-driven output-only case [28]. As a result, the computational cost for the variance computation of an entire stabilization diagram analysis is reduced significantly. This was mandatory for realistic applications on large structures like in [29, 30] and a prerequisite for implementation in modal analysis software.

While this approach has shown to be working both in theory and practice for the variance computation of modal parameters, it is only available for *output-only covariance-driven subspace identification* [27, 28]. In the present paper, this covariance computation approach is generalized to a wider class of subspace algorithms in both output-only and input/output frameworks, and for covariance-driven and data-driven methods, while keeping the advantages of the previously published method in [28] in terms of efficiency and precision. Notice that, while the computation of the modal mass is possible with the input/output algorithms [31], we focus on the covariance computation for frequencies, damping ratios and mode shapes. In this paper, new covariance computation schemes are developed for four well-known subspace algorithms. One of them is an output-only algorithm:

- Output-only orthogonal projection data-driven, which is the UPC algorithm (Unweighted Principal Component) [2, Sec. 3.3.2], [14].

Furthermore, we consider three input/output algorithms:

- Input/output covariance-driven [32],

- Input/output orthogonal projection data-driven, related to the projection algorithm [2, Sec. 2.3.2] and MOESP (Multivariable Output Error State Space) [33], and

- Input/output oblique projection data-driven, which is the N4SID algorithm (Numerical algorithms for Subspace State Space System IDentification) [2, 34].

The extension from the output-only covariance-driven algorithm to these four subspace algorithms is performed by a thorough sensitivity analysis of the underlying auto- and cross-covariance matrices in the formulation of the subspace

algorithms. In particular, the extension to the data-driven algorithms is carried out by developing an equivalent formulation in a covariance-driven form, regarding the estimation of frequencies, damping ratios and mode shapes. The new algorithms are validated in extensive Monte-Carlo simulations and in an application on an aircraft.

This paper is organized as follows. In Section 2, the underlying vibration modeling is recalled. In Section 3, the general properties of subspace identification algorithms are described and the five considered subspace algorithms are detailed. An equivalent covariance-driven formulation of the data-driven algorithms is discussed in Section 4, which is the basis for the subsequent derivation of the covariance computation for the data-driven algorithms. Then variances for the considered algorithms are derived in Section 5. The validation of the methods in Monte-Carlo simulations is presented in Section 6 in an academic example. The application on an aircraft structure is presented in Sections 7 and 8 using simulations and real data, respectively, before discussing and concluding the paper in Sections 9 and 10.

Nomenclature			
$\otimes$	Kronecker product	n	system order
vec	column stacking vectorization operator	N	data length
$\dagger$	Moore-Penrose pseudoinverse	N_i, N_o	number of inputs, outputs
Y/X	orthogonal projection	n_b	number of data blocks
$Y/_Z X$	oblique projection	$\mathcal{H}_*$	subspace matrix for the methods:
π_X	orthogonal projector on X	OOcov	output-only covariance-driven
$X^\perp$	orthogonal complement of X	OOort	output-only orthogonal projection data-driven
$\mathbf{E}$	expectation	IOcov	input/output covariance-driven
$\mathbb{R}$	set of real numbers	IOort	input/output orthogonal projection data-driven
ΔX	first order perturbation on X	IOobl	input/output oblique projection data-driven
Σ_X	covariance of vec(X)	Γ_p	observability matrix
$\mathcal{J}_{Y,X}$	sensitivity of vec(Y) wrt. vec(X)	$(\cdot)^-, (\cdot)^+$	past and future time horizon
I_a	identity matrix of size $a \times a$	q, p	length of past and future time horizon
$0_{a,b}$	zero matrix of size $a \times b$	$\mathcal{U}^-, \mathcal{U}^+$	data Hankel matrices of inputs
$\mathcal{P}_{a,b}$	permutation, vec(X^T) = $\mathcal{P}_{a,b}$vec(X)	$\mathcal{Y}^-, \mathcal{Y}^+$	data Hankel matrices of outputs
A, C	state transition, output matrices	$\mathcal{W}^-$	stacked matrix of $\mathcal{U}^-$ and $\mathcal{Y}^-$
B, D	output, feedthrough matrices	$\widetilde{\mathcal{U}}^-, \widetilde{\mathcal{U}}^+$	data Hankel matrices of unknown inputs
u_k, y_k	known inputs, outputs	$\mathcal{V}^-, \mathcal{V}^+$	data Hankel matrices of sensor noise
$\tilde{u}_k, v_k$	unknown inputs, sensor noise	$\mathcal{R}_i$	auto- or cross-covariance matrix
f_i, ξ_i	frequency, damping ratio		

2. Vibration modeling

2.1. Mechanical and state space models

The vibration behavior of a linear time-invariant mechanical structure, which is observed at some sensor positions, can be described by the equations [1]

$$\begin{cases} \mathcal{M}\ddot{q}(t) + C\dot{q}(t) + \mathcal{K}q(t) = \mathcal{F}u(t) + \tilde{u}(t) \\ y(t) = L_a\ddot{q}(t) + L_v\dot{q}(t) + L_dq(t) + v(t) \end{cases} \tag{1}$$

where $\mathcal{M}$, C and $\mathcal{K} \in \mathbb{R}^{m \times m}$ are mass, stiffness and damping matrices, and m is the number of degrees of freedom. The vector $q(t) \in \mathbb{R}^{m \times 1}$ contains the displacements at the degrees of freedom generated by the known inputs $u(t) \in \mathbb{R}^{N_i \times 1}$ and unknown inputs $\tilde{u}(t) \in \mathbb{R}^{m \times 1}$, where N_i is the number of known inputs, and $\mathcal{F} \in \mathbb{R}^{m \times N_i}$ represents how the known inputs are applied to the system. The vector $y(t) \in \mathbb{R}^{N_o \times 1}$ contains the observed outputs, with N_o being the number of sensors. The matrices L_a, L_v and $L_d \in \mathbb{R}^{N_o \times m}$ represent how accelerations, velocities and displacements are obtained from the model degrees of freedom. The vector $v(t) \in \mathbb{R}^{N_o \times 1}$ is the sensor noise. Both $\tilde{u}(t)$ and $v(t)$ are assumed to be white noise with finite fourth moments and uncorrelated with the known inputs [2, 35].

From Eq. (1), the following continuous-time state space model can be obtained:

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) + \tilde{B}_c \tilde{u}(t) \\ y(t) = C_c x(t) + D_c u(t) + \tilde{D}_c \tilde{u}(t) + v(t), \end{cases} \tag{2}$$

where $x(t) = [\dot{q}(t)^T \quad q(t)^T]^T \in \mathbb{R}^{n \times 1}$ is the state vector, $n = 2m$ is the model order and

$$A_c = \begin{bmatrix} -\mathcal{M}^{-1}C & -\mathcal{M}^{-1}\mathcal{K} \\ I_m & 0_{m,m} \end{bmatrix} \in \mathbb{R}^{n \times n}, \ B_c = \begin{bmatrix} \mathcal{M}^{-1}\mathcal{F} \\ 0_{m,N_i} \end{bmatrix} \in \mathbb{R}^{n \times N_i}, \ C_c = \begin{bmatrix} L_v - L_a\mathcal{M}^{-1}C & L_d - L_a\mathcal{M}^{-1}\mathcal{K} \end{bmatrix} \in \mathbb{R}^{N_o \times n}$$

and $D_c = L_a\mathcal{M}^{-1}\mathcal{F} \in \mathbb{R}^{N_o \times N_i}$ are the state transition, input, output and feedthrough matrices, respectively. The matrices $\tilde{B}_c$ and $\tilde{D}_c$ are defined similarly with respect to the unknown inputs. The matrices $I_a \in \mathbb{R}^{a \times a}$ and $0_{a,b} \in \mathbb{R}^{a \times b}$ are the identity and the zero matrix, respectively.

Sampling Eq. (2) at rate τ and assuming zero-order hold for inputs $u(t)$ and $\tilde{u}(t)$ yield the discrete-time state space representation [9]

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + \tilde{B}\tilde{u}_k \\ y_k = Cx_k + Du_k + \tilde{D}\tilde{u}_k + v_k, \end{cases} \tag{3}$$

where $A = e^{A_c\tau}$, $B = (A - I)A_c^{-1}B_c$, $\tilde{B} = (A - I)A_c^{-1}\tilde{B}_c$, $C = C_c$, $D = D_c$, and $\tilde{D} = \tilde{D}_c$.

The modal parameters of system (1) are equivalently found in system (3) as follows. Let λ_i and ϕ_i be eigenvalues and eigenvectors of A, for $i = 1, \ldots, n$. Then the poles μ_i of systems (1) and (2), modal frequencies f_i, damping ratios ξ_i and mode shapes φ_i are obtained by

$$\mu_i = \frac{\log(\lambda_i)}{\tau}, \ f_i = \frac{|\mu_i|}{2\pi}, \ \xi_i = \frac{-\mathrm{Re}(\mu_i)}{|\mu_i|}, \ \varphi_i = C\phi_i.$$

5

2.2. Stochastic nature of known inputs

Until now, no assumption has been stated on the known input $\{u_k\}_{k=1,\ldots,N}$ for the case of input/output algorithms. Known inputs are used in what is called deterministic [2] or combined deterministic/stochastic [2, 31] system identification in the literature. In the designation of these methods, there is equivalence between "deterministic" and "the input sequence is known", and between "stochastic" and "there are unknown inputs and/or noise".

However, these designations should not be mistaken with the nature of the *known inputs*. The known input sequence itself may be a *deterministic* signal, or it may be a realization of a *stochastic* process. For the considered input/output methods in this paper, we assume that known inputs are realizations of a stochastic process satisfying mild ergodicity and mixing properties. This assumption is in particular coherent with the ergodic-algebraic framework of [35] and cross-covariance estimates in [13] for covariance-driven input/output algorithms. In combination with standard white noise assumptions for unknown inputs and measurement noise, this assumption allows the consistent computation of auto-covariances of the inputs, the outputs and of cross-covariances between inputs and outputs, which is a basic step in subspace identification.

These auto- and cross-covariances contribute to the modal parameter estimates. Then, the corresponding uncertainties for these auto- and cross-covariances play together in a non-trivial way to obtain the modal parameter uncertainties, for which the expressions are derived in this paper. Note that assuming known inputs as stochastic is no contradiction for the particular case of a system with no unknown inputs and no sensor noise. In this case, the modal parameters can be identified exactly from the known inputs and outputs, and their variance assuming stochastic inputs is indeed zero, as will be shown in Section 6.

3. Subspace identification

Subspace algorithms are based on projections of output or input/output data Hankel matrices with a "past" and "future" time horizon. These projections are designed in a way that the column space of the resulting matrix – we call it subspace matrix $\mathcal{H}$ – is defined by the observability matrix $\Gamma_p = [C^T \ \ldots \ (CA^{p-1})^T]^T$ of system (3). All five subspace algorithms considered in this paper have the respective factorization property $\mathcal{H} = \Gamma_p \mathcal{Z}$, where $\mathcal{Z}$ depends on the chosen algorithm [2]. This property leads to the estimation of Γ_p from measurement data and subsequently to the system matrices A and C, from where the desired modal parameters are identified.

After defining the underlying data Hankel matrices and the respective auto- and cross-covariances, the three steps of modal identification with subspace algorithms are detailed in this section: 1/ Estimation of auto- and cross-covariance matrices related to "past" and "future" data matrix products for the chosen subspace algorithm, 2/ Computation of the subspace matrix $\mathcal{H}$ for the five subspace algorithms considered in this paper, and finally 3/ Estimation of the modal parameters from $\mathcal{H}$.

Definition 1 (Data Hankel matrix [2]). *Let N_t samples $a_l \in \mathbb{R}^{r \times 1}$ of a discrete signal a be given, and integers i and j with $i \leq j \leq N_t - N + 1$. Then the data Hankel matrix $\mathcal{A}_{i|j}$ is defined as*

$$\mathcal{A}_{i|j} \stackrel{\text{def}}{=} \begin{bmatrix} a_i & a_{i+1} & \ldots & a_{i+N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_j & a_{j+1} & \ldots & a_{j+N-1} \end{bmatrix} \in \mathbb{R}^{(j-i+1)r \times N}.$$

Let $N + p + q$ be the number of available samples for the known inputs u_k and the outputs y_k, where q and p are parameters that define a "past" and "future" time horizon. They are most often equal, and assumed to be large enough to satisfy the condition $\min\{(p-1)N_o, qN_o\} \geq n$. Under the above Definition 1, the "past" ($^-$) and "future" ($^+$) data Hankel matrices containing the known data are defined as in [2],

$$\mathcal{U}^- \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{U}_{0|q-1}, \ \mathcal{U}^+ \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{U}_{q|q+p-1}, \ \mathcal{Y}^- \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{Y}_{0|q-1}, \ \mathcal{Y}^+ \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{Y}_{q|q+p-1}. \tag{4}$$

Consider similarly the unknown signals, i.e. the unknown input $\tilde{u}$, the sensor noise v and the state x, defining the corresponding matrices $\widetilde{\mathcal{U}}^- \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\widetilde{\mathcal{U}}_{0|q-1}$, $\widetilde{\mathcal{U}}^+ \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\widetilde{\mathcal{U}}_{q|q+p-1}$, $\mathcal{V}^- \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{V}_{0|q-1}$, $\mathcal{V}^+ \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{V}_{q|q+p-1}$ and finally $\mathcal{X}^- \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{X}_{0|0}$, $\mathcal{X}^+ \stackrel{\text{def}}{=} \frac{1}{\sqrt{N}}\mathcal{X}_{q|q}$, which are needed for deriving the equations of the model. Then Eq. (3) can be extended to [2]

$$\begin{cases} \mathcal{Y}^- = \Gamma_q \mathcal{X}^- + H_q \mathcal{U}^- + \tilde{H}_q \widetilde{\mathcal{U}}^- + \mathcal{V}^- \\ \mathcal{Y}^+ = \Gamma_p \mathcal{X}^+ + H_p \mathcal{U}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+ \\ \mathcal{X}^+ = A^q \mathcal{X}^- + \Delta_q \mathcal{U}^- + \tilde{\Delta}_q \widetilde{\mathcal{U}}^-. \end{cases} \tag{5}$$

The matrices Δ_q, $\tilde{\Delta}_q$, H_p and $\tilde{H}_p$ are defined in [2].

3.1. First step: estimation of auto- and cross-covariances

Known data (outputs and, if available, inputs) are used to estimate auto- and cross-covariances in order to compute the subspace matrix $\mathcal{H}$. Let a and b be placeholders for two of these signals (e.g. u and y, y and y or u and u). Then, their cross-covariance estimate at lag i is defined as

$$R_{[a,b]_i} = \frac{1}{N} \sum_{k=1}^{N} a_{k+i} b_k^T, \tag{6}$$

considering that a and b are ergodic and zero mean random processes. The term auto-covariance is used for the cross-covariance of the same signal, i.e. $a = b$ in (6). Note that due to the finite data length, these auto- and cross-covariance estimates are random variables with a covariance [36], which will be the starting point of the variance propagation to the modal parameter estimates in Section 5.

Besides the direct computation as in Eq. (6), an efficient way to compute auto- and cross-covariance functions at all lags i is to first estimate the PSD (Power Spectral Density) functions of signals by Welch's method [9] and to compute the inverse discrete Fourier transform of the result, using the Fast Fourier Transform (FFT), as detailed in Appendix A.

Using "past" and "future" data Hankel matrices as defined in Eq. (4) brings some useful properties, since their products are related to the auto- and cross-covariance estimates. For example, it holds

$$
\mathcal{U}^+ \mathcal{Y}^{-T} = \begin{bmatrix} R_{[u,y]_q} & R_{[u,y]_{q-1}} & \cdots & R_{[u,y]_1} \\ R_{[u,y]_{q+1}} & R_{[u,y]_q} & \cdots & R_{[u,y]_2} \\ \vdots & \vdots & \vdots & \vdots \\ R_{[u,y]_{q+p-1}} & R_{[u,y]_{q+p-2}} & \cdots & R_{[u,y]_p} \end{bmatrix},
$$
(7)

and similarly

$$
\mathcal{U}^- \mathcal{Y}^{-T} = \begin{bmatrix} R_{[u,y]_0} & R_{[u,y]_{-1}} & \cdots & R_{[u,y]_{1-q}} \\ R_{[u,y]_1} & R_{[u,y]_0} & \cdots & R_{[u,y]_{2-q}} \\ \vdots & \vdots & \ddots & \vdots \\ R_{[u,y]_{q-1}} & R_{[u,y]_{q-2}} & \cdots & R_{[u,y]_0} \end{bmatrix}, \quad \mathcal{U}^+ \mathcal{Y}^{+T} = \begin{bmatrix} R_{[u,y]_0} & R_{[u,y]_{-1}} & \cdots & R_{[u,y]_{1-p}} \\ R_{[u,y]_1} & R_{[u,y]_0} & \cdots & R_{[u,y]_{2-p}} \\ \vdots & \vdots & \ddots & \vdots \\ R_{[u,y]_{p-1}} & R_{[u,y]_{p-2}} & \cdots & R_{[u,y]_0} \end{bmatrix}
$$
(8)

The products $\mathcal{Y}^- \mathcal{Y}^{-T}$, $\mathcal{Y}^+ \mathcal{Y}^{+T}$, $\mathcal{Y}^+ \mathcal{Y}^{-T}$, $\mathcal{U}^- \mathcal{U}^{-T}$, $\mathcal{U}^+ \mathcal{U}^{+T}$ and $\mathcal{U}^+ \mathcal{U}^{-T}$ are computed analogously. These products based on auto- and cross-covariances are elementary blocks for the computation of the subspace matrices needed by the subspace algorithms of Section 3.2.

Remark 2 (Cross-covariance with unknown inputs and sensor noise). *Products between data Hankel matrices containing outputs or known inputs and noise Hankel matrices are established analogously as above. These cross-covariance based products will appear in the derivation of subspace algorithms in the following section. Some of these cross-covariances tend to zero due to the assumption that unknown inputs and sensor noise are white and uncorrelated with the known inputs. In particular it holds $\widetilde{\mathcal{U}}^+ \mathcal{U}^{+T} = 0$ and $\mathcal{V}^+ \mathcal{U}^{+T} = 0$ for an infinite amount of data. Under these assumptions, past outputs are uncorrelated with future unknown inputs and with future sensor noise, and it holds similarly $\widetilde{\mathcal{U}}^+ \mathcal{Y}^{-T} = 0$ and $\mathcal{V}^+ \mathcal{Y}^{-T} = 0$. These properties are required for the development of the subspace methods in the next section.*

3.2. Second step: computation of subspace matrix $\mathcal{H}$

Each subspace algorithm is related to a subspace matrix $\mathcal{H}$ with factorization property $\mathcal{H} = \Gamma_p \mathcal{Z}$, where $\mathcal{Z}$ depends on the algorithm. Subspace algorithms may be divided into two families: covariance-driven and data-driven. Covariance-driven subspace matrices are products of auto- or cross-covariance matrices. Data-driven matrices can be described as a product of covariance-driven matrices with a data Hankel matrix [8]. The main difference between both kinds of algorithms is the size of the associated subspace matrix. When the number of samples increases, covariance-driven matrices converge to their expected value, while their column dimension does not increase. Concerning data-driven matrices, their covariance components converge as the number of samples increases, but also their column dimension increases due to the increasing dimension of the data-matrix part. In both cases, the column space of the respective subspace matrices converges to the subspace defined by the observability matrix Γ_p.

