

RESEARCH ARTICLE

Open Access



Dynamic data-driven model reduction: adapting reduced models from incomplete data

Benjamin Peherstorfer* and Karen Willcox

*Correspondence:
pehersto@mit.edu
Department of Aeronautics and
Astronautics, MIT, 77
Massachusetts Avenue,
Cambridge 02139, USA

Abstract

This work presents a data-driven online adaptive model reduction approach for systems that undergo dynamic changes. Classical model reduction constructs a reduced model of a large-scale system in an offline phase and then keeps the reduced model unchanged during the evaluations in an online phase; however, if the system changes online, the reduced model may fail to predict the behavior of the changed system. Rebuilding the reduced model from scratch is often too expensive in time-critical and real-time environments. We introduce a dynamic data-driven adaptation approach that adapts the reduced model from incomplete sensor data obtained from the system during the online computations. The updates to the reduced models are derived directly from the incomplete data, without recourse to the full model. Our adaptivity approach approximates the missing values in the incomplete sensor data with gappy proper orthogonal decomposition. These approximate data are then used to derive low-rank updates to the reduced basis and the reduced operators. In our numerical examples, incomplete data with 30–40 % known values are sufficient to recover the reduced model that would be obtained via rebuilding from scratch.

Keywords: Model reduction, Online adaptivity, Dynamic data-driven reduced models, Incomplete sensor data, Gappy proper orthogonal decomposition, Dynamic data-driven application systems

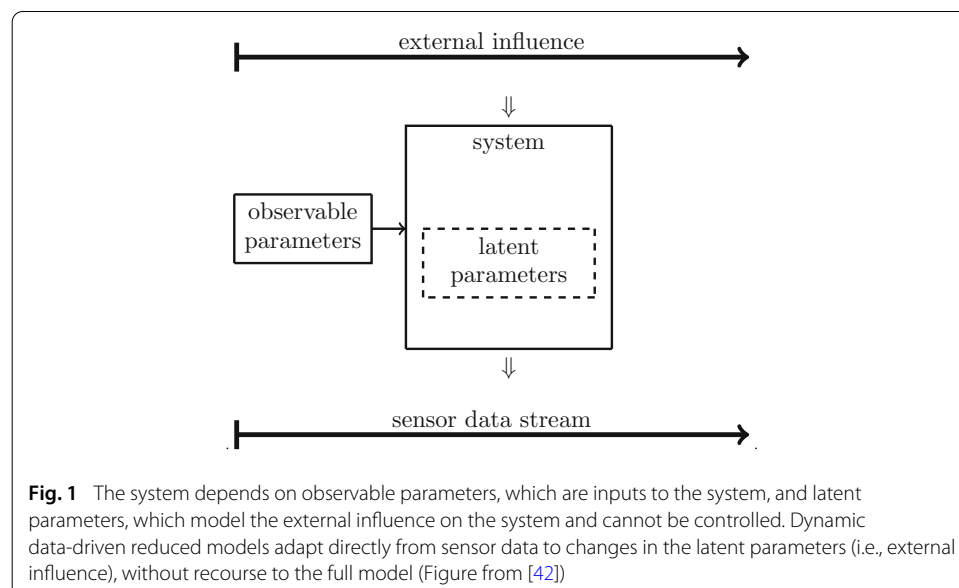
Background

Dynamic online (near real-time) capability estimation is a pivotal component of future autonomous systems to dynamically observe, orient, decide, and act in complex and changing environments. We consider the situation where the dynamics of the system are modeled by a parametrized partial differential equation (PDE) and sensor data are generated that provide information on the current state of the system. The system dynamics are approximated by a large-scale parametrized computer model, the so-called *full model*, resulting from the discretization of the underlying PDE. We rely on (projection-based) model reduction [7, 29, 45] to derive a low-cost *reduced model* of the full model to meet the real-time demands of online capability estimation. Reduced models are typically built with one-time high-computational costs in an offline phase and then stay unchanged while they are repeatedly evaluated in an online phase. However, in changing environ-

ments, the properties and the behavior of the system might change even during the online phase. Rebuilding the reduced model from scratch to take into account the changes in the system is often too time consuming. We therefore rely on dynamic data-driven reduced models, as introduced in [42]. Dynamic data-driven reduced models adapt directly from sensor data to changes in the underlying system, without recourse to the full model; however, the dynamic data-driven approach as presented in [42] requires sensor samples that measure the full large-scale state of the system.

Here, we present an extension to the dynamic data-driven approach that handles incomplete sensor samples. We consider the situation where we might have the ability to sense the full large-scale state of the system, but where we can afford to process only a subset of the sensor data. For example, new sensor technologies (e.g., “sensor skins”) provide high-resolution sensor data of an entire component (e.g., an aircraft wing) but processing these tremendous amounts of data online is computationally challenging. Note that this is in contrast to settings where we have sparse sensors that are in fixed locations. Our methodology processes a selection of the sensor data—an incomplete sensor sample—that contains the essential information for updating the reduced model. Furthermore, we can dynamically change this selection of the sensor data during the online phase, so that at each step we process the subset of sensor data that are most informative to the event at hand.

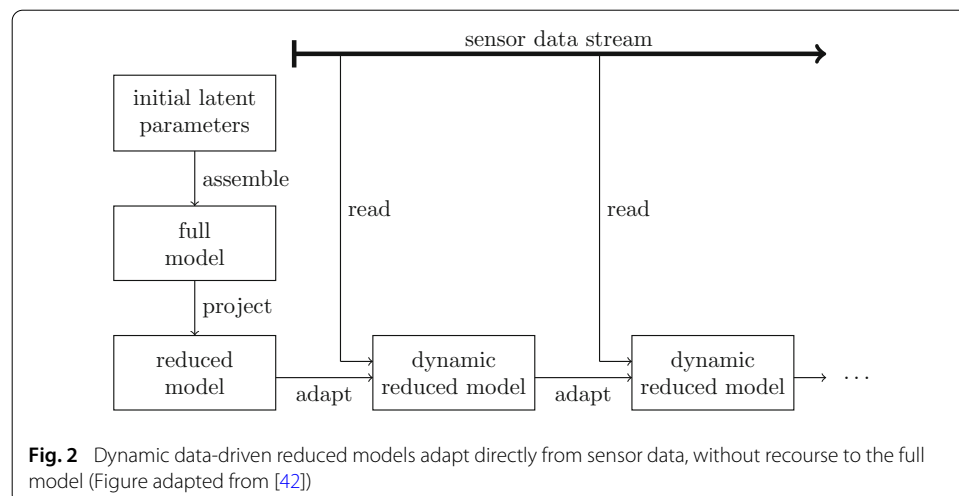
To model changes in the system, the parameters of the system are split into observable and latent parameters, see Fig. 1. The observable parameters are inputs to the system and therefore the values of these parameters are known. Latent parameters describe external influences on the system (e.g., damage, fatigue, erosion). The values of the latent parameters are unknown, except for the nominal latent parameters that describe the nominal state of the system (e.g., no-damage condition). Since the values of the latent parameters are unknown, a reduced model can be built in the offline phase for the nominal latent parameter only. If the latent parameters change online (e.g., the system gets damaged), the reduced model fails to predict the behavior of the system. Rebuilding the reduced



model from scratch requires inferring the value of the changed latent parameters from the sensor data with a model of the changed system, then assembling the full model operators corresponding to the inferred latent parameters, and deriving the reduced model. Rebuilding from scratch therefore is often too expensive in the context of online capability estimation, see, e.g., [1, 33, 37, 42] for a discussion. The dynamic data-driven approach introduced in [42] exploits the sensor data of the system to adapt the reduced model to changes in the latent parameters online, without the computationally expensive inference step and without assembling the full model operators for the inferred latent parameters, see Fig. 2.

There are several online adaptation approaches for reduced models. We distinguish between approaches that solely rely on pre-computed quantities for the adaptation and approaches that adapt the reduced model from new data that are generated during the online phase. Interpolation between reduced operators and reduced models [2, 18, 39, 51], localization approaches [3, 9, 11, 19–21, 36, 40, 46], and dictionary approaches [30, 35] rely on pre-computed quantities but do not incorporate information from new data into the reduced model online. In [4], local reduced models are adapted from partial data online to smooth the transition between the local models. In [12], an h -adaptive refinement is presented that splits basis vectors based on an unsupervised learning algorithm and residuals that become available online. The online adaptive approach [43] adapts the approximation of nonlinear terms from sparse data of the full model. There is also a body of work that rebuilds reduced models from scratch, e.g., in optimization [27, 32, 50], inverse problems [17, 25], and multiscale methods [38]. We also mention that reduced models have been used in the context of dynamical data-driven application systems (DDDAS), which dynamically incorporate data into an executing application, and, in reverse, dynamically steer the measurement process. In [26], proper generalized decomposition [16] is used in a DDDAS to recover from device malfunctions by reconfiguring the simulation process. In [28], online parameter identification from measurements is considered for DDDAS with proper generalized decomposition. The work [1, 33, 34, 37] considers model reduction for structural health monitoring in DDDAS.

Our extension to handle incomplete sensor samples in the dynamic data-driven reduced model adaptation builds on gappy proper orthogonal decomposition (POD), which is a



method to approximate unknown or missing values in vector-valued data [22]. Gappy POD reconstructs the unknown values by representing the data vector as a linear combination of POD basis vectors. Applications of gappy POD in model reduction include flow field reconstruction [10,49], acceleration of efficient approximations of nonlinear terms [5,13,24], and forecasting for time-dependent problems [14]. In our adaptation approach, we first construct a gappy POD basis from incomplete sensor samples using an incremental POD basis generation algorithm. The missing values of the incomplete sensor samples are then approximated in the space spanned by the obtained gappy POD basis. These approximate sensor samples are used in the dynamic data-driven adaptation to derive updates to the reduced model.

This paper is organized as follows. “Preliminaries and adaptation from complete data” section introduces the full model and the dynamic data-driven adaptation. “Incomplete sensor samples” section defines incomplete sensor samples and describes the problem setup in detail. “Dynamic data-driven adaptation from incomplete sensor samples” section introduces the extension to the dynamic data-driven adaptation approach that handles incomplete sensor samples. The numerical results in “Numerical results” section demonstrate that in our examples 30–40 % of the values of the sensor samples are sufficient to recover reduced models that accurately capture the changes in the latent parameters. “Summary and future work” section gives concluding remarks.

Preliminaries and adaptation from complete data

This section briefly discusses model reduction for systems with observable and latent parameters and summarizes the dynamic data-driven adaptation approach presented in [42].

