A surrogate modeling approach to support real-time structural assessment and decision-making

L. Mainini * and K. Willcox †

Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, USA‡

This paper proposes a data-driven strategy to assist online rapid decision-making for an unmanned aerial vehicle that uses sensed data to estimate its structural state, uses this estimate to update its corresponding flight capabilities, and then dynamically re-plans its mission accordingly. Our approach comprises offline and online computational phases constructed to address the sense-plan-act information flow while avoiding a costly online inference step. During the offline phase, high-fidelity finite element simulations are used to construct reduced-order models and classification criteria: proper orthogonal decomposition approximations and self-organizing maps are combined to realize a fast mapping from measured quantities to system capabilities. During the online phase, the surrogate mapping is employed to directly estimate the vehicle's evolving structural capability from sensor data. The approach is demonstrated for a test problem of a composite wing panel on an unmanned aerial vehicle that undergoes degradation in structural properties.

Nomenclature

a_{ij}^m	ith coefficient of the response surface for the jth POD coefficient to approximate $\mathbf{q}_m$
b_{ij}^c	ith coefficient of the local response surface for the jth POD coefficient to approximate $\mathbf{s}_c$
C	Number of capability quantities of interest
d_d	Number of remaining plies (damage depth)
E	Number of elements of the FEM model
e_1	Normalized root mean square error between the original snapshot $\mathbf{q}$ and its approximation $\tilde{\mathbf{q}}$
e_2	Normalized root mean square error between the original $\mathbf{FI}_{\max}$ and its approximation $\tilde{\mathbf{FI}}_{\max}$
e_g^m	Error between reconstructed and sensed $\mathbf{q}_m$ using gappy norm
$\mathbf{f}^m$	Gappy measurement vector for $\mathbf{q}_m$
FI	Failure Index
$\mathbf{FI}$	Failure Index vector
$FI_{\max}$	Maximum value of Falure Index in vector $\mathbf{FI}$
$\mathbf{FI}_{\max}$	Vector of all $FI_{\max}$ in the validation set
$\mathbf{G}^m$	Gappy POD matrix for $\mathbf{q}_m$
l	Panel loads
ℓ_c	Number of retained POD modes to approximate $\mathbf{s}_c$
M	Number of measured quantities of interest
N_α	Number of reconstructed measurement POD coefficients
N_β	Number of evaluated capability POD coefficients
N_c	Number of SOM clusters
n_m	Number of retained POD modes to approximate $\mathbf{q}_m$
n_{pod}	Total number of POD modes over all quantities of interest
n_{s}	Number of snapshots of the evaluation set computed for each quantity of interest
n_{sv}	Number of snapshots of the validation set computed for each quantity of interest
n_v	Number of state variables x_i that constitute the state vector $\mathbf{x}$

*Postdoctoral Associate, Department of Aeronautics and Astronautics, MIT, AIAA Member.

†Professor, Department of Aeronautics and Astronautics, MIT, AIAA Associate Fellow.

‡An earlier draft of this paper was presented at the 2014 AIAA Scitech conference, AIAA-2014-1488.

n_w	Number of SOM weight vectors
$\mathbf{q}_m$	mth measured quantity of interest
$\mathbf{s}_c$	cth capability quantity of interest
$\bar{t}$	Mean computational runtime computed over the evaluation set
$\mathbf{T}$	SOM training matrix with n_s training row vectors $\boldsymbol{\tau}_i$
$\mathbf{w}_j$	jth SOM weight vector
x_i	ith component of state vector $\mathbf{x}$
$\mathbf{x}$	Vehicle state vector
y_d	Damage centroid location along y-direction
Δy	Damage size along y-direction
z_d	Damage centroid location along z-direction
Δz	Damage size along z-direction
α_j^m	jth POD modal coefficient for $\mathbf{q}_m$
$\boldsymbol{\alpha}$	Vector of all N_α measurement POD coefficients
$\boldsymbol{\alpha}_g$	Vector of all N_α measurement POD coefficients reconstructed with gappy POD
$\boldsymbol{\alpha}_g^m$	Vector of gappy reconstruction of POD coefficients for $\mathbf{q}_m$
β_j^c	jth POD modal coefficient for $\mathbf{s}_c$
ε	Strain component ($\mu\varepsilon$)
λ_j^m	jth POD eigenvalue for $\mathbf{q}_m$
$\boldsymbol{\Lambda}$	Diagonal matrix of normalized POD eigenvalues
μ_j^c	jth POD eigenvalue for $\mathbf{s}_c$
σ	Experienced stress
$\sigma_{\max}$	Maximum allowable stress
φ_j^m	jth POD basis vector for $\mathbf{q}_m$
ψ_j^c	jth POD basis vector for $\mathbf{s}_c$
$\boldsymbol{\tau}_i$	ith SOM training vector collecting POD coefficients over all quantities of interest

Subscript

max	Maximum value in a snapshot
min	Minimum value in a snapshot
n_1	Normal component along first main orthotropic axis
n_2	Normal component along second main orthotropic axis
s_{12}	Shear component on ply plane

Superscript

c	Capability quantity of interest number
m	Measured quantity of interest number
test	Test case

I. Introduction

This paper develops computational methods to support real-time decision-making in the face of dynamic data. Our setting is the specific challenge of an unmanned aerial vehicle (UAV) that uses sensed structural data to estimate its structural state, uses this estimate to update its corresponding flight capabilities, and then dynamically re-plans its mission accordingly. Figure 1 depicts the overall sense-infer-plan-act framework, in which we seek not just to make estimates but also to characterize their uncertainties. This paper focuses specifically on structural assessment at the wing panel level. Our strategy is to conduct simulations in an offline phase and build a surrogate model that maps sensed quantities (strain) to structural capabilities (failure indices). In this way, we avoid conducting an expensive inference problem in the online phase, although we capture its essence through the offline construction of the surrogate models. The surrogate models are approximate and intended to yield rapid first-cut estimates to support online decision-making.

Structural health monitoring and structural reliability analysis have received considerable research attention in the past decades. As stated by Ref. 1, structural health monitoring can be defined as "the process of implementing a damage-detection strategy"; on the other hand, structural reliability assessment relates to

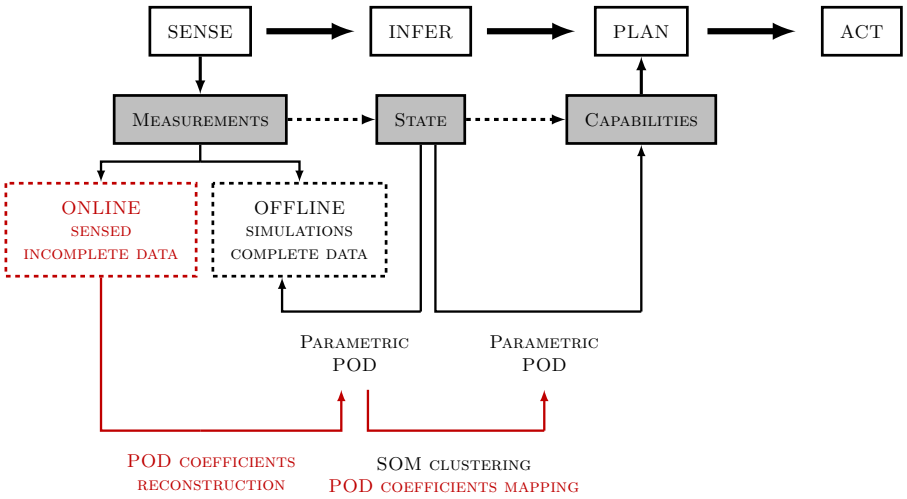


Figure 1. The sense-infer-plan-act framework and our mapping strategy (offline process in black and online process in red).

the evaluation of the probability of failure of the system. Thus, the former focuses on methods for damage identification, while the latter deals with methods for the identification of limit state functions for binary (failure/no-failure) problems. Since our work targets online structural assessment in order to enable dynamic mission re-planning, we require both damage identification and system capability evaluation, thus drawing on both fields.

There is a wide range of approaches to model-based structural health assessment. Many of these works rely on vibration-based strategies that evaluate changes in structural properties through dynamic analysis and modal characterization.[2–4] The estimation of structural parameters via static responses (in terms of displacements[5–9] or strain[10,11]) is becoming more popular due to a growing interest in real-time continuous structural monitoring and improvements in distributed sensing systems.[12,13] Both dynamic and static assessment strategies aim to solve a parameter identification problem with signature analysis, pattern recognition, system identification and model updating being the main approaches used.[12,14] Ref. 15 outlines four levels of increasing knowledge about the condition of a damaged structure: (i) damage detection, (ii) damage localization (iii) damage severity/size quantification, and (iv) remaining life prediction. The first three steps concern the diagnosis phase, while the fourth step relates to prognosis. During the last decade, many efforts have extended damage diagnosis to the cases of multiple cracks[16–19] and delamination problems for composite structures;[13,20] other works have focused on strategies for the quantification of the uncertainties that affect the measurements and the different levels of knowledge about the damage.[21–23] In addition, investigations using neural networks and surrogate modeling techniques have been employed for rapid online structural assessment. A neural network approach interprets damage identification as a pattern recognition problem,[1,4,9,24–26] while surrogate modeling allows efficient representations of the behavior of damaged structures.[16–18,27]

A traditional approach to structural reliability assessment relies on first and second order reliability methods (FORM and SORM),[28] which approximate the limit state function around the most probable points of failure. FORM and SORM work well for smooth failure surfaces characterized by one design point, but become inefficient for more complex problems involving multiple design points. Monte Carlo sampling is a more general but more costly approach to identify the failure surface. Many samples are needed, especially to estimate low probability of failures (rare events), which becomes prohibitively expensive when considering detailed models of complex structures and/or nonlinear problems. This challenge has been addressed by using surrogate modeling techniques, such as polynomial regression, moving least-squares, Gaussian process based interpolations (kriging models), polynomial chaos expansions, radial basis functions and artificial neural networks, to approximate the limit state function.[29–34] In particular, Ref. 30 interprets the estimation of limit state functions as a classification problem and opens a spectrum of possible approaches based on statistical learning. It groups the methods in three main categories: (i) probabilistic methods (e.g., Bayesian approaches, Fisher discriminant, radial basis functions, probabilistic neural networks), (ii) distance based methods (e.g., nearest neighbor, classification trees, constrained topological mapping, self-organizing

maps), (iii) boundary methods (e.g., multi-layer perceptron, support vector machines). Several strategies for structural reliability analysis using support vector machines can be found in the literature.[33,35–39] More recent work combines reduced-order modeling (to compress the information about the state using a small number of characteristic features), clustering algorithms (to group different response behaviors), and neural network boundary methods (to obtain an identification of the limit state function) to address computational cost for dynamic nonlinear problems.[38,40]

In this paper we focus on the use of static strain data rather than dynamic structural responses. We propose a formulation that avoids solving an inverse for damage identification—we bypass identification of the damage itself, and instead identify directly an estimate of the damage impact on the structural capability. Our methodology is non-intrusive—that is, it applies generally to black-box simulation models—and it combines computational modeling outputs with real system data. Our specific problem setup is characterized by measured quantities of interest (strain data), a set of parameters that define the structural state (damage location and damage extent), and output quantities of interest that represent vehicle capability (failure indices). In the offline phase, we use high-fidelity finite element plate models and simulated data to obtain the surrogate models and to train a clustering algorithm. In the online phase, we use parametric proper orthogonal decomposition (POD)[41–45] to obtain a reduced-order model representing the structural response. This reduced-order model is combined with a self-organizing map (SOM),[46–49] which provides a rapid classification criterion to map measurements directly to capabilities.

Section II of this paper describes our surrogate modeling approach. In Section III we describe our specific problem setup, which considers structural assessment of a composite wing panel, and present results for our approach. Section IV discusses the results, and Section V presents concluding remarks.

II. Offline/online surrogate modeling approach

Our proposed approach comprises two phases: an offline phase and an online phase (Figure 2). The following subsections describe the steps and associated methods that characterize these two phases.