3.2.1. Orthogonal and oblique projections

Orthogonal and oblique projections of the data Hankel matrices are essential in the definition of subspace methods [2, 8, 13]. In the following, they are defined and some of their properties are recalled, before they are used in the subsequent sections.

Definition 3. *Let M, P and Q be three matrices with the same number of columns. The orthogonal projection M/P of the row space of M onto the row space of P and the oblique projection $M/_Q P$ of the row space of M on the row space of P along the row space of Q are defined by*

$$M/P \stackrel{\text{def}}{=} M\pi_P \stackrel{\text{def}}{=} MP^T \left(PP^T\right)^\dagger P, \quad M/_Q P \stackrel{\text{def}}{=} M\pi_{Q^\perp} \left(P\pi_{Q^\perp}\right)^\dagger P,$$

where π_P is the orthogonal projector on P, $Q^\perp$ is the orthogonal complement of Q and † denotes the Moore-Penrose pseudoinverse.

Recall the properties of an orthogonal projector: It holds $\pi_{P^\perp} = I - \pi_P$, $\pi_P^T = \pi_P$, $\pi_P \pi_P = \pi_P$ and $P\pi_{P^\perp} = 0$. These properties are always exact and no convergence assumption is required [37].

These geometric projections have a stochastic interpretation, since their definition coincides with the definition of the projection of Gaussian random vectors. Indeed, the projection of such vectors is defined through the conditional expectation. For the particular case of centered Gaussian random vectors X, Y, and Z it holds

$$Y/X \stackrel{\text{def}}{=} \mathbf{E}(Y/X) = Y\pi_X \stackrel{\text{def}}{=} \text{Cov}(Y, X)\text{Cov}(X)^\dagger X, \quad Y/_Z X \stackrel{\text{def}}{=} Y\pi_{Z^\perp} \left(X\pi_{Z^\perp}\right)^\dagger X. \tag{9}$$

Finally, let $\mathcal{A}$ and $\mathcal{B}$ be placeholders for any of the data Hankel matrices $\mathcal{U}^-$, $\mathcal{U}^+$, $\mathcal{Y}^-$ or $\mathcal{Y}^+$ (see Eq. (4)) that are filled with the discrete-time samples of the known input or output data, or any of the Hankel matrices in Section 2 related to the unknown signals. Then, $\mathcal{A}\mathcal{B}^T$ tends to a matrix filled with auto- or cross-covariances as in Eqs. (7)–(8) and the orthogonal projection $\mathcal{A}/\mathcal{B} = \mathcal{A}\mathcal{B}^T(\mathcal{B}\mathcal{B}^T)^\dagger \mathcal{B}$ relates to the conditional expectation in Eq. (9) when the number of data samples goes to infinity. In particular, if $\mathcal{A}$ and $\mathcal{B}$ are uncorrelated and covariances have converged, then $\mathcal{A}\mathcal{B}^T = 0$ and $\mathcal{A}/\mathcal{B} = 0$.

3.2.2. Output-only algorithms

These algorithms are based on the assumptions that the unknown white noise input $\tilde{u}(t)$ excites the system, and no known inputs are available ($u(t) = 0$). The observations may be corrupted by an unknown white sensor noise $v(t)$. Then Eq. (5) becomes:

$$\begin{cases} \mathcal{Y}^- = \Gamma_q \mathcal{X}^- + \tilde{H}_q \widetilde{\mathcal{U}}^- + \mathcal{V}^- \\ \mathcal{Y}^+ = \Gamma_p \mathcal{X}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+ \\ \mathcal{X}^+ = A^q \mathcal{X}^- + \tilde{\Delta}_q \widetilde{\mathcal{U}}^- \end{cases}$$

3.2.2.a. OOcov: Output-only covariance-driven

This algorithm [8, 14, 38] is based on the auto-covariances between the N_o outputs as in Eq. (7), which can be computed as a product of the future and past output data matrices,

$$\mathcal{H}_{\text{OOcov}} \overset{\text{def}}{=} \mathcal{Y}^+ \mathcal{Y}^{-^T} \tag{10}$$
$$= (\Gamma_p \mathcal{X}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+) \mathcal{Y}^{-^T} = \Gamma_p \mathcal{X}^+ \mathcal{Y}^{-^T} = \Gamma_p \mathcal{Z}_{\text{OOcov}}$$

3.2.2.b. OOort: Output-only orthogonal projection data-driven

It is also known as UPC algorithm [2, 8, 14], and is based on the projection of the row space of the future output data onto the row space of the past output data,

$$\mathcal{H}_{\text{OOort}} \overset{\text{def}}{=} \mathcal{Y}^+ / \mathcal{Y}^- = \mathcal{Y}^+ \mathcal{Y}^{-^T} \left(\mathcal{Y}^- \mathcal{Y}^{-^T} \right)^\dagger \mathcal{Y}^- \tag{11}$$
$$= \Gamma_p \mathcal{Z}_{\text{OOcov}} \left(\mathcal{Y}^- \mathcal{Y}^{-^T} \right)^\dagger \mathcal{Y}^- = \Gamma_p \mathcal{Z}_{\text{OOort}}$$

3.2.3. Input/output algorithms

These algorithms are based on the assumption that the system is excited by known inputs $u(t)$ and unknown white noise inputs $\tilde{u}(t)$, and measurements are corrupted by white sensor noise $v(t)$. It holds

$$\begin{cases} \mathcal{Y}^- = \Gamma_q \mathcal{X}^- + H_q \mathcal{U}^- + \tilde{H}_q \widetilde{\mathcal{U}}^- + \mathcal{V}^- \\ \mathcal{Y}^+ = \Gamma_p \mathcal{X}^+ + H_p \mathcal{U}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+ \\ \mathcal{X}^+ = A^q \mathcal{X}^- + \Delta_q \mathcal{U}^- + \tilde{\Delta}_q \widetilde{\mathcal{U}}^- \end{cases}$$

3.2.3.a. IOcov: Input/output covariance-driven

This algorithm is based on the data-driven projection algorithm [2, Sec. 2.3.2] right multiplied by the past output data matrix $\mathcal{Y}^{-T}$ in order to obtain a covariance-driven algorithm [32],

$$\mathcal{H}_{\text{IOcov}} \overset{\text{def}}{=} (\mathcal{Y}^+ / \mathcal{U}^{+\perp}) \mathcal{Y}^{-^T} \tag{12}$$
$$= \left((\Gamma_p \mathcal{X}^+ + H_p \mathcal{U}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+) / \mathcal{U}^{+\perp} \right) \mathcal{Y}^{-^T}$$
$$= \left(\Gamma_p \mathcal{X}^+ / \mathcal{U}^{+\perp} + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+ \right) \mathcal{Y}^{-^T} = \Gamma_p \left(\mathcal{X}^+ / \mathcal{U}^{+\perp} \right) \mathcal{Y}^{-^T} = \Gamma_p \mathcal{Z}_{\text{IOcov}},$$

using the assumptions that known inputs are uncorrelated with unknown inputs and with sensor noise, and that past outputs are uncorrelated with future unknown inputs and with future sensor noise (see Remark 2).

3.2.3.b. IOort: Input/output orthogonal projection data-driven

This algorithm is described as the projection algorithm in [2, Sec. 2.3.2] for the "no noise case" when there are no unknown inputs and no sensor noise,

$$\mathcal{H}_{\text{IOort}} \overset{\text{def}}{=} \mathcal{Y}^+ / \mathcal{U}^{+\perp} \tag{13}$$
$$= (\Gamma_p \mathcal{X}^+ + H_p \mathcal{U}^+ + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+) / \mathcal{U}^{+\perp} = \Gamma_p \mathcal{X}^+ / \mathcal{U}^{+\perp} + \tilde{H}_p \widetilde{\mathcal{U}}^+ + \mathcal{V}^+$$

Indeed, there is no factorization property $\mathcal{H}_{\text{IOort}} = \Gamma_p \mathcal{Z}_{\text{IOort}}$ in the presence of unknown inputs or sensor noise. Hence, the theory of this algorithm is only adapted to Experimental Modal Analysis framework. Note that the formulation of this algorithm is equivalent to the MOESP algorithm [2, Sec. 4.3.2] if there are no unknown inputs and no sensor noise.

3.2.3.c. IOobl: Input/output oblique projection data-driven

This data-driven algorithm is known as N4SID [2, Sec. 4.3.1] and is again adapted to the input/output framework with unknown inputs and sensor noise. It yields

$$\mathcal{H}_{\text{IOobl}} \overset{\text{def}}{=} \mathcal{Y}^+ /_{\mathcal{U}^+} \mathcal{W}^- = \left(\mathcal{Y}^+ / \mathcal{U}^{+\perp} \right) \left(\mathcal{W}^- / \mathcal{U}^{+\perp} \right)^\dagger \mathcal{W}^- \tag{14}$$
$$= \Gamma_p \mathcal{Z}_{\text{IOobl}}$$

where $\mathcal{W}^- \overset{\text{def}}{=} \begin{bmatrix} \mathcal{U}^- \\ \mathcal{Y}^- \end{bmatrix}$ and $\mathcal{Z}_{\text{IOobl}}$ is defined in [2].

3.2.4. Some remarks about the considered algorithms

3.2.4.a. $\mathcal{H}_{\text{OOcov}}$ and $\mathcal{H}_{\text{IOcov}}$

Assuming $\mathcal{U}^+$ does not depend on the past, $\mathcal{Z}_{\text{OOcov}}$ and $\mathcal{Z}_{\text{IOcov}}$ are both product of future states and past observations. Both algorithms differ by the knowledge of a component of the excitation. For the output-only covariance-driven algorithm the whole excitation is considered to be unknown. The unknown input term in Eq. (10) is canceled by the product with $\mathcal{Y}^{-T}$ as future unknown inputs are not correlated to past outputs (see Remark 2). The product converges to zero when the number of samples tends to infinity. The difference between Eq. (10) and Eq. (12) is how the input terms are canceled. In both Eq. (10) and Eq. (12), canceling $\widetilde{\mathcal{U}}^+$ is based on cross-covariance convergence. The known part $\mathcal{U}^+$ in Eq. (12) is canceled from the orthogonal projection as $\mathcal{U}^+ / \mathcal{U}^{+\perp}$ is zero. The cancellation through the orthogonal projection is satisfied all the time, whereas cancellation by the cross-covariance product is satisfied only when the number of samples tends to infinity.

3.2.4.b. $\mathcal{H}_{\text{IOcov}}$ and $\mathcal{H}_{\text{IOort}}$

As seen in Section 3.2.3.b, $\mathcal{H}_{\text{IOort}}$ is theoretically a good candidate for retrieving the observability matrix only if there are no unknown inputs and no noise on sensors. Nevertheless, robustness to sensor noise is tested in the application. One can notice that $\mathcal{H}_{\text{IOcov}} = \mathcal{H}_{\text{IOort}} \mathcal{Y}^{-T}$, where the multiplication by $\mathcal{Y}^{-T}$ removes the effect of both unknown inputs and sensor noise assuming convergence for the cross-covariance estimates (see Remark 2).

3.3. Third step: estimation of the modal parameters from $\mathcal{H}$

Assume $\mathcal{H}$ is computed according to some subspace algorithm of interest. The final step is to retrieve modal parameters from $\mathcal{H}$. First notice that all presented subspace methods have been defined such that $\mathcal{H}$ factorizes as $\mathcal{H} = \Gamma_p \mathcal{Z}$ at convergence. The objective is then to obtain the observability matrix Γ_p from $\mathcal{H}$, since $\Gamma_p =$

$[C^T \quad \ldots \quad (CA^{p-1})^T]^T$ contains the system matrices A and C. Thanks to the factorization property, $\mathcal{H}$ and Γ_p share the same column space [2]. Thus, Γ_p can be obtained by the singular value decomposition (SVD) [37]

$$\mathcal{H} = USV^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{15}$$

as $\Gamma_p = U_1 S_1^{\frac{1}{2}}$, where the SVD is truncated at the desired model order n [2]. The output matrix C is then obtained by direct extraction of the first block row of Γ_p as $C = \begin{bmatrix} I_{N_o} & 0_{N_o,(p-1)N_o} \end{bmatrix} \Gamma_p$. The state transition matrix A is obtained from the shift invariance property of Γ_p as the least squares solution $A = \Gamma_{up}^{\dagger} \Gamma_{dw}$ [14], where Γ_{dw} and Γ_{up} are obtained from Γ_p by removing the first and last block row, respectively, as

$$\Gamma_{dw} = \begin{bmatrix} CA \\ \vdots \\ CA^{p-1} \end{bmatrix}, \quad \Gamma_{up} = \begin{bmatrix} C \\ \vdots \\ CA^{p-2} \end{bmatrix}.$$

Finally, the modal parameters are obtained from matrices A and C as stated in Section 2. Hence, the computation of the modal parameters is straightforward when the subspace matrix $\mathcal{H}$ is computed.

4. From data-driven to covariance-driven subspace algorithms

In contrast to the covariance-driven algorithms, the number of columns of matrix $\mathcal{H}$ depends on the number of data samples N for the data-driven algorithms. Hence, the matrix $\mathcal{H}$ does not converge to a fixed limit for data-driven algorithms for $N \to \infty$, which is a problem for the subsequent covariance analysis. In this section, we propose a simple way to transform data-driven algorithms to covariance-driven algorithms for the sake of covariance computation, where the resulting covariance-driven algorithm yields *identical* estimates of the observability matrix Γ_p as the original data-driven algorithm. Hence, the variance of the resulting modal parameter estimates is also identical between both algorithms. Note that only Γ_p is required for the estimation of the modal parameters (frequencies, damping ratios and mode shapes) from the system matrices (A, C), and the identification of (B, D) and the modal mass is beyond the scope of this paper.

Let $\mathcal{H}_{\text{dat}}$ be the subspace matrix of a data-driven algorithm. Then, the corresponding covariance-driven algorithm is defined through the *squared subspace matrix*

$$\mathcal{H}_{\text{cov}} = \mathcal{H}_{\text{dat}} \mathcal{H}_{\text{dat}}^T$$

Notice that this was already exploited in [8] to relate the consistency of both data-driven and covariance-driven algorithms together. It is easy to see that both matrices $\mathcal{H}_{\text{cov}}$ and $\mathcal{H}_{\text{dat}}$ indeed have the same column space: let the thin SVD of $\mathcal{H}_{\text{dat}} = USV^T$ be given, then $\mathcal{H}_{\text{cov}} = \mathcal{H}_{\text{dat}} \mathcal{H}_{\text{dat}}^T = US^2U^T$, and hence both $\mathcal{H}_{\text{cov}}$ and $\mathcal{H}_{\text{dat}}$ have the same left singular vectors (up to a change of basis). Thus, subspace-based system identification using $\mathcal{H}_{\text{cov}}$ or $\mathcal{H}_{\text{dat}}$ leads to identical results (see Section 3.3), and the covariance computation for data-driven algorithms can be based on $\mathcal{H}_{\text{cov}}$.

Based on the data-driven algorithms described in Sections 3.2.2.b, 3.2.3.b, 3.2.3.c, the respective covariance-driven algorithms can be formulated as follows.

4.1. OOort^2: Output-only orthogonal projection data-driven squared

With the definition of the underlying UPC algorithm in Section 3.2.2.b, set $\mathcal{H}_{\text{OOort}^2} \overset{\text{def}}{=} \mathcal{H}_{\text{OOort}} \mathcal{H}_{\text{OOort}}^T$. It follows from projector properties

$$\mathcal{H}_{\text{OOort}^2} = \mathcal{Y}^+ \pi_{\mathcal{Y}^-} \mathcal{Y}^{+^T} = \mathcal{H}_{\text{OOort}} \mathcal{Y}^{+^T} = \left(\mathcal{Y}^+ \mathcal{Y}^{-^T} \right) \left(\mathcal{Y}^- \mathcal{Y}^{-^T} \right)^{\dagger} \left(\mathcal{Y}^+ \mathcal{Y}^{-^T} \right)^T. \tag{16}$$

4.2. IOort^2: Input/output orthogonal projection data-driven squared

With the definition of the underlying projection algorithm in Section 3.2.3.b, set $\mathcal{H}_{\text{IOort}^2} \overset{\text{def}}{=} \mathcal{H}_{\text{IOort}} \mathcal{H}_{\text{IOort}}^T$, and projector properties yield

$$\mathcal{H}_{\text{IOort}^2} = \mathcal{Y}^+ \pi_{\mathcal{U}^{+\perp}} \mathcal{Y}^{+^T} = \mathcal{H}_{\text{IOort}} \mathcal{Y}^{+^T}. \tag{17}$$

Note that this right multiplication with $\mathcal{Y}^{+^T}$ does not remove the effect of unknown inputs and sensor noise, whereas this would be the case with $\mathcal{Y}^{-^T}$ (cf. Section 3.2.3.a). Equivalently to $\mathcal{H}_{\text{IOort}}$, $\mathcal{H}_{\text{IOort}^2}$ should not be used under unknown inputs and sensor noise conditions.

4.3. IOobl^2: Input/output oblique projection data-driven squared

With the definition of the underlying N4SID algorithm in Section 3.2.3.c, set $\mathcal{H}_{\text{IOobl}^2} \overset{\text{def}}{=} \mathcal{H}_{\text{IOobl}} \mathcal{H}_{\text{IOobl}}^T$. Using the pseudoinverse property $\mathcal{A}^{\dagger} = \mathcal{A}^T (\mathcal{A} \mathcal{A}^T)^{\dagger}$ for an arbitrary matrix $\mathcal{A}$, the pseudoinverse in Eq. (14) can be replaced, leading to

$$\mathcal{H}_{\text{IOobl}} = \left(\mathcal{Y}^+ / \mathcal{U}^{+\perp} \right) \left(\mathcal{W}^- / \mathcal{U}^{+\perp} \right)^T \left[\left(\mathcal{W}^- / \mathcal{U}^{+\perp} \right) \left(\mathcal{W}^- / \mathcal{U}^{+\perp} \right)^T \right]^{\dagger} \mathcal{W}^-.$$

Squaring this matrix and using projector properties leads finally to the covariance-driven matrix

$$\mathcal{H}_{\text{IOobl}^2} = \mathcal{Y}^+ \pi_{\mathcal{U}^{+\perp}} \mathcal{W}^{-^T} \left(\mathcal{W}^- \pi_{\mathcal{U}^{+\perp}} \mathcal{W}^{-^T} \right)^{\dagger} \mathcal{W}^- \mathcal{W}^{-^T} \left(\mathcal{W}^- \pi_{\mathcal{U}^{+\perp}} \mathcal{W}^{-^T} \right)^{\dagger} \mathcal{W}^- \pi_{\mathcal{U}^{+\perp}} \mathcal{Y}^{+^T}. \tag{18}$$

5. Variance estimation

In this section, the notations and basic principles of the covariance computation are introduced based on [27, 28]. Until now this method was developed only for the output-only covariance-driven subspace algorithm. Here, the variance estimation algorithms are extended to the considered data-driven and input/output subspace algorithms by developing expressions for the covariance of the corresponding subspace matrices.