Systems with latent parameters

Consider a parametrized system of equations stemming from the discretization of a parametrized PDE

$$\mathbf{A}_\eta(\boldsymbol{\mu})\mathbf{y}_\eta(\boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}). \quad (1)$$

The full model (1) depends on the observable parameter $\boldsymbol{\mu} \in \mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^d$ with $d \in \mathbb{N}$, and on the latent parameter $\boldsymbol{\eta} \in \mathcal{E}$, where $\mathcal{E} \subset \mathbb{R}^{d'}$ with $d' \in \mathbb{N}$. In general, the value of the latent parameter is unknown, only the value of a nominal latent parameter $\boldsymbol{\eta}_0 \in \mathcal{E}$ is known, see “Background” section. The linear operator $\mathbf{A}_\eta(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ is an $N \times N$ matrix, where $N \in \mathbb{N}$ is the number of degrees of freedom of the full model (1). The linear operator $\mathbf{A}_\eta(\boldsymbol{\mu})$ depends on the observable and on the latent parameter. The operator $\mathbf{A}_\eta(\boldsymbol{\mu})$ has an affine parameter dependence with respect to the observable parameter

$$\mathbf{A}_\eta(\boldsymbol{\mu}) = \sum_{i=1}^{l_A} \Theta_A^{(i)}(\boldsymbol{\mu})\mathbf{A}_\eta^{(i)},$$

where $l_A \in \mathbb{N}$ and $\Theta_A^{(1)}, \dots, \Theta_A^{(l_A)} : \mathcal{D} \rightarrow \mathbb{R}$. The linear operators $\mathbf{A}_\eta^{(1)}, \dots, \mathbf{A}_\eta^{(l_A)} \in \mathbb{R}^{N \times N}$ are independent of the observable parameter. Note that an affine parameter dependence with respect to $\boldsymbol{\mu}$ can be approximated with sparse sampling methods, e.g., [5,6,13,15,22]. Note further that no affine parameter dependence with respect to the latent parameter is required. The state $\mathbf{y}_\eta(\boldsymbol{\mu}) \in \mathbb{R}^N$ is an N -dimensional vector. The right-hand side

$\mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^N$ depends on the observable parameter but is independent of the latent parameter. The right-hand side has an affine parameter dependence with respect to $\boldsymbol{\mu}$

$$\mathbf{f}(\boldsymbol{\mu}) = \sum_{i=1}^{l_f} \Theta_f^{(i)}(\boldsymbol{\mu}) \mathbf{f}^{(i)},$$

with $l_f \in \mathbb{N}$, $\Theta_f^{(1)}, \dots, \Theta_f^{(l_f)} : \mathcal{D} \rightarrow \mathbb{R}$, and the $\boldsymbol{\mu}$ -independent vectors $\mathbf{f}^{(1)}, \dots, \mathbf{f}^{(l_f)} \in \mathbb{R}^N$.

Classical model reduction for systems with latent parameters

Let $\mathbf{Y}_{\eta_0} \in \mathbb{R}^{N \times M}$ be the snapshot matrix that contains as columns $M \in \mathbb{N}$ state vectors $\mathbf{y}_{\eta_0}(\boldsymbol{\mu}_1), \dots, \mathbf{y}_{\eta_0}(\boldsymbol{\mu}_M) \in \mathbb{R}^N$ of the full model (1) corresponding to the observable parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M \in \mathcal{D}$ and the nominal latent parameter $\eta_0 \in \mathcal{E}$. The POD basis matrix $\mathbf{V}_{\eta_0} \in \mathbb{R}^{N \times n}$ contains as columns the first $n \in \mathbb{N}$ left-singular vectors of the snapshot matrix \mathbf{Y}_{η_0} that correspond to the largest singular values. The POD basis vectors, i.e., the columns of the POD basis matrix \mathbf{V}_{η_0} , span the n -dimensional POD space \mathcal{V}_{η_0} .

The reduced linear operator $\tilde{\mathbf{A}}_{\eta_0}(\boldsymbol{\mu}) \in \mathbb{R}^{n \times n}$ is obtained via Galerkin projection of the equations of the full model onto the POD space \mathcal{V}_{η_0} . Consider therefore the projected $\boldsymbol{\mu}$ -independent operators

$$\tilde{\mathbf{A}}_{\eta_0}^{(i)} = \mathbf{V}_{\eta_0}^T \mathbf{A}_{\eta_0}^{(i)} \mathbf{V}_{\eta_0}, \quad i = 1, \dots, l_A.$$

By exploiting the affine parameter dependence of the linear operator $\mathbf{A}_{\eta_0}(\boldsymbol{\mu})$ on the observable parameter $\boldsymbol{\mu} \in \mathcal{D}$, the reduced linear operator $\tilde{\mathbf{A}}_{\eta_0}(\boldsymbol{\mu})$ is

$$\tilde{\mathbf{A}}_{\eta_0}(\boldsymbol{\mu}) = \sum_{i=1}^{l_A} \Theta_A^{(i)}(\boldsymbol{\mu}) \tilde{\mathbf{A}}_{\eta_0}^{(i)}.$$

Similarly, the reduced right-hand side is

$$\tilde{\mathbf{f}}_{\eta_0}(\boldsymbol{\mu}) = \sum_{i=1}^{l_f} \Theta_f^{(i)}(\boldsymbol{\mu}) \tilde{\mathbf{f}}^{(i)},$$

where $\tilde{\mathbf{f}}^{(i)} = \mathbf{V}_{\eta_0}^T \mathbf{f}^{(i)} \in \mathbb{R}^n$ for $i = 1, \dots, l_f$. The reduced model for the latent parameter η_0 is

$$\tilde{\mathbf{A}}_{\eta_0}(\boldsymbol{\mu}) \tilde{\mathbf{y}}_{\eta_0}(\boldsymbol{\mu}) = \tilde{\mathbf{f}}_{\eta_0}(\boldsymbol{\mu}), \tag{2}$$

where $\tilde{\mathbf{y}}_{\eta_0}(\boldsymbol{\mu}) \in \mathbb{R}^n$ is the reduced state. The reduced right-hand side $\tilde{\mathbf{f}}_{\eta_0}(\boldsymbol{\mu}) \in \mathbb{R}^n$ in the reduced model (2) depends on the latent parameter η_0 because of the projection onto the POD space \mathcal{V}_{η_0} , in contrast to the right-hand side vector $\mathbf{f}(\boldsymbol{\mu}) \in \mathbb{R}^N$ in the full model (1).

Dynamic data-driven adaptation for reduced models

The reduced model (2) is derived from snapshots with the latent parameter $\eta = \eta_0$ set to the nominal latent parameter η_0 . This means that if the latent parameter changes online, the reduced model (2) cannot predict the behavior of the system. In [41,42], a dynamic data-driven adaptation approach is presented that successively adapts a reduced model in $M' \in \mathbb{N}$ adaptivity steps to changes in the latent parameter. Consider therefore the $h = 1, \dots, M'$ adaptivity steps, in which the reduced model is adapted from the nominal

latent parameter η_0 to the changed latent parameter, say, $\eta_1 \in \mathcal{E}$.¹ In each adaptivity step h , a sensor sample $\hat{y}_{\eta_1}(\mu_{M+h}) \in \mathbb{R}^N$ is received. The sensor sample $\hat{y}_{\eta_1}(\mu_{M+h})$ is an approximation of the state $y_{\eta_1}(\mu_{M+h})$ for the changed latent parameter η_1 and an observable parameter $\mu_{M+h} \in \mathcal{D}$. The difference $\|\hat{y}_{\eta_1}(\mu_{M+h}) - y_{\eta_1}(\mu_{M+h})\|$ between the sensor sample and the state in a norm $\|\cdot\|$ is noise, measurement error, and the discrepancy of the full model and reality (model discrepancy [31]). At step h , the sensor samples matrix $S_h \in \mathbb{R}^{N \times h}$ contains the received sensor samples $\hat{y}_{\eta_1}(\mu_{M+1}), \dots, \hat{y}_{\eta_1}(\mu_{M+h}) \in \mathbb{R}^N$ as columns

$$S_h = [\hat{y}_{\eta_1}(\mu_{M+1}), \dots, \hat{y}_{\eta_1}(\mu_{M+h})] \in \mathbb{R}^{N \times h}.$$

At each adaptivity step $h = 1, \dots, M'$, the dynamic data-driven adaptation first adapts the POD basis and then the reduced operators. Consider the POD basis adaptation first. At step $h = 1$, the first snapshot, i.e., the first column, in the snapshot matrix Y_{η_0} is replaced with the sensor sample $\hat{y}_{\eta_1}(\mu_{M+1}) \in \mathbb{R}^N$ and the snapshot matrix at step $h = 1$ is obtained

$$Y_1 = [\hat{y}_{\eta_1}(\mu_{M+1}), y_{\eta_0}(\mu_2), \dots, y_{\eta_0}(\mu_M)] \in \mathbb{R}^{N \times M}.$$

Note that there is no particular ordering of the snapshots in the snapshots matrix. We replace the first column of Y_{η_0} because we are at step $h = 1$. By reordering the columns of Y_{η_0} , any other snapshot can be replaced at step $h = 1$. The matrix Y_1 is the result of an additive rank-one update to the snapshot matrix Y_{η_0} . Let $e_i \in \{0, 1\}^N$ be the canonical unit vector with 1 at component i and 0 at all other components for $i = 1, \dots, N$. Then, the snapshot matrix Y_1 is

$$Y_1 = Y_{\eta_0} + \mathbf{a}e_1^T,$$

where $\mathbf{a} = \hat{y}_{\eta_1}(\mu_{M+1}) - y_{\eta_0}(\mu_1) \in \mathbb{R}^N$. Therefore, the POD basis matrix $V_1 \in \mathbb{R}^{N \times n}$ corresponding to the snapshot matrix Y_1 can be approximately derived from V_{η_0} via the adaptation algorithm [8]. The algorithm extracts the components $\alpha = \mathbf{a} - V_{\eta_0} V_{\eta_0}^T \mathbf{a}$ and $\beta = e_1 - V_{\eta_0} V_{\eta_0}^T e_1$ of \mathbf{a} and e_1 , respectively, that are orthogonal to V_{η_0} . The vectors α and β are used to derive a rotation matrix $V' \in \mathbb{R}^{n \times n}$ of size $n \times n$ and an additive rank-one update $\gamma\delta^T$ with $\gamma \in \mathbb{R}^n$ and $\delta \in \mathbb{R}^n$. Computing the rotation matrix and the rank-one update requires computing the singular value decomposition (SVD) of an $(n+1) \times (n+1)$ matrix. The adapted POD basis matrix V_1 is then given by

$$V_1 = V_{\eta_0} V' + \gamma\delta^T.$$

Note that an SVD of a typically small $(n+1) \times (n+1)$ matrix is required by the adaptation algorithm, instead of the SVD of an $N \times M$ matrix if the POD basis matrix were computed directly from Y_1 without reusing V_{η_0} . We refer to [8] for details on the adaptation of the POD basis matrix. The adaptation algorithm is summarized in [42, Algorithm 1] for the case of the dynamic data-driven adaptation.