II.A. The offline phase

In the offline phase, high-fidelity models are used to obtain detailed information about the system and its possible responses for different conditions. The simulated information is then used to build a fast mapping via surrogate models. We consider two kinds of quantities of interest:

- The quantities that are measured during flight, referred to as the *measured quantities of interest*. We denote these by $\mathbf{q}_m(\mathbf{x}),\ m = 1, \ldots, M$, where $\mathbf{q}_m$ is the mth measured quantity of interest (generally a vector representing a discretized field quantity) and we consider M such quantities.
- The quantities employed in the decision process that give information about capability and performance constraints, referred to as the *capability quantities of interest*. We denote these by $\mathbf{s}_c(\mathbf{x}),\ c = 1, \ldots, C$, where $\mathbf{s}_c$ is the cth capability quantity of interest (generally a vector representing a discretized field quantity) and we consider C such quantities.

Both quantities of interest are functions of the vehicle state, $\mathbf{x}$. Two sets of reduced-order models (ROMs) are obtained by analyzing simulations of the quantities of interest (Figure 1). We define M ROMs that map vehicle state to measured quantities of interest, and a second group of C ROMs that map vehicle state into capability quantities of interest.

II.A.1. Parametric proper orthogonal decomposition for measurement and capability quantities of interest

We derive the two sets of ROMs using parametric proper orthogonal decomposition (POD). This requires sampling over different vehicle states, $\{\mathbf{x}^i\}_{i=1}^{n_\mathrm{s}}$, and computing the corresponding measurement vectors $\{\mathbf{q}_m^i\}_{i=1}^{n_\mathrm{s}},\ m = 1, \ldots, M$, and capability vectors $\{\mathbf{s}_c^i\}_{i=1}^{n_\mathrm{s}},\ c = 1, \ldots, C$. These sampled vectors are referred to as "snapshots", where the superscript i denotes the ith snapshot, $\mathbf{q}_m^i \equiv \mathbf{q}_m(\mathbf{x}^i)$, $\mathbf{s}_c^i \equiv \mathbf{s}_c(\mathbf{x}^i)$, and we compute n_s snapshots for each quantity of interest. Then for each quantity of interest we compute a POD basis in the standard way: we assemble the snapshot matrix, which has n_s columns containing the snapshots of that quantity of interest; we compute the singular value decomposition of the snapshot matrix; and we retain the dominant left singular vectors as the POD basis vectors, ranked in order of importance

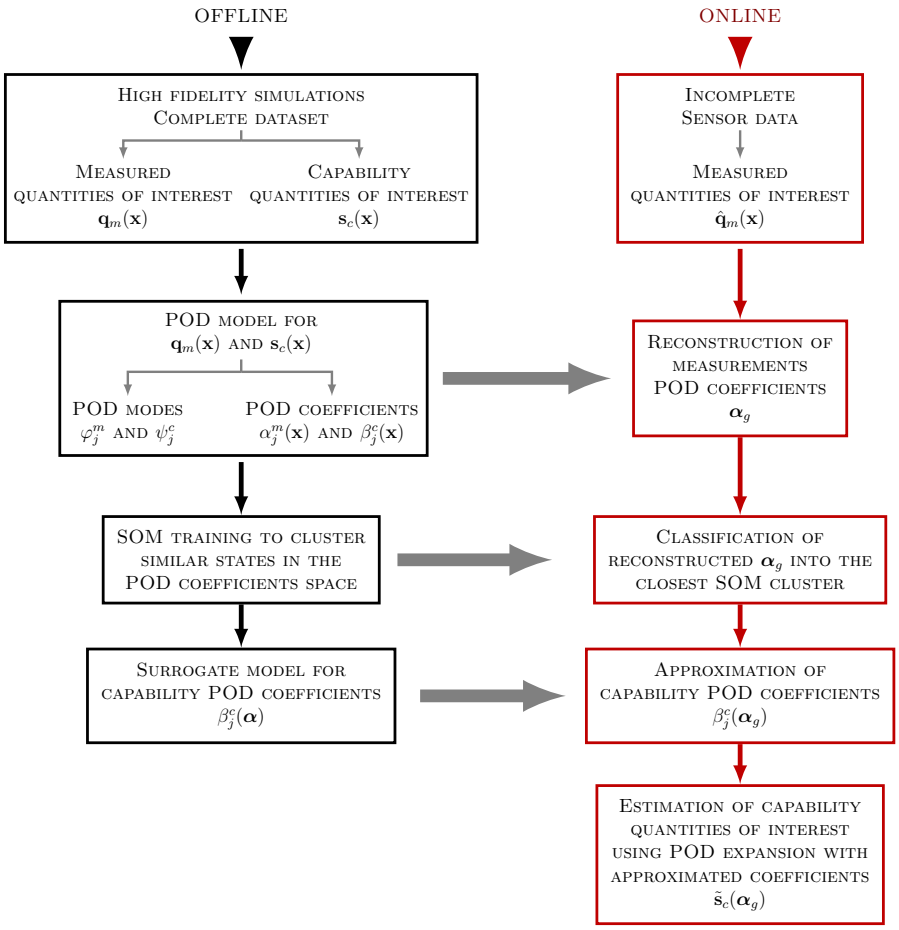


Figure 2. Diagram of the offline (black boxes) and online (red boxes) steps of our approach: POD models and clusters are obtained offline and employed online as indicated by the gray arrows.

by the magnitude of their corresponding singular value. To improve numerical conditioning and avoid the magnitude of the first singular value numerically dominating all others, we first compute the average quantity of interest over its snapshot set and subtract that mean from each snapshot. The POD was chosen because it has been shown to be effective in compressing high-dimensional information arising from discretized fields such as those in our problems of interest. For problems of the kind we consider here, the POD often leads to efficient yet accurate low-dimensional representations; however, our modeling and reconstruction approach is general and could be applied to other representations of the quantities of interest (e.g., other choices of the basis). The key is that the representation be low-dimensional, since online costs will scale with the number of coefficients in the approximations.

For each measurement quantity of interest, $\mathbf{q}_m$, we denote the POD basis by $\{\varphi_j^m\}_{j=1}^{n_s}$, where φ_j^m is the jth POD basis vector for $\mathbf{q}_m$. The corresponding POD eigenvalues are given by the squares of the singular values, and for $\mathbf{q}_m$ are denoted as $\{\lambda_j^m\}_{j=1}^{n_s}$. The importance of a given POD mode is often assessed by the normalized magnitude of its eigenvalue, that is by the quantity $\lambda_j^m/\bar{\lambda}^m$, where $\bar{\lambda}^m = \sum_{j=1}^{n_s} \lambda_j^m$. The ratio $\lambda_j^m/\bar{\lambda}^m$ represents the relative energy (measured in terms of L_2-norm) captured by the jth basis vector, thus the quantity $\sum_{i=1}^{j} \lambda_i^m/\bar{\lambda}^m$ is referred to as the cumulative energy recovered by the first j dominant modes. Similarly, for each capability quantity of interest, $\mathbf{s}_c$, the POD basis is denoted $\{\psi_j^c\}_{j=1}^{n_s}$, with corresponding POD eigenvalues $\{\mu_j^c\}_{j=1}^{n_s}$. We also define $\bar{\mu}^c = \sum_{j=1}^{n_s} \mu_j^c$, giving the relative importance of POD mode ψ_j^c as $\mu_j^c/\bar{\mu}^c$ and the related cumulative energy as $\sum_{i=1}^{j} \mu_i^c/\bar{\mu}^c$.

The surrogate model for each quantity of interest is defined using its POD basis. For the measurement

quantities of interest, the POD models are:

$$\mathbf{q}_m(\mathbf{x}) \approx \bar{\mathbf{q}}_m + \sum_{j=1}^{n_m} \alpha_j^m(\mathbf{x})\varphi_j^m, \quad m = 1, \ldots, M, \tag{1}$$

where $\bar{\mathbf{q}}_m$ is the average value of $\mathbf{q}_m$ over the snapshots $\{\mathbf{q}_m^i\}_{i=1}^{n_\mathrm{s}}$, $n_m \leq n_\mathrm{s}$ is the number of POD basis vectors retained in the approximation of $\mathbf{q}_m$, and $\alpha_j^m(\mathbf{x})$ is the POD modal coefficient for the jth POD basis vector φ_j^m. Similarly, for the capability quantities of interest, the POD model is:

$$\mathbf{s}_c(\mathbf{x}) \approx \bar{\mathbf{s}}_c + \sum_{j=1}^{\ell_c} \beta_j^c(\mathbf{x})\psi_j^c, \quad c = 1, ..., C, \tag{2}$$

where $\bar{\mathbf{s}}_c$ is the average value of $\mathbf{s}_c$ over the snapshots $\{\mathbf{s}_c^i\}_{i=1}^{n_\mathrm{s}}$, $\ell_c \leq n_\mathrm{s}$ is the number of POD basis vectors retained in the approximation of $\mathbf{s}_c$, and $\beta_j^c(\mathbf{x})$ is the POD modal coefficient for the jth POD basis vector ψ_j^c.

II.A.2. Clustering of similar states in the POD coefficients space using self-organizing maps

The next step in our approach is to create a model that maps the measurement POD coefficients $\alpha_j^m(\mathbf{x})$ to the capability POD coefficients $\beta_j^c(\mathbf{x})$. This mapping could be done in a variety of ways. Here, we use a neural network classifier combined with a polynomial mapping of the coefficients. In particular self-organizing maps (SOM) are chosen as the clustering algorithm.[46,48] To train the SOMs, we must define the training matrix, which collects all the training data. In our case, the training data are the POD modal coefficients corresponding to all the snapshots. That is, for each sampled $\mathbf{x}^i$, we have the corresponding coefficients $\alpha_j^m(\mathbf{x}^i)$, $j = 1, \ldots, n_m,\ m = 1, \ldots, M$, and $\beta_j^c(\mathbf{x}^i)$, $j = 1, \ldots, \ell_c,\ c = 1, \ldots, C$. We define $N_\alpha = \sum_{m=1}^{M} n_m$ to be the total number of retained measurement POD modes and, similarly, $N_\beta = \sum_{c=1}^{C} \ell_c$ to be the total number of retained capability POD modes; accordingly, $n_\mathrm{pod} = N_\alpha + N_\beta$ is the total number of POD modes over all quantities of interest. For each snapshot $\mathbf{x}^i$, we collect all its POD coefficients in the n_pod-dimensional vector $\boldsymbol{\tau}_i$. The training matrix $\mathbf{T}$ is then the $n_\mathrm{s} \times n_\mathrm{pod}$ matrix defined as

$$\mathbf{T} = [\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_{n_\mathrm{s}}]^\top. \tag{3}$$

The SOM is built from this set of training data. In the training phase, the SOM computes a set of n_w weight vectors, $\mathbf{w}_j,\ j = 1, \ldots, n_w$, each of dimension n_pod. These weight vectors (the neurons) are computed using an iterative unsupervised learning process that clusters similar training vectors.[50–52] Each cluster is characterized with a representative weight vector (neuron) that can be interpreted as the centroid/prototype of the cluster, although not all the n_w weight vectors of the SOM network end up leading a cluster ($N_c \leq n_w$). For further details about SOMs we refer the reader to Ref. 46–49.

During the SOM training phase, a given training vector $\boldsymbol{\tau}_i$ is associated with its closest cluster, found by minimizing a distance metric. Typically, the cluster k chosen for $\boldsymbol{\tau}_i$ is found by

$$k = \underset{j \in 1, \ldots, n_w}{\arg\min} \{\|\boldsymbol{\tau}_i - \mathbf{w}_j\|\}. \tag{4}$$

In our particular setting, we modify the distance metric to have better coverage along the most important directions (i.e., those directions corresponding to the dominant POD modes for each quantity of interest). To achieve this, we define a scaled L_2-norm using the POD eigenvalues:

$$\|\boldsymbol{\tau}_i - \mathbf{w}_k\|_{\boldsymbol{\Lambda}} = \sqrt{(\boldsymbol{\tau}_i - \mathbf{w}_k)^\top \boldsymbol{\Lambda} (\boldsymbol{\tau}_i - \mathbf{w}_k)}, \tag{5}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{n_\mathrm{pod} \times n_\mathrm{pod}}$ is a diagonal matrix whose entries are the normalized POD eigenvalues:

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\left\{ \frac{\{\lambda_j^m\}_{j=1,\ldots,n_m}}{\bar{\lambda}^m} \right\}_{m=1,\ldots,M}, \left\{ \frac{\{\mu_j^c\}_{j=1,\ldots,\ell_c}}{\bar{\mu}^c} \right\}_{c=1,\ldots,C} \right). \tag{6}$$

The distance defined by (5) is the criterion by which SOM groups the input dataset in $\mathbf{T}$ into N_c clusters.