13

5.1. Theory and definitions

The uncertainty of the modal parameter estimates comes from the fact that only output or input/output sequences of finite length N are known. Expressions for their variance based on an input/output data sequence of length N are hence also estimates, and the central limit theorem (CLT) ensures that these expressions are asymptotically correct (i.e. for $N \to \infty$) [9, Ch. 9].

The computation of the modal parameter covariance results from the propagation of sample covariances on auto- or cross-covariance estimates (involving measured outputs, and for the case of input/output algorithms also the known inputs) through all steps of the modal identification algorithm. These sample covariances reflect in particular the unknown inputs due to non-measurable excitation sources and the sensor noise, and they contribute in a non-trivial way to the covariance of the modal parameter estimates. The propagation to the modal parameter estimates is based on the delta method [39], where the analytical sensitivity matrices are obtained from perturbation theory [26, 27].

Definition 4. *Let $\theta(0)$ be the expected value of a parameter θ and $\theta(\varepsilon)$ its estimate, where ε denotes a small perturbation. A first order Taylor approximation of $\theta(\varepsilon)$ is given by $\theta(\varepsilon) \approx \theta(0) + \frac{\partial \theta}{\partial \varepsilon}\big|_{\varepsilon=0} \varepsilon$. Define $\Delta\theta \overset{\text{def}}{=} \theta(\varepsilon) - \theta(0) \approx \frac{\partial \theta}{\partial \varepsilon}\big|_{\varepsilon=0} \varepsilon$.*

Lemma 5. *Let g be a continuous and differentiable function of θ. A first order Taylor approximation of $g(\theta(\varepsilon))$ is given by $g(\theta(\varepsilon)) \approx g(\theta(0)) + \frac{\partial g(\theta)}{\partial \theta}\big|_{\theta=\theta(\varepsilon)} (\theta(\varepsilon) - \theta(0))$, then $\Delta g(\theta) = g(\theta(\varepsilon)) - g(\theta(0)) \approx \frac{\partial g(\theta)}{\partial \theta}\big|_{\theta=\theta(\varepsilon)} (\theta(\varepsilon) - \theta(0))$. Then*

$$\Delta g(\theta) \approx \frac{\partial g(\theta)}{\partial \theta}\bigg|_{\theta=\theta(\varepsilon)} \Delta\theta,$$

and the covariance yields

$$\text{Cov}\left(g(\theta)\right) = \mathbf{E}\left[\Delta g(\theta) \Delta g(\theta)^T\right] \approx \frac{\partial g(\theta)}{\partial \theta}\bigg|_{\theta=\theta(\varepsilon)} \text{Cov}(\theta) \left.\frac{\partial g(\theta)}{\partial \theta}\right|^T_{\theta=\theta(\varepsilon)}. \tag{19}$$

A sample covariance is easily computed from the measured data (see Section 5.2). With Eq. (19), it can be propagated through the whole computation chain of the subspace algorithms (see Sections 5.3 and 5.4). Since the auto- and cross-covariances of the data are asymptotically Gaussian (for $N \to \infty$) [36], the delta method ensures that each variable in this computation chain – down to the modal parameters – is also asymptotically Gaussian [39]. Hence the respective covariance estimate in Eq. (19) is consistent.

The challenge is to obtain the analytical sensitivity matrices for each step of the algorithm by perturbation theory that are required in Eq. (19). To illustrate with an example, let $M = PQ$ be a matrix product of $P \in \mathbb{R}^{k \times l}$ and $Q \in \mathbb{R}^{l \times m}$. The perturbation theory consists then on relating the covariance of M to the covariances of P and Q. Let vec(.) be the vectorization operator which stacks the columns of a matrix. With $\Delta M = \Delta P Q + P \Delta Q$, it follows using the vec properties $\text{vec}(\Delta M) = (Q^T \otimes I_k)\text{vec}(\Delta P) + (I_m \otimes P)\text{vec}(\Delta Q)$, and finally

$$\text{vec}(\Delta M) = \mathcal{J} \begin{bmatrix} \text{vec}(\Delta P) \\ \text{vec}(\Delta Q) \end{bmatrix} \quad \text{where} \quad \mathcal{J} = \begin{bmatrix} (Q^T \otimes I_k) & (I_m \otimes P) \end{bmatrix},$$

$$\text{Cov}(\text{vec}(M)) = \mathbf{E}\left[\text{vec}(\Delta M)\text{vec}(\Delta M)^T\right] = \mathcal{J} \, \text{Cov}\left(\begin{bmatrix} \text{vec}(P) \\ \text{vec}(Q) \end{bmatrix}\right) \mathcal{J}^T.$$

14

In the following, some further properties of the involved matrix operations are needed. Define the permutation matrix $\mathcal{P}_{a,b} \stackrel{\text{def}}{=} \sum_{k=1}^{a} \sum_{l=1}^{b} E_{k,l}^{a,b} \otimes E_{l,k}^{b,a}$, 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 [40]. This matrix satisfies $\mathcal{P}_{a,b}\text{vec}(G) = \text{vec}(G^T)$, for $G \in \mathbb{R}^{a \times b}$. Finally, we need to derive the sensitivities of components containing the pseudo-inverse in the projection terms within the definition of the subspace methods. While the relation $\Delta M^{-1} = -M^{-1} \Delta M M^{-1}$ holds for invertible matrices M, a similar relationship for the pseudo-inverse requires some stronger assumptions and is established for our case in the following lemma.

Lemma 6. *Let $\mathcal{A}$, $\mathcal{B}$ and C be placeholders for any of the data Hankel matrices $\mathcal{U}^-$, $\mathcal{U}^+$, $\mathcal{Y}^-$ or $\mathcal{Y}^+$, and let $\mathcal{R}_a = \mathcal{A}\mathcal{B}^T$, $\mathcal{R}_b = \mathcal{B}\mathcal{B}^T$ and $\mathcal{R}_c = \mathcal{B}C^T$ be the respective auto- or cross-covariance matrices. Then,*

$$\Delta(\mathcal{R}_a \mathcal{R}_b^\dagger \mathcal{R}_c) = \Delta \mathcal{R}_a \mathcal{R}_b^\dagger \mathcal{R}_c - \mathcal{R}_a \mathcal{R}_b^\dagger \Delta \mathcal{R}_b \mathcal{R}_b^\dagger \mathcal{R}_c + \mathcal{R}_a \mathcal{R}_b^\dagger \Delta \mathcal{R}_c.$$

Proof. See Appendix C. $\qquad\square$

5.2. First step: covariance estimation of the auto- and cross-covariance matrices

The starting point of the variance propagation to the modal parameters is the covariance of the auto- and cross-covariance matrices that are involved in a chosen subspace algorithm. For the considered algorithms in this paper, the auto- and cross-covariance matrices are the following:

$$\mathcal{R}_1 \stackrel{\text{def}}{=} \mathcal{Y}^+ \mathcal{Y}^{+T} \in \mathbb{R}^{pN_o \times pN_o}, \quad \mathcal{R}_2 \stackrel{\text{def}}{=} \mathcal{Y}^+ \mathcal{Y}^{-T} \in \mathbb{R}^{pN_o \times qN_o}, \quad \mathcal{R}_3 \stackrel{\text{def}}{=} \mathcal{Y}^- \mathcal{Y}^{-T} \in \mathbb{R}^{qN_o \times qN_o} \quad \mathcal{R}_4 \stackrel{\text{def}}{=} \mathcal{Y}^+ \mathcal{U}^{+T} \in \mathbb{R}^{pN_o \times pN_i},$$

$$\mathcal{R}_5 \stackrel{\text{def}}{=} \mathcal{Y}^- \mathcal{U}^{+T} \in \mathbb{R}^{qN_o \times pN_i}, \quad \mathcal{R}_6 \stackrel{\text{def}}{=} \mathcal{Y}^+ \mathcal{U}^{-T} \in \mathbb{R}^{pN_o \times qN_i}, \quad \mathcal{R}_7 \stackrel{\text{def}}{=} \mathcal{Y}^- \mathcal{U}^{-T} \in \mathbb{R}^{qN_o \times qN_i}, \quad \mathcal{R}_8 \stackrel{\text{def}}{=} \mathcal{U}^+ \mathcal{U}^{+T} \in \mathbb{R}^{pN_i \times pN_i},$$

$$\mathcal{R}_9 \stackrel{\text{def}}{=} \mathcal{Y}^+ \mathcal{W}^{-T} \in \mathbb{R}^{pN_o \times q(N_o+N_i)}, \quad \mathcal{R}_{10} \stackrel{\text{def}}{=} \mathcal{W}^- \mathcal{U}^{+T} \in \mathbb{R}^{pN_i \times q(N_o+N_i)}, \quad \mathcal{R}_{11} \stackrel{\text{def}}{=} \mathcal{W}^- \mathcal{W}^{-T} \in \mathbb{R}^{q(N_o+N_i) \times q(N_o+N_i)}$$

Notice that the convergence of these matrices depends on the data length. The central limit theorem ensures that they are asymptotically Gaussian and afflicted with uncertainty for a finite data length [36].

Hence, the computation of $\text{Cov}(\text{vec}(\mathcal{R}_i), \text{vec}(\mathcal{R}_j))$ between any of the matrices $\mathcal{R}_i$ and $\mathcal{R}_j$ for the considered subspace algorithm (including $i = j$) is required. This computation follows the lines of the OOcov algorithm, i.e. for $\text{Cov}(\text{vec}(\mathcal{R}_2))$, as described in detail in [27, 28], and is generalized as follows.

Let $\mathcal{A}$, $\mathcal{B}$, C and $\mathcal{D}$ be placeholders for any of the data Hankel matrices $\mathcal{U}^-$, $\mathcal{U}^+$, $\mathcal{Y}^-$ or $\mathcal{Y}^+$ (see Eq. (4)), such that $\mathcal{R}_i = \mathcal{A}\mathcal{B}^T$ and $\mathcal{R}_j = C\mathcal{D}^T$ are the respective auto- or cross-covariance matrices. Divide the data matrices into n_b blocks and normalize them with respect to their length, such that

$$\sqrt{N}\mathcal{A} = \sqrt{N_b}\begin{bmatrix} \mathcal{A}^1 & \mathcal{A}^2 & \dots & \mathcal{A}^{n_b} \end{bmatrix}, \quad \sqrt{N}\mathcal{B} = \sqrt{N_b}\begin{bmatrix} \mathcal{B}^1 & \mathcal{B}^2 & \dots & \mathcal{B}^{n_b} \end{bmatrix},$$

$$\sqrt{N}C = \sqrt{N_b}\begin{bmatrix} C^1 & C^2 & \dots & C^{n_b} \end{bmatrix}, \quad \sqrt{N}\mathcal{D} = \sqrt{N_b}\begin{bmatrix} \mathcal{D}^1 & \mathcal{D}^2 & \dots & \mathcal{D}^{n_b} \end{bmatrix},$$

where each block $\mathcal{A}^k$, $\mathcal{B}^k$, C^k and $\mathcal{D}^k$ 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 these blocks, the respective auto-

or cross-covariance estimate can be computed as $\mathcal{R}_i^k = \mathcal{A}^k \mathcal{B}^{kT}$ and $\mathcal{R}_j^k = C^k \mathcal{D}^{kT}$, which can be assumed to be i.i.d., yielding

$$\mathcal{R}_i = \frac{1}{n_b} \sum_{k=1}^{n_b} \mathcal{R}_i^k, \quad \mathcal{R}_j = \frac{1}{n_b} \sum_{k=1}^{n_b} \mathcal{R}_j^k \, .$$

It follows $\mathrm{Cov}\left(\mathrm{vec}(\mathcal{R}_*)\right) = \frac{1}{n_b}\mathrm{Cov}\left(\mathrm{vec}(\mathcal{R}_*^k)\right)$, and the covariance between the auto- or cross-covariance matrices can be computed from the usual sample covariance as

$$\mathrm{Cov}(\mathrm{vec}(\mathcal{R}_i), \mathrm{vec}(\mathcal{R}_j)) = \frac{1}{n_b(n_b-1)} \sum_{k=1}^{n_b} \left(\mathrm{vec}(\mathcal{R}_i^k) - \mathrm{vec}(\mathcal{R}_i)\right)\left(\mathrm{vec}(\mathcal{R}_j^k) - \mathrm{vec}(\mathcal{R}_j)\right)^T . \tag{20}$$

With this expression, covariance estimates for covariance-driven subspace identification are achievable. It is not directly applicable for data-driven algorithms, where the size of the matrices changes with the number of available samples, requiring the squaring of the subspace matrix as shown in Section 4. Further subspace algorithms besides the ones investigated in the current paper can fit to this approach as long as the corresponding subspace matrix $\mathcal{H}$ can be expressed with respect to similar auto- or cross-covariance matrices as $\mathcal{R}_i, i = 1, \ldots, 11$.

5.3. Second step: sensitivity of $\mathcal{H}$ for each subspace algorithm

In this section, the sensitivities of the subspace matrices $\mathcal{H}$ are derived with respect to the underlying auto- and cross-covariance matrices in their computation, depending on each considered subspace algorithm. With the sample covariance of the auto- and cross-covariance matrices from the previous section, the covariance of the subspace matrix may then be computed based on Eq. (19), and propagated to the system matrices A and C and the modal parameters in the next step.

5.3.1. Output-only algorithms

5.3.1.a. OOcov: Output-only covariance-driven

Due to Eq. (10), it holds $\mathcal{H}_{\mathrm{OOcov}} = \mathcal{R}_2$ and thus

$$\mathrm{vec}(\Delta \mathcal{H}_{\mathrm{OOcov}}) = \mathrm{vec}(\Delta \mathcal{R}_2).$$

5.3.1.b. OOort^2: Output-only orthogonal projection data-driven

From its definition in Eq. (16), it follows $\mathcal{H}_{\mathrm{OOort}^2} = \mathcal{R}_2 \mathcal{R}_3^\dagger \mathcal{R}_2^T$ and thus, together with Lemma 6,

$$\Delta \mathcal{H}_{\mathrm{OOort}^2} = \Delta \mathcal{R}_2 \mathcal{R}_3^\dagger \mathcal{R}_2^T - \mathcal{R}_2 \mathcal{R}_3^\dagger \Delta \mathcal{R}_3 \mathcal{R}_3^\dagger \mathcal{R}_2^T + \mathcal{R}_2 \mathcal{R}_3^\dagger \Delta \mathcal{R}_2^T,$$

$$\mathrm{vec}\left(\Delta \mathcal{H}_{\mathrm{OOort}^2}\right) = \mathcal{J}_{\mathcal{H}_{\mathrm{OOort}^2}, \mathcal{R}} \begin{bmatrix} \mathrm{vec}\left(\Delta \mathcal{R}_2\right) \\ \mathrm{vec}\left(\Delta \mathcal{R}_3\right) \end{bmatrix},$$

where $\mathcal{J}_{\mathcal{H}_{\mathrm{OOort}^2}, \mathcal{R}} = \left[\mathcal{R}_2 \mathcal{R}_3^\dagger \otimes I_{pN_o} + (I_{pN_o} \otimes \mathcal{R}_2 \mathcal{R}_3^\dagger) \mathcal{P}_{pN_o, qN_o} \quad -\mathcal{R}_2 \mathcal{R}_3^\dagger \otimes \mathcal{R}_2 \mathcal{R}_3^\dagger \right]$.

5.3.2. Input/output algorithms

5.3.2.a. IOcov: Input/output covariance-driven

From its definition in Eq. (12) it follows $\mathcal{H}_{\text{IOcov}} = \mathcal{Y}^+ \mathcal{Y}^{-T} - \mathcal{Y}^+ \mathcal{U}^{+T} \left(\mathcal{U}^+ \mathcal{U}^{+T} \right)^\dagger \mathcal{U}^+ \mathcal{Y}^{-T} = \mathcal{R}_2 - \mathcal{R}_4 \mathcal{R}_8^\dagger \mathcal{R}_5^T$, and thus

$$\Delta \mathcal{H}_{\text{IOcov}} = \Delta \mathcal{R}_2 - \Delta \mathcal{R}_4 \mathcal{R}_8^\dagger \mathcal{R}_5^T + \mathcal{R}_4 \mathcal{R}_8^\dagger \Delta \mathcal{R}_8 \mathcal{R}_8^\dagger \mathcal{R}_5^T - \mathcal{R}_4 \mathcal{R}_8^\dagger \Delta \mathcal{R}_5^T,$$

$$\text{vec}(\Delta \mathcal{H}_{\text{IOcov}}) = \mathcal{J}_{\mathcal{H}_{\text{IOcov}},\mathcal{R}} \begin{bmatrix} \text{vec}(\Delta \mathcal{R}_2) \\ \text{vec}(\Delta \mathcal{R}_4) \\ \text{vec}(\Delta \mathcal{R}_8) \\ \text{vec}(\Delta \mathcal{R}_5) \end{bmatrix},$$

where $\mathcal{J}_{\mathcal{H}_{\text{IOcov}},\mathcal{R}} = \begin{bmatrix} I_{pN_o q N_o} & -\mathcal{R}_5 \mathcal{R}_8^\dagger \otimes I_{pN_o} & \mathcal{R}_5 \mathcal{R}_8^\dagger \otimes \mathcal{R}_4 \mathcal{R}_8^\dagger & -(I_{qN_o} \otimes \mathcal{R}_4 \mathcal{R}_8^\dagger) \mathcal{P}_{qN_o, pN_i} \end{bmatrix}$.