Consider now the adaptation of the operators at step $h = 1$. The goal is to approximate the reduced operators

$$\tilde{A}_{\eta_1}^{(i)} = V_1^T A_{\eta_1}^{(i)} V_1, \quad i = 1, \dots, l_A,$$

¹Note that the adaptation can be repeated to adapt from η_1 to $\eta_2 \in \mathcal{E}$ and so on.

without assembling the full operators $A_{\eta_1}^{(1)}, \dots, A_{\eta_1}^{(l_A)} \in \mathbb{R}^{N \times N}$ corresponding to the changed latent parameter η_1 . Therefore, at adaptivity step $h = 1$, the operators

$$\tilde{A}_1^{(i)} = V_1^T A_{\eta_0}^{(i)} V_1, \quad i = 1, \dots, l_A, \tag{3}$$

are constructed. The operator $\tilde{A}_1^{(i)}$ is the full operator $A_{\eta_0}^{(i)}$ for latent parameter η_0 projected onto the adapted POD space \mathcal{V}_1 with the adapted POD basis matrix V_1 , for $i = 1, \dots, l_A$. Note that (3) projects the full operators corresponding to the nominal latent parameter η_0 , and not the operators corresponding to the changed latent parameter η_1 . Then, additive updates $\delta\tilde{A}_1^{(1)}, \dots, \delta\tilde{A}_1^{(l_A)} \in \mathbb{R}^{n \times n}$ are derived from the sensor sample matrix S_1 with the optimization problem

$$\min_{\delta\tilde{A}_h^{(1)}, \dots, \delta\tilde{A}_h^{(l_A)} \in \mathbb{R}^{n \times n}} \sum_{j=1}^h \left\| \sum_{i=1}^{l_A} \Theta_A^{(i)}(\mu_{M+j}) \left(\tilde{A}_h^{(i)} + \delta\tilde{A}_h^{(i)} \right) V_h^T \hat{y}_{\eta_1}(\mu_{M+j}) - \tilde{f}_h(\mu_{M+j}) \right\|_2^2, \tag{4}$$

where $\tilde{f}_h(\mu_{M+j}) \in \mathbb{R}^n$ is the reduced right-hand side with respect to the POD basis V_h . Note that the optimization problem (4) is formulated for general $h \geq 1$, and not only for $h = 1$. The solution of the optimization problem (4) are the updates $\delta\tilde{A}_h^{(1)}, \dots, \delta\tilde{A}_h^{(l_A)}$ that best-fit the sensor samples in the sensor sample matrix S_h . The optimization problem (4) is a least-squares problem that can be solved with, e.g., the QR decomposition. For $h < l_A n$, the least-squares problem is underdetermined, and only low-rank approximations of $\delta\tilde{A}_h^{(1)}, \dots, \delta\tilde{A}_h^{(l_A)}$ are computed [42].

At step $h = 1$, the adapted operators are

$$\tilde{A}_1^{(i)} = \tilde{A}_{\eta_0}^{(i)} + \delta\tilde{A}_1^{(i)}, \quad i = 1, \dots, l_A,$$

and the adapted reduced operator $\tilde{A}_1(\mu) \in \mathbb{R}^{n \times n}$ can be assembled using the affine parameter dependence as

$$\tilde{A}_1(\mu) = \sum_{i=1}^{l_A} \Theta_{l_A}^{(i)}(\mu) \tilde{A}_1^{(i)}.$$

In each adaptivity step $h = 1, \dots, M'$, this POD basis and operator adaptation is repeated. This means, at step h , the POD basis matrix is adapted from V_{h-1} to V_h by exploiting that the snapshot matrix Y_h at step h is the result of a rank-one update to the snapshot matrix Y_{h-1} from the previous step. The adapted reduced operator $\tilde{A}_h(\mu)$ is derived via the additive rank-one updates $\delta\tilde{A}_h^{(1)}, \dots, \delta\tilde{A}_h^{(l_A)} \in \mathbb{R}^{n \times n}$, which are obtained via optimization from the sensor samples matrix $S_h = \left[\hat{y}_{\eta_1}(\mu_{M+1}), \dots, \hat{y}_{\eta_1}(\mu_{M+h}) \right] \in \mathbb{R}^{N \times h}$. For sufficiently many sensor samples, and if the sensor samples are noise-free, the reduced operator $\tilde{A}_{\eta_1}(\mu)$ with respect to the POD basis matrix V_h equals the adapted reduced operator $\tilde{A}_h(\mu)$, see [42].

Incomplete sensor samples

The dynamic data-driven adaptation derives updates to a reduced model from sensor samples. We consider here the situation where we receive incomplete sensor samples, i.e., partial measurements of the state. This section mathematically defines incomplete sensor samples, and the next section develops the extension to the dynamic data-driven adaptation to handle incomplete sensor samples.

Let $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N$ be the (complete) sensor sample that is received at adaptivity step h . Let $k \in \mathbb{N}$ with $k < N$ and let $p_1^h, \dots, p_k^h \in \{1, \dots, N\}$ be pairwise distinct indices of the sensor sample $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N$. The indices p_1^h, \dots, p_k^h give rise to a point selection matrix

$$\mathbf{P}_h = [\mathbf{e}_{p_1^h}, \dots, \mathbf{e}_{p_k^h}] \in \mathbb{R}^{N \times k}.$$

The point selection matrix \mathbf{P}_h selects the components with indices p_1^h, \dots, p_k^h . For example, consider the vector $\mathbf{x} = [x_1, \dots, x_N]^T \in \mathbb{R}^N$, then we have

$$\begin{bmatrix} x_{p_1^h} \\ \vdots \\ x_{p_k^h} \end{bmatrix} = \mathbf{P}_h^T \mathbf{x}.$$

From the point selection matrix \mathbf{P}_h , we derive the matrix $\mathbf{Q}_h \in \mathbb{R}^{N \times (N-k)}$ that selects the components of the (complete) sensor sample $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h})$ that are missing in the incomplete sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h})$. The matrices \mathbf{P}_h and \mathbf{Q}_h lead to the decomposition

$$\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}) = \mathbf{P}_h \mathbf{P}_h^T \hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}) + \mathbf{Q}_h \mathbf{Q}_h^T \hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}).$$

The matrix $\mathbf{P}_h \mathbf{P}_h^T$ selects all components that correspond to the indices p_1^h, \dots, p_k^h and sets the components at all other indices $\{1, \dots, N\} \setminus \{p_1^h, \dots, p_k^h\}$ to zero. The matrix $\mathbf{Q}_h \mathbf{Q}_h^T$ has the opposite effect and selects all components with indices in $\{1, \dots, N\} \setminus \{p_1^h, \dots, p_k^h\}$ and sets the components with indices $\{p_1^h, \dots, p_k^h\}$ to zero.

We define the incomplete sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h})$ of the (complete) sensor sample $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h})$ corresponding to the point selection matrix \mathbf{P}_h as

$$\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) = \mathbf{P}_h \mathbf{P}_h^T \hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N. \tag{5}$$

The values at the components of the incomplete sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h})$ with indices p_1^h, \dots, p_k^h are set to the corresponding components of the (complete) sensor sample $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+h})$. All other components are missing in the incomplete sensor sample and their values in $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h})$ are zero through the definition (5).

Dynamic data-driven adaptation from incomplete sensor samples

We propose an extension to the dynamic data-driven adaptation approach that handles incomplete sensor samples. Consider the adaptation from the nominal latent parameter η_0 to the latent parameter η_1 in the M' adaptivity steps $h = 1, \dots, M'$. At each adaptivity step $h = 1, \dots, M'$, we receive incomplete sensor samples $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N$ and the corresponding point selection matrices \mathbf{P}_h . The point selection matrix depends on h and might change at each adaptivity step, see the discussion on future sensor technologies in “Background” section. The number of known components k is independent of h and stays constant for all $h = 1, \dots, M'$.

We split the adaptivity steps $M' = M^{\text{basis}} + M^{\text{update}}$ into $M^{\text{basis}} \in \mathbb{N}$ and $M^{\text{update}} \in \mathbb{N}$ steps. In the first $h = 1, \dots, M^{\text{basis}}$ steps, a gappy POD basis is derived from the incomplete sensor samples $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1}), \dots, \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+M^{\text{basis}}}) \in \mathbb{R}^N$. At the subsequent M^{update} steps $h = M + M^{\text{basis}} + 1, \dots, M'$, the missing values of the incomplete sensor samples $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+M^{\text{basis}}+h}) \in \mathbb{R}^N$ are approximated using gappy POD with the obtained gappy POD

basis. The approximations of the missing values and the components in the incomplete sensor sample are combined to approximate the complete sensor sample. The dynamic data-driven adaptation is then applied to these approximate sensor samples to update the reduced model. “Deriving the gappy POD basis” section discusses the construction of the gappy POD basis and “Dynamic data-driven adaptation from approximate sensor samples” section presents the adaptation of the reduced model from the approximate sensor samples. “Computational procedure” section summarizes the procedure and presents Algorithm 1.

Deriving the gappy POD basis

In the first $h = 1, \dots, M^{\text{basis}}$ adaptivity steps, we derive a gappy POD basis from the incomplete sensor samples. Let $r \in \mathbb{N}$ be the dimension of the gappy POD basis with gappy POD basis matrix $\mathbf{U}_h \in \mathbb{R}^{N \times r}$. The initial gappy POD basis matrix $\mathbf{U}_0 \in \mathbb{R}^{N \times r}$ contains as columns the r -dimensional POD basis vectors corresponding to the snapshot matrix \mathbf{Y}_{η_0} .

At step $h = 1$, we receive the incomplete sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1})$ and the corresponding point selection matrix $\mathbf{P}_1 \in \mathbb{R}^{N \times k}$ with $\mathbf{Q}_1 \in \mathbb{R}^{N \times (N-k)}$. We use the initial gappy POD basis matrix \mathbf{U}_0 to derive the approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+1}) \in \mathbb{R}^N$ using gappy POD [10, 22, 49]

$$\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+1}) = \mathbf{Q}_1 \mathbf{Q}_1^T \mathbf{U}_0 \left(\mathbf{P}_1^T \mathbf{U}_0 \right)^+ \mathbf{P}_1^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1}) + \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1}). \quad (6)$$

The matrix $(\mathbf{P}_1^T \mathbf{U}_0)^+ \in \mathbb{R}^{r \times k}$ is the Moore–Penrose pseudoinverse of the matrix $\mathbf{P}_1^T \mathbf{U}_0 \in \mathbb{R}^{k \times r}$. Since $\mathbf{P}_1^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1}) = \mathbf{P}_1^T \hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+1})$, we have that $(\mathbf{P}_1^T \mathbf{U}_0)^+ \mathbf{P}_1^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1})$ is the solution of the regression problem

$$\arg \min_{\mathbf{c} \in \mathbb{R}^r} \left\| \mathbf{P}_1^T \left(\mathbf{U}_0 \mathbf{c} - \hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+1}) \right) \right\|_2^2. \quad (7)$$

Note that the regression problem is overdetermined and has a unique solution if the matrix $\mathbf{P}_1^T \mathbf{U}_0$ has full column rank, which we typically ensure by selecting $k > r$. Therefore, the vector $\mathbf{U}_0 (\mathbf{P}_1^T \mathbf{U}_0)^+ \mathbf{P}_1^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+1}) \in \mathbb{R}^N$ is the best approximation with respect to (7) of the complete sensor sample $\hat{\mathbf{y}}_{\eta_1}(\boldsymbol{\mu}_{M+1})$ in the space spanned by the columns of the POD basis matrix \mathbf{U}_0 . The approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+1})$ combines this best approximation and the known values in the incomplete sensor sample. The values at the components corresponding to the missing components of the incomplete sensor sample are set to the best approximation, and the values at all other components are set to the values obtained from the incomplete sensor sample.