II.A.3. Local response surface models for capability POD coefficients $\beta_j^c(\boldsymbol{\alpha})$

In the last step of the offline phase, each cluster is characterized by models that express each retained coefficient β_j^c as a function of the measurement POD coefficients. For each cluster, we build a local polynomial response surface model (noting again that other representations could be employed) for each capability POD coefficient $\beta_j^c(\mathbf{x})$ as a function of the measurement POD coefficients $\boldsymbol{\alpha} = \{\alpha_j^m(\mathbf{x})\}_{\substack{j=1,\ldots,n_m \\ m=1,\ldots,M}}$. This then completes the mapping from measurement quantities of interest to capability quantities of interest.

II.B. The online phase

The online phase is characterized by the need to process information rapidly in order to make a decision. During this phase, sensors provide data on the measurement quantities of interest. We use the gappy POD method[53,54] to rapidly reconstruct the measurement POD coefficients α_j^m given the sensor measurement data $\hat{\mathbf{q}}_m$.

II.B.1. Reconstruction of measurement POD coefficients using gappy POD

Consider reconstruction of the mth measured quantity of interest $\mathbf{q}_m$. Given a partial measurement $\hat{\mathbf{q}}_m$, which contains only those elements of $\mathbf{q}_m$ that correspond to sensor locations, we wish to reconstruct a full estimate $\tilde{\mathbf{q}}_m$. Using the POD model, $\tilde{\mathbf{q}}_m$ can be expressed as follows:

$$\tilde{\mathbf{q}}_m = \bar{\mathbf{q}}_m + \sum_{j=1}^{n_m} \alpha_j^m \varphi_j^m. \tag{7}$$

The gappy POD method estimates the POD coefficients α_j^m by minimizing the so-called "gappy norm" of the error e_g^m between the reconstructed $\tilde{\mathbf{q}}_m$ and the sensed data $\hat{\mathbf{q}}_m$:

$$e_g^m = \|\hat{\mathbf{q}}_m - \tilde{\mathbf{q}}_m\|_g^2, \tag{8}$$

where g indicates the gappy norm which computes the L_2-norm only over those elements of the vectors corresponding to available sensed data. The POD coefficients $\boldsymbol{\alpha}_g^m = [\alpha_1^m, \alpha_2^m, \ldots, \alpha_{n_m}^m]^\top$ that minimize (8) can be shown to satisfy the linear system of equations:

$$\mathbf{G}^m \boldsymbol{\alpha}_g^m = \mathbf{f}^m, \tag{9}$$

where the ijth entry of the matrix $\mathbf{G}^m$ is $G_{ij}^m = \left(\varphi_i^m, \varphi_j^m\right)_g$, the ith entry of the vector $\mathbf{f}^m$ is $f_i^m = \left(\hat{\mathbf{q}}_m, \varphi_i^m\right)_g$, and $(\cdot,\cdot)_g$ denotes the gappy inner product which again considers only those elements in the vectors that correspond to the available sensed data. The POD coefficients α_j^m are reconstructed for the M measured quantities of interest and collected in $\boldsymbol{\alpha}_g = \left[(\boldsymbol{\alpha}_g^1)^\top, ..., (\boldsymbol{\alpha}_g^M)^\top\right]$.

II.B.2. Classification of the reconstructed measurement POD coefficients $\boldsymbol{\alpha}_g$ into the closest SOM cluster

The estimated POD coefficients $\boldsymbol{\alpha}_g$ are then used to assign the sensed condition to one of the N_c SOM clusters. To perform this classification, the first step is to choose the closest cluster, using the distance metric weighted with the POD eigenvalues, as defined in (5), such that:

$$k = \underset{j \in 1,\ldots,N_c}{\arg\min} \{\|\boldsymbol{\tau}_i - \mathbf{w}_j\|_{\Lambda_\alpha}\}, \tag{10}$$

where $\|\cdot\|_{\Lambda_\alpha}$ indicates that the weighted norm is computed only over the coefficients $\boldsymbol{\alpha}_g$, that is, over the first N_α elements of the vectors. More specifically, recall that $\boldsymbol{\tau}_i = [\boldsymbol{\alpha}_g, \beta_1^1, \beta_2^1, ..., \beta_{\ell_1}^1, \beta_1^2, ..., \beta_{\ell_C}^C]$, and at this stage of the online mapping we know only the values of the elements that correspond to the measurement coefficients reconstructed via gappy POD and collected in $\boldsymbol{\alpha}_g$. The last N_β elements of $\boldsymbol{\tau}_i$ are still unknown (and will be estimated in the next step); thus, they are not used in computing the distance in (10).

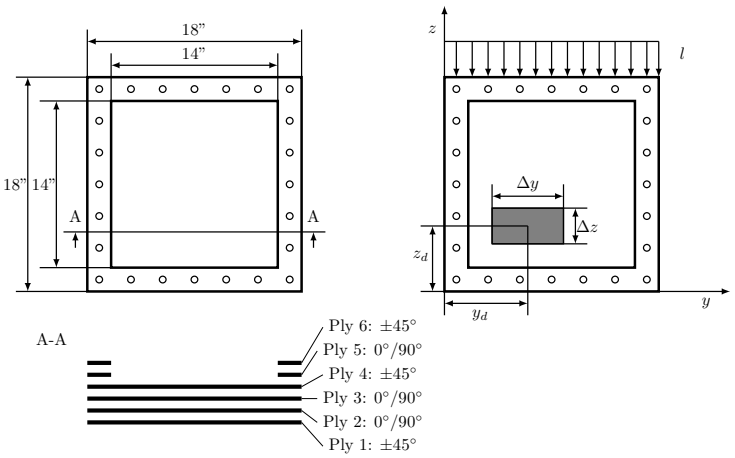


Figure 3. Panel layout and layer sequence. Panel state variables: damage location, damage size, and load definition.

II.B.3. Approximation of the capability POD coefficients $\beta_j^c(\boldsymbol{\alpha}_g)$ using local response surface models

In the offline phase, we characterized each cluster by a set of polynomial response surface models that express each retained coefficient β_j^c as function of the N_α measurement POD coefficients. Therefore, once $\boldsymbol{\alpha}_g$ is obtained and the closest cluster k is identified, using these response surface models the unknown capability coefficients are estimated as polynomial functions of the reconstructed measurement coefficients such that $\beta_j^c(\boldsymbol{\alpha}_g) \approx \beta_j^c(\mathbf{x})$.

II.B.4. Estimation of capability quantity of interest $\tilde{\mathbf{s}}_c(\boldsymbol{\alpha}_g)$

In the last step of the online phase, given the approximated β_j^c coefficients, the capability quantities of interest are estimated as a combination of the corresponding POD modes using the expansion (2):

$$\tilde{\mathbf{s}}_c(\boldsymbol{\alpha}_g) = \bar{\mathbf{s}}_c + \sum_{j=1}^{\ell_c} \beta_j^c(\boldsymbol{\alpha}_g)\psi_j^c, \quad c = 1, ..., C. \tag{11}$$

The entire process described in Sections II.B.1 to II.B.4 exploits the map from measurements directly to capability quantities of interest, thus avoiding the need to infer the actual state of the system. The process is efficient because it focuses on reconstruction of a small number of POD coefficients α_j^m and β_j^c. Figure 2 summarizes the offline and online steps of the overall approach.

III. Application

The methodology described in Section II is applied to a structural capability assessment problem. We first describe the problem setup, then apply our approach. We provide a detailed assessment of errors and present computational runtime results.

III.A. Problem setup and data collection

Our specific test problem considers an 18 × 18 square-inch composite wing panel. The panel is made up of four plain weave carbon fiber plies with the symmetric stacking sequence $[45°, 0°, 0°, 45°]$ such that the whole panel behaves in a quasi-isotropic manner. Four clamped edges define boundary conditions that simulate the presence of fastening bolts along the panel perimeter. The border area that hosts the holes is reinforced with two additional woven plies with orientation 0° and 45°. Figure 3 shows the panel layout and a sketch of the section with the layer sequence.

A high-fidelity finite element method (FEM) model of the panel is employed to simulate and analyze the panel behavior under different loading and damage scenarios: the specific FEM solver we use is NASTRAN. The model has $E = 3921$ two-dimensional laminate plate elements: mostly quadrilateral elements with four

nodes and a few triangular elements with three nodes for transitional locations in proximity of the bolt holes, characterized by the properties of the carbon fiber composite layup. The loading is specified, where the loads are defined by a prescribed aircraft maneuver. For the simulations presented in this paper, the loading conditions (indicated with l in Figure 3) are fixed and represent a uni-axial compression load applied in terms of imposed nodal displacements of -0.01 in. The panel has four clamped edges and fastening bolts placed in the holes along the perimeters. The presence of damage is simulated by weakening the stiffness properties of the FEM elements that belong to the prescribed damaged area. This model is used to explore the state space and generate the snapshot sets by evaluating, for each condition, the measured quantities of interest and the capability quantities of interest.

This problem setup provides a representative test case that relates to our motivating application of a wing panel on a UAV. This particular choice of geometry, boundary conditions and material properties is also motivated by ongoing experimental efforts that will provide experimental data to further investigate our approach. However, it is also important to note that the methodology proposed in this paper is general; our method is directly applicable to other problem setups, including those that might stretch beyond structural analysis. In particular, one attractive feature of our approach is that the simulation data used to build the POD models and the coefficient mappings can be obtained from black-box simulation models, with no need to access the details of the governing equations or the specific numerical simulation implementation.

The complete structural state is defined by three classes of parameters: (1) the panel layout and boundary conditions, (2) the maneuver and related load cases, and (3) the characteristics of the damage. The first two sets of parameters indicate the panel features and loading (l); in our analysis, these are considered fixed as shown in Figure 3. The damage characteristics then define our state variable vector $\mathbf{x}$. Referring to Figure 3, the state vector $\mathbf{x}$ is given by five components prescribing damage size along the y-direction (Δy), damage size along the z-direction (Δz), damage centroid location along the y-direction (y_d), damage centroid location along the z-direction (z_d), and damage depth in terms of number of remaining undamaged plies (d_d):

$$\mathbf{x} = [\Delta y, \Delta z, y_d, z_d, d_d]^\top. \tag{12}$$

The measured quantities of interest are the components of strain. We consider a specific case where the strain gauges are placed on ply 4 and distributed so that three components of strain can be measured: the normal components along the main orthotropic axes of the ply ($\varepsilon_{\mathrm{n}_1}$ and $\varepsilon_{\mathrm{n}_2}$), and the shear component ($\varepsilon_{\mathrm{s}_{12}}$) on the ply plane. Our three measurement quantity of interest vectors, $\mathbf{q}_1(\mathbf{x}), \mathbf{q}_2(\mathbf{x}), \mathbf{q}_3(\mathbf{x})$, contain the finite element representations of the discretized strain fields $\varepsilon_{\mathrm{n}_1}(\mathbf{x})$, $\varepsilon_{\mathrm{n}_2}(\mathbf{x})$ and $\varepsilon_{\mathrm{s}_{12}}(\mathbf{x})$, respectively. Thus, each $\mathbf{q}_m$ vector has dimension equal to the number of elements in the FEM mesh.