5.3.2.b. IOort[2]: Input/output orthogonal projection data-driven squared

From its definition in Eq. (17), it follows $\mathcal{H}_{\text{IOort}^2} = \mathcal{Y}^+ \mathcal{Y}^{+T} - \mathcal{Y}^+ \mathcal{U}^{+T} \left(\mathcal{U}^+ \mathcal{U}^{+T} \right)^\dagger \mathcal{U}^+ \mathcal{Y}^{+T} = \mathcal{R}_1 - \mathcal{R}_4 \mathcal{R}_8^\dagger \mathcal{R}_4^T$ and thus

$$\Delta \mathcal{H}_{\text{IOort}^2} = \Delta \mathcal{R}_1 - \Delta \mathcal{R}_4 \mathcal{R}_8^\dagger \mathcal{R}_4^T + \mathcal{R}_4 \mathcal{R}_8^\dagger \Delta \mathcal{R}_8 \mathcal{R}_8^\dagger \mathcal{R}_4^T - \mathcal{R}_4 \mathcal{R}_8^\dagger \Delta \mathcal{R}_4^T,$$

$$\text{vec}(\Delta \mathcal{H}_{\text{IOort}^2}) = \mathcal{J}_{\mathcal{H}_{\text{IOort}^2},\mathcal{R}} \begin{bmatrix} \text{vec}(\Delta \mathcal{R}_1) \\ \text{vec}(\Delta \mathcal{R}_4) \\ \text{vec}(\Delta \mathcal{R}_8) \end{bmatrix},$$

where $\mathcal{J}_{\mathcal{H}_{\text{IOort}^2},\mathcal{R}} = \begin{bmatrix} I_{pN_o pN_o} & -\mathcal{R}_4 \mathcal{R}_8^\dagger \otimes I_{pN_o} - (I_{pN_o} \otimes \mathcal{R}_4 \mathcal{R}_8^\dagger) \mathcal{P}_{pN_o, pN_i} & \mathcal{R}_4 \mathcal{R}_8^\dagger \otimes \mathcal{R}_4 \mathcal{R}_8^\dagger \end{bmatrix}$.

5.3.2.c. IOobl[2]: Input/output oblique projection data-driven squared

From its definition in Eq. (18), it follows $\mathcal{H}_{\text{IOobl}^2} = \Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \Omega_1^T$, where $\Omega_1 = \mathcal{Y}^+ \pi_{\mathcal{U}^{+\perp}} \mathcal{W}^{-T} = \mathcal{R}_9 - \mathcal{R}_4 \mathcal{R}_8^\dagger \mathcal{R}_{10}^T \in$ $\mathbb{R}^{pN_o \times q(N_o+N_i)}$ and $\Omega_2 = \mathcal{W}^- \pi_{\mathcal{U}^{+\perp}} \mathcal{W}^{-T} = \mathcal{R}_{11} - \mathcal{R}_{10} \mathcal{R}_8^\dagger \mathcal{R}_{10}^T \in \mathbb{R}^{q(N_o+N_i) \times q(N_o+N_i)}$. After perturbation and vectorization of $\mathcal{H}_{\text{IOobl}^2}$,

$$\text{vec}(\Delta \mathcal{H}_{\text{IOobl}^2}) = \mathcal{J}_{\mathcal{H}_{\text{IOobl}^2},\mathcal{R}} \begin{bmatrix} \text{vec}(\Delta \mathcal{R}_9) \\ \text{vec}(\Delta \mathcal{R}_4) \\ \text{vec}(\Delta \mathcal{R}_8) \\ \text{vec}(\Delta \mathcal{R}_{10}) \\ \text{vec}(\Delta \mathcal{R}_{11}) \end{bmatrix}, \tag{21}$$

where $\mathcal{J}_{\mathcal{H}_{\text{IOobl}^2},\mathcal{R}} = \begin{bmatrix} \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(1)} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(2)} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(3)} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(4)} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(5)} \end{bmatrix}$ is given in Appendix D.

5.4. Third step: covariance of $\mathcal{H}$

In the previous section, the sensitivity of the subspace matrix was related to the sensitivities of the underlying auto- and cross-covariance matrices for each considered subspace algorithm, and the covariance $\text{Cov}(\text{vec}(\mathcal{R}_i), \text{vec}(\mathcal{R}_j))$ between the auto- and cross-covariance matrices was obtained in Section 5.2. To finally compute the covariance of the

subspace matrix, Eq. (19) is applied. For example, the OOcov algorithm requires the computation of $\text{Cov}(\text{vec}(\mathcal{H}_{\text{OOcov}})) = \text{Cov}(\text{vec}(\mathcal{R}_2))$ (see Section 5.3.1.a), or the OOort^2 algorithm yields

$$
\begin{aligned}
\text{Cov}(\text{vec}\,(\mathcal{H}_{\text{OOort}^2})) &= \mathbf{E}\left(\mathcal{J}_{\mathcal{H}_{\text{OOort}^2},\mathcal{R}}\begin{bmatrix}\text{vec}\,(\Delta\mathcal{R}_2)\\ \text{vec}\,(\Delta\mathcal{R}_3)\end{bmatrix}\begin{bmatrix}\text{vec}\,(\Delta\mathcal{R}_2)\\ \text{vec}\,(\Delta\mathcal{R}_3)\end{bmatrix}^T \mathcal{J}^T_{\mathcal{H}_{\text{OOort}^2},\mathcal{R}}\right)\\
&= \mathcal{J}_{\mathcal{H}_{\text{OOort}^2},\mathcal{R}}\begin{bmatrix}\text{Cov}(\text{vec}(\mathcal{R}_2)) & \text{Cov}(\text{vec}(\mathcal{R}_2),\text{vec}(\mathcal{R}_3))\\ \text{Cov}(\text{vec}(\mathcal{R}_2),\text{vec}(\mathcal{R}_3))^T & \text{Cov}(\text{vec}(\mathcal{R}_3))\end{bmatrix}\mathcal{J}^T_{\mathcal{H}_{\text{OOort}^2},\mathcal{R}}\,,
\end{aligned}
\tag{22}
$$

requiring thus the computation of $\text{Cov}(\text{vec}(\mathcal{R}_2))$, $\text{Cov}(\text{vec}(\mathcal{R}_3))$ and $\text{Cov}(\text{vec}(\mathcal{R}_2),\text{vec}(\mathcal{R}_3))$ (see Section 5.3.1.b). With the covariance of the subspace matrix, the covariance of the modal parameters is obtained as described in Appendix B.

5.5. Algorithmic summary

In this section, a summary of the variance computation for the modal parameters from the subspace methods of this paper is given. Starting point of the computation is the subspace matrix $\mathcal{H}$ computed on the output dataset or input/output dataset for the chosen subspace method and the resulting modal parameters (see Section 3). Then the following steps are taken:

1. **Covariance of auto- and cross-covariance matrices:** For the chosen subspace method, $\mathcal{H}$ depends on a number of auto- or cross-covariance matrices $\mathcal{R}_i$ (see Sections 5.3.1 and 5.3.2). The covariance of and between all involved auto- or cross-covariance matrices is computed on n_b data blocks of the available output or input/output data as shown in Eq. (20).

2. **Covariance of subspace matrix:** The sensitivity matrix $\mathcal{J}_{\mathcal{H},\mathcal{R}}$ for the chosen subspace method is computed in Sections 5.3.1 and 5.3.2. Then, the covariance of the subspace matrix $\text{Cov}(\text{vec}\,(\mathcal{H}))$ is computed as in Eq. (22).

3. **Covariance of modal parameters:** The covariance of the identified modal parameters is then computed as outlined in Appendix B. A detailed computation is described in [27, 28].

Note that an efficient numerical implementation of these steps can be formulated based on [28, 41].

6. Numerical validation: mass-spring chain

A simulation study was made on a damped mass-spring chain model of six degrees of freedom (see Figure 1) to validate the variance computation for the different subspace methods in an academic example. The mass of all elements is $m_i = 1/20$, stiffnesses are $k_1 = k_3 = k_5 = 100, k_2 = k_4 = k_6 = 200$ and damping is defined such that all modes have a damping ratio of $\xi_i = 0.020$.

6.1. Monte Carlo simulations including unknown excitation and sensor noise

Gaussian white noise excitation force at element 1 is simulated, which is considered as the known input for the input/output algorithms, plus additional (unknown) Gaussian white noise excitation force at all six elements with 10% of the standard deviation of the known input. From these inputs, acceleration output time series of length $N = 10,000$ are simulated with time step $\tau = 0.02$ s at elements 1, 2 and 5, where Gaussian white sensor noise with 5% of the standard deviation of each output is added, respectively.

These simulations are repeated 10,000 times. Each time the modal parameters and their variances are estimated with the five output-only and input/output subspace algorithms from this paper. The parameters used for the identification and variance computation in the subspace algorithms are:

- system order $n = 12$ (see Section 2 and Eq. (15)),

- future and past time horizon $p = q = 16$ (see Eq. (4)),

- number of data blocks for variance computation $n_b = 50$ (see Section 5.2).

The means of the identified modal parameters with the different subspace algorithms are shown together with the theoretical modal parameters from the mass-spring chain model in Table 1. The identified values are very close to the theoretical ones, except for the IOort^2 method that is not adapted to noisy data (see Section 3.2.3.b), and except for the first mode in the OOcov method that would have required longer datasets.

In each simulation, the standard deviations of the modal parameters are computed for the different subspace algorithms with the methods in Section 5. They are obtained from the square root of the respective modal parameter
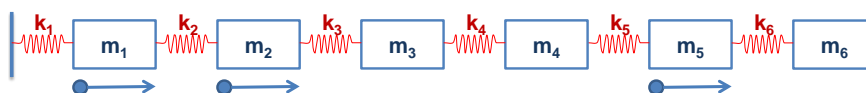

Figure 1: Mass-spring chain (with modal damping), three sensors, for simulations.

Table 1: Means for frequencies and damping ratios from modal identification in 10,000 Monte Carlo simulations.

	Frequency (Hz)						Damping ratio (‰)					
Algorithm	1	2	3	4	5	6	1	2	3	4	5	6
Actual model	1.930	5.618	8.682	14.49	15.85	17.01	20.0	20.0	20.0	20.0	20.0	20.0
OOcov	1.937	5.618	8.683	14.49	15.85	17.01	29.4	20.1	20.1	20.0	20.0	20.0
OOort^2	1.933	5.618	8.683	14.49	15.84	17.01	21.7	20.1	20.0	20.0	19.7	19.9
IOcov	1.930	5.618	8.682	14.49	15.85	17.01	20.2	20.0	20.0	20.0	20.0	20.0
IOort^2	2.016	5.623	8.684	14.52	15.85	17.00	36.1	19.0	19.2	18.0	18.3	17.1
IOobl^2	1.930	5.618	8.682	14.49	15.85	17.01	20.1	20.0	20.0	20.0	20.0	20.0

Table 2: Mean of estimated standard deviations ($\overline{\sigma}_{\text{est}}$) and sample standard deviations (σ_{MC}) for frequencies and damping ratios of the six modes in 10,000 Monte Carlo simulations.

Algorithm		Standard deviation of frequency (Hz $\cdot 10^{-2}$)					
		1	2	3	4	5	6
OOcov	$\overline{\sigma}_{\text{est}}$	2.02 ± 1.80	1.10 ± 0.15	1.55 ± 0.21	2.60 ± 0.36	2.64 ± 0.38	3.51 ± 0.48
	σ_{MC}	1.99	1.14	1.59	2.63	2.65	3.48
OOort^2	$\overline{\sigma}_{\text{est}}$	0.74 ± 0.12	1.05 ± 0.14	1.41 ± 0.18	2.45 ± 0.33	2.42 ± 0.33	3.01 ± 0.42
	σ_{MC}	0.69	1.08	1.45	2.46	2.44	2.99
IOcov	$\overline{\sigma}_{\text{est}}$	0.42 ± 0.06	0.24 ± 0.03	0.29 ± 0.03	0.66 ± 0.08	0.49 ± 0.05	0.87 ± 0.10
	σ_{MC}	0.41	0.25	0.29	0.68	0.50	0.87
IOort^2	$\overline{\sigma}_{\text{est}}$	2.00 ± 1.17	0.23 ± 0.03	0.27 ± 0.03	0.62 ± 0.07	0.45 ± 0.05	0.75 ± 0.08
	σ_{MC}	2.74	0.24	0.27	0.63	0.46	0.75
IOobl^2	$\overline{\sigma}_{\text{est}}$	0.34 ± 0.05	0.23 ± 0.03	0.27 ± 0.03	0.62 ± 0.07	0.45 ± 0.05	0.77 ± 0.09
	σ_{MC}	0.33	0.23	0.27	0.64	0.46	0.78

Algorithm		Standard deviation of damping ratio (‰)					
		1	2	3	4	5	6
OOcov	$\overline{\sigma}_{\text{est}}$	16.46 ± 26.64	1.98 ± 0.28	1.80 ± 0.25	1.70 ± 0.22	1.67 ± 0.23	2.09 ± 0.28
	σ_{MC}	17.12	2.07	1.85	1.71	1.71	2.09
OOort^2	$\overline{\sigma}_{\text{est}}$	4.08 ± 0.81	1.83 ± 0.26	1.60 ± 0.21	1.61 ± 0.21	1.50 ± 0.20	1.81 ± 0.23
	σ_{MC}	3.91	1.91	1.64	1.62	1.53	1.81
IOcov	$\overline{\sigma}_{\text{est}}$	2.13 ± 0.34	0.43 ± 0.05	0.33 ± 0.04	0.45 ± 0.05	0.31 ± 0.04	0.52 ± 0.06
	σ_{MC}	2.07	0.44	0.34	0.45	0.31	0.53
IOort^2	$\overline{\sigma}_{\text{est}}$	10.46 ± 12.20	0.39 ± 0.05	0.30 ± 0.03	0.39 ± 0.04	0.27 ± 0.03	0.42 ± 0.05
	σ_{MC}	15.69	0.40	0.30	0.39	0.27	0.42
IOobl^2	$\overline{\sigma}_{\text{est}}$	1.72 ± 0.25	0.40 ± 0.05	0.31 ± 0.04	0.42 ± 0.05	0.28 ± 0.03	0.47 ± 0.05
	σ_{MC}	1.65	0.41	0.31	0.43	0.29	0.48

variances. The mean $\overline{\sigma}_{\text{est}}$ of these 10,000 estimated standard deviations from each of the Monte Carlo simulations is shown for each modal parameter and algorithm in Table 2. These values are compared to the sample standard deviations σ_{MC} of the 10,000 estimated modal parameters in Table 2. An excellent agreement between the estimated standard deviations with the methods from this paper and the sample statistics from the Monte Carlo simulations can be found for all subspace methods, validating the new methods for variance computation from a theoretical
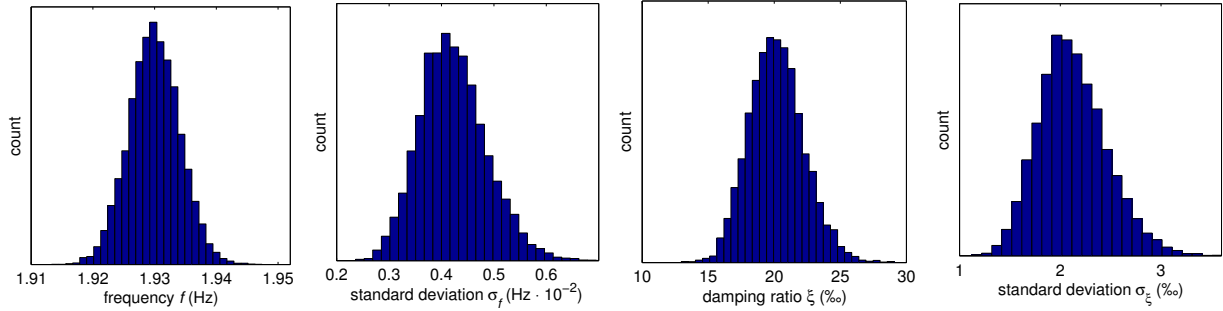
Figure 2: Histograms for first mode identified with IOcov from Monte Carlo simulations: estimated frequencies, estimated standard deviations on frequencies, estimated damping ratios and estimated standard deviations on damping ratios.

point of view. Note that the estimates of the standard deviations show good agreement even for the cases where the identification results were not satisfying (in particular for the first mode in the OOcov method and all modes in the IOort[2] method). Note also that the standard deviations are significantly lower with the input/output methods, compared to the output-only methods, roughly by factor 4 in this application.

Table 2 also shows the one sigma sample standard deviation of the estimated standard deviations during the Monte Carlo simulations. It is mostly smaller than 15% of their mean value (except for the badly estimated modes), both for frequencies and damping ratios. This means that the estimation error of the standard deviation on a single dataset is in most cases less than 15% with the proposed methods, which is very reasonable for a second-order statistics. As examples, the histograms of the identified frequencies and damping ratios of the first mode, and the histograms of their estimated standard deviations with the methods from this paper are shown in Figure 2 for the Monte Carlo simulations with the IOcov method. The histograms validate empirically the Gaussian assumption for the distribution of the frequency and damping estimates, and they show the sample distribution of the their estimated standard deviations that also seems to be approximately Gaussian. The frequencies seem to fit better to a Gaussian distribution than the damping ratios, which is expected considering frequencies are easier to estimate [42]. Still, the distribution for the damping ratios is satisfactory and well centered.

6.2. Identification without unknown inputs and without sensor noise

For the particular case where all the inputs are known (and not corrupted by noise) and the measured outputs are not corrupted by sensor noise, the modal identification with input/output algorithms should be perfect if the true model order is known, and if (even short) data of some minimum length is available. The identified modal parameters should match exactly the theoretical values in this case, and the uncertainty on their estimation should thus be zero. This is verified in this section.

For this purpose, Gaussian white noise excitation at element 1 of length $N = 1,000$ is simulated with time step $\tau = 0.02\,\text{s}$, which is considered as the known input of the system. From this input, one acceleration output dataset

at elements 1, 2 and 5 is simulated without adding sensor noise. From this input/output dataset, modal identification with the estimation of their standard deviations is performed with the same parameters as in the previous section. In Table 3, the errors between the identified modal parameters and the theoretical parameters from the model, as well as the estimated standard deviation σ_f and σ_ξ from the methods of this paper are shown. It can be seen that both the errors and the standard deviations are zero as expected, up to numerical errors, validating the uncertainty quantification with the input/output algorithms for this "no noise" case.

Note that in the present variance computation methods it is assumed that both the input and the output signals are (known) stochastic sequences. The standard deviations (σ_f and σ_ξ) in Table 3 are correctly estimated at zero in this

Table 3: Identification error and estimated standard deviations on input/output data without noise in one simulation.