We then use the approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+1})$ to adapt the gappy POD basis from \mathbf{U}_0 to \mathbf{U}_1 . Consider therefore the snapshot matrix \mathbf{Y}_0 and note that \mathbf{U}_0 is the k -dimensional POD basis derived from \mathbf{Y}_0 . We adapt the snapshot matrix \mathbf{Y}_0 to $\mathbf{Y}_1 \in \mathbb{R}^{N \times M}$ via a rank-one update that replaces column 1 of \mathbf{Y}_0 with the approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+1}) \in \mathbb{R}^N$. Since \mathbf{Y}_1 is the result of a rank-one update to \mathbf{Y}_0 , the k -dimensional POD basis corresponding to \mathbf{Y}_1 can be approximated in a computationally efficient manner using the incremental POD algorithm [8]. Note that this is the same approach as used in the dynamic data-driven adaptation, see “Dynamic data-driven adaptation for reduced models” section. Thus, the adapted gappy POD basis matrix \mathbf{U}_1 can be derived cheaply from the basis matrix \mathbf{U}_0 .

At step $h = 2$, the approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+2})$ is constructed with the gappy POD basis matrix \mathbf{U}_1 , which is then used to adapt from \mathbf{U}_1 to \mathbf{U}_2 . This process is continued until step $h = M^{\text{basis}}$, where the gappy POD basis matrix $\mathbf{U}_{M^{\text{basis}}}$ is derived. Note that the number of columns in the snapshot matrix is fixed and that columns are replaced following the first-in-first-out principle if $h > M$.

Dynamic data-driven adaptation from approximate sensor samples

In the M^{update} steps $h = M + M^{\text{basis}} + 1, \dots, M'$, we adapt the reduced model from approximate sensor samples using the dynamic data-driven adaptation. Consider therefore an adaptivity step $h > M^{\text{basis}}$, at which the incomplete sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N$ and the corresponding point selection matrix $\mathbf{P}_h \in \mathbb{R}^{k \times N}$ are received. We use the gappy POD basis $\mathbf{U}_{M^{\text{basis}}}$ to derive the approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h})$ of the complete sensor sample with the gappy POD basis $\mathbf{U}_{M^{\text{basis}}}$. The approximate sensor sample $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h})$ is then used to adapt the reduced model with the dynamic data-driven adaptation as described in “Dynamic data-driven adaptation for reduced models” section.

Computational procedure

Algorithm 1 summarizes the dynamic data-driven adaptation that can handle incomplete sensor samples. Inputs of Algorithm 1 are the POD basis matrix \mathbf{V}_{h-1} , the operators $\tilde{\mathbf{A}}_{h-1}^{(1)}, \dots, \tilde{\mathbf{A}}_{h-1}^{(l_A)}$, and the right-hand sides $\tilde{\mathbf{f}}_{h-1}^{(1)}, \dots, \tilde{\mathbf{f}}_{h-1}^{(l_f)}$ derived at the previous adaptivity step $h-1$. If $h \leq M^{\text{basis}}$, the algorithm adapts the gappy POD basis from \mathbf{U}_{h-1} to \mathbf{U}_h using the approach presented in “Deriving the gappy POD basis” section. First, the approximate sensor sample is constructed with gappy POD. Then, the adapted basis matrix \mathbf{U}_h is computed with the incremental POD algorithm [8]. Only the gappy POD basis is adapted and the reduced model is returned unchanged. If $h > M^{\text{basis}}$, the approximate sensor sample is derived with gappy POD and $\mathbf{U}_{M^{\text{basis}}}$. The approximate sensor sample is then used with the dynamic data-driven adaptation to derive the adapted POD basis \mathbf{V}_h , the adapted operators $\tilde{\mathbf{A}}_h^{(1)}, \dots, \tilde{\mathbf{A}}_h^{(l_A)}$, and the adapted right-hand sides $\tilde{\mathbf{f}}_h^{(1)}, \dots, \tilde{\mathbf{f}}_h^{(l_f)}$.

Algorithm 1 Dynamic data-driven adaptation from incomplete sensor samples

```

1: procedure ADAPTINCOMPLETE( $M^{\text{basis}}, \mathbf{U}_{h-1}, \mathbf{V}_{h-1}, \tilde{\mathbf{A}}_{h-1}^{(1)}, \dots, \tilde{\mathbf{A}}_{h-1}^{(l_A)}, \tilde{\mathbf{f}}_{h-1}^{(1)}, \dots, \tilde{\mathbf{f}}_{h-1}^{(l_f)}$ )
2:   Receive incomplete sensor sample  $\hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) \in \mathbb{R}^N$  and point selection matrix  $\mathbf{P}_h \in \mathbb{R}^{k \times N}$ 
3:   Construct matrix  $\mathbf{Q}_h \in \mathbb{R}^{(N-k) \times N}$ 
4:   if  $h \leq M^{\text{basis}}$  then
5:     Compute approximate sensor sample using gappy POD basis matrix  $\mathbf{U}_{h-1}$ 
6:     
$$\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h}) = \mathbf{Q}_h \mathbf{Q}_h^T \mathbf{U}_{h-1} (\mathbf{P}_h^T \mathbf{U}_{h-1})^+ \mathbf{P}_h^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) + \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}).$$

7:     Update snapshot matrix  $\mathbf{Y}_{h-1}$  with  $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h})$  to obtain  $\mathbf{Y}_h$ 
8:     Derive adapted gappy POD basis matrix  $\mathbf{U}_h$  from  $\mathbf{U}_{h-1}$  using [8]
9:     Set  $\tilde{\mathbf{A}}_h^{(i)} = \tilde{\mathbf{A}}_{h-1}^{(i)}$  for  $i = 1, \dots, l_A$ 
10:    Set  $\tilde{\mathbf{f}}_h^{(i)} = \tilde{\mathbf{f}}_{h-1}^{(i)}$  for  $i = 1, \dots, l_f$ 
11:    Set  $\mathbf{V}_h = \mathbf{V}_{h-1}$ 
12:   else
13:     Compute approximate sensor sample using basis  $\mathbf{U}_{M^{\text{basis}}}$ 
14:     
$$\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h}) = \mathbf{Q}_h \mathbf{Q}_h^T \mathbf{U}_{M^{\text{basis}}} (\mathbf{P}_h^T \mathbf{U}_{M^{\text{basis}}})^+ \mathbf{P}_h^T \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}) + \hat{\mathbf{y}}_{\eta_1}^{\text{incp}}(\boldsymbol{\mu}_{M+h}).$$

15:     Get  $\mathbf{V}_h, \tilde{\mathbf{A}}_h^{(1)}, \dots, \tilde{\mathbf{A}}_h^{(l_A)}, \tilde{\mathbf{f}}_h^{(1)}, \dots, \tilde{\mathbf{f}}_h^{(l_f)}$  from dynamic adaptation with  $\hat{\mathbf{y}}_{\eta_1}^{\text{apprx}}(\boldsymbol{\mu}_{M+h})$ 
16:   end if
17:   return  $\mathbf{V}_h, \tilde{\mathbf{A}}_h^{(1)}, \dots, \tilde{\mathbf{A}}_h^{(l_A)}, \tilde{\mathbf{f}}_h^{(1)}, \dots, \tilde{\mathbf{f}}_h^{(l_f)}$ 
18: end procedure

```

Numerical results

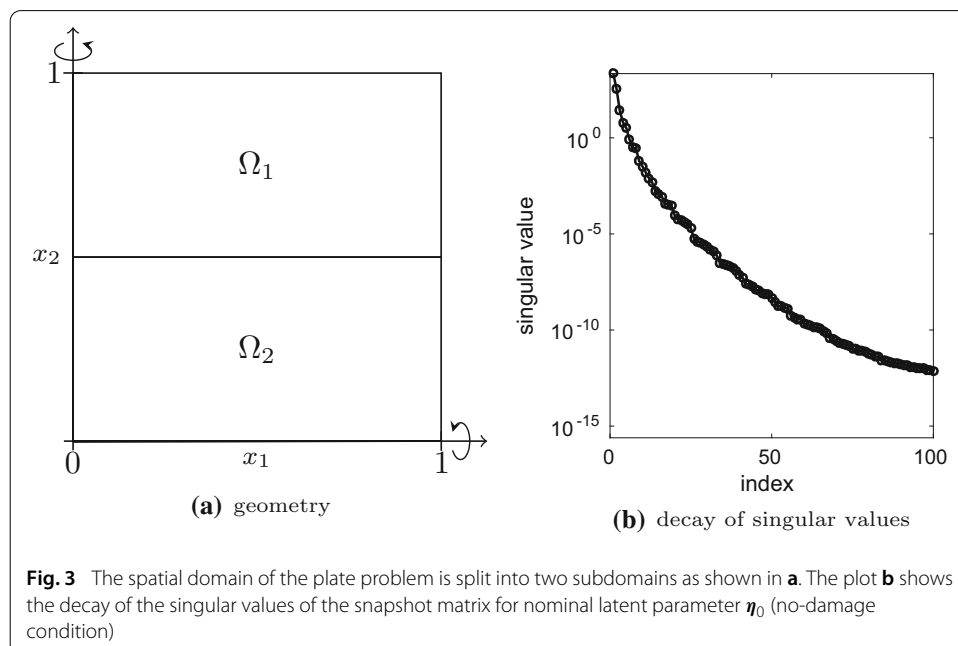
This section demonstrates the dynamic data-driven adaptation from incomplete sensor samples on a model of a bending plate. The latent parameter describes damage of the plate. The damage is a local decrease of the thickness of the plate. The model is based on the Mindlin plate theory [23,47] that takes into account transverse shear deformations but neglects important nonlinear effects such as postbuckling behavior. Therefore, the model that we use in this section is a simple description of a plate in bending. We use the plate model only to provide a proof of concept of our adaptation approach. More advanced plate models are used in real-world engineering applications. We refer to “Summary and future work” section for a discussion on further applications of our adaptation approach.

We first build a reduced model for the nominal problem, i.e., the latent parameter is set to the nominal latent parameter $\eta_0 \in \mathcal{D}$ that corresponds to the no-damage condition. We then decrease the thickness of the plate stepwise and adapt the reduced model. After each change in the latent parameter, synthetic incomplete sensor samples are computed with the full model, which are used to adapt the reduced model. The following sections give details on the problem setup and report the numerical results.

Plate problem

We consider the static analysis of a plate in bending. The plate is clamped into a frame and a load is applied. Our problem is an extension of the plate problems introduced in [23,42,44]. The geometry of our plate problem is shown in Fig. 3a. The spatial domain $\Omega \in [0, 1]^2 \subset \mathbb{R}^2$ is split into two subdomains $\Omega = \Omega_1 \cup \Omega_2$. The problem has three observable parameters $\mu = [\mu_1, \mu_2, \mu_3]^T \in \mathcal{D}$ with $\mathcal{D} = [0.05, 0.1]^2 \times [1, 100]$. The observable parameters μ_1 and μ_2 control the nominal thickness of the plate in the subdomain Ω_1 and Ω_2 , respectively. The third observable parameter μ_3 defines the load on the plate.