The capability quantity of interest is the failure index (FI), which is an indicator of the structural condition that is translated into a scaling factor for maneuver parameters. FI is defined as the ratio between the experienced stress σ and the maximum allowable stress $\sigma_{\max}$ (i.e., the compression/tension/shear strengths that are characteristic properties of the material) for each element in the FEM mesh. For each ply, five failure modes and their related maximum allowable stresses are considered, namely compression along the main orthotropic axes of the ply ($\mathrm{C}_{\mathrm{n}_1}$ and $\mathrm{C}_{\mathrm{n}_2}$), tension along the main orthotropic axes of the ply ($\mathrm{T}_{\mathrm{n}_1}$ and $\mathrm{T}_{\mathrm{n}_2}$), and shear on ply plane ($\mathrm{S}_{\mathrm{s}_{12}}$). The application of these criteria to the stress field, obtained by running the FEM model of the panel, provides a set of $FI(\mathbf{x})$ values for each discrete plate element ($e = 1, ..., E$), for each ply (ply $= 1, ..., 6$), and for each failure mode (mode $= \mathrm{C}_{\mathrm{n}_1}, \mathrm{C}_{\mathrm{n}_2}, \mathrm{T}_{\mathrm{n}_1}, \mathrm{T}_{\mathrm{n}_2}, \mathrm{S}_{\mathrm{s}_{12}}$) considered:

$$FI^{\mathrm{mode}}(e, \mathrm{ply}, \mathbf{x}) = \frac{\sigma^{\mathrm{mode}}(e, \mathrm{ply}, \mathbf{x})}{\sigma^{\mathrm{mode}}_{\max}}. \tag{13}$$

We condense the resulting large set of data to define a vector $\mathbf{FI}$ that contains for each element the maximum failure index computed over the five failure modes and over all plies:

$$\mathbf{FI}(\mathbf{x}) = \left\{ \max_{\mathrm{ply},\ \mathrm{mode}} \left(FI^{\mathrm{mode}}(e, \mathrm{ply}, \mathbf{x}) \right) \right\}_{e=1}^{E}. \tag{14}$$

The vector $\mathbf{FI}(\mathbf{x})$ then defines our single capability quantity of interest, $\mathbf{s}_1(\mathbf{x})$, which has dimension equal to the number of elements (E) in the FEM mesh.

Table 1 summarizes the state variables, measured quantities of interest, and capability quantities of interest for our particular problem setup. In order to apply and test our proposed offline/online approach, we use the FEM model to generate samples of the quantities of interest: three measurement quantities $\varepsilon_{\mathrm{n}_1}$,

Table 1. Problem setup: state variables, measurement quantities of interest, and capability quantities of interest.

State variable $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]^\top$		**notation**	**unit**	**dimension**
x_1	damage centroid coordinate along y-direction	y_d	[in]	1
x_2	damage centroid coordinate along z-direction	z_d	[in]	1
x_3	damage size along y-direction	Δy	[in]	1
x_4	damage size along z-direction	Δz	[in]	1
x_5	damage depth as undamaged remaining plies	d_d	[plies]	1

Measured quantities of interest $\{\mathbf{q}_m\}_{m=1}^M \quad M = 3$		**notation**	**unit**	**dimension**
$\mathbf{q}_1$	normal strain component along first orthotropic axis	$\varepsilon_{\mathrm{n}_1}$	$[\mu\varepsilon]$	3921
$\mathbf{q}_2$	normal strain component along second orthotropic axis	$\varepsilon_{\mathrm{n}_2}$	$[\mu\varepsilon]$	3921
$\mathbf{q}_3$	shear strain component on ply plane	$\varepsilon_{\mathrm{s}_{12}}$	$[\mu\varepsilon]$	3921

Capability quantities of interest $\{\mathbf{s}_c\}_{c=1}^C \quad C = 1$		**notation**	**unit**	**dimension**
$\mathbf{s}_1$	failure index	**FI**	[-]	3921

$\varepsilon_{\mathrm{n}_2}$ and $\varepsilon_{\mathrm{s}_{12}}$, and one capability quantity **FI**. Two Latin hypercube explorations of the state-space over the ranges defined in Table 2 yield reference data for different cases of damage size and damage locations. We use the Latin hypercube sampling implemented in the MATLAB® function *lhsdesign*. In particular, we choose the criterion *correlation* such that the sampling is progressively improved with an iterative reduction of samples correlation, leading to a Latin hypercube design with improved orthogonality properties.[55–57] We generate a first set of $n_{\mathrm{s}} = 3000$ snapshots, which constitutes the evaluation set used to obtain our surrogate models during the offline phase, and a second validation set of $n_{\mathrm{sv}} = 500$ snapshots, which are used to test the performance of the online mapping strategy.

Table 2. State variables and related bounds of variation for snapshot generation. Evaluation set: $n_{\mathrm{s}} = 3000$ snapshots, sampled using a Latin hypercube. Validation set: 500 snapshots, sampled using a Latin hypercube.

state variable	**min value**	**max value**	**units**
Δy	2	8	[in]
Δz	2	8	[in]
y_d	4	14	[in]
z_d	4	14	[in]
d_d	1	3	[plies]

III.B. Application of the surrogate modeling approach

According to the procedure described in Section II.A.1, ROMs are obtained for each quantity of interest, $\mathbf{q}_1(\mathbf{x}) = \varepsilon_{\mathrm{n}_1}(\mathbf{x})$, $\mathbf{q}_2(\mathbf{x}) = \varepsilon_{\mathrm{n}_2}(\mathbf{x})$, $\mathbf{q}_3(\mathbf{x}) = \varepsilon_{\mathrm{s}_{12}}(\mathbf{x})$, and $\mathbf{s}_1(\mathbf{x}) = \mathbf{FI}(\mathbf{x})$. Figure 4 shows the POD eigenvalue spectra for all four quantities of interest, while the related cumulative energies (defined in Section II.A.1) are plotted in Figure 5. The circles in Figure 5 indicate the number of modes retained for each POD approximation to recover the 95% of cumulative energy, with the specific numerical results quoted in each figure. Note that, as discussed below, we can employ fewer POD modes than those indicated in Figure 5 in the final mapping to estimate the capability with $\beta_j^c(\boldsymbol{\alpha})$.

Figures 6, 7 and 8 compare the FEM and the POD approximations of the three strain components for a case where the panel has a 7.85×7.91 square-inch damaged region located at $(y_d = 5.47, z_d = 7.15)$ that involves plies 4 to 6 (i.e., $\mathbf{x}^{\mathrm{test}} = [7.85, 7.91, 5.47, 7.15, 3]$). For the same case, Figure 9 shows the reference FEM estimate of **FI** and Figure 10 plots its representation in the POD basis (i.e., the approximation using the actual POD coefficients, but not yet using any online reconstruction). Figures 6–8 demonstrate that

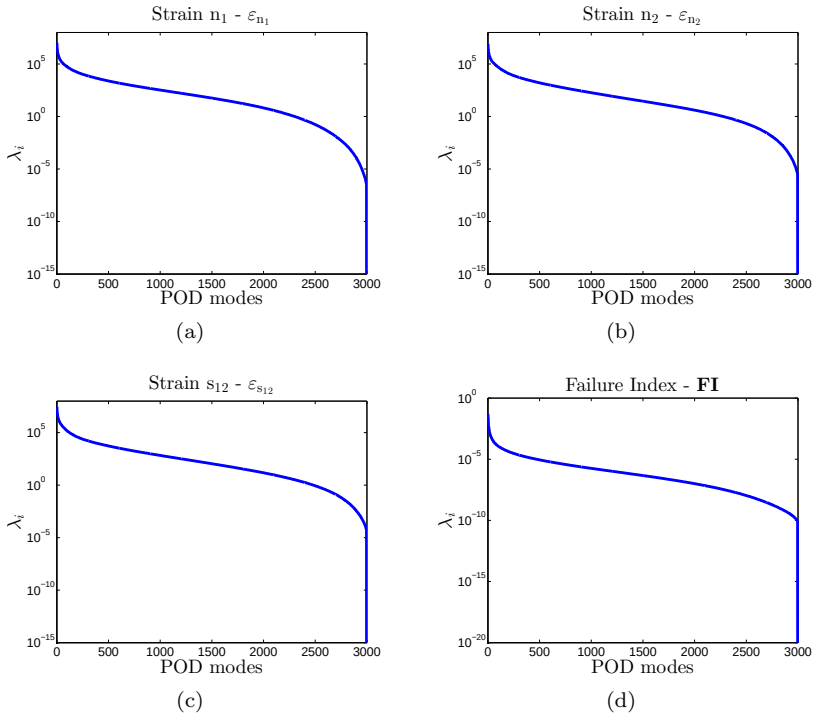


Figure 4. POD eigenvalues for the modes computed from $n_s = 3000$ snapshots for each quantity of interest.

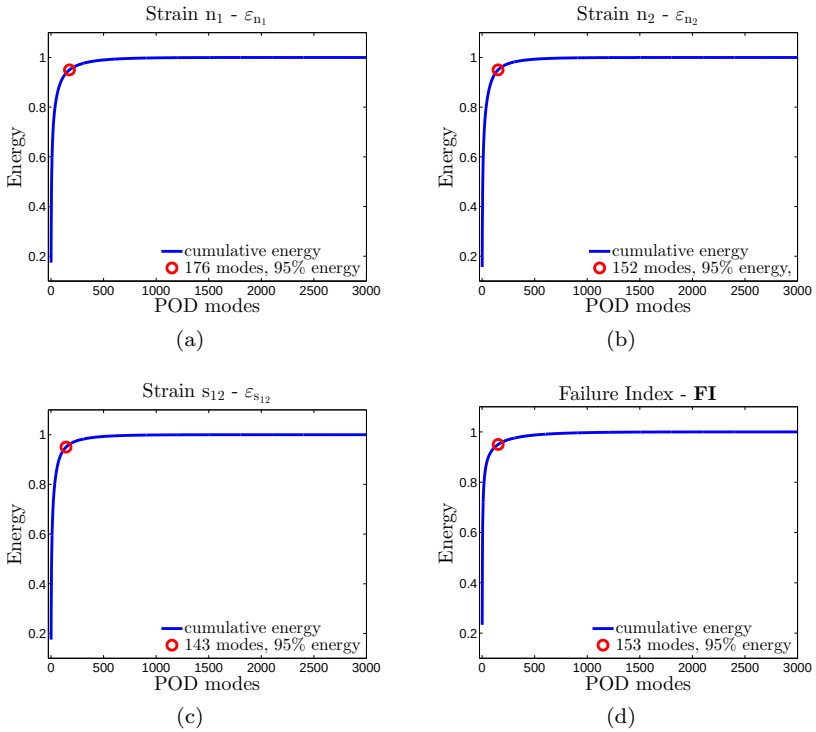


Figure 5. Cumulative energy, $\sum_{j=1}^{i} \lambda_j^m / \bar{\lambda}^m$ and $\sum_{j=1}^{i} \mu_j^c / \bar{\mu}^c$, for each POD mode i computed from the $n_s = 3000$ snapshots for each quantity of interest. Circles denote the number of modes retained for each POD approximation, recovering the 95% of the cumulative energy.

the POD models do a reasonable job of approximating the strain fields, although they do not represent accurately the local fields around the damaged region. However, our goal is not to recover these fields fully, but rather to use sparse sensor measurements of strain to infer the coefficients of the failure index POD model, and ultimately to assess the panel's structural integrity.

We next assess the error induced by approximating the **FI** POD coefficients as function of the strain POD coefficients. For this step, we find that we can use many fewer POD modes to create an acceptable mapping than are needed for full reconstruction of the field. Thereafter, we set a filter to down-select the modes to be retained for the online overall mapping: for each cluster, we preserve only the POD coefficients that achieved a reconstruction percentage error less than 30% on the evaluation set. This error level is set according to the evaluation data during the offline training; although it seems quite a high value by itself, we will see in Section III.C that it leads to a final mapping characterized by small errors. In the particular test case under consideration, this criterion translates into including in the mapping only the dominant seven POD modes for the capability quantity of interest ($N_\beta = 7$). However, the number of POD modes N_β retained for the capability estimate changes for different cases, depending on the cluster into which they are classified.

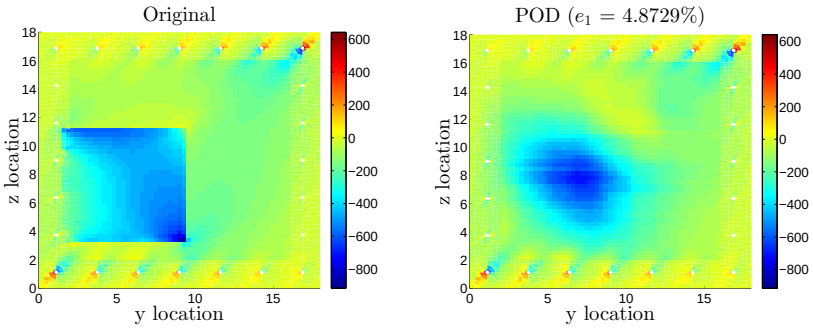
The next step is to use the response surface models that characterize the closest cluster to estimate the unknown capability POD coefficients. In this example, we use linear response surface models, so that each capability POD coefficient is represented as a linear function of the retained measurement POD coefficients. In this example, we obtain appropriate models for $\beta_j^c = \beta_j^c(\boldsymbol{\alpha})$ in each cluster (that is, on each subdomain in the POD coefficient space), using local response surface models that depend only on six measurement POD coefficients, the two dominant POD modes for each measurement quantity of interest:

$$\beta_j^1(\boldsymbol{\alpha}) = b_{0j}^1 + b_{1j}^1\alpha_1^1 + b_{2j}^1\alpha_2^1 + b_{3j}^1\alpha_1^2 + b_{4j}^1\alpha_2^2 + b_{5j}^1\alpha_1^3 + b_{6j}^1\alpha_2^3 \qquad j = 1, ..., N_\beta. \tag{15}$$

Figure 11 shows the resulting approximation of the **FI** distribution given by the POD with the SOM mapping. Section IV) provides more discussion on the tradeoffs of selecting the number of POD coefficients to include in the mapping.