Algorithm		Frequency (Hz)					
		1	2	3	4	5	6
IOcov	$\lvert f - f_{\mathrm{model}} \rvert$	$7.8 \cdot 10^{-16}$	$3.6 \cdot 10^{-16}$	$1.1 \cdot 10^{-15}$	$1.8 \cdot 10^{-16}$	$7.1 \cdot 10^{-16}$	$1.1 \cdot 10^{-15}$
	σ_f	$3.9 \cdot 10^{-16}$	$6.8 \cdot 10^{-16}$	$1.8 \cdot 10^{-15}$	$6.1 \cdot 10^{-15}$	$6.9 \cdot 10^{-15}$	$3.2 \cdot 10^{-15}$
	σ_f^{d}	$6.3 \cdot 10^{-3}$	$7.4 \cdot 10^{-3}$	$6.0 \cdot 10^{-3}$	$2.7 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$	$1.9 \cdot 10^{-2}$
IOort^2	$\lvert f - f_{\mathrm{model}} \rvert$	$3.3 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$3.6 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$3.6 \cdot 10^{-15}$
	σ_f	$5.5 \cdot 10^{-16}$	$4.6 \cdot 10^{-16}$	$3.8 \cdot 10^{-16}$	$6.3 \cdot 10^{-16}$	$6.0 \cdot 10^{-16}$	$8.2 \cdot 10^{-16}$
	σ_f^{d}	$6.1 \cdot 10^{-2}$	$9.8 \cdot 10^{-3}$	$2.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-1}$	$3.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$
IOobl^2	$\lvert f - f_{\mathrm{model}} \rvert$	$1.4 \cdot 10^{-14}$	$4.4 \cdot 10^{-15}$	0	$7.1 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$1.4 \cdot 10^{-14}$
	σ_f	$1.4 \cdot 10^{-15}$	$4.9 \cdot 10^{-16}$	$4.0 \cdot 10^{-16}$	$2.1 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$	$5.5 \cdot 10^{-15}$
	σ_f^{d}	$6.5 \cdot 10^{-3}$	$6.4 \cdot 10^{-3}$	$6.0 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$	$2.0 \cdot 10^{-2}$

Algorithm		Damping ratio (‰)					
		1	2	3	4	5	6
IOcov	$\lvert \xi - \xi_{\mathrm{model}} \rvert$	$1.9 \cdot 10^{-12}$	$1.2 \cdot 10^{-12}$	$3.0 \cdot 10^{-13}$	$1.2 \cdot 10^{-12}$	$1.1 \cdot 10^{-12}$	$5.9 \cdot 10^{-13}$
	σ_ξ	$2.8 \cdot 10^{-13}$	$9.7 \cdot 10^{-14}$	$8.6 \cdot 10^{-13}$	$3.7 \cdot 10^{-13}$	$5.7 \cdot 10^{-13}$	$3.9 \cdot 10^{-13}$
	σ_ξ^{d}	3.1	0.9	1.3	0.3	0.6	0.5
IOort^2	$\lvert \xi - \xi_{\mathrm{model}} \rvert$	$1.4 \cdot 10^{-12}$	$5.1 \cdot 10^{-14}$	$5.2 \cdot 10^{-13}$	$2.6 \cdot 10^{-13}$	$8.9 \cdot 10^{-14}$	$6.3 \cdot 10^{-13}$
	σ_ξ	$2.7 \cdot 10^{-13}$	$6.3 \cdot 10^{-14}$	$5.2 \cdot 10^{-14}$	$1.2 \cdot 10^{-13}$	$3.2 \cdot 10^{-14}$	$1.5 \cdot 10^{-13}$
	σ_ξ^{d}	32	4.9	2.8	9.0	10	15
IOobl^2	$\lvert \xi - \xi_{\mathrm{model}} \rvert$	$5.9 \cdot 10^{-12}$	$1.0 \cdot 10^{-12}$	$9.3 \cdot 10^{-14}$	$6.8 \cdot 10^{-13}$	$5.5 \cdot 10^{-13}$	$9.8 \cdot 10^{-13}$
	σ_ξ	$9.3 \cdot 10^{-13}$	$2.1 \cdot 10^{-13}$	$6.9 \cdot 10^{-14}$	$1.8 \cdot 10^{-13}$	$6.9 \cdot 10^{-14}$	$2.3 \cdot 10^{-13}$
	σ_ξ^{d}	3.5	0.9	1.1	0.3	0.6	0.6

"no noise" case, which shows that assuming at the same time *known* and *stochastic* inputs is no contradiction in the derivation of the uncertainty quantification methods. On the contrary, the derivation of the terms related to the auto- or cross-covariances involving the inputs as it is done in the present methods is necessary to correctly obtain this result.

To show the effect of neglecting the stochastic nature of the known inputs in the derivation of the uncertainty quantification methods, let us assume that the terms related to uncertainties of the input covariances are zero, i.e. $\Delta\mathcal{R}_8 = 0$, and $\Delta(\mathcal{U}^-\mathcal{U}^{+T}) = 0$ and $\Delta(\mathcal{U}^-\mathcal{U}^{-T}) = 0$ in $\Delta\mathcal{R}_{10}$ and $\Delta\mathcal{R}_{11}$ (see definitions in Section 5.2). The corresponding standard deviations are denoted by σ_f^{d} and σ_ξ^{d} and shown in Table 3. Note that these values are distinctively different from zero in our example, which again supports the assumption on using the stochastic nature of the inputs in the derivation.

7. Numerical validation: aircraft model

In this section, the five considered subspace algorithms and the new methods for their variance computation are applied using a complex numerical model of an unnamed aircraft, simulating a ground vibration test. The performance of the methods is evaluated in an OMA(X) context. In particular, robustness to noise on sensors is tested.

7.1. Benchmark

The simulation of time series was performed on a linear time invariant system, from a finite element model of an unnamed aircraft with around 200,000 degrees of freedom using the Newmark numerical scheme. A constant damping ratio of 15 ‰ was chosen for each mode. The inputs of the system are simulated as white noise excitation triggered by control surfaces of the aircraft. The responses of the system are observed at 46 sensors (outputs of the system) along the aircraft. Simulations of a length of 128 s were made and sampled at 2048 Hz. Such a high sampling frequency was necessary to reach scheme convergence until 40 Hz. The 262,144 samples of each time signal are divided into ten consecutive blocks. Power spectral densities are estimated as described in Appendix A for frequencies from 0 Hz to 40 Hz. Finally, auto- and cross-covariances are estimated using 10 blocks. The simulations are performed in order to respect ground vibration test conditions.

7.2. Strategy for uncertainty validation

The purpose of this section is to validate the variance computation based on one realization of each input and output signals in a more realistic context than in Section 6. Similarly, if the variance computation procedure is valid, it will coincide with the variance computed by the much more computationally intensive Monte Carlo approach. Notice that the Monte Carlo approach is not feasible in an operational context. Notice also that the proposed sensitivity-based approach will only need one sequence of each input and output signals since they are ergodic signals. This section focuses on the uncertainty validation for frequencies and damping ratios.

To validate the developed algorithms, vibration simulations of $N_s = 200$ datasets are performed. Each simulation provides responses of the system under white noise excitation triggered by control surfaces. The input signals are

23

different realizations for all simulations but all have the same statistical properties. From these simulations, $N_s = 200$ modal identifications are carried out. Since the system order is rarely known (never when using real-life measured data), the modal parameters are estimated at several orders in a stabilization diagram. Considering a specific order, each identification provides a set of estimated modes (f, ξ and φ) and their estimated standard deviation (σ_f, σ_ξ and σ_φ). From these 200 sets of modal parameters, the Monte Carlo validation is performed. Note that the different estimates are i.i.d. random variables for each mode.

To perform the validation of the variance computation, we compare the sample covariance σ_{MC}^2 of the 200 Monte Carlo estimates of a quantity with the mean of their computed variances $\overline{\sigma}_{\mathrm{est}}^2$ using the new methods of this paper, similar as in Section 6.1. Then the following ratio is computed for each estimated quantity m (frequency or damping ratio) at each model order for validation, to compare both variances:

$$C_m = \frac{\min\left(\sigma_{\mathrm{MC},m}^2, \overline{\sigma}_{\mathrm{est},m}^2\right)}{\max\left(\sigma_{\mathrm{MC},m}^2, \overline{\sigma}_{\mathrm{est},m}^2\right)} \in [0, 1]$$

The criterion to validate the variance computation for a quantity is set at $C_m \geq 0.7$, i.e. the computed variance with the methods of this paper deviates in mean less than 30 % from the Monte Carlo sample variance, and the respective standard deviations (as the square root of the variance) deviate less than 16 %. Note that these thresholds are empirical.

Plotting frequencies (or damping ratios) for each order yields so-called frequency (or damping) stabilization diagrams [7, 14]. The solid lines in the following plots correspond to the exact values from the model. Stable modes are modes that appear at multiple model orders unlike spurious modes. Frequencies are normalized by the highest displayed frequency value. Axes are adjusted from 0 to 1 for the frequency range and for orders 1 to 100. In our uncertainty validation framework, results are presented in particular stabilization diagrams corresponding to the Monte Carlo simulations: instead of plotting the identified values, their means obtained from the 200 identifications are plotted. To find the same mode in the different identifications, a criterion was used based on the frequency ratio and the MAC value with the theoretical values from the model. An empirical identification threshold is set up, which needs to be exceeded for mode pairing. If a mode is not found in each identification, i.e., it is always below the identification threshold, it is removed from the analysis. Empty circles correspond to values with $C_m < 0.7$, for which the variance computation is not validated. Circles filled with stars correspond to values with $C_m \geq 0.7$, where the variance computation is validated.

7.3. Validation results

7.3.1. OOcov: Output-only covariance-driven – Figure 3

In terms of identification: Seven stable modes are identified, and one more mode seems to appear around order 90. Both frequencies and damping ratios stabilize well. Several spurious modes appear when model order becomes too high. Sensor noise has no significant effect on stable modes. This algorithm, in this specific case, is robust to noise on sensors. Note that the number of spurious modes seems to decrease when adding noise on sensors. However,

(a) Normalized frequencies – without noise on sensors

(b) Normalized frequencies – with noise on sensors

(c) Damping ratios (‰) – without noise on sensors

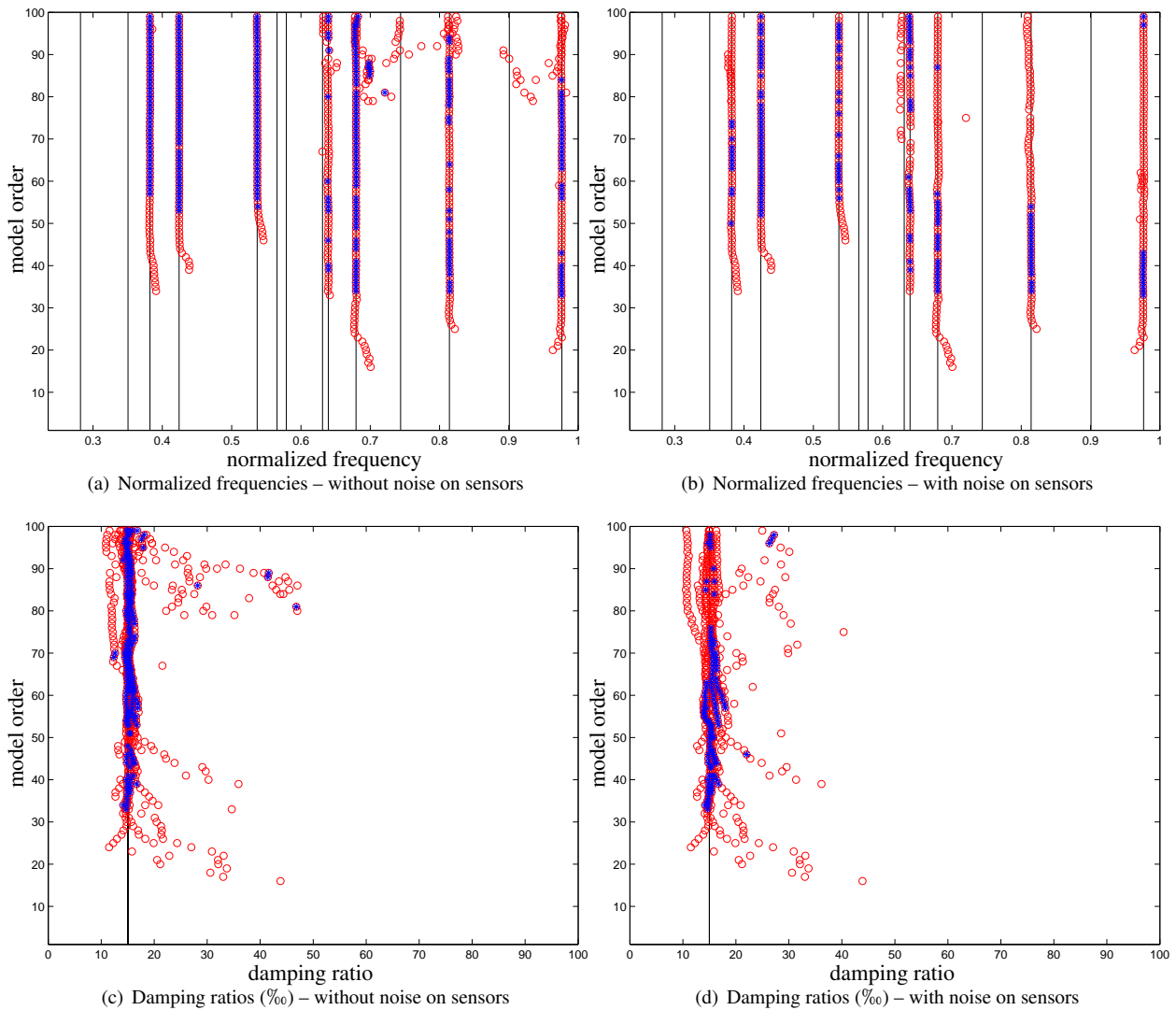(d) Damping ratios (‰) – with noise on sensors

Figure 3: OOcov – Monte Carlo stabilization charts with variance validation values for orders 1 to 100

recall that the averages of 200 identifications are shown in the figures. The number of spurious modes does not really decrease at each identification. However these modes become much more unstable comparing all identifications. The automated selection removes them, when they are not found in every identification.

In terms of uncertainty estimations: With or without noise on sensors, uncertainty estimations on spurious modes are rarely validated. It seems that the uncertainty estimation is more accurate when the identification is more stable. Noise on sensors reduces the number of validated uncertainty estimations. This algorithm, in this specific case, does not seem to be robust to noise on sensors in terms of uncertainty estimations for both frequencies and damping ratios.

25

7.3.2. OOort^2*: Output-only orthogonal projection data-driven squared – Figure 4*

(a) Normalized frequencies – without noise on sensors

(b) Normalized frequencies – with noise on sensors

(c) Damping ratios (‰) – without noise on sensors
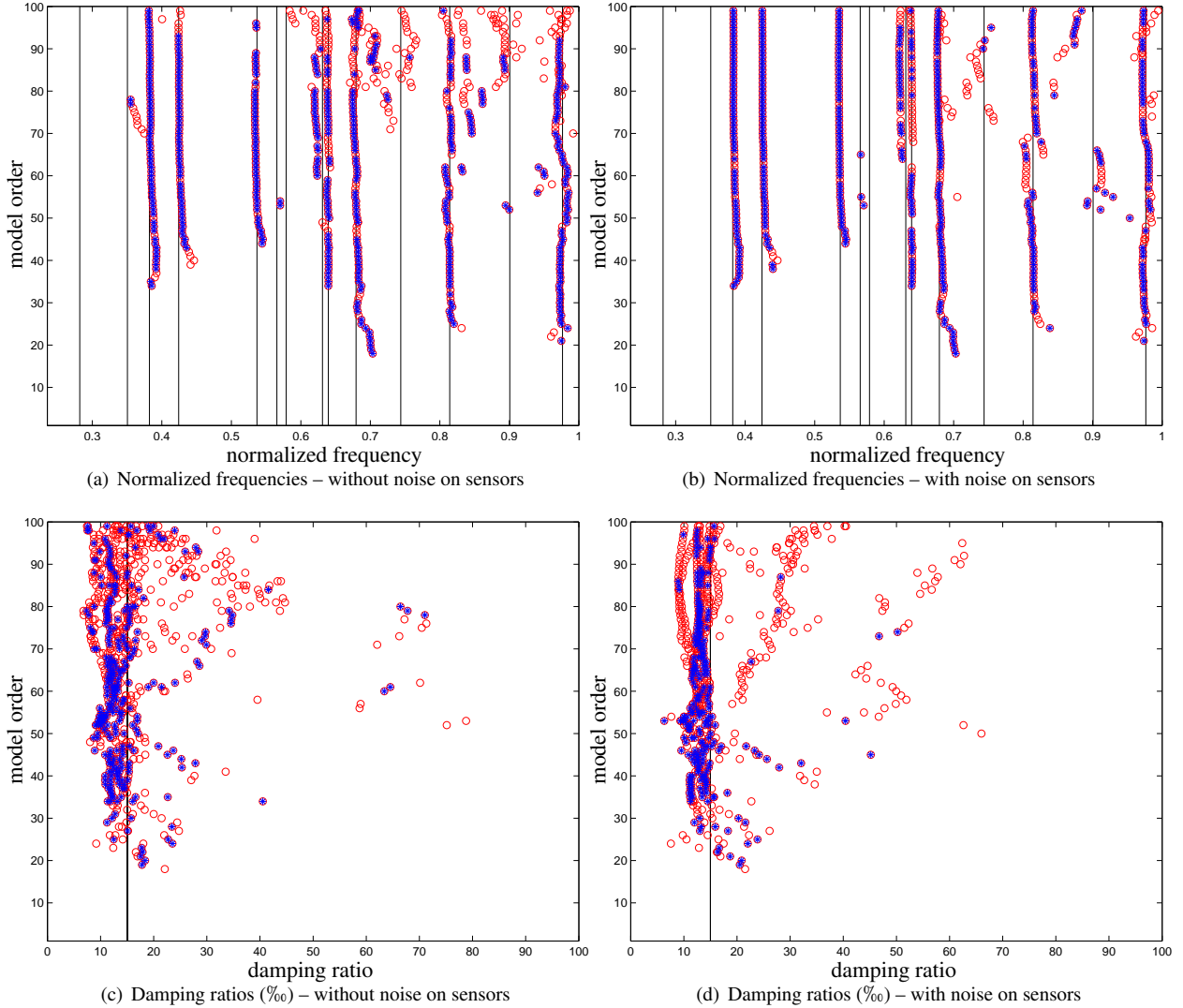
(d) Damping ratios (‰) – with noise on sensors

Figure 4: OOort^2 - Monte Carlo stabilization charts with success test values for orders 1 to 100

In terms of identification: In this specific case, the identification is not well stabilized and numerous spurious modes appear. In particular, the damping identification is quite biased. This may be due to ill-conditioning of $\mathcal{Y}^-\mathcal{Y}^{-^T}$. Noise on sensors seems to help with stabilization. This effect can be explained by a sensors decorrelation and then an improvement of the conditioning of $\mathcal{Y}^-\mathcal{Y}^{-^T}$. As with the OOcov algorithm, noise on sensors reduces stability of spurious modes comparing all identifications and then reduces the number of spurious modes that are kept. Notice that even if this algorithm provides badly stabilized identifications, it is robust to and even improved by noise on sensors both for frequencies and damping ratios.