The latent parameter $\eta = [\eta_1, \eta_2]^T \in \mathcal{E}$ controls the damage of the plate, i.e., the latent parameter defines the local decrease of the thickness that corresponds to the damage. The



domain of the latent parameter is $\mathcal{E} = [0, 0.2] \times (0, 0.05]$. The thickness of the plate at position $\mathbf{x} \in \Omega$ is given by the function $t : \Omega \times \mathcal{D} \times \mathcal{E} \rightarrow \mathbb{R}$ with

$$t(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\eta}) = t_0(\mathbf{x}; \boldsymbol{\mu}) \left(1 - \eta_1 \exp \left(-\frac{1}{2\eta_2^2} \|\mathbf{x} - \mathbf{z}\|_2^2 \right) \right),$$

and

$$t_0(\mathbf{x}; \boldsymbol{\mu}) = \begin{cases} \mu_1 & \text{if } x_1 > 0.5 \\ \mu_2 & \text{if } x_1 \leq 0.5 \end{cases},$$

with position $\mathbf{z} = [0.7, 0.4]^T \in \Omega$. The function t is nonlinear in \mathbf{x} , $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$. We set the nominal latent parameter $\boldsymbol{\eta}_0$ to $\boldsymbol{\eta}_0 = [0, 0.01]^T \in \mathcal{E}$ that corresponds to no local decrease of the thickness and therefore to the no-damage condition.

The full model of the plate problem is a finite element model, see [23]. The corresponding system of equations is of the form (1), where $l_A = 4$, $l_f = 1$, $\Theta_f^{(1)}(\boldsymbol{\mu}) = \mu_3$,

$$\Theta_A^{(1)}(\boldsymbol{\mu}) = \mu_1^3, \quad \Theta_A^{(2)}(\boldsymbol{\mu}) = \mu_2^3,$$

and

$$\Theta_A^{(3)}(\boldsymbol{\mu}) = \mu_1, \quad \Theta_A^{(4)}(\boldsymbol{\mu}) = \mu_2$$

The system of equations has $N = 4719$ degrees of freedom. The thickness of the plate with $\boldsymbol{\mu} = [0.08, 0.07, 50]^T \in \mathcal{D}$ and with $\boldsymbol{\eta} = \boldsymbol{\eta}_0$ is visualized in Fig. 4a and the deflection in

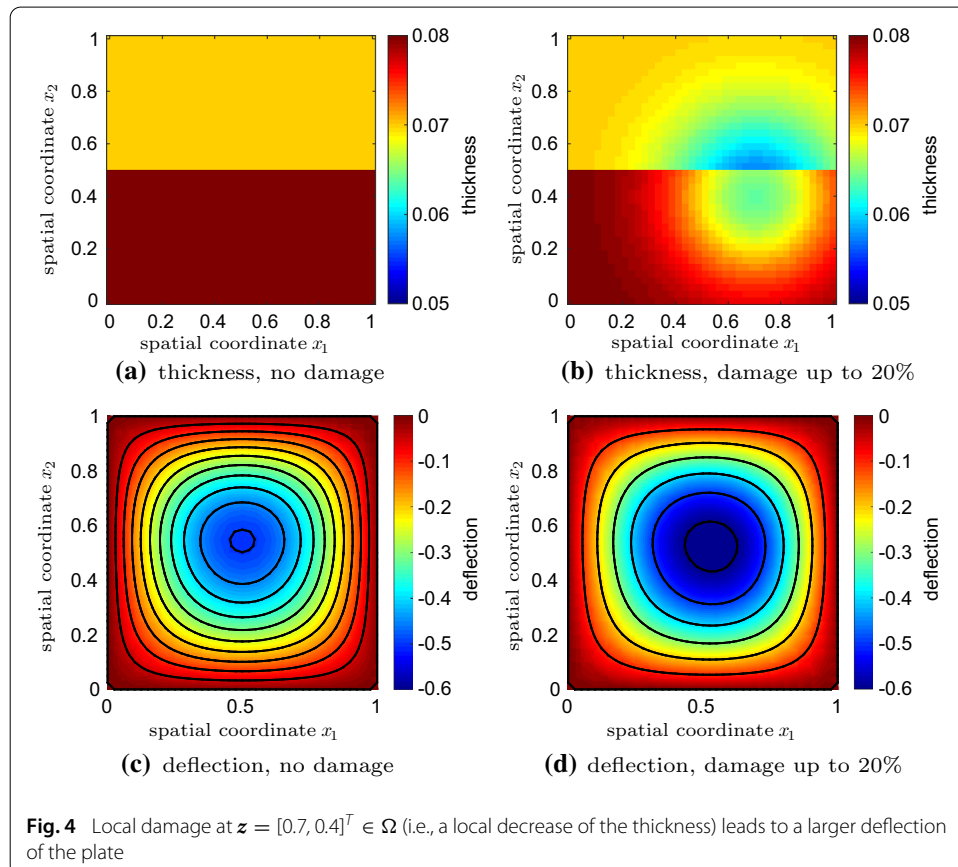


Fig. 4c. The thickness and the deflection of the plate with a damage up to 20 %, i.e., a local decrease of the thickness of the plate at \mathbf{z} by 20 %, is shown in Fig. 4b and d, respectively.

We draw $M = 1000$ observable parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_M \in \mathcal{D}$ uniformly in \mathcal{D} and compute the corresponding state vectors with the full model to assemble the snapshot matrix

$$Y_{\eta_0} = [\hat{\mathbf{y}}_{\eta_0}(\boldsymbol{\mu}_1), \dots, \hat{\mathbf{y}}_{\eta_0}(\boldsymbol{\mu}_M)] \in \mathbb{R}^{N \times M}.$$

Note that the latent parameter $\boldsymbol{\eta} = \boldsymbol{\eta}_0$ is set to the nominal latent parameter $\boldsymbol{\eta}_0$. Figure 3b plots the decay of the singular values of the snapshot matrix Y_{η_0} . We construct a reduced model via Galerkin projection onto the space spanned by the first $n = 8$ POD basis vectors of Y_{η_0} .

Setup of numerical experiments

We now describe the details of our numerical experiments. We have ten latent parameters $\boldsymbol{\eta}_0, \boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_9 \in \mathcal{E}$, where $\boldsymbol{\eta}_0$ is the nominal latent parameter corresponding to the no-damage condition and

$$\boldsymbol{\eta}_i = \left[\frac{2i}{90}, \frac{2i}{360} \right]^T \in \mathcal{E}, \quad i = 1, \dots, 9.$$

This means that from latent parameter $\boldsymbol{\eta}_{i-1}$ to $\boldsymbol{\eta}_i$ the thickness at position \mathbf{z} is decreased by a factor of two, for $i = 1, \dots, 9$. After each change of the latent parameter, the sensor window is flushed and $M' \in \mathbb{N}$ incomplete sensor samples are received to adapt the reduced model.

Number of sensor samples

We receive incomplete sensor samples, and therefore we use the extension to the dynamic data-driven adaptation described in “Dynamic data-driven adaptation from incomplete sensor samples” section. This means that the adaptivity steps $h = 1, \dots, M'$ required for adapting from latent parameter $\boldsymbol{\eta}_{i-1}$ to $\boldsymbol{\eta}_i$ are split into $M^{\text{basis}} \in \mathbb{N}$ steps to derive the gappy POD basis and $M^{\text{update}} \in \mathbb{N}$ steps to update the reduced model. We chose M^{basis} and M^{update} conservatively in the following, because we are primarily interested in studying the effect of the number of missing components in the incomplete sensor samples onto the adaptation, rather than the number of sensor samples; see [42] for studies on the effect of the number of samples on the dynamic data-driven adaptation in the case with complete sensor samples. We set $M^{\text{basis}} = 5000$ and therefore derive the gappy POD basis from $M^{\text{basis}} = 5000$ incomplete sensor samples. We buffer 50 incomplete sensor samples and use them in the incremental basis generation procedure described in “Deriving the gappy POD basis” section.

The theory of the dynamic data-driven adaptation with complete sensor samples gives guidance on the selection of M^{update} . In case of complete sensor samples, setting $M^{\text{update}} = l_A \times n$ is sufficient to recover the reduced model that would be obtained via rebuilding from scratch [42]. Note that $l_A = 4$ is the number of $\boldsymbol{\mu}$ -independent operators and $n = 8$ the dimension of the POD basis space. We set $M^{\text{update}} = 5 \times l_A \times n = 160$ since we adapt from incomplete sensor samples and therefore expect that the approximation of the missing values introduces additional error into the adaptation. In total, we receive $M' = M^{\text{basis}} + M^{\text{update}} = 5160$ incomplete sensor samples to adapt from $\boldsymbol{\eta}_{i-1}$ to $\boldsymbol{\eta}_i$ for $i = 1, \dots, 9$.

Sensor sample generation

The number of missing components $N - k$ in the incomplete sensor samples is controlled by the number of known components k . To discuss the effect of k on the adaptation, we introduce separate numbers of known components $k^{\text{basis}} \in \mathbb{N}$ and $k^{\text{update}} \in \mathbb{N}$ for the gappy POD basis construction and the update, respectively. Furthermore, we introduce the sensor rates

$$\rho^{\text{basis}} = \frac{k^{\text{basis}}}{N} \times 100, \quad \rho^{\text{update}} = \frac{k^{\text{update}}}{N} \times 100,$$

which are the percent of the number of known components of the total number of components N in the incomplete sensor samples. Thus, for example, $\rho^{\text{basis}} = 100\%$ means that all components are known and therefore that we have a complete sensor sample.

We synthetically generate incomplete sensor samples with the full model at each step $h = 1, \dots, M'$. We therefore first draw uniformly an observable parameter μ_{M+h} in \mathcal{D} and compute the state vector $y_\eta(\mu_{M+h})$ with the full model for the current latent parameter μ . We then draw $k \in \mathbb{N}$ unique indices uniformly in $\{1, \dots, N\}$ and construct the point selection matrix $P_h \in \mathbb{R}^{N \times k}$. The incomplete sensor sample is $\hat{y}_\eta^{\text{incp}}(\mu_{M+h}) = P_h P_h^T y_\eta(\mu_{M+h})$.

Error computation

We compare three reduced models:

- A static reduced model that is built as described in “Classical model reduction for systems with latent parameters” section. The static reduced model is not adapted to changes in the latent parameter.
- A rebuilt reduced model that is derived as in “Classical model reduction for systems with latent parameters” section but from M^{update} complete sensor samples corresponding to the current changed latent parameter. This requires repeating the computation of the POD basis and the operator projections, which is prohibitively expensive to conduct online.
- An online adaptive reduced model that is adapted to changes in the latent parameter from incomplete sensor samples with the dynamic data-driven adaptation described in Algorithm 1.