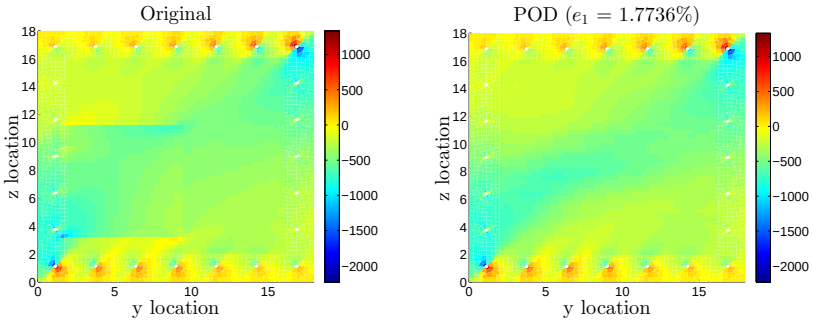
Finally, Figure 12 shows the full online approach (Section II.B): the coefficients of the measurement POD approximations are reconstructed from sparse strain sensor data using the gappy POD, and then the capability POD coefficients are estimated from the SOM mapping. For all figures, the test case is again $\mathbf{x}^{\text{test}}$ and the gappy POD estimate is obtained from incomplete distributions of strain components missing 50% information, that is, we consider the case where we have online sensor measurements at half the FEM grid points.

To summarize the results, Figure 9 shows the original FEM estimate of the **FI** over the panel and Figure 12 shows the final result using our approach described in Section II. Figures 10 and 11 demonstrate the intermediate cumulative errors due to the POD approximation and the SOM mapping. Section III.C provides a quantitative assessment of the method's performance over the full validation set, giving the reader a better idea of how the method performs for difference damage cases. For completeness, we also include field plots for two additional test cases within the validation set in the appendix. Figures 16 and 18 in the appendix show the original data, while Figures 17 and 19 display the final approximations provided by the complete online rapid mapping procedure. Overall, as will be seen quantitatively in the next section, the reconstructions of the failure indices over a large range of test cases are sufficiently accurate to provide a first-cut assessment for online decision-making.



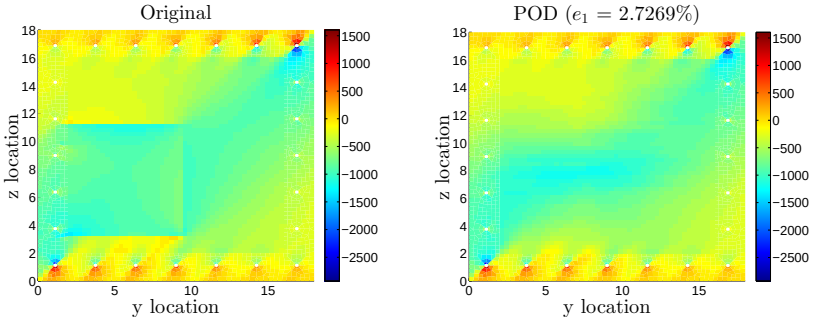
(a) Ply 4 - Original FEM data (b) Ply 4 - POD approximation

Figure 6. Strain ε_{n_1} (microstrain $[\mu\varepsilon]$) for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: comparison between FEM reference data $\mathbf{q}_1(\mathbf{x}^{\text{test}}) = \varepsilon_{n_1}(\mathbf{x}^{\text{test}})$ and POD approximation $\tilde{\mathbf{q}}_1(\mathbf{x}^{\text{test}}) = \bar{\mathbf{q}}_1 + \sum_{j=1}^{n_1} \alpha_j^m(\mathbf{x})\varphi_j^m$, $n_1 = 176$. e_1 is the normalized root mean square error defined in Eq. (16).



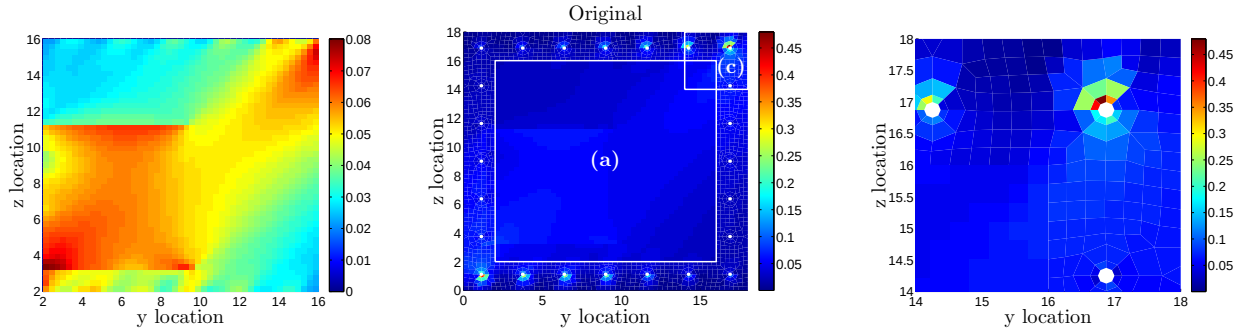
(a) Ply 4 - Original FEM data (b) Ply 4 - POD approximation

Figure 7. Strain ε_{n_2} (microstrain $[\mu\varepsilon]$) for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: comparison between FEM reference data $\mathbf{q}_2(\mathbf{x}^{\text{test}}) = \varepsilon_{n_2}(\mathbf{x}^{\text{test}})$ and POD approximation $\tilde{\mathbf{q}}_2(\mathbf{x}^{\text{test}}) = \bar{\mathbf{q}}_2 + \sum_{j=1}^{n_2} \alpha_j^m(\mathbf{x})\varphi_j^m$, $n_2 = 152$. e_1 is the normalized root mean square error defined in Eq. (16).



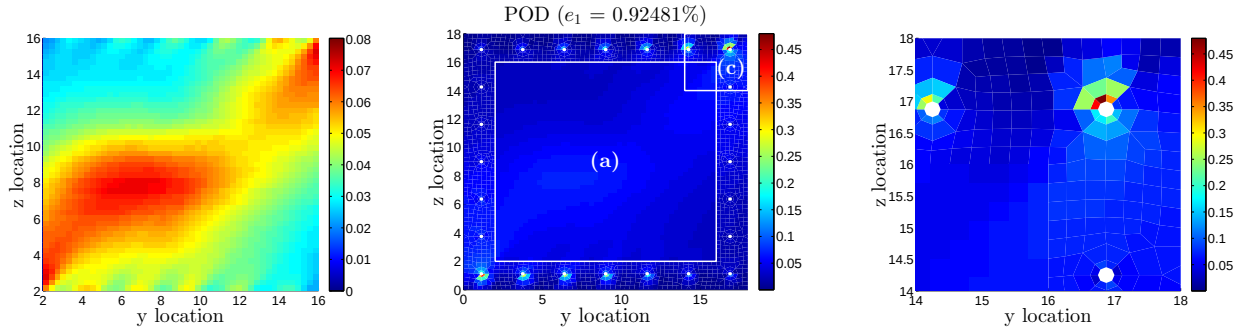
(a) Ply 4 - Original FEM data (b) Ply 4 - POD approximation

Figure 8. Strain $\varepsilon_{s_{12}}$ (microstrain $[\mu\varepsilon]$) for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: comparison between original FEM data $\mathbf{q}_3(\mathbf{x}^{\text{test}}) = \varepsilon_{s_{12}}(\mathbf{x}^{\text{test}})$ and POD approximation $\tilde{\mathbf{q}}_3(\mathbf{x}^{\text{test}}) = \bar{\mathbf{q}}_3 + \sum_{j=1}^{n_3} \alpha_j^m(\mathbf{x})\varphi_j^m$, $n_3 = 143$. e_1 is the normalized root mean square error defined in Eq. (16).



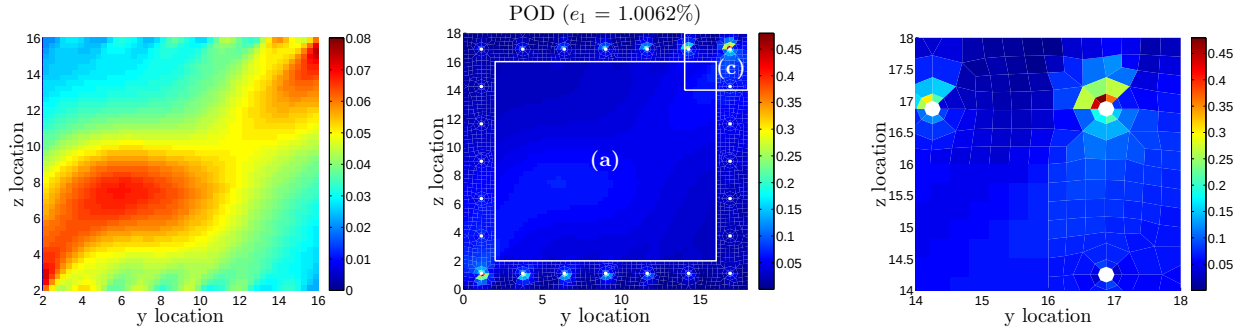
(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 9. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: original FEM data $\mathbf{s}_1(\mathbf{x}^{\text{test}}) = \mathbf{FI}(\mathbf{x}^{\text{test}})$.



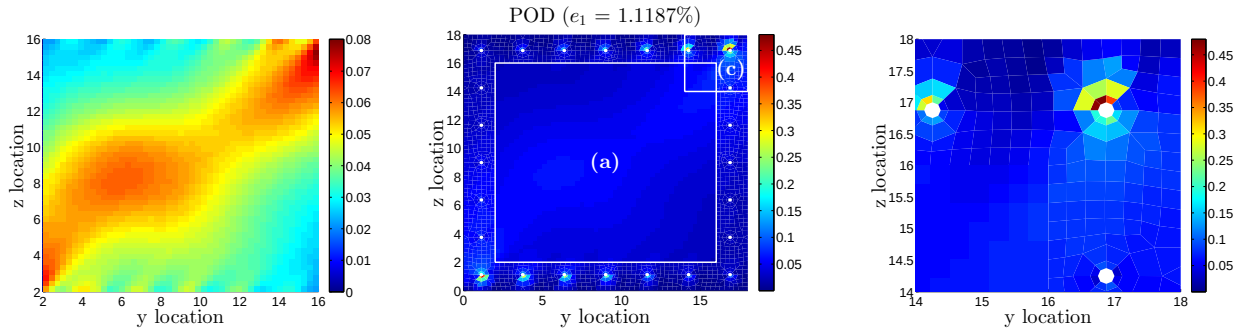
(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 10. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: approximation with 153 POD modes, computed with direct projection from the FEM solution (no reconstruction), $\tilde{\mathbf{s}}_1(\mathbf{x}^{\text{test}}) = \bar{\mathbf{s}}_1 + \sum_{j=1}^{\ell_1} \beta_j^c(\mathbf{x})\psi_j^c$, $\ell_1 = 153$. e_1 is the normalized root mean square error defined in Eq. (16).



(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 11. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: approximation using POD coefficients computed from SOM mapping, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha}) = \bar{\mathbf{s}}_1 + \sum_{j=1}^{N_\beta} \beta_j^c(\boldsymbol{\alpha})\psi_j^c$, $N_\beta = 7$. e_1 is the normalized root mean square error defined in Eq. (16).



(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 12. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.85, 7.91, 5.47, 7.15, 3]$: approximation using POD coefficients reconstructed from sensor data using gappy POD and SOM mapping, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha}_g) = \bar{\mathbf{s}}_1 + \sum_{j=1}^{N_\beta} \beta_j^c(\boldsymbol{\alpha}_g)\psi_j^c$, $N_\beta = 7$. e_1 is the normalized root mean square error defined in Eq. (16).