In terms of uncertainty estimations: As with the OOcov algorithm, the accuracy of the uncertainty estimations

increases with the stability of the modes. However, numerous modes can be unstable with respect to model order but still validated in terms of uncertainty estimations. Concerning noise effects, adding noise on sensors improves uncertainty estimations, like for the identification. These results show that uncertainty estimations may be accurate even if identification is not stable with respect to the model order.
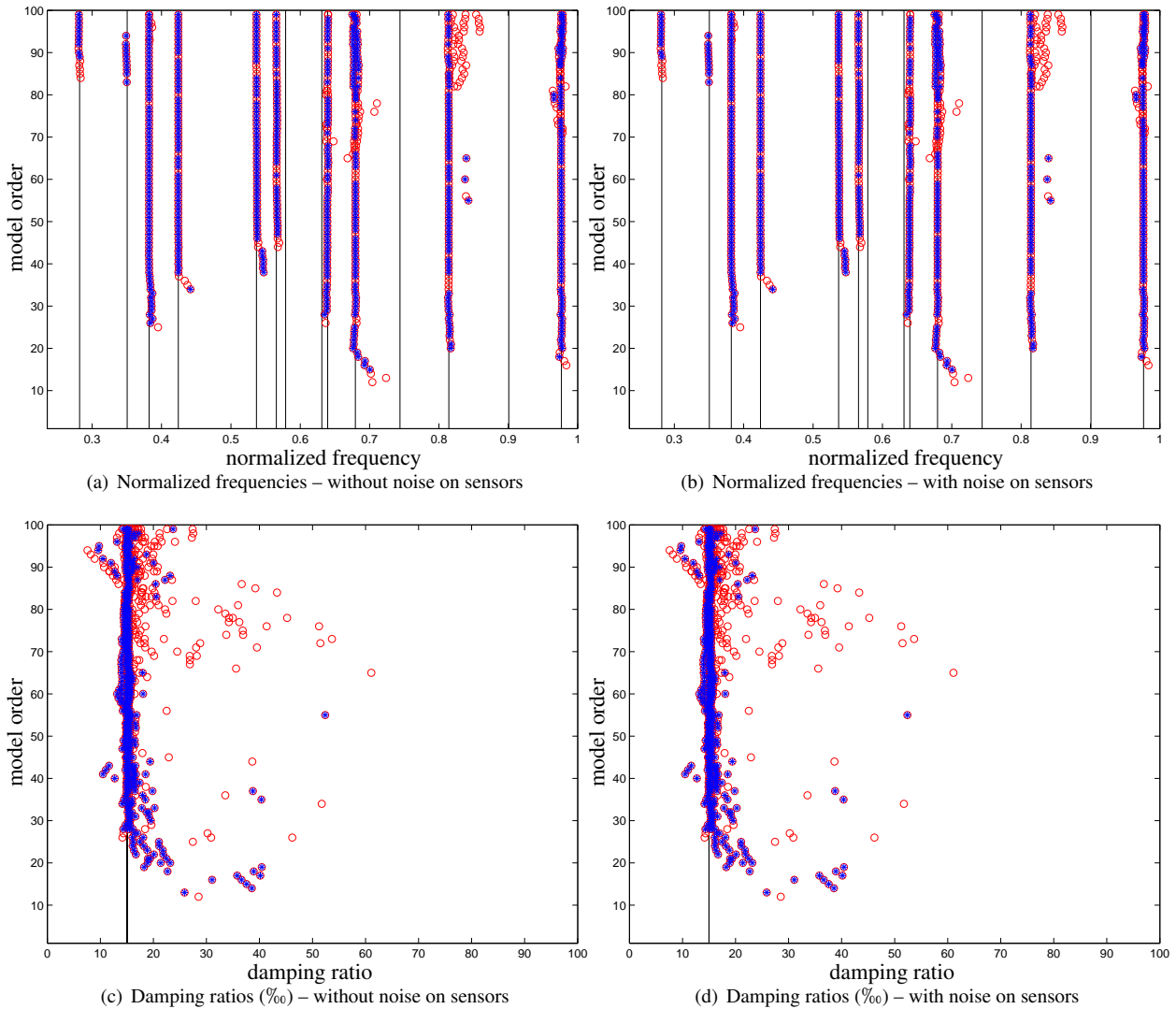
7.3.3. IOcov*: Input/output covariance-driven – Figure 5*

(a) Normalized frequencies – without noise on sensors

(b) Normalized frequencies – with noise on sensors

(c) Damping ratios (‰) – without noise on sensors

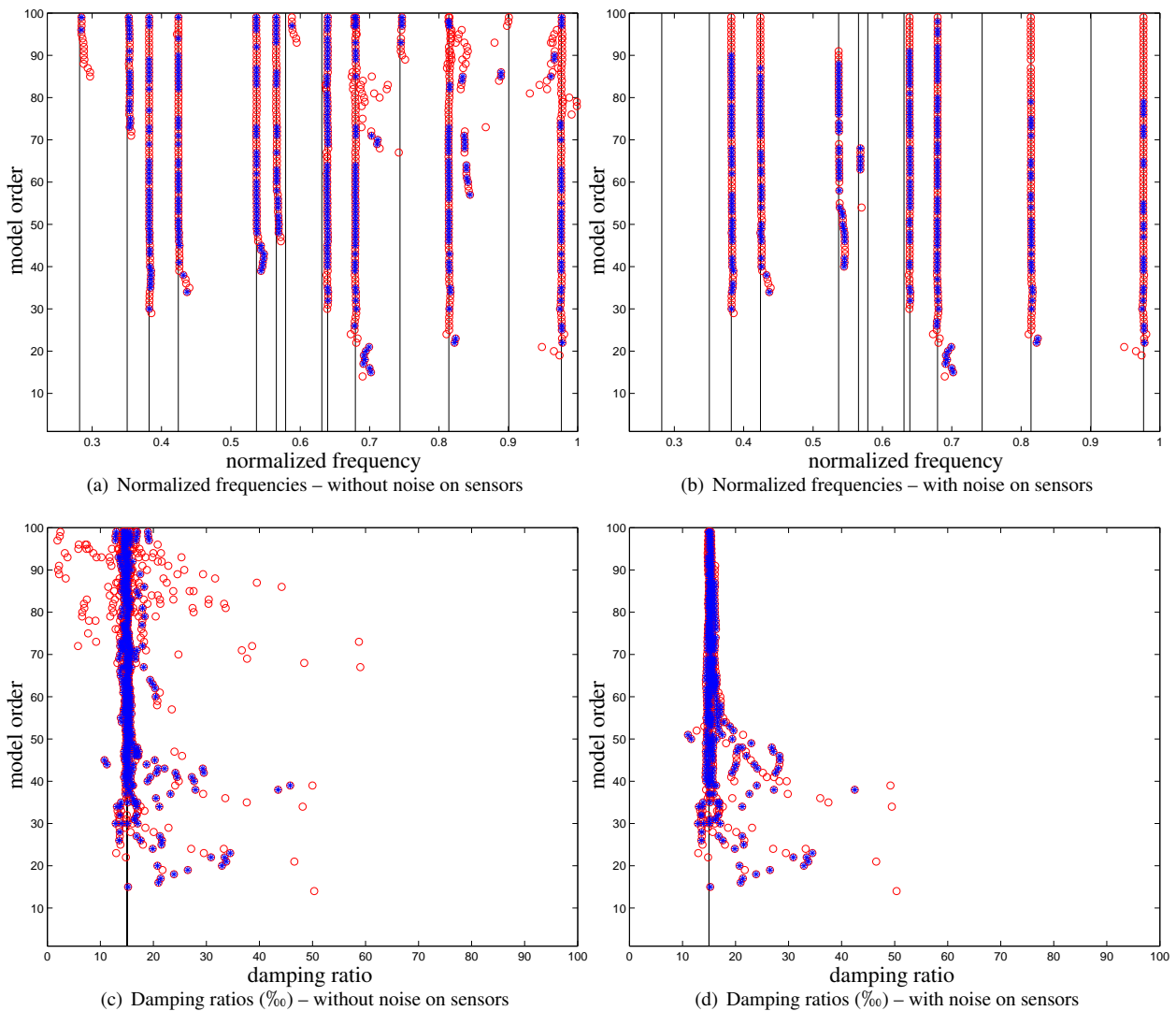(d) Damping ratios (‰) – with noise on sensors

Figure 5: IOcov – Monte Carlo stabilization charts with success test values for orders 1 to 100

In terms of identification: In this specific case, ten modes are identified, three more than in output-only algorithms, and modes are found "sooner" (i.e. for lower orders). This may be due to the use of input information which eases modal identification. Noise on sensors seems to have no impact at all on the identification. We will see that it is

the only algorithm which is completely robust to noise on sensors.

In terms of uncertainty estimations: Except for some spurious modes, uncertainty estimations on the modal identification are almost all validated for both frequencies and damping ratios. As for identification, noise on sensors has no effect on the validity of uncertainty estimations. In this specific case, the IOcov algorithm gives much better results than the output-only algorithms.

7.3.4. IOort[2]: Input/output orthogonal projection data-driven squared – Figure 6

(a) Normalized frequencies – without noise on sensors

(b) Normalized frequencies – with noise on sensors

(c) Damping ratios (‰) – without noise on sensors

(d) Damping ratios (‰) – with noise on sensors

Figure 6: IOort[2] - Monte Carlo stabilization charts with success test values for orders 1 to 100

In terms of identification: In this specific case, the same ten modes are identified as with the IOcov algorithm, plus one additional mode (the eleventh physical mode of the system) which becomes stable around order 90. Two

other modes (the seventh and thirteenth physical modes of the system) seem to appear around order 95. However, the frequency of the seventh mode is not well identified and what looks like the thirteenth mode might only be a spurious mode. Depending on the mode, identification using the IOort^2 algorithm is either equivalent or better than using the IOcov algorithm. However, when noise is applied on sensors, the number of identified modes decreases. Identification becomes close to the performance of the OOcov algorithm but remains better in terms of stability with respect to model order, especially for damping ratio estimations. This algorithm is not as robust to noisy sensor conditions as the IOcov algorithm due its theoretical properties (see Section 3.2.3.b) but it remains accurate enough to identify well excited modes.

In terms of uncertainty estimations: This algorithm is not really accurate in terms of uncertainty estimations. It barely validates half the identified modes. However, conclusions stated about previous algorithms are still valid. Uncertainty estimation is more accurate on physical modes than on spurious modes. Noise on sensors seems to have no significant effect on uncertainty estimation. This algorithm can be considered robust to noise on sensors but in this specific case less modes are validated than using the IOcov algorithm.

7.3.5. IOobl^2: Input/output oblique projection data-driven squared – Figure 7

In terms of identification: Identification results are similar to the IOort^2 algorithm in terms of both number and stability of identified modes. Even the sensor noise effect is equivalent for these two algorithms. It significantly reduces the number of identified modes. However, when noise is applied on sensors the IOobl^2 algorithm gives better identification results than the IOort^2 algorithm, especially for the fifth and sixth physical modes of the system.

In terms of uncertainty estimations: In this specific case, this algorithm is the worst proposed algorithm for uncertainty estimations. As with IOort^2, less than half of the identified modes are validated in terms of uncertainty estimations. However, when noise is applied on sensors, it becomes one of the best algorithm to estimate uncertainty both for frequencies and damping ratios. Results become equivalent to results obtained using the IOcov algorithm, except that less (physical and spurious) modes are identified with IOobl^2 than with IOcov.

7.3.6. Global conclusions on the simulations

In this section five modal identification algorithms have been tested in terms of both identification and uncertainty estimations. Note that these conclusions should not be generalized and may be specific to the studied case.

In terms of identification: Based on Figure 3–7 it is clear that using input information eases modal identification. Modes are found sooner (i.e. for lower orders) and are more numerous in the input/output framework than in the output-only framework. In the output-only framework, the OOcov algorithm provides a more stable identification in general and better damping ratios than the OOort^2 algorithm. This may be due to ill conditioning of $\mathcal{Y}^-\mathcal{Y}^{-^T}$ (see Eq. (11)). In the input/output framework, the three studied algorithms are equivalent. However, a slight distinction can be made concerning the seventh and the eleventh physical modes. Only the IOort^2 algorithm identifies both of them. Until order 100, the IOobl^2 algorithm identifies only the seventh one and the IOcov algorithm none of them.

29

(a) Normalized frequencies – without noise on sensors

(b) Normalized frequencies – with noise on sensors

(c) Damping ratios (‰) – without noise on sensors
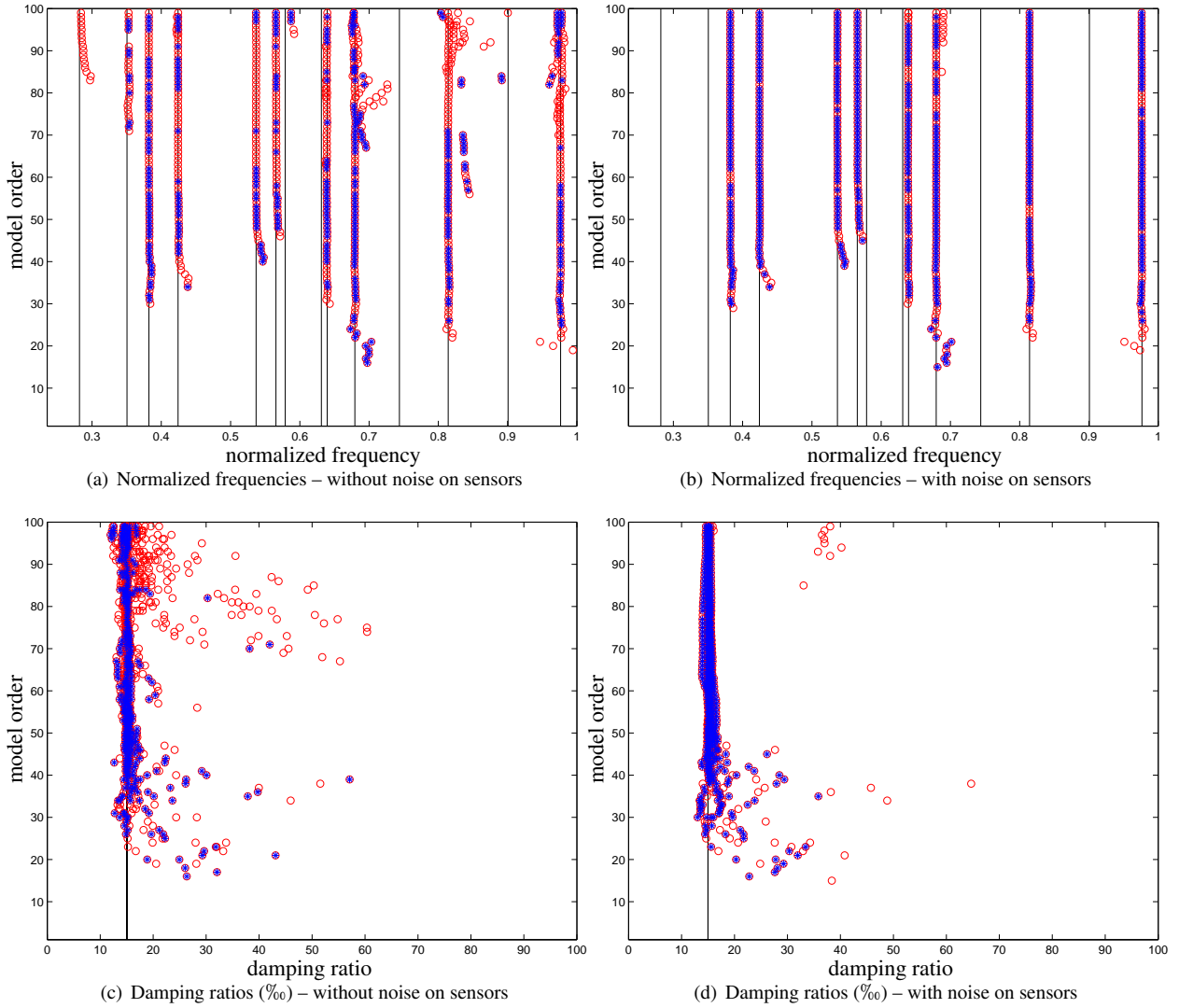
(d) Damping ratios (‰) – with noise on sensors

Figure 7: IOobl^2 – Monte Carlo stabilization charts with success test values for orders 1 to 100

Concerning noise on sensors effect, it was shown that it strongly depends on the algorithm. Remind that the presented results are an average of stable mode estimates for different identifications. Based on that, noise has no effect on the IOcov algorithm and almost no effect on the OOcov algorithm. It reduces the number of modes and improves their stability when using OOort^2, IOort^2 and IOobl^2.

In terms of uncertainty estimations: Concerning the use of input information, it seems to have no clear effect on uncertainty estimations. Validations are not better in the input/output framework than in the output-only framework. Both the OOort^2 and IOcov algorithms provide much more validated uncertainty estimations than OOcov, IOort^2 and IOobl^2. The OOcov algorithm is the only studied algorithm which is not robust to noise on sensors. Noise has absolutely no influence on uncertainty estimated by the IOcov algorithm. It has no clear impact when using
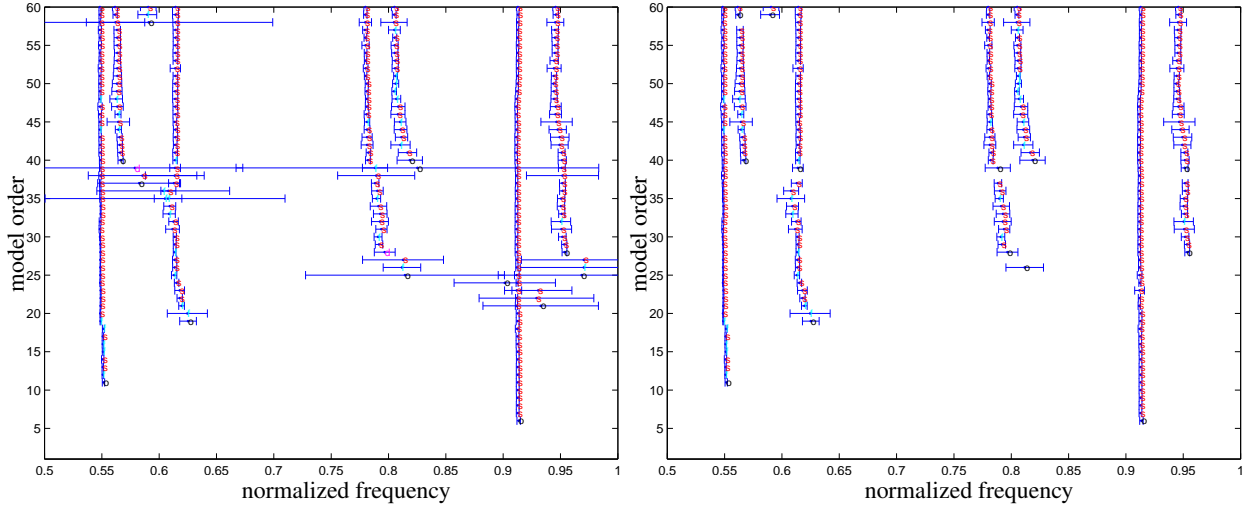
Figure 8: OOcov: Output-only covariance-driven – stabilization charts of normalized frequencies (left: full, right: cleaned)

either OOort^2 or IOort^2. Finally, noise on sensors makes IOobl^2 the best proposed algorithm in terms of uncertainty estimations.