To assess the quality of the reduced models quantitatively, we draw ten observable parameters $\mu'_1, \dots, \mu'_{10} \in \mathcal{D}$ uniformly in \mathcal{D} and compute the relative L_2 error with respect to the full model

$$e_r = \frac{1}{10} \sum_{i=1}^{10} \frac{\|\hat{y}_\eta(\mu'_i) - y_\eta(\mu'_i)\|_2}{\|y_\eta(\mu'_i)\|_2}, \tag{8}$$

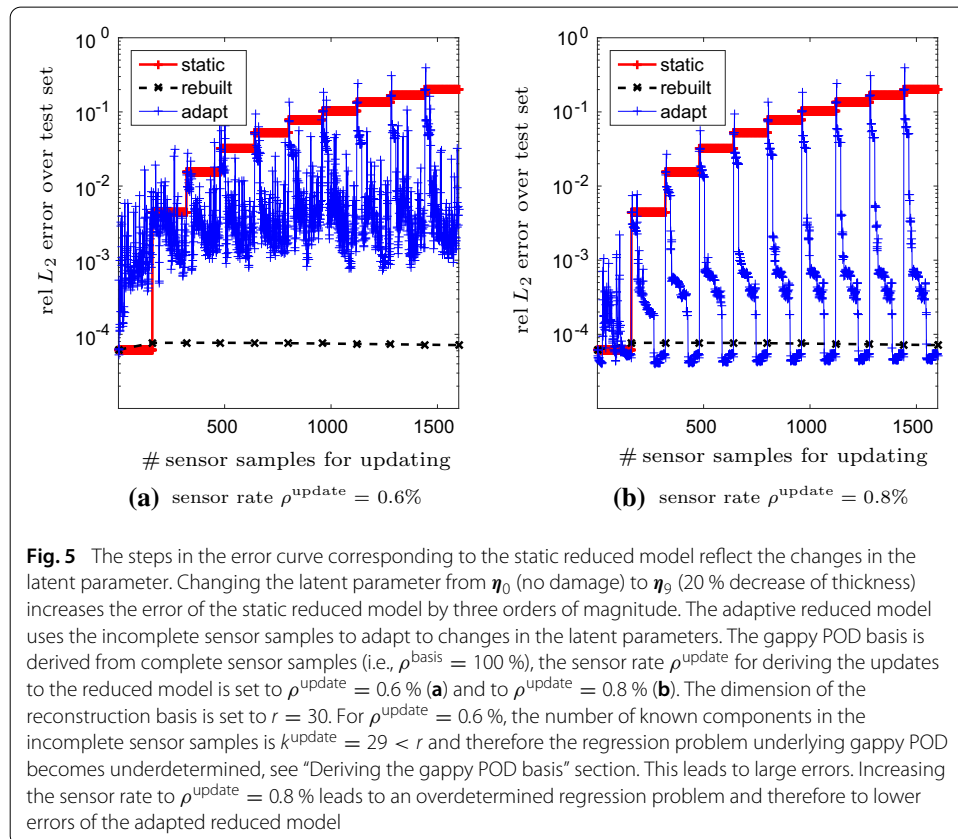
where η is the current latent parameter and $\hat{y}_\eta(\mu'_1), \dots, \hat{y}_\eta(\mu'_{10}) \in \mathbb{R}^n$ are the state vectors obtained with either the static, the rebuilt, or the adapted reduced model.

Gappy POD basis from complete sensor samples

We first consider the situation where $\rho^{\text{basis}} = 100\%$ is fixed and the sensor rate ρ^{update} varies. This means that we have available complete sensor samples (without missing components) for deriving the gappy POD basis in the first M^{basis} steps but incomplete sensor samples for updating the reduced model in the final M^{update} adaptivity steps.

Figures 5 and 6 demonstrate the effect of the sensor rate ρ^{update} on the dynamic data-driven adaptation. First consider the static reduced model. As the latent parameter changes from η_0 (no damage) to η_9 (20 % decrease of thickness), the error of the static reduced model increases by three orders of magnitude. The steps in the error curve reflect the changes in the latent parameter. The error of the rebuilt reduced model stays near 10^{-4} . Consider now the adaptive reduced model. The dimension of the gappy POD basis is set to $r = 30$. Figure 5 shows that a sensor rate $\rho^{\text{update}} = 0.6\%$ leads to an adapted reduced model with large errors. A sensor rate of $\rho^{\text{update}} = 0.6\%$ means that $k^{\text{update}} = 29$ components of the incomplete sensor sample are known, and therefore $k^{\text{update}} < r$. This violates the condition of gappy POD that requires a full-column rank $\mathbf{P}_h^T \mathbf{U}_{M^{\text{basis}}}$, see “Deriving the gappy POD basis” section. For a slightly larger sensor rate $\rho^{\text{update}} = 0.8\%$, and $k^{\text{update}} > r$, our dynamic data-driven adaptation from incomplete sensor samples recovers the rebuilt reduced model. Figure 6 indicates that increasing the sensor rate ρ^{update} reduces the error of the adapted reduced model in the first few adaptivity steps after a change in the latent parameter, cf. Fig. 5.

Note that the adapted reduced model achieves a slightly lower error than the rebuilt reduced model in Figs. 5 and 6. The dynamic data-driven adaptation constructs the adapted operators with an optimization problem from the sensor samples projected onto the POD space. This projection and the optimization cause the difference in the error of the adapted and the rebuilt reduced model, if the dimension of the reduced model is low. The difference decreases if the dimension of the reduced model is increased, see [42, Theorem 1].



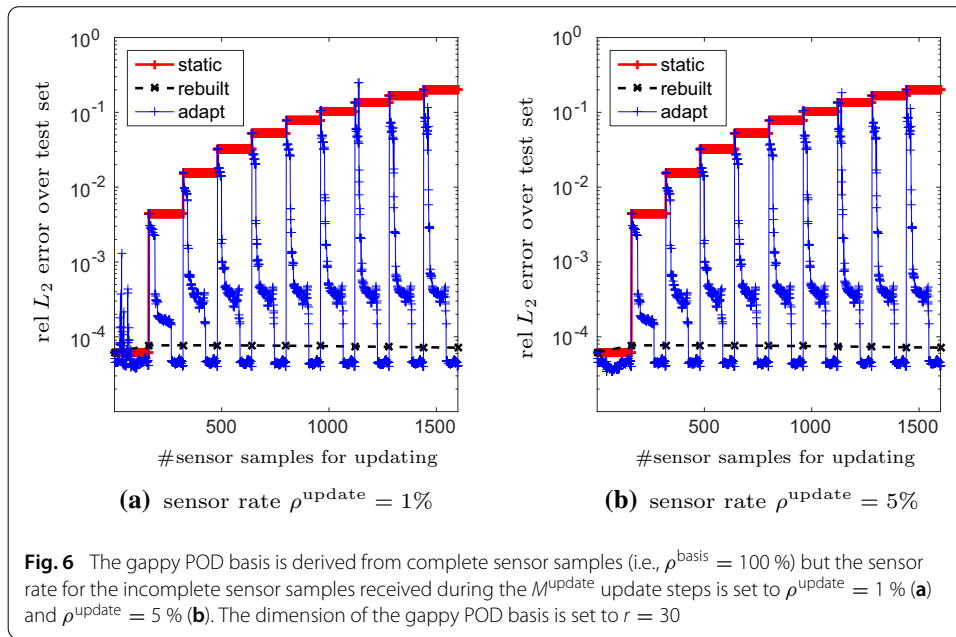
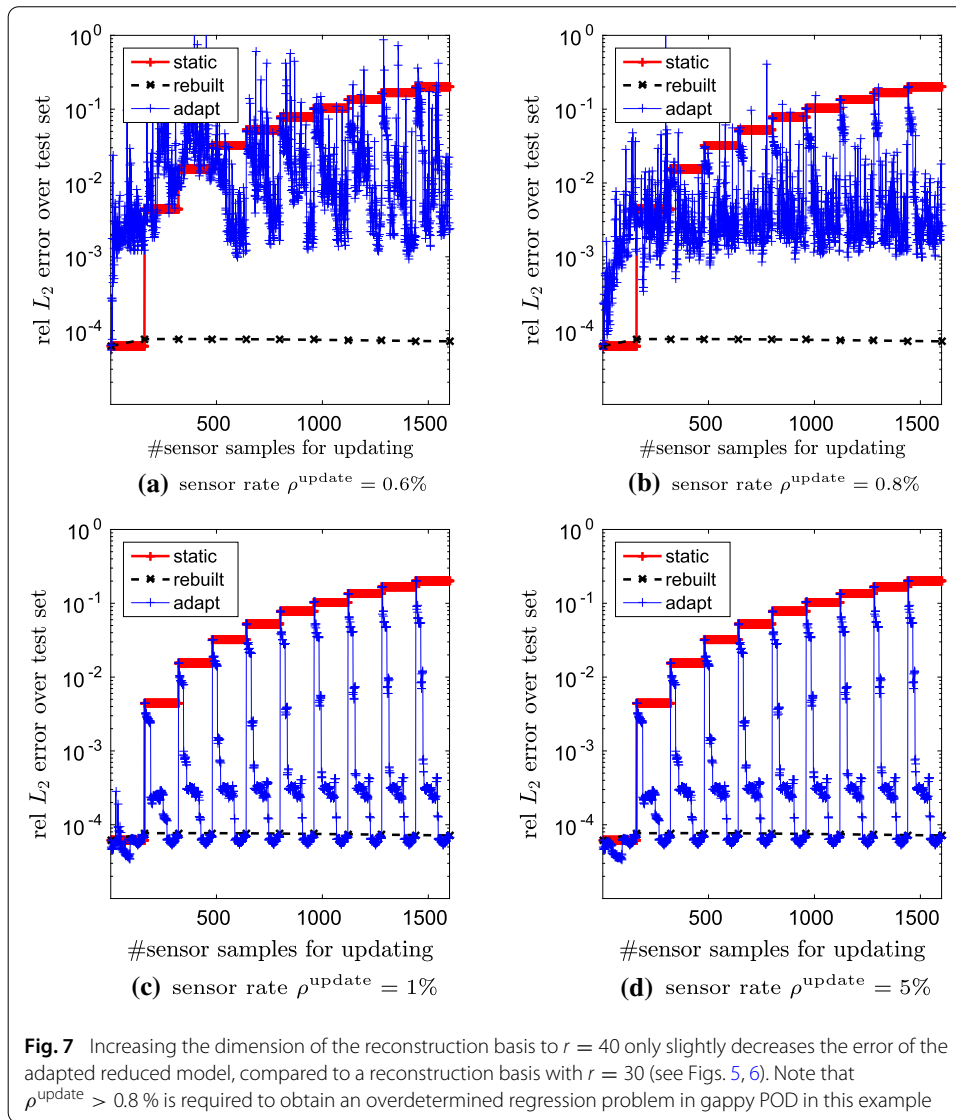


Figure 7 reports the error behavior of an adapted reduced model that uses a gappy POD basis of dimension $r = 40$. For $\rho^{\text{update}} = 0.6\%$ and $\rho^{\text{update}} = 0.8\%$, we again obtain the situation $k^{\text{update}} < r$ and therefore obtain an underdetermined least-squares problem that introduces large errors in the adaptation. However, if the sensor rate ρ^{update} is increased, the approximation quality of the adapted reduced model increases too. The results in Fig. 6 for $r = 30$ are similar to the result obtained in Fig. 7 for $r = 40$. This shows that a gappy POD basis with $r = 30$ dimensions is sufficient in this example.