III.C. Error assessment

An analysis of the kind reported in Section III.B is replicated for each of the $n_{\text{sv}} = 500$ snapshots in the validation set. We define the normalized root mean square error (e_1) between an approximated quantity of interest $\tilde{\mathbf{q}}$ and its corresponding "truth" solution $\mathbf{q}$ as

$$e_1 = \frac{\|\tilde{\mathbf{q}} - \mathbf{q}\|_2}{\sqrt{E}(q_{\max} - q_{\min})} \times 100\%. \tag{16}$$

In this expression, the generic vector $\mathbf{q}$ represents the original FEM snapshot corresponding to any one of the measured or capability quantities of interest (i.e., the three strain components ε_{n_1}, ε_{n_2}, $\varepsilon_{\text{s}_{12}}$ or the failure index **FI**). The quantities $q_{\max}$ and $q_{\min}$ are, for each snapshot $\mathbf{q}$, the maximum and minimum values among its E elements, respectively. The generic vector $\tilde{\mathbf{q}}$ represents our approximation of $\mathbf{q}$. This approximation could be achieved in a number of different ways: it could be the POD approximation of ε_{n_1}, ε_{n_2} and $\varepsilon_{\text{s}_{12}}$ for the measured quantities of interest, or it could be the different estimates of the capability quantity of interest **FI** given by POD, POD+SOM or POD+gappy+SOM. Its specific meaning will be clear when we report numerical results. All approximations are compared to the original values provided by the FEM simulations and normalized with the quantity $\sqrt{E}(q_{\max} - q_{\min})$.

Figure 13 plots histograms of the normalized root mean square error e_1 for the entire validation set ($n_{\text{sv}} = 500$ cases). Table 3 shows the minimum, maximum and mean values of e_1 for each of these cases, where $\bar{e}_1$ denotes the mean value of e_1 over the validation set.

Next, we assess our ability to estimate the maximum value of **FI** experienced by the structure over all elements, which we denote as $FI_{\max}$:

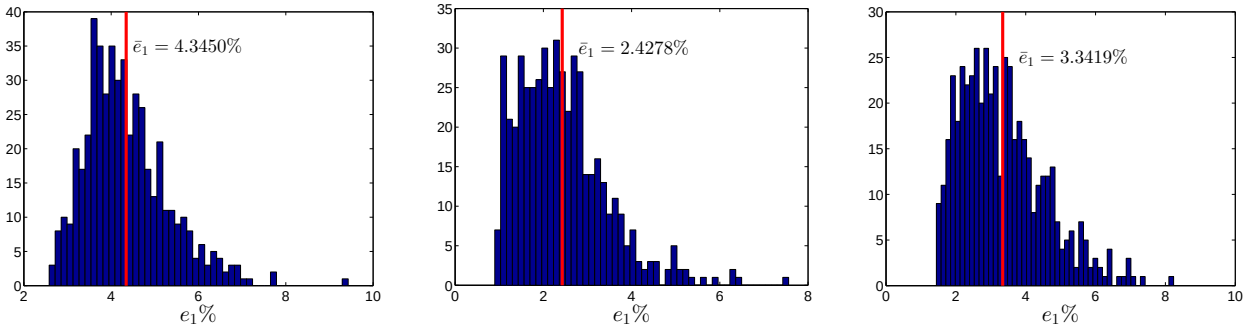
$$FI_{\max} = \max_e\{\mathbf{FI}\} \qquad e = 1, ..., E. \tag{17}$$

For this specific quantity we consider the normalized root mean square error e_2 evaluated over the validation set and defined as follows:

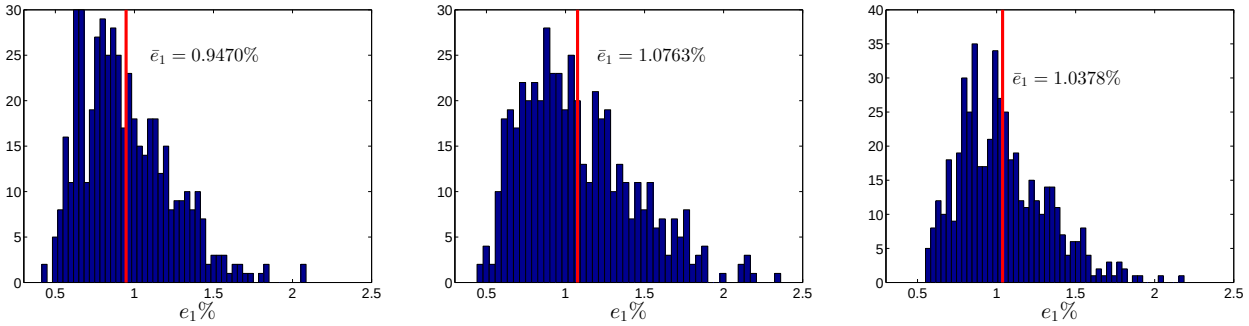
$$e_2 = \frac{\|\tilde{\mathbf{FI}}_{\max} - \mathbf{FI}_{\max}\|_2}{\sqrt{n_{\text{sv}}}\,((FI_{\max})_{\max} - (FI_{\max})_{\min})} \times 100\%. \tag{18}$$

Table 3. Normalized root mean square errors (e_1): mean ($\bar{e}_1$), maximum, and minimum values computed over the 500 snapshots in the validation set for various levels of approximation of the POD coefficients.

	ε_{n_1} POD	ε_{n_2} POD	$\varepsilon_{\text{s}_{12}}$ POD	**FI** POD	**FI** POD+SOM	**FI** POD+gappy+SOM
$\bar{e}_1$ (%)	4.3450	2.4278	3.3419	0.9470	1.0763	1.0378
maximum e_1 (%)	9.4305	7.5604	8.2350	2.0857	2.3610	2.1860
minimum e_1 (%)	2.5804	0.8986	1.4425	0.4144	0.4401	0.5506



(a) POD representation of ε_{n_1}, $\tilde{\mathbf{q}}_1(\mathbf{x})$ (b) POD representation of ε_{n_2}, $\tilde{\mathbf{q}}_2(\mathbf{x})$ (c) POD representation of $\varepsilon_{s_{12}}$, $\tilde{\mathbf{q}}_3(\mathbf{x})$



(d) POD representation of **FI**, $\tilde{\mathbf{s}}_1(\mathbf{x})$ (e) SOM mapping to POD representation of **FI**, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha})$ (f) Gappy POD reconstruction plus SOM mapping to POD representation of **FI**, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha}_g)$

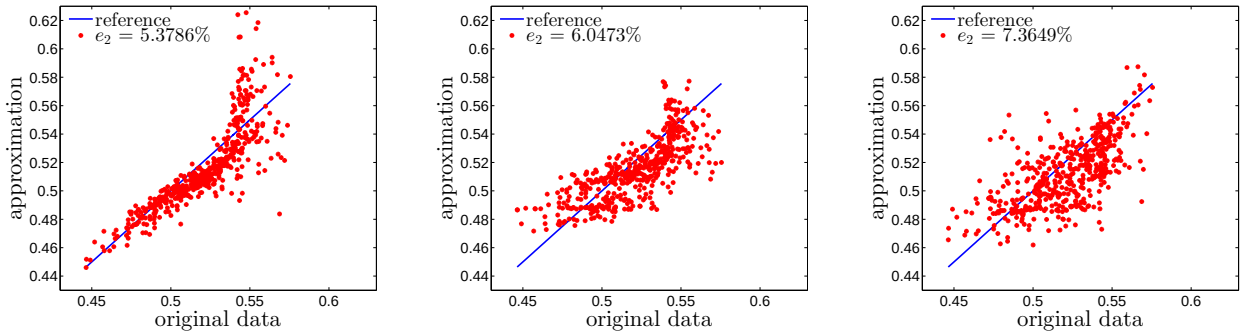
Figure 13. Normalized root mean square error (e_1) distributions for the 500 snapshots in the validation set.

$\mathbf{FI}_{\max} = \{FI^i_{\max}\}_{i=1}^{n_{sv}}$ is a vector collecting the $FI_{\max}$ "truth" values provided by the FEM reference data for the $n_{sv} = 500$ points in the validation set. Similarly, the vector $\tilde{\mathbf{FI}}_{\max} = \{\tilde{FI}^i{}_{\max}\}_{i=1}^{n_{sv}}$ collects the related approximations provided by POD, POD+SOM, or POD+gappy+SOM. $(FI_{\max})_{\max}$ and $(FI_{\max})_{\min}$ are the maximum and minimum values of the reference $FI_{\max}$ among the n_{sv} points of the validation set. Figure 14 compares the values of $FI_{\max}$ estimated by the FEM and the POD approximations for the snapshots in the validation set. The figure also shows the corresponding normalized root mean square error e_2, indicating promising performance of the strategy.

Note that the approximation of **FI** obtained with POD and POD+SOM are intermediate steps evaluated here to provide insight to the strategy, but would not actually be used as approximations in a practical application of our method. These two intermediate approximations do not use the gappy POD to reconstruct the measurement POD coefficients, $\boldsymbol{\alpha}$; instead each $\alpha_j^m(\mathbf{x})$ is approximated with a different polynomial quadratic response surface model:

$$\tilde{\alpha}_j^m(\mathbf{x}) = a_{0j}^m + \sum_{i=1}^{n_v} a_{ij}^m x_i + \sum_{i<k}^{n_v} a_{ikj}^m x_i x_k + \sum_{i=1}^{n_v} a_{iij}^m x_i^2 \approx \alpha_j^m(\mathbf{x}),$$

where a_i are the coefficients of the response surface model and $\mathbf{x} = [x_1, \ldots, x_{n_v}]^\top$ is the n_v-dimensional state vector. For this particular example, reconstruction of the measurement coefficients via gappy POD leads to a final better approximation of the overall capability snapshot than using the $\tilde{\boldsymbol{\alpha}}$ given by the set of response surface models defined above. This is the reason for the smaller values on average of e_1 for the POD+gappy+SOM estimates (Figure 13(f)) with respect to the POD+SOM approximations (Figure 13(e)) and the corresponding smaller mean value $\bar{e_1}$ reported in Table 3. In general, the relative errors of the various levels of approximation will depend on the case at hand, as well as the quantity and quality of available sensor data.



(a) **FI** POD approximation

(b) **FI** POD approximation with SOM mapping

(c) **FI** POD approximation with SOM mapping and gappy reconstruction

Figure 14. Comparison of estimated $FI_{\max}$ between FEM and POD models.

III.D. Computational runtimes

The procedure has been implemented in MATLAB® and tested on an Intel Core i7-2600 @ 3.40 GHz. Table 4 shows the runtime for each step of the online phase, as well as the cumulative runtime for the complete mapping. The table reports the mean values $\bar{t}$ over the entire validation set. On average, the entire online procedure mapping from data to capability estimates executes in 7.5 ms. In the next section we discuss how this computational cost scales with the size of the models and representations.

Table 4. Mean runtime ($\bar{t}$) evaluated over the 500 snapshots in the validation set for each step of the online phase.

Online mapping phase	$\bar{t}$ [ms]
$\boldsymbol{\alpha}_g$ reconstruction	6.579
coefficients clustering	0.799
β_j^c RS approximation	0.038
capability final estimate	0.081
Complete online mapping	7.498

IV. Discussion

This section discusses the role of the tuning parameters N_α, N_β and N_c, including a discussion of the criteria that drive their choice and the way they affect the procedure in terms of cost and accuracy.

For the general approach presented in Section II, the cost of the online process depends on three parameters: (i) the overall number of measurement POD coefficients to be reconstructed from the sensed data, N_α, (ii) the overall number of capability POD coefficients to be evaluated as a function of measurement POD coefficients, N_β, and (iii) the number of clusters that characterize the space of POD coefficients, N_c. Recall that, given sensor data, we first recover the N_α measurement POD coefficients via gappy POD, then classify the result into the closest cluster among the N_c available clusters. Then the N_β capability POD coefficients are evaluated as functions of the N_α measurement POD coefficients according to the polynomial relationships associated with the chosen cluster.