8. Application to real data from in-flight vibration tests

After validation on $N_s = 200$ simulations, the five considered algorithms are used to compute identifications with uncertainty estimations based on real measured data. These data have been obtained during in-flight vibration tests of an unnamed aircraft. Known excitations are symmetric white noises performed by both left and right control surfaces of the aircraft during $T = 180$ s and sampled at rate $F_s = 250$ Hz. Responses are obtained by $N_o = 50$ sensors along the aircraft under both the known excitation and in addition unknown external environmental excitations. To estimate auto- and cross-covariances (see Appendix A), the $N = 44,828$ time samples of each input and output are divided into 10 blocks. Estimation results are presented within classical stabilization diagrams added by confidence intervals $(f \pm \sigma_f)$. Only frequencies are presented. Modes are normalized by the highest displayed frequency value. Estimates are then plotted from 0 to 1 for the frequency range and for orders from 1 to 60.

It is empirically observed that spurious modes often have higher uncertainty values than physical ones. Confidence intervals can then be used to clean up stabilization charts by applying a threshold based on uncertainty values. In the figures, the original uncleaned stabilization diagrams are shown on the left, whereas the cleaned stabilization diagrams using the uncertainty-based threshold are shown. Thresholds on uncertainties have been set to 0.25 Hz for frequencies and 100 ‰ for damping ratios. Results are as follows:

• OOcov, Figure 8: Seven modes are clearly identified. An eighth one appears at order 58. There are few spurious modes. Uncertainty-based clean up is relevant. It removes only spurious or unstable modes. One can observe that more stable modes have a smaller uncertainty estimation. Moreover, uncertainties are coherent with operationally
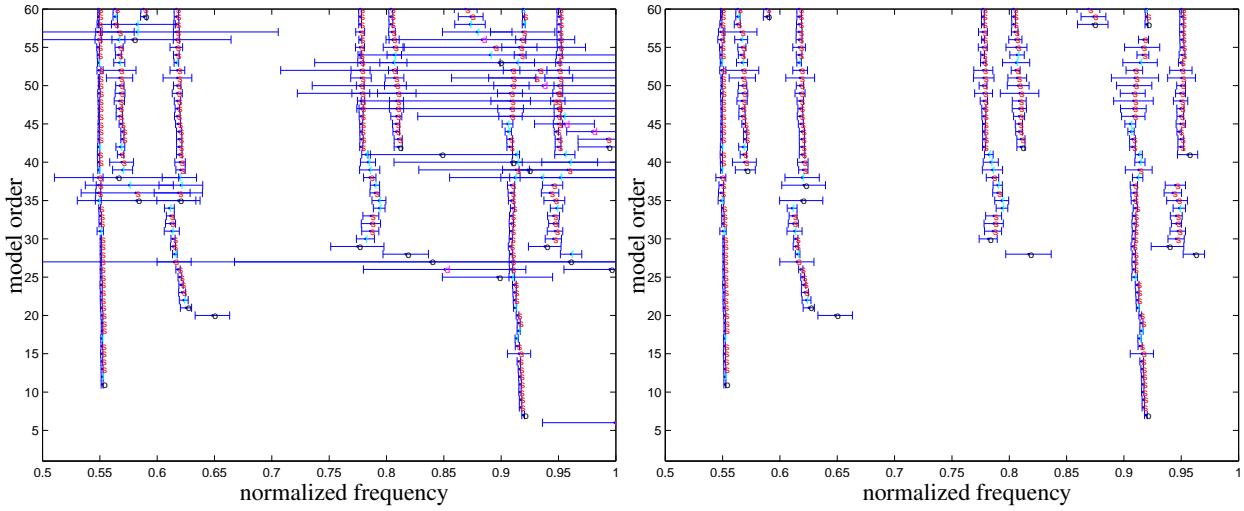
31

Figure 9: OOort^2: Output-only orthogonal projection data-driven squared – stabilization charts of normalized frequencies (left: full, right: cleaned)
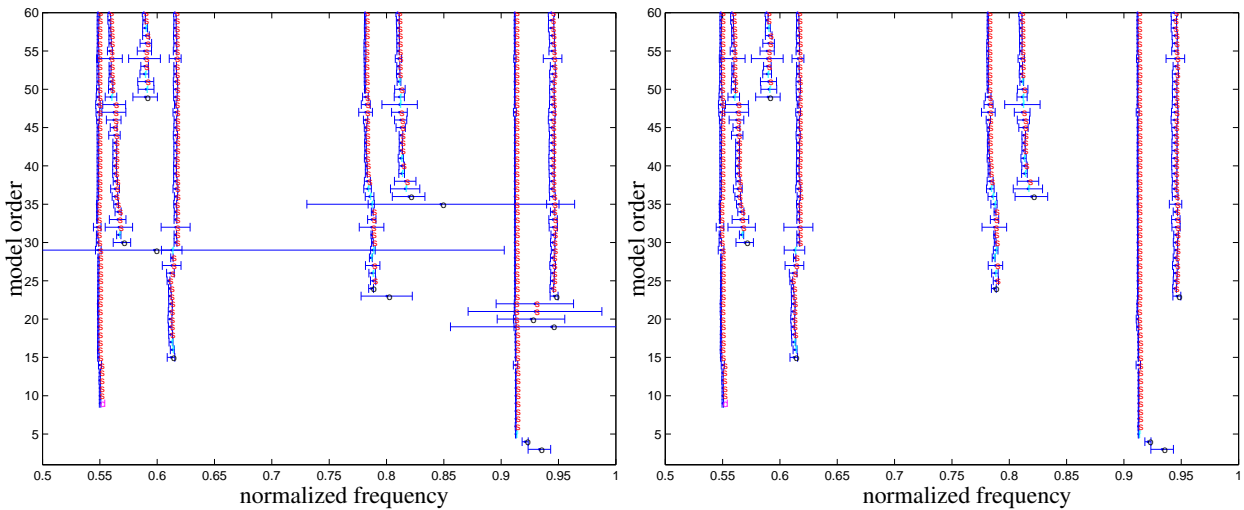

Figure 10: IOcov: Input/output covariance-driven – stabilization charts of normalized frequencies (left: full, right: cleaned)

observed deviations.

• OOort^2, Figure 9: As with the OOcov algorithm, seven modes are clearly identified. However, they are much less stable with respect to model order. Moreover, this identification contains more spurious modes than the identification obtained with OOcov. This example shows how useful the uncertainty-based clean up may be.

• IOcov, Figure 10: It is the first presented algorithm using input information. On can observe that modes appear "sooner" (i.e. for lower orders) when input information is used. Here, modes are also more stable and have smaller uncertainty values. Only a few spurious modes are identified (only 7 points in the whole stabilization charts). Once again, as with previous algorithms, one can observe that uncertainty estimates are smaller for more stable modes.

• IOort^2, Figure 11: This algorithm gives the worst identification and higher uncertainty values. However it clearly
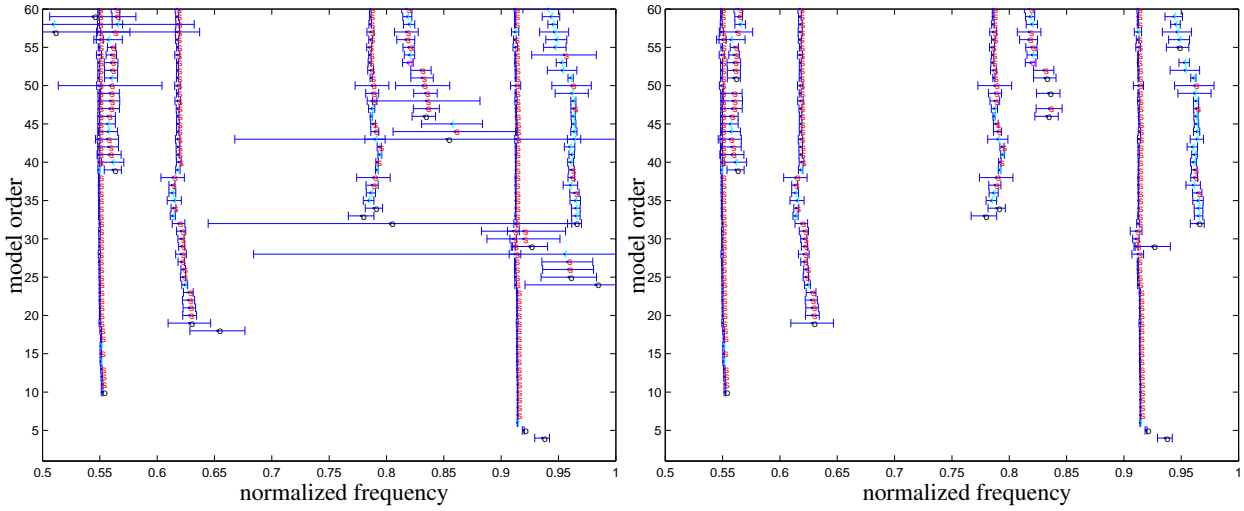
Figure 11: IOort^2: Input/output orthogonal projection data-driven squared – stabilization charts of normalized frequencies (left: full, right: cleaned)

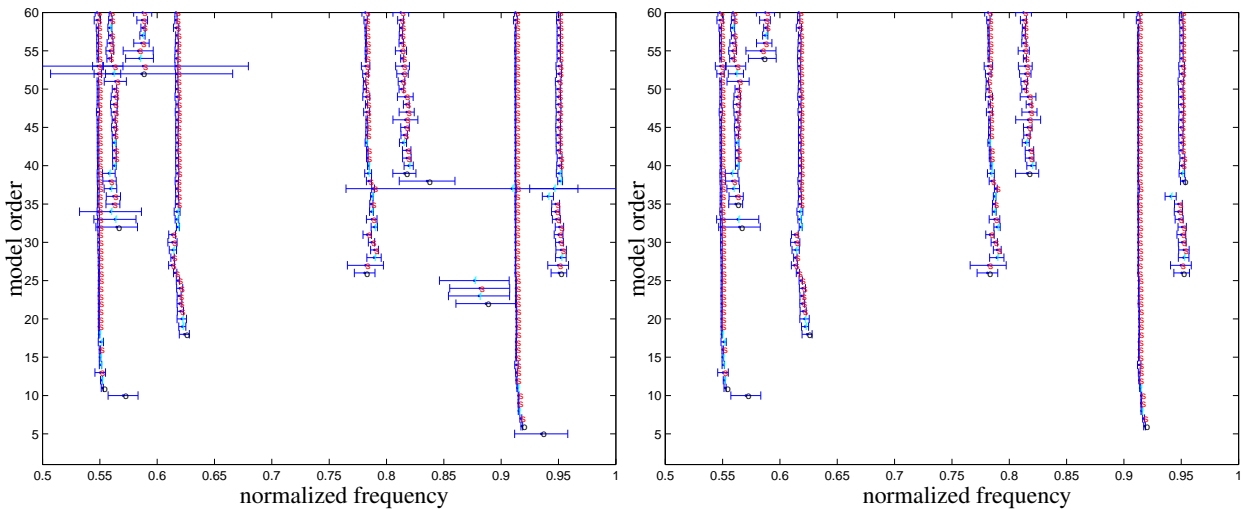shows as well that the more stable a mode, the smaller its uncertainty estimation.

Figure 12: IOobl^2: Input/output oblique projection data-driven squared – stabilization charts of normalized frequencies (left: full, right: cleaned)

• IOobl^2, Figure 12: Identification is similar to the IOcov algorithm. In this specific case, these two algorithms provide the best identification and small uncertainty values.

9. Discussion

In this paper, variance estimates were developed for output-only and input/output system identification schemes generalizing [27, 28] previously developed for the OOcov (output-only covariance-driven) subspace algorithm. Based on this method, variance expressions have been developed for OOort^2 (output-only orthogonal projection data-driven

33

squared), IOcov (input/output covariance-driven), IOort^2 (input/output orthogonal projection data-driven squared) and IOobl^2 (input/output oblique projection data-driven squared) algorithms. Providing also a computation scheme for the mode shapes, the paper focusses on the estimation of the modal frequencies and damping ratios, which are parameters of interest for operational modal analysis during in flight tests.

These five modal identification algorithms have been validated in terms of both identification and variance estimations. In this validation, their robustness to sensor noise has also been tested. Finally they have been applied on real measurements. Besides an academic test case on a mass-spring chain, simulations and real data were provided on an unnamed aircraft from industrial cases and results have been normalized to allow publication.

The validation of the modal identification algorithms and their variance computation were based on Monte Carlo simulations. Output-only and input/output algorithms have been compared. In the investigated examples, the number of identified modes increases and modes appear at lower orders when using input information. Furthermore, the variance decreases with input information when using the appropriate algorithms. The validation of the variance computation was equally comparable with or without using inputs. In the academic test case, the results of the variance computation were close to the respective variances from the Monte Carlo simulations for all algorithms, including the use of sensor noise. In the industrial test cases, the effect of sensor noise has been shown strongly dependent on the respective algorithms. For the OOcov algorithm a degradation of both identification and variance estimations could be observed with sensor noise. For the OOort^2 algorithm, it was a slight improvement of both identification in terms of stability with respect to model order and variance estimations. Sensor noise did not change validation results for the IOcov algorithm. In terms of identification, impact of noise was similar with IOort^2 and IOobl^2 algorithms. One could observe a reduction of the number of identified modes. Only the more stable ones were found again with sensor noise. However, in terms of uncertainty validation, a huge improvement was observed for the IOobl^2 algorithm.

After validation, the considered algorithms have been tested on real-life measured data obtained during in-flight vibration tests. It appeared that non-physical poles (i.e., spurious modes) have often much higher variance values than physical ones (real modes of the structure). This allows to use variance estimation to clean up stabilization charts and then to ease their interpretation during identification. Furthermore, the mode selection from the stabilization diagram could be supported by the actual computed uncertainty bounds. While different model orders indeed help to reliably identify the modes, the results show that the model order should not be too high in same cases. There, the results deteriorate slightly or the uncertainty estimation is less validated for high orders, which might be due to overfitting or worse conditioning of the involved matrices for (pseudo-)inverting in the subspace methods at high orders.

Concerning the efficiency of the algorithms, it seems that input information eases identification and reduces uncertainty values. However, if only outputs are available, output-only algorithms give relevant results. If the five algorithms had to be classified in terms of both identification and uncertainty estimations, their ranking could be as follows:

$$\text{IOcov} > \text{IOobl}^2 > \text{OOcov} > \text{IOort}^2 > \text{OOort}^2$$

This classification has to be taken carefully. It cannot be generalized and it could be specific to the test case.

Recall that the chosen approach for variance computation cannot be used directly on data-driven algorithms. Indeed, the column dimension of data-driven matrices for subspace identification increases with the number of samples. Hence, they are not compatible with a sample covariance formulation and the perturbation-based approach from [26–28] cannot be applied directly. Fortunately, as shown in the current paper, the singular value decomposition properties ensure that the proposed variance computation scheme for the squared subspace algorithms provides equivalent uncertainty bounds for the pair (A, C) obtained from the data-driven algorithms. Thus, frequency, damping ratio and mode shape estimates are exactly the same using either data-driven algorithms or their squared counterpart. This allows to use the proposed scheme on the squared subspace matrices to estimate variances of modal parameters from data-driven algorithms. Still, the actual implementation of the data-driven algorithms may differ from the direct computation of the subspace matrices and implies usually computation procedures such as QR decomposition [2], which is a supplementary numerical step before the SVD to obtain the observability matrix. While such a step does not change the variance expressions thanks to the equivalence of the algorithms, it should be incorporated as in [41]. Furthermore, the efficient numerical framework for the variance computation of [28] should be applied to the new methods from this paper. With this framework, computation times for all the methods are less than a minute for a whole stabilization diagram containing model orders until 100 on a standard desktop computer and hardly differ between the different subspace methods.

Finally, we recall that the input/output methods in this paper have been derived under the premise that the known inputs are the realization of a stochastic process. The Monte Carlo simulations have shown that this assumption gives coherent variance estimates with the methods of this paper. The stochastic properties of the inputs contribute to the variance estimates of the modal parameters. They do not necessarily mean an increase of the variance estimates of the modal parameters. Taking them into account is necessary for a correct variance estimation. In particular, it has been demonstrated in Section 6.2 that the variance is correctly estimated at zero in the case of no unknown inputs and no sensor noise when taking the stochastic properties of the inputs into account. Neglecting the stochastic properties of the inputs has led to a variance estimate different from zero in this case.

Deterministic inputs are not considered in this paper. Under a few assumptions for the known inputs such as persistent excitation [2], quasi-stationarity for deterministic signals such as in [21], and under the assumption of both known deterministic input and unknown stochastic noise, the current approach would be applicable provided the uncertainty is derived with respect to the noise only, and as long as auto- and cross-covariances can be defined for the known input as in [9]. The corresponding variance computation could be handled by the current approach as a special case, where the variance with respect to the functions of the deterministic and known inputs is taken as zero.

10. Conclusion

Modal identification is commonly used in industry in the design process for a structure. The uncertainty information related to the identified modal parameters is at least as valuable as the modal parameters themselves in order to judge their quality and confidence. Moreover, in some highly dangerous and expensive vibration tests (e.g. flutter tests [43]), uncertainty estimations may be very helpful in terms of both safety and cost reduction. In this paper, the main results are new expressions for variance estimation for four output-only and input/output subspace-based modal identification algorithms. It has been shown that considering the stochastic properties of the known input leads to a correct formulation of the variance estimation scheme. Concerning the considered subspace algorithms, it has been shown how data-driven algorithms as found in [2, 32] can be related to an equivalent covariance-driven form yielding identical estimates for the pair (A, C). From this remark, covariance estimates for (A, C) and subsequently for frequencies, damping ratios and mode shapes were derived. The validation of this approach has been carried out with Monte Carlo simulations, and it has been applied on real in-flight aircraft data. Further work should include the variance estimation for the pair (B, D) as well as for the modal mass.