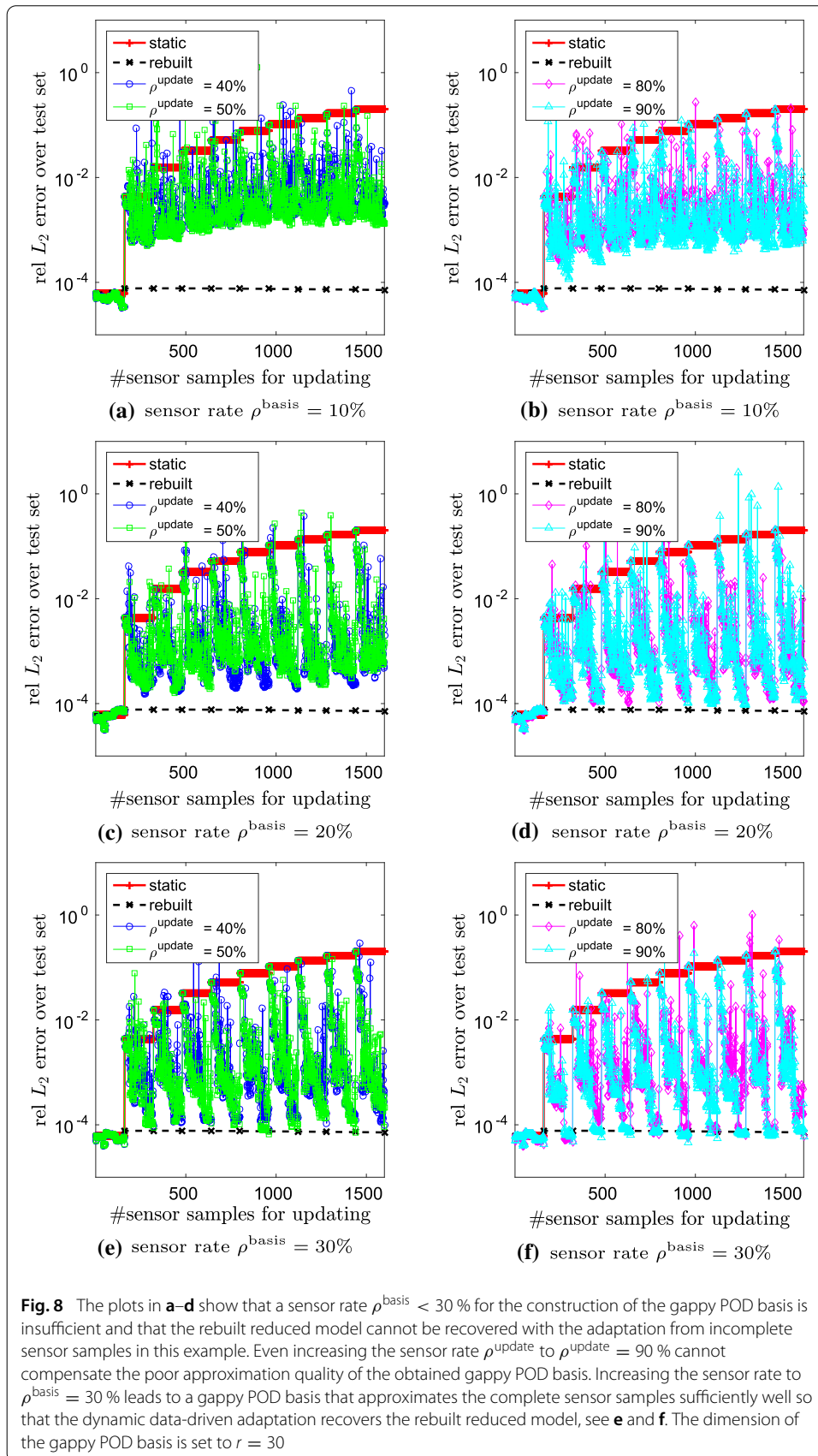
Gappy POD basis from incomplete sensor samples

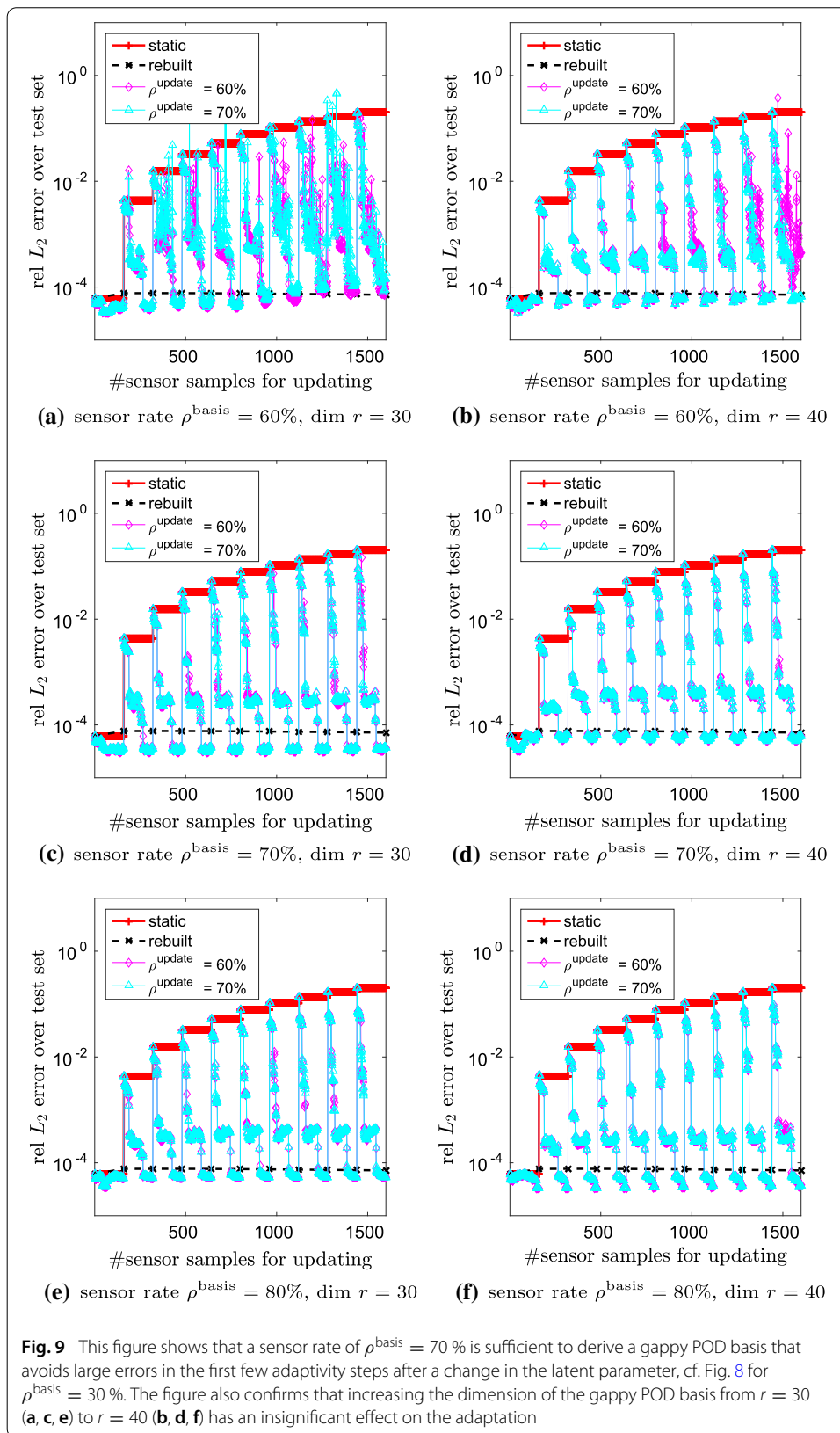
We now consider the situation where $\rho^{\text{basis}} < 100\%$ and $\rho^{\text{update}} < 100\%$, i.e., the gappy POD basis is derived from incomplete sensor samples and the updates to the reduced models are obtained from incomplete sensor samples as well. Figure 8 shows the effect of the sensor rate ρ^{basis} on the adaptation. Figures 8a and b demonstrate that a sensor rate $\rho^{\text{basis}} = 10\%$ is too low to recover the rebuilt reduced model with the adapted reduced model in this example. Even setting the sensor rate for the update to $\rho^{\text{update}} = 90\%$ (i.e., generating the gappy POD basis from incomplete samples with $\rho^{\text{basis}} = 10\%$ and updating the reduced model from approximate sensor samples with $\rho^{\text{update}} = 90\%$) cannot compensate the inadequate sensor rate $\rho^{\text{basis}} = 10\%$. Increasing the sensor rate for the gappy POD basis construction to $\rho^{\text{basis}} = 30\%$ leads to an adapted reduced model that recovers the rebuilt reduced model. However, with $\rho^{\text{basis}} = 30\%$ there still are outliers that lead to a reduced model with a large error. Figure 9 shows that increasing the sensor rate to $\rho^{\text{basis}} = 70\%$ reduces those outliers significantly. Again, increasing the dimension of the gappy POD basis from $r = 30$ to $r = 40$ only slightly reduces the error of the adapted reduced model, compare Fig. 9a, c, e with Fig. 9b, d, f.

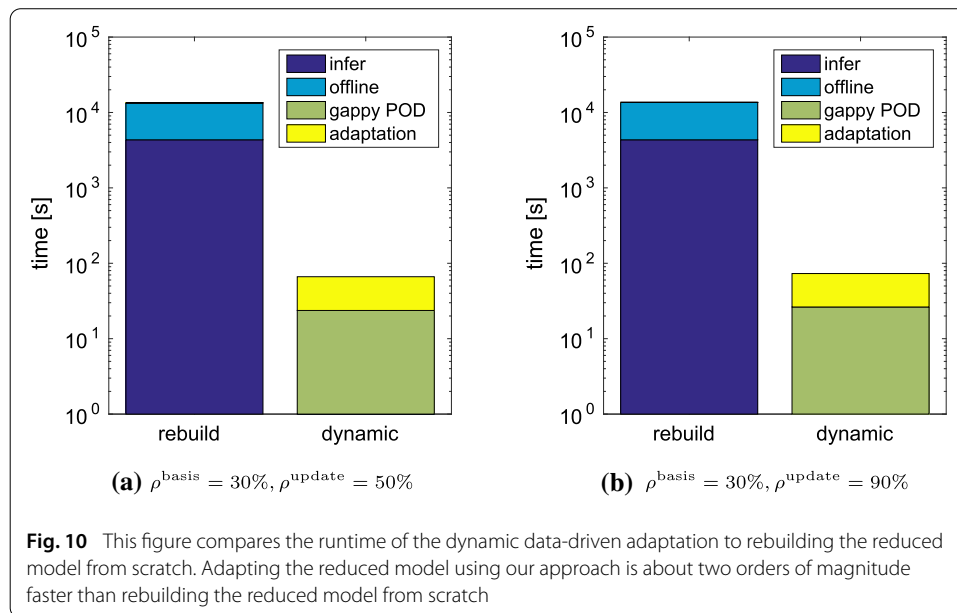
Figure 10 reports the runtime of the dynamic data-driven adaptation for $\rho^{\text{basis}} = 30\%$, $\rho^{\text{update}} = 50\%$ and $\rho^{\text{basis}} = 30\%$, $\rho^{\text{update}} = 90\%$. The latent parameter changes



from η_0 to η_9 in nine steps. For each of the nine latent parameters η_1, \dots, η_9 , the gappy POD basis is derived and the reduced model is adapted in M^{update} steps to the incomplete sensor samples. Thus, in total, nine gappy POD bases are derived and $9 \times M^{\text{update}}$ adaptivity steps are performed for adapting from η_0 to η_9 . Figure 10 reports the total runtime split into the runtime of the gappy POD basis construction and the adaptation. The runtime of the dynamic data-driven adaptation is compared to the runtime of rebuilding the reduced model from scratch in each of the $9 \times M^{\text{update}}$ adaptivity steps. The runtime of rebuilding the reduced model is split into the runtime of inferring the latent parameter from the sensor samples and the runtime of the offline phase where the reduced operators are constructed, see “Background” section. The dynamic data-driven approach achieves a speedup of about two orders of magnitude compared to rebuilding the reduced model from scratch. Increasing ρ^{update} from $\rho^{\text{update}} = 50\%$ to $\rho^{\text{update}} = 90\%$ only slightly changes the runtime of the dynamic data-driven adaptation. The runtime measurements were performed on an i5-3570 CPU.