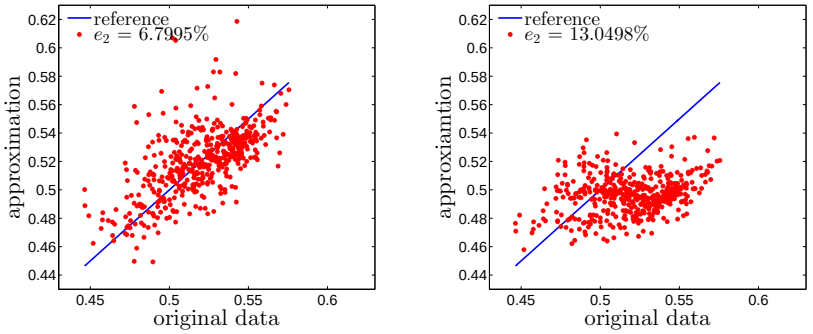
The accuracy of the overall mapping relies not only on the number of POD modes we consider in our approximations, but also on the quality of the estimated POD coefficients. Therefore, the selected α_j^m and β_j^c should be well reconstructed and should also represent the dominant POD modes. In our numerical experiments, these two conditions largely overlap—most likely because the dominant POD modes generally represent the smoothest spatial modes, which are most informed by sparse sensor data. Because of this, choosing a small number N_α of POD measurement coefficients to reconstruct does not introduce significant errors in the final result. Similarly, choosing a small N_β captures the most important information and yields satisfactory results. In contrast, the choice of N_c is a trade-off between a good characterization of the training set and avoidance of over-fitting.

The measurement POD coefficients α_j^m contribute to the online mapping in two ways: (i) the identification of the closest SOM cluster according to (10), and (ii) the response surface approximations of capability POD coefficients $\beta_j^c(\boldsymbol{\alpha})$. The choice of N_α is thus a tradeoff between computational cost (cost of the mapping scales with the number of coefficients), and the risk of misclassification for task (i) and poor estimation for task (ii). We note that it is not necessary to use the same number of coefficients for both tasks. In the examples shown in Section III, we found that a successful strategy is to use a higher number of coefficients to identify the SOM cluster and a reduced number of coefficients to construct the response surface approximations.

For the capability POD coefficients β_j^c, the number of modes retained for the final estimate of the capabilities varies for different cases, according to the cluster into which they are classified. Thus, the methodology allows naturally for adaptivity over the state space. This is determined offline, where we employ the training vectors grouped into a given cluster to determine the set of characteristic response surface models for $\beta_j^c(\boldsymbol{\alpha})$. The specific elements of β_j^c included in the model for a given cluster (and the corresponding number of coefficients) are determined by evaluating the reconstruction error using the set of evaluation snapshots. This means that N_β differs for each cluster and the retained modes are not necessarily those corresponding to the largest POD eigenvalues. In our test cases, we observe that it is usually the dominant POD modes that are selected, while the reconstruction accuracy decreases for higher order modes, leading them to be discarded.

Finally, the number of clusters N_c is determined offline and represents an output of the SOM training. Since each neuron of the SOM network can become the centroid of a cluster, the maximum number of possible clusters is given by the overall number of neurons n_w that constitute the network, that is $N_c \leq n_w$. Therefore, N_c is limited by the size of the network chosen to characterize the space of the POD coefficients. Moreover, since the number of training vectors in the evaluation set is limited, the final value of N_c must guarantee a minimum number of vectors in each cluster in order to have the necessary samples with which to construct the local response surface models.

To demonstrate some of these tradeoffs for our test problem, Figure 15 shows two limiting cases, both assessed for all snapshots in the validation set. In Figure 15(a), instead of considering a POD approximation of the full capability quantity of interest **FI**, we map directly to the scalar value of $FI_{\max}$. In this case SOM is trained with a matrix $\mathbf{T}$ that includes the $FI_{\max}$ values for each snapshot but excludes the capability POD coefficients β_j^c. This option reduces the number of operations in the online phase, but does not provide any information about the distribution of **FI**. It shows a small improvement over Figure 14(c). In Figure 15(b), we use only a single cluster, $N_c = 1$. This amounts to using the same mapping relationship between β_j^c and the α_j^m coefficients for the entire space of coefficients. In other words, we replace the SOM with a simple polynomial mapping between measurement and capability POD coefficients. Figure 15(b) shows that this simplification leads to a larger error. This approach could be suitable for simpler problems, but does not perform well in our application case.



(a) $FI_{\max}$ estimated directly from the SOM with no POD approximation of **FI**

(b) $FI_{\max}$ estimated using a single cluster, i.e., with POD approximation but no SOM

Figure 15. Comparison of $FI_{\max}$ between FEM and POD approximation for limiting cases.

The discussion in this section provides qualitative guidance in selecting the three algorithm tuning parameters N_α, N_β and N_c. For a given problem, quantitative selection can be achieved in a systematic way through careful analysis of the method's performance on the set of evaluation snapshots. It is also important to note that the specific tradeoffs that inform these choices will depend not only on the problem at hand, but

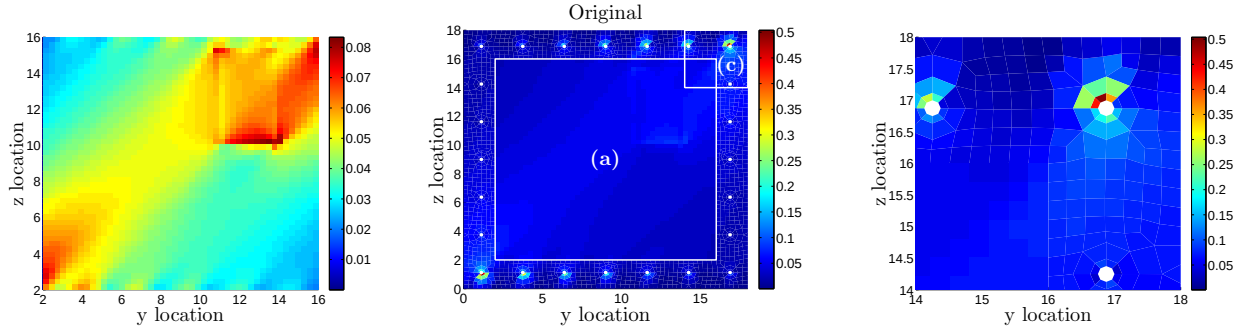
also on the available resources (quantity/quality of sensor data and online computational time requirements).

V. Conclusions

This paper proposes an offline/online strategy for the data-to-decision problem of on-board structural assessment. The strategy invokes multiple sources of information—high-fidelity finite element model simulations, sensor data, low-fidelity approximations of the quantities of interest—and reduced-order modeling techniques. Our approach combines proper orthogonal decomposition and self-organizing maps to obtain a fast mapping from measurements to capabilities. The approach is demonstrated for assessment of a damaged composite panel of a UAV wing, in which strain components are measured and the failure index is the capability quantity of interest. The methodology permits a choice among several levels of assessment of a damaged structure in order to meet different requirements or different constraints in resource availability. For example, one could pursue a complete diagnosis of the damage using a reconstruction of the failure index distribution over the panel, or one could assess the structure capability directly with an estimate of maximum failure index. Furthermore, the accuracy of the assessment can be controlled through the complexity of the models invoked, from expensive finite element models to cheaper reduced-order approximations. The demonstration of structural assessment of a wing panel presented here shows that the approach is a promising strategy to enable on-board data-driven decision-making. Further developments require a deeper investigation of the interface with actual sensor data, consideration of sensor noise, consideration of computational resource allocation, and the extension of the state variable to consider loading provided by realistic maneuver simulations. Another important open question relates to the fidelity of the FEM model itself, here taken to be our reference solution. The use of actual sensor data will provide an avenue to investigate this question and its impact on decision-making.

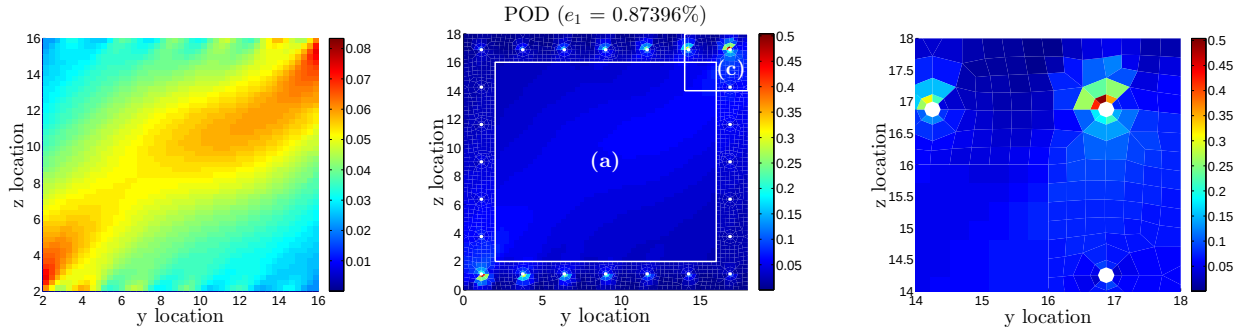
Appendix

We present here detailed results for two additional test cases within the validation set. Compared to the results presented in the main body of the paper, the first additional test case here considers a smaller damage region but with two plies unaffected. The damage state vector for this case is $\mathbf{x}^{\text{test}} = [2.99, 5.38, 12.55, 12.71, 2]$. The second additional test case considers a medium damage region again with two plies unaffected, with $\mathbf{x}^{\text{test}} = [7.81, 4.87, 4.99, 11.45, 2]$. Figures 16 and 18 show the FEM failure index data, while Figures 17 and 19 display the final approximations provided by our complete online mapping procedure.



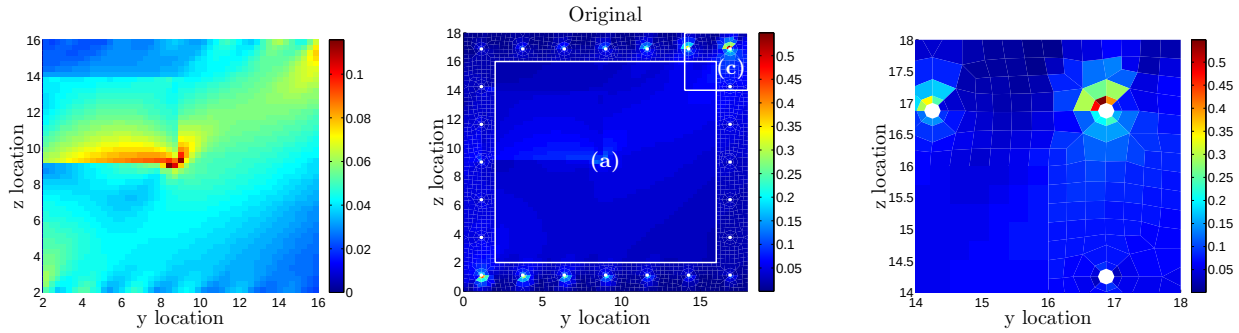
(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 16. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [2.99, 5.38, 12.55, 12.71, 2]$: original FEM data $\mathbf{s}_1(\mathbf{x}^{\text{test}}) = \mathbf{FI}(\mathbf{x}^{\text{test}})$.



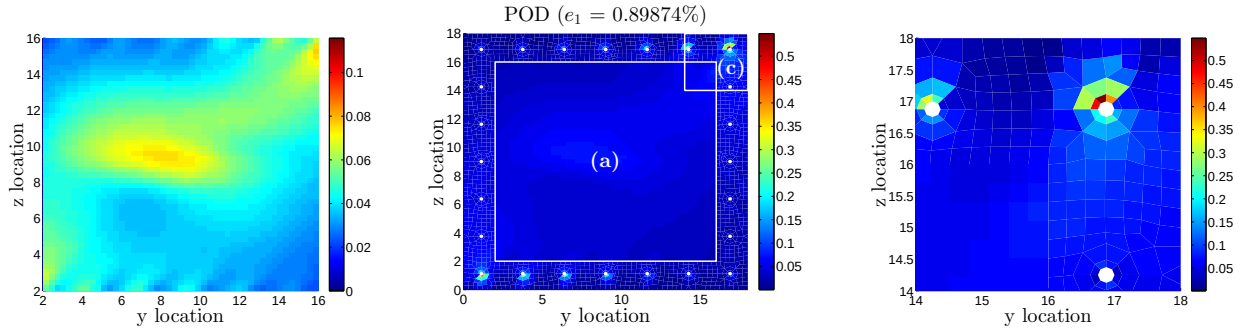
(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 17. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [2.99, 5.38, 12.55, 12.71, 2]$: approximation using POD coefficients reconstructed from sensor data using gappy POD and SOM mapping, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha}_g) = \bar{\mathbf{s}}_1 + \sum_{j=1}^{N_\beta} \beta_j^c(\boldsymbol{\alpha}_g)\psi_j^c$, $N_\beta = 9$.