Appendix A. Computation of auto- and cross-covariance functions

Let $\mathcal{F}$ be the discrete Fourier transform operator, and let a and b be placeholders for two discrete-time signals (e.g. u and y, or y and y) whose cross-covariance shall be computed. The signals are divided into n_b segments $a^{(k)}$ and $b^{(k)}$, $k = 1, \ldots, n_b$, of equal size (with or without overlap). An apodization window is often applied to these segments to reduce the leakage effect. Fourier transformations of $a^{(k)}$ and $b^{(k)}$ are given by $a_{\mathcal{F}}^{(k)} = \mathcal{F}[a^{(k)}]$ and $b_{\mathcal{F}}^{(k)} = \mathcal{F}[b^{(k)}]$. PSD functions of each segment k between $a_{\mathcal{F}}^{(k)}$ and $b_{\mathcal{F}}^{(k)}$ are then $S_{[a^{(k)}, b^{(k)}]} = a_{\mathcal{F}}^{(k)} (b_{\mathcal{F}}^{(k)})^*$, where * denotes the conjugate transpose, and PSD functions of a and b are estimated by averaging as $\overline{S}_{[a,b]} \overset{\text{def}}{=} \frac{1}{n_b} \sum_{k=1}^{n_b} S_{[a^{(k)}, b^{(k)}]}$. Finally, cross-covariance functions are obtained by an inverse Fourier transformation as $R_{[a,b]} = \mathcal{F}^{-1}[\overline{S}_{[a,b]}]$. An advantage of this approach for computing cross-covariance functions is the possibility to reduce the frequency band by applying a rectangular window on the estimation of the power spectral density functions. Note that, due to the linearity of $\mathcal{F}$, one can also first apply the Fourier transform to the n_b PSD functions and then average these n_b cross-covariance functions to obtain the estimation $R_{[a,b]}$. Also, an important remark concerns the stochastic properties of $\overline{S}_{[a,b]}$. If $a^{(k)}$ and $b^{(k)}$ are ergodic random processes, $a_{\mathcal{F}}^{(k)}$ and $b_{\mathcal{F}}^{(k)}$ are also ergodic random processes by linearity of the Fourier transformation. The n_b PSD functions $S_{[a^{(k)}, b^{(k)}]}$ are n_b independent and identically distributed (i.i.d.) ergodic random processes having the same expectation $\mathbf{E}[S_{[a^{(k)}, b^{(k)}]}] = S_{[a,b]}$ and same covariance $\Sigma_{S_{[a^{(k)}, b^{(k)}]}} = \Sigma_{S_{[a', b']}}$. Their mean $\overline{S}_{[a,b]}$ is an estimate of $S_{[a,b]}$ and, by the CLT, follows a normal distribution with expectation $\mathbf{E}[\overline{S}_{[a,b]}] = S_{[a,b]}$ and covariance $\Sigma_{\overline{S}_{[a,b]}} = \frac{1}{n_b} \Sigma_{S_{[a', b']}}$.

Appendix B. From covariance of $\mathcal{H}$ to the modal parameters

Let $\mathcal{H}$ be the subspace matrix corresponding to a chosen subspace algorithm, which is given together with an estimate of $\text{Cov}(\text{vec}(\mathcal{H}))$. This covariance is propagated to the observability matrix Γ_p, then to the system matrices A and C and finally to the modal parameters, as outlined as follows. Matrix Γ_p is obtained from an SVD of $\mathcal{H}$ in Eq. (15) as $\Gamma_p = U_1 S_1^{1/2}$. The sensitivity of Γ_p is obtained by perturbation as $\Delta\Gamma_p = \Delta U_1 S_1^{1/2} + U_1 \Delta S_1^{1/2}$ and finally, using $\mathcal{J}_{\Gamma,\mathcal{H}}$ as detailed in [27, 28],

$$\text{Cov}(\text{vec}(\Gamma_p)) = \mathbf{E}\left[\text{vec}(\Delta\Gamma_p)\text{vec}(\Delta\Gamma_p)^T\right] = \mathcal{J}_{\Gamma,\mathcal{H}}\text{Cov}(\text{vec}(\mathcal{H}))\mathcal{J}_{\Gamma,\mathcal{H}}^T$$

Then, A and C are obtained from Γ_p, and perturbations on A an C are obtained by perturbations of the equations in Section 3.3. Then $\text{vec}(\Delta A) = \mathcal{J}_{A,\Gamma}\text{vec}(\Delta\Gamma_p)$ and $\text{vec}(\Delta C) = \mathcal{J}_{C,\Gamma}\text{vec}(\Delta\Gamma_p)$. The covariances of A and C write as

$$\Sigma_{A,C} = \text{Cov}\left(\begin{bmatrix}\text{vec}(A)\\\text{vec}(C)\end{bmatrix}\right) = \begin{bmatrix}\mathcal{J}_{A,\Gamma}\\\mathcal{J}_{C,\Gamma}\end{bmatrix}\mathcal{J}_{\Gamma,\mathcal{H}}\Sigma_{\mathcal{H}}\mathcal{J}_{\Gamma,\mathcal{H}}^T\begin{bmatrix}\mathcal{J}_{A,\Gamma}^T & \mathcal{J}_{C,\Gamma}^T\end{bmatrix}$$

where $\mathcal{J}_{A,\Gamma}$ and $\mathcal{J}_{C,\Gamma}$ are detailed in [27, 28]. Covariances of modal parameters are finally as in [27, 28].

Appendix C. Proof of Lemma 6

Deriving the product $\mathcal{R}_a\mathcal{R}_b^\dagger\mathcal{R}_c$, it follows $\Delta(\mathcal{R}_a\mathcal{R}_b^\dagger\mathcal{R}_c) = \Delta\mathcal{R}_a\mathcal{R}_b^\dagger\mathcal{R}_c + \mathcal{R}_a\Delta(\mathcal{R}_b^\dagger)\mathcal{R}_c + \mathcal{R}_a\mathcal{R}_b^\dagger\Delta\mathcal{R}_c$, and it remains to show the identity

$$\mathcal{R}_a\Delta(\mathcal{R}_b^\dagger)\mathcal{R}_c = -\mathcal{R}_a\mathcal{R}_b^\dagger\Delta(\mathcal{R}_b)\mathcal{R}_b^\dagger\mathcal{R}_c \tag{C.1}$$

to conclude the proof. Let the SVD of $\mathcal{B}$ be given,

$$\mathcal{B} = \begin{bmatrix}U_1 & U_2\end{bmatrix}\begin{bmatrix}S_1 & 0\\0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\V_2^T\end{bmatrix} = U_1 S_1 V_1^T,$$

where S_1 contains all non-zero singular values, and $U_1^T U_1 = I$. From the SVD of $\mathcal{B}$ it follows the SVD of $\mathcal{R}_b = \mathcal{B}\mathcal{B}^T = U_1 S_1^2 U_1^T$ and its pseudo-inverse $\mathcal{R}_b^\dagger = U_1 S_1^{-2} U_1^T$. Both sides of (C.1) are now developed. It holds

$$\begin{aligned}\mathcal{R}_a\Delta(\mathcal{R}_b^\dagger)\mathcal{R}_c &= \mathcal{A}\mathcal{B}^T\Delta(\mathcal{B}\mathcal{B}^T)^\dagger\mathcal{B}\mathcal{C}^T\\
&= \mathcal{A}V_1 S_1 U_1^T\Delta(U_1 S_1^{-2} U_1^T)U_1 S_1 V_1^T\mathcal{C}^T\\
&= \mathcal{A}V_1 S_1 U_1^T\left(\Delta U_1 S_1^{-2} U_1^T + U_1\Delta S_1^{-2} U_1^T + U_1 S_1^{-2}\Delta U_1^T\right)U_1 S_1 V_1^T\mathcal{C}^T\\
&= \mathcal{A}V_1 S_1\left(U_1^T\Delta U_1 S_1^{-2} - S_1^{-2}\Delta S_1^2 S_1^{-2} + S_1^{-2}\Delta U_1^T U_1\right)S_1 V_1^T\mathcal{C}^T\end{aligned} \tag{C.2}$$

and

$$\begin{aligned}-\mathcal{R}_a\mathcal{R}_b^\dagger\Delta(\mathcal{R}_b)\mathcal{R}_b^\dagger\mathcal{R}_c &= -\mathcal{A}(U_1 S_1 V_1^T)^T(U_1 S_1^{-2} U_1^T)\Delta(U_1 S_1^2 U_1^T)(U_1 S_1^{-2} U_1^T)(U_1 S_1 V_1^T\mathcal{C}^T)\\
&= \mathcal{A}V_1 S_1 S_1^{-2} U_1^T\left(-\Delta U_1 S_1^2 U_1^T - U_1\Delta S_1^2 U_1^T - U_1 S_1^2\Delta U_1^T\right)U_1 S_1^{-2}S_1 V_1^T\mathcal{C}^T\\
&= \mathcal{A}V_1 S_1\left(-S_1^{-2}U_1^T\Delta U_1 - S_1^{-2}\Delta S_1^2 S_1^{-2} - \Delta U_1^T U_1 S_1^{-2}\right)S_1 V_1^T\mathcal{C}^T.\end{aligned} \tag{C.3}$$

Since $U_1^T U_1 = I$, it follows $\Delta U_1^T U_1 = -U_1^T \Delta U_1$ and hence (C.2) and (C.3) are equivalent. Thus, (C.1) holds, proving the assertion.

Appendix D. Sensitivity of IOobl^2: Input/output orthogonal projection data-driven squared

To derive $\text{vec}(\Delta \mathcal{H}_{\text{IOobl}^2})$ in Eq. (21) with $\mathcal{H}_{\text{IOobl}^2} = \Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \Omega_1^T$ and

$$\Omega_1 = \left(\mathcal{Y}^+/\mathcal{U}^{+\perp}\right)\left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)^T, \quad \Omega_2 = \left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)\left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)^T, \quad \text{and} \quad \mathcal{R}_{11} = \mathcal{W}^- \mathcal{W}^{-T},$$

we first show

$$\Delta(\Omega_1 \Omega_2^\dagger \mathcal{R}_{11}) = \Delta(\Omega_1)\Omega_2^\dagger \mathcal{R}_{11} - \Omega_1 \Omega_2^\dagger \Delta(\Omega_2)\Omega_2^\dagger \mathcal{R}_{11} + \Omega_1 \Omega_2^\dagger \Delta(\mathcal{R}_{11}). \tag{D.1}$$

First, consider the "noisy" case with the presence of non-zero unmeasured inputs (process noise) $\tilde{u}$ and sensor noise v. For this case, it has been shown in [44] that the term $\left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)$ is of full row rank. Hence, Ω_2 is invertible and Eq. (D.1) holds since $\Omega_2^\dagger = \Omega_2^{-1}$ and due to the property $\Delta(\Omega_2^{-1}) = -\Omega_2^{-1}\Delta(\Omega_2)\Omega_2^{-1}$.

Second, consider the "no noise" case with no unmeasured inputs and no sensor noise. For this case, it has been shown in [2, Appendix A.1] that the term $\left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)$ is indeed rank deficient, but that $\left(\mathcal{W}^-/\mathcal{U}^{+\perp}\right)$ and $\mathcal{W}^-$ have the same column space. Due to this property, Lemma 6 can be applied to show Eq. (D.1).

Hence, Eq. (D.1) holds for both the "noisy" and "no noise" cases. Then, a subsequent relationship for $\Delta \mathcal{H}_{\text{IOobl}^2} = \Delta(\Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \Omega_1^T)$ is straightforward to show, and the relationship for $\text{vec}(\Delta \mathcal{H}_{\text{IOobl}^2})$ in Eq. (21) follows after vectorization and re-ordering, where

$$\begin{cases} \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(1)} = \mathcal{J}_{\mathcal{H},\Omega_1} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(3)} = \mathcal{J}_{\mathcal{H},\Omega_1}\mathcal{J}_{\Omega_1,\mathcal{R}}^{(2)} + \mathcal{J}_{\mathcal{H},\Omega_2}\mathcal{J}_{\Omega_2,\mathcal{R}}^{(2)} \\ \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(2)} = \mathcal{J}_{\mathcal{H},\Omega_1}\mathcal{J}_{\Omega_1,\mathcal{R}}^{(1)} & \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(4)} = \mathcal{J}_{\mathcal{H},\Omega_1}\mathcal{J}_{\Omega_1,\mathcal{R}}^{(3)} + \mathcal{J}_{\mathcal{H},\Omega_2}\mathcal{J}_{\Omega_2,\mathcal{R}}^{(1)} \end{cases} \quad \mathcal{J}_{\mathcal{H},\mathcal{R}}^{(5)} = \mathcal{J}_{\mathcal{H},\Omega_2} + \mathcal{J}_{\mathcal{H},\mathcal{R}_{11}}$$

$$\begin{cases} \mathcal{J}_{\mathcal{H},\Omega_1} = \Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \otimes I_{pN_o} + (I_{pN_o} \otimes \Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger)\mathcal{P}_{pN_o,q(N_o+N_i)} \\ \mathcal{J}_{\mathcal{H},\Omega_2} = -\Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \otimes \Omega_1 \Omega_2^\dagger - \Omega_1 \Omega_2^\dagger \otimes \Omega_1 \Omega_2^\dagger \mathcal{R}_{11} \Omega_2^\dagger \\ \mathcal{J}_{\mathcal{H},\mathcal{R}_{11}} = \Omega_1 \Omega_2^\dagger \otimes \Omega_1 \Omega_2^\dagger \end{cases}$$

$$\begin{cases} \mathcal{J}_{\Omega_1,\mathcal{R}}^{(1)} = -\mathcal{R}_{10}\mathcal{R}_8^\dagger \otimes I_{pN_o} \\ \mathcal{J}_{\Omega_1,\mathcal{R}}^{(2)} = \mathcal{R}_{10}\mathcal{R}_8^\dagger \otimes \mathcal{R}_4\mathcal{R}_8^\dagger \\ \mathcal{J}_{\Omega_1,\mathcal{R}}^{(3)} = -(I_{q(N_o+N_i)} \otimes \mathcal{R}_4\mathcal{R}_8^\dagger)\mathcal{P}_{q(N_o+N_i),pN_i} \end{cases} \qquad \begin{aligned} \mathcal{J}_{\Omega_2,\mathcal{R}}^{(1)} &= -\mathcal{R}_{10}\mathcal{R}_8^\dagger \otimes I_{q(N_o+N_i)} - (I_{q(N_o+N_i)} \otimes \mathcal{R}_{10}\mathcal{R}_8^\dagger)\mathcal{P}_{q(N_o+N_i),pN_i} \\ \mathcal{J}_{\Omega_2,\mathcal{R}}^{(2)} &= \mathcal{R}_{10}\mathcal{R}_8^\dagger \otimes \mathcal{R}_{10}\mathcal{R}_8^\dagger \end{aligned}$$

References

[1] W. Heylen, S. Lammens, P. Sas, Modal Analysis Theory and Testing, Katholieke Universiteit Leuven, Belgium, 1998.

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

[3] E. Reynders, System identification methods for (operational) modal analysis: review and comparison, Archives of Computational Methods in Engineering 19 (1) (2012) 51–124.

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

[5] N. Mastronardi, D. Kressner, V. Sima, P. V. Dooren, S. V. Huffel, A fast algorithm for subspace state-space system identification via exploitation of the displacement structure, Journal of Computational and Applied Mathematics 132 (1) (2001) 71–81.

[6] E. Gandino, S. Marchesiello, Identification of a duffing oscillator under different types of excitation, Mathematical Problems in Engineering 2010.

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

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

[9] L. Ljung, System identification, Wiley Online Library, 1999.

[10] M. Jansson, B. Wahlberg, A linear regression approach to state-space subspace system identification, Signal Processing 52 (2) (1996) 103–129.

[11] C. T. Chou, M. Verhaegen, Subspace algorithms for the identification of multivariable dynamic errors-in-variables models, Automatica 33 (10) (1997) 1857–1869.

[12] T. Gustafsson, Subspace identification using instrumental variable techniques, Automatica 37 (12) (2001) 2005–2010.

[13] E. Gandino, L. Garibaldi, S. Marchesiello, Covariance-driven subspace identification: A complete input–output approach, Journal of Sound and Vibration 332 (26) (2013) 7000–7017.

[14] 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.

[15] M. Basseville, L. Mevel, M. Goursat, Statistical model-based damage detection and localization: subspace-based residuals and damage-to-noise sensitivity ratios, Journal of Sound and Vibration 275 (3) (2004) 769–794.

[16] M. Döhler, L. Mevel, Subspace-based fault detection robust to changes in the noise covariances, Automatica 49 (9) (2013) 2734–2743.

[17] L. Mevel, M. Basseville, A. Benveniste, M. Goursat, Merging sensor data from multiple measurement set-ups for non-stationary subspace-based modal analysis, Journal of Sound and Vibration 249 (4) (2002) 719–741.

[18] A. Guyader, L. Mevel, Covariance-driven subspace methods: input/output vs. output-only, in: Proceedings of the 21st International Modal Analysis Conference, Kissimmee, FL, USA, 2003.

[19] 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 identificationapplication to in-flight data analysis, Journal of Sound and Vibration 295 (3) (2006) 531–552.

[20] M. Döhler, L. Mevel, Modular subspace-based system identification from multi-setup measurements, IEEE Transactions on Automatic Control 57 (11) (2012) 2951–2956.

[21] M. Viberg, B. Wahlberg, B. Ottersten, Analysis of state space system identification methods based on instrumental variables and subspace fitting, Automatica 33 (9) (1997) 1603–1616.

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

[23] A. Chiuso, G. Picci, The asymptotic variance of subspace estimates, Journal of Econometrics 118 (1) (2004) 257–291.

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

[25] M. Raffy, C. Gontier, Statistical asymptotic error on modal parameters in combined deterministic–stochastic identification algorithm, Mechanical Systems and Signal Processing 19 (4) (2005) 714–735.

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

[27] 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.

[28] M. Döhler, L. Mevel, Efficient multi-order uncertainty computation for stochastic subspace identification, Mechanical Systems and Signal Processing 38 (2) (2013) 346–366.

[29] M. Döhler, X.-B. Lam, L. Mevel, Uncertainty quantification for modal parameters from stochastic subspace identification on multi-setup

measurements, Mechanical Systems and Signal Processing 36 (2) (2013) 562–581.

[30] M. Döhler, F. Hille, L. Mevel, W. Rücker, Structural health monitoring with statistical methods during progressive damage test of S101 Bridge, Engineering Structures 69 (2014) 183–193.

[31] E. Reynders, G. D. Roeck, Reference-based combined deterministic–stochastic subspace identification for experimental and operational modal analysis, Mechanical Systems and Signal Processing 22 (3) (2008) 617–637.

[32] A. Esna Ashari, L. Mevel, Input-output subspace-based fault detection, in: Proc. 8th IFAC Symposium on fault detection, diagnosis and safety of technical processes (SAFEPROCESS), Mexico City, Mexico, 2012, pp. 204–209.

[33] M. Verhaegen, Application of a subspace model identification technique to identify lti systems operating in closed-loop, Automatica 29 (4) (1993) 1027–1040.

[34] P. Van Overschee, B. De Moor, N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems, Automatica 30 (1) (1994) 75–93.

[35] M. Verhaegen, Identification of the deterministic part of MIMO state space models given in innovations form from input-output data, Automatica 30 (1) (1994) 61–74.

[36] E. Hannan, Multiple time series, Wiley New York, 1970.

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

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

[39] G. Casella, R. Berger, Statistical inference, Duxbury Press, 2002.

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

[41] M. Döhler, X.-B. Lam, L. Mevel, Multi-order covariance computation for estimates in stochastic subspace identification using QR decompositions, in: Proc. 19th IFAC World Congress, Cape Town, South Africa, 2014.

[42] W. Gersch, On the achievable accuracy of structural system parameter estimates, Journal of Sound and Vibration 34 (1) (1974) 63–79.

[43] C. R. Pickrel, P. J. White, Flight flutter testing of transport aircraft: in-flight modal analysis, in: Proc. 21st International Modal Analysis Conference, 2003.

[44] P. Van Overschee, B. De Moor, A unifying theorem for three subspace system identification algorithms, Automatica 31 (12) (1995) 1853–1864.