Summary and future work

We proposed an extension to the dynamic data-driven adaptation that handles incomplete sensor samples, i.e., partial measurements of the large-scale state. In our approach, a gappy POD basis is derived from incomplete sensor samples. The missing values of the incomplete sensor samples are approximated with gappy POD in the space spanned by the gappy POD basis. The reduced model is then adapted using the gappy POD approximations of the complete sensor samples with the dynamic data-driven adaptation. The numerical results confirm that about 30–40 % of the total number of components of the sensor samples are sufficient to recover the reduced model that would be obtained via rebuilding from scratch.

Future sensing technologies (e.g., “sensor skins”) of next-generation engineering systems will provide high-resolution measurements. Processing these large data sets will be computationally challenging. In big data analytics, sublinear algorithms are currently developed that look at only a subset of the given data set to meet runtime requirements [48]. Our approach follows a similar paradigm. We selectively process sensor data that are most informative for deriving the update to the reduced model and ignore large parts of the received data that are irrelevant in the current situation. Our approach is applicable even if the selection of the high-resolution sensor data is dynamically changing online, e.g., due to new damage events.

We considered here real-time structural assessment and decision-making but sensor data are available in many other applications. For example, in control, the goal is to design a controller that stabilizes a dynamical system. However, if the dynamical system passes through multiple regimes with different system characteristics, a single controller might be insufficient to stabilize the system. If sensor data, e.g., sparse measurements of the state of the dynamical system, are available, the controller can be adapted to the sensor data to take into account the changes in the underlying dynamical system. We also mention system identification as a potential application of our adaptation approach. Instead of starting with reduced operators derived in an offline phase, one could start with initial operators

that have all components set to zero, and then adapt these operators to the available data. Such a system identification approach would derive a reduced model directly from data. In general, our approach is applicable to DDDAS for which massive amounts of sensor data are available.

Authors' contributions

BP and KW developed the methodology, performed the numerical investigations, and wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work was supported in part by the AFOSR MURI on multi-information sources of multi-physics systems under Award Number FA9550-15-1-0038, program manager Jean-Luc Cambier, and by the United States Department of Energy Applied Mathematics Program, Awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center. Some of the numerical examples were computed on the computer cluster of the Munich Centre of Advanced Computing.

Competing interests

The authors declare that they have no competing interests.

Received: 28 November 2015 Accepted: 8 March 2016

Published online: 21 March 2016

References

- Allaire D, Chambers J, Cowlagi R, Kordonowy D, Lecerf M, Mainini L, Ulker F, Willcox K. An offline/online DDDAS capability for self-aware aerospace vehicles. *Procedia Comput Sci.* 2013;18:1959–68.
- Amsallem D, Farhat C. An online method for interpolating linear parametric reduced-order models. *SIAM J Sci Comput.* 2011;33(5):2169–98.
- Amsallem D, Zahr M, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *Int J Numer Methods Eng.* 2012;92(10):891–916.
- Amsallem D, Zahr M, Washabaugh K. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. Special issue on model reduction of parameterized systems (MoRePaS). *Adv Comput Math.* 2014. (accepted).
- Astrid P, Weiland S, Willcox K, Backx T. Missing point estimation in models described by proper orthogonal decomposition. *Autom Control IEEE Trans.* 2008;53(10):2237–51.
- Barrault M, Maday Y, Nguyen N, Patera A. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Math.* 2004;339(9):667–72.
- Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* 2015;57(4):483–531.
- Brand M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl.* 2006;415(1):20–30.
- Brunton SL, Tu JH, Bright I, Kutz JN. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM J Appl Dyn Syst.* 2014;13(4):1716–32.
- Bui-Thanh T, Damodaran M, Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J.* 2004;42(8):1505–16.
- Burkardt J, Gunzburger M, Lee H-C. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Comput Methods Appl Mech Eng.* 2006;196(1–3):337–55.
- Carlberg K. Adaptive h-refinement for reduced-order models. *Int J Numer Methods Eng.* 2015;102(5):1192–210.
- Carlberg K, Bou-Mosleh C, Farhat C. Efficient non-linear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations. *Int J Numer Methods Eng.* 2011;86(2):155–81.
- Carlberg K, Ray J, van Bloemen Waanders B. Decreasing the temporal complexity for nonlinear, implicit reduced-order models by forecasting. *Comput Methods Appl Mech Eng.* 2015;289:79–103.
- Chaturantabut S, Sorensen D. Nonlinear model reduction via discrete empirical interpolation. *SIAM J Sci Comput.* 2010;32(5):2737–64.
- Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng.* 2011;18(4):395–404.
- Cui T, Marzouk YM, Willcox KE. Data-driven model reduction for the bayesian solution of inverse problems. *Int J Numer Methods Eng.* 2015;102(5):966–90.
- Degroote J, Vierendeels J, Willcox K. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int J Numer Methods Fluids.* 2010;63(2):207–30.
- Dihlmann M, Drohmann M, Haasdonk B. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. In: Aubry D, Díez P, Tie B, Parés N, editors, Proceedings of the international conference on adaptive modeling and simulation; 2011. p. 156–67.
- Eftang J, Patera A. Port reduction in parametrized component static condensation: approximation and a posteriori error estimation. *Int J Numer Methods Eng.* 2013;96(5):269–302.
- Eftang J, Stamm B. Parameter multi-domain hp empirical interpolation. *Int J Numer Methods Eng.* 2012;90(4):412–28.
- Everson R, Sirovich L. Karhunen–Loève procedure for gappy data. *J Opt Soc Am A.* 1995;12(8):1657–64.
- Ferreira A. MATLAB codes for finite element analysis. Berlin: Springer; 2008.

24. Galbally D, Fidkowski K, Willcox K, Ghattas O. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *Int J Numer Methods Eng.* 2010;81(12):1581–608.
25. Garmatter D, Haasdonk B, Harrach B. A reduced landweber method for nonlinear inverse problems. Technical report, University of Stuttgart; 2015. Available at Arxiv.
26. Ghnatios C, Masson F, Huerta A, Leygue A, Cueto E, Chinesta F. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Comput Methods Appl Mech Eng.* 2012;213–216:29–41.
27. Gogu C. Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. *Int J Numer Methods Eng.* 2015;101(4):281–304.
28. González D, Masson F, Poulhaon F, Leygue A, Cueto E, Chinesta F. Proper generalized decomposition based dynamic data driven inverse identification. *Math Comput Simul.* 2012;82(9):1677–95.
29. Gugercin S, Antoulas A. A survey of model reduction by balanced truncation and some new results. *Int J Control.* 2004;77(8):748–66.
30. Kaulmann S, Haasdonk B. Online greedy reduced basis construction using dictionaries. In: Troch I, Breitenacker F, editors, Proceedings of 7th Vienna international conference on mathematical modelling; 2012. p. 112–7.
31. Kennedy M, O'Hagan A. Bayesian calibration of computer models. *J R Stat Soc Series B (Stat Methodol).* 2001;63(3):425–64.
32. Lass O. Reduced order modeling and parameter identification for coupled nonlinear PDE systems. PhD thesis, University of Konstanz; 2014.
33. Mainini L, Karen E. Willcox. A surrogate modeling approach to support real-time structural assessment and decision-making. In 10th AIAA multidisciplinary design optimization conference, AIAA SciTech. American Institute of Aeronautics and Astronautics; 2014.
34. Lecerf M, Allaire D, Willcox K. Methodology for dynamic data-driven online flight capability estimation. *AIAA J.* 2015;53(10):3073–87.
35. Maday Y, Stamm B. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J Sci Comput.* 2013;35(6):A2417–41.
36. Mainini L, Willcox K. Sensitivity analysis of surrogate-based methodology for real time structural assessment. In AIAA modeling and simulation technologies conference, AIAA SciTech 2015, AIAA. AIAA; 2015. Paper 1362.
37. Mainini L, Willcox K. Surrogate modeling approach to support real-time structural assessment and decision making. *AIAA J.* 2015;53(6):1612–26.
38. Ohlberger M, Schindler F. Error control for the localized reduced basis multi-scale method with adaptive on-line enrichment. *SIAM J Sci Comput.* 2015. (accepted).
39. Panzer H, Mohring J, Eid R, Lohmann B. Parametric model order reduction by matrix interpolation. at *Automatisierungstechnik.* 2010;58(8):475–84.
40. Peherstorfer B, Butnaru D, Willcox K, Bungartz H-J. Localized discrete empirical interpolation method. *SIAM J Sci Comput.* 2014;36(1):A168–92.
41. Peherstorfer B, Willcox K. Detecting and adapting to parameter changes for reduced models of dynamic data-driven application systems. *Procedia Comput Sci.* 2015;51:2553–62.
42. Peherstorfer B, Willcox K. Dynamic data-driven reduced-order models. *Comput Methods Appl Mech Eng.* 2015;291:21–41.
43. Peherstorfer B, Willcox K. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM J Sci Comput.* 2015;37(4):A2123–50.
44. Peherstorfer B, Willcox K, Gunzburger M. Optimal model management for multifidelity Monte Carlo estimation. Technical report 15–2, Aerospace Computational Design Laboratory, MIT; 2015.
45. Rozza G, Huynh D, Patera A. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch Comput Methods Eng.* 2007;15(3):1–47.
46. Sargsyan S, Brunton SL, Kutz JN. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Phys Rev E.* 2015;92:033304.
47. Ventsel E, Krauthammer T. Thin plates and shells. Boca Raton: CRC Press; 2001.
48. Wang D, Han Z. Sublinear algorithms for big data applications. Berlin: Springer; 2015.
49. Willcox K. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Comput Fluids.* 2006;35(2):208–26.
50. Zahr M, Farhat C. Progressive construction of a parametric reduced-order model for pde-constrained optimization. *Int J Numer Methods Eng.* 2015;102(5):1111–35.
51. Zimmermann R. A locally parametrized reduced-order model for the linear frequency domain approach to time-accurate computational fluid dynamics. *SIAM J Sci Comput.* 2014;36(3):B508–37.