(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 18. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.81, 4.87, 4.99, 11.45, 2]$: original FEM data $\mathbf{s}_1(\mathbf{x}^{\text{test}}) = \mathbf{FI}(\mathbf{x}^{\text{test}})$.



(a) Zoom on the middle (b) Complete panel (c) Zoom on the top right corner

Figure 19. Failure Index FI, Ply 4 for $\mathbf{x}^{\text{test}} = [7.81, 4.87, 4.99, 11.45, 2]$: approximation using POD coefficients reconstructed from sensor data using gappy POD and SOM mapping, $\tilde{\mathbf{s}}_1(\boldsymbol{\alpha}_g) = \bar{\mathbf{s}}_1 + \sum_j \beta_j^c(\boldsymbol{\alpha}_g)\psi_j^c$, $j = 1, 3, 4, 5, 6, 8, 9$.

Acknowledgements

This work was supported by AFOSR grant FA9550-11-1-0339 under the Dynamic Data-Driven Application Systems Program, Program Manager Dr. Frederica Darema.

References

[1]Farrar, C. R., Doebling, S. W., and Nix, D. A., "Vibration-based structural damage identification," *Philosophical Transactions of The Royal Society Lond. A*, Vol. 359, No. 1778, January 2001, pp. 131–149.

[2]Cawley, P. and Adams, R. D., "The location of defects in structures from measurements of natural frequencies," *Journal of Strain Analysis for Engineering Design*, Vol. 14, No. 2, April 1979, pp. 49–57.

[3]Salawu, O. S., "Detection of structural damage through changes in frequency: a review," *Engineering Structures*, Vol. 19, No. 9, 1997, pp. 718–723.

[4]Yan, Y., Cheng, L., Wu, Z. Y., and Yam, L., "Development in vibration based structural damage detection technique," *Mechanical Systems and Signal Processing*, Vol. 21, 2007, pp. 2198–2211.

[5]Sanayei, M. and Onipede, O., "Damage assessment of structures using static test data," *AIAA Journal*, Vol. 29, No. 7, July 1991, pp. 1174–1179.

[6]Sanayei, M., Onipede, O., and Babu, S. R., "Selection of noisy measurement locations for error reduction in static parametr identification," *AIAA Journal*, Vol. 30, No. 9, September 1992, pp. 2299–2309.

[7]Banan, M. R., Banan, M. R., and Hjelmstad, K. D., "Parameter estimation of structures from static response. I. Computational aspects," *Journal of Structural Engineering*, Vol. 120, No. 11, November 1994, pp. 3243–3258.

[8]Banan, M. R., Banan, M. R., and Hjelmstad, K. D., "Parameter estimation of structures from static response. II. Numerical simulation studies," *Journal of Structural Engineering*, Vol. 120, No. 11, November 1994, pp. 3259–3283.

[9]Szewczyk, Z. P. and Hajela, P., "Damage detection in structures based on feature-sensitive neural networks," *Journal of Computing in Civil Engineering*, Vol. 8, No. 2, April 1994, pp. 163–178.

[10]Sanayei, M. and Saletnik, M. J., "Parameter estimation of structures from static strain measurements. I: Formulation," *Journal of Structural Engineering*, Vol. 122, No. 5, May 1996, pp. 555–562.

[11]Sanayei, M. and Saletnik, M. J., "Parameter estimation of structures from static strain measurements. II: Error sensitivity analysis," *Journal of Structural Engineering*, Vol. 122, No. 5, May 1996, pp. 563–572.

[12]Housner, G. W., Bergman, L. A., Caughey, T. K., Chassiakos, A. G., Claus, R. O., Masri, S. F., Skelton, R. E., Soong, T. T., Spencer, B. F., and Yao, J. T. P., "Structural control: Past, present and future," *Journal of Engineering Mechanics*, Vol. 123, No. 9, September 1997, pp. 897–971.

[13]Zou, Y., Tong, L., and Steven, G. P., "Vibration-based model-dependent damage (delamination) identification and health monitoring for composite structures - A review," *Journal of Sound and Vibration*, Vol. 230, No. 2, 2000, pp. 357–378.

[14]Raghavan, A. and Cesnik, C. E. S., "Review of guided-wave structural health monitoring," *The Shock and Vibration Digest*, Vol. 39, No. 2, 2007, pp. 91–114.

[15]Rytter, A., *Vibration based inspection of civil engineering structures*, Ph.D. thesis, Aalborg University, Denmark, 1993.

[16]Casciati, S., "Response surface models to detect and localize distributed cracks in a complex continuum," *Journal of Engineering Mechanics*, Vol. 136, No. 9, September 2010, pp. 1131–1142.

[17]Fang, S. E. and Perera, R., "Damage identification by response surface based model updating using D-optimal desing," *Mechanical Systems and Signal Processing*, Vol. 25, 2011, pp. 717–733.

[18]Gao, H. Y., Guo, X. L., and Hu, X. F., "Crack identification based on Kriging surrogate model," *Structural Engineering and Mechanics*, Vol. 41, No. 1, 2012, pp. 25–41.

[19]Fu, Y. Z., Lu, Z. R., and Liu, J. K., "Damage identification in plates using finite element model updating in time domain," *Journal of Sound and Vibration*, Vol. 332, 2013, pp. 7018–7032.

[20]Kessler, S. S., Spearing, S. M., Atalla, M. J., Cesnik, C. E. S., and Soutis, C., "Structural health monitoring in composite materials using frequency response methods," *Proc. SPIE, Nondestructive Evaluation of Materials and Composites V*, Vol. 4336, Newport Beach, CA, March 4 2001.

[21]Papadopoulos, L. and Garcia, E., "Structural Damage Identification: A probabilistic approach," *AIAA Journal*, Vol. 36, No. 11, November 1998, pp. 2137–2145.

[22]Sankararaman, S. and Mahadevan, A., "Uncertainty quantification in structural health monitoring," *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, May 4-7 2009.

[23]Peng, T., He, J., Liu, Y., Saxena, A., Celaya, J., and Goebel, K., "Fatigue damage diagnosis and prognosis using Bayesian updating," *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston,MA, April 8-11 2013.

[24]Rhim, J. and Lee, S. W., "A neural network approach for damage detection and identification of structures," *Computational Mechanics*, Vol. 16, No. 6, 1995, pp. 437–443.

[25]Yam, L. H., Yan, Y. J., and Jiang, J. S., "Vibration-based damage detection for composite structures using wavelt transform and neural network identification," *Composite Structures*, Vol. 60, 2003, pp. 403–412.

[26]Lee, J. J., Lee, J. W., Yi, J. H., Yun, C. B., and Jung, H. Y., "Neural networks-based damage detection for bridges considering errors in baseline finite element models," *Journal of Sound and Vibration*, Vol. 280, 2005, pp. 555–578.

[27]Zang, C. and Imregun, M., "Structural damage detection using artificial neural networks and measured FRF data reduced via principal component projection," *Journal of Sound and Vibration*, Vol. 242, No. 5, 2001, pp. 813–827.

[28]Rackwitz, R., "Reliability abalysis - a review and some perspectives," *Structural Safety*, Vol. 23, 2001, pp. 365–395.

[29]Bucher, C. G. and Bourgund, U., "A fast and efficient response surface approach for structural reliability problems," *Structural Safety*, Vol. 7, 1990, pp. 57–66.

[30]Hurtado, J., "An examination of methods for approximating implicit limit state functions from the viwpoint of statistical learning theory," *Structural Safety*, Vol. 26, 2004, pp. 271–293.

[31]Most, T. and Bucher, C., "Adaptive response surface approach using artificial neural networks and moving least squares," *Proceedings of the 17th International Conference of Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, Bauhaus-University Weimar, April 8-11 2006.

[32]Bucher, C. and Most, T., "A comparison of approximate response functions in structural reliability analysis," *Probabilistic Engineering Mechanics*, Vol. 23, 2008, pp. 154–163.

[33]Sudret, B., "Meta-models for structural reliability and uncertainty quantification," *Proceedings of the 5th Asian-Pacific Symp. Struct. Reliab. (APSSRA'2012)*, Singapore, 2012.

[34]Ghosh, J., Padgett, J. E., and nas Osorio, L. D., "Surrogate modeling and failure surface visualization for efficient seismic vulnerability assessment of highway bridges," *Probabilistic Engineering Mechanics*, Vol. 34, 2013, pp. 189–199.

[35]Basudhar, A., Missoum, S., and Sanchez, A. H., "Limit state function identification using Support Vector Machines for discontinuous responses and disjoint failure domains," *Probabilistic Engineering Mechanics*, Vol. 23, 2008, pp. 1–11.

[36]Basudhar, A. and Missoum, S., "Adaptive explicit decision functions for probabilistic design and optimization using support vector machines," *Computers and Structures*, Vol. 86, 2008, pp. 1904–1917.

[37]Basudhar, A. and Missoum, S., "Two alternative schemes to update SVM approximations for identification of Explicit Decision Functions," *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia Canada, September 10-12 2008.

[38]Basudhar, A. and Missoum, S., "A sampling-based approach for probabilistic design with random fields," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, 2009, pp. 3647–3655.

[39]Bourinet, J. M., Deheeger, F., and Lemaire, M., "Assessing small failure probabilities by combined subset simulation and support vector machines," *Structural Safety*, Vol. 33, 2011, pp. 343–353.

[40]Missoum, S., "Controlling structural failure modes dring an impact in the presence of uncertainties," *Structural and Multidisciplinary Optimization*, Vol. 34, 2007, pp. 463–472.

[41]Lumley, J. L., "The Structure of Inhomogeneous Turbulent Flows," *Atmospheric Turbulence and Radio WavePropagation*, edited by Yaglom and Tatarsky, Macmillian, Moscow and Toulouse, 1967, pp. 166–178.

[42]Sirovich, L., "Turbulence and the Dynamics of Coherent Structures. Part 1: Coherent Structures," *Quarterly of Applied Mathematics*, Vol. 45, No. 3, October 1987, pp. 561–571.

[43]Holmes, P. J., Lumley, J. L., Berkooz, G., Mattingly, J., and Wittenberg, R. W., "Low-dimensional Models of Coherent Structures in Turbulence," *Physics Reports*, Vol. 287, No. 4, 1997, pp. 337–384.

[44]Ly, H. V. and Tran, H. T., "Modeling and Control of Physical Processes using Proper Orthogonal Decomposition," *Journal of Mathematical and Computer Modeling*, 1999.

[45]Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper Orthogonal Decomposition Extensions for Parametric Applications in Transonic Aerodynamics," *21st Applied Aerodynamics Conference*, AIAA 2003-4213, Orlando, Florida, June 23-26 2003.

[46]Kohonen, T., *Self-Organizing Maps*, Springer-Verlag, New York, 3rd ed., 2001.

[47]Erwin, E., Obermayer, K., and Schulten, K., "Self-organizing maps: ordering, convergence properties and energy functions," *Biological Cybernetics*, Vol. 67, 1992, pp. 47–55.

[48]Kohonen, T., "Things you haven't heard about the self-organizing map," *Proc. IEEE Int. Conf on Neural Networks*, San Francisco, 1993, pp. 1147–1156.

[49]Kohonen, T., Oja, E., Simula, O., Visa, A., and Kangas, J., "Engineering applications of the Self-organizing map," *Proceedings of the IEEE*, Vol. 84, No. 10, October 1996, pp. 1358–1384.

[50]Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

[51]Anderson, J. A., *Introduction to Neural Networks*, MIT Press, Cambridge, MA, 1995.

[52]Haykin, S., *Neural Networks and Learning Machines*, Prentice Hall, 3rd ed., 2009.

[53]Everson, R. and Sirovich, L., "The Karhunen-Loeve Procedure for Gappy Data," *Journal of Optical Society of America*, Vol. 12, No. 8, 1995, pp. 1657–1664.

[54]Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, August 2004, pp. 1505–1516.

[55]McKay, M. D., Beckman, R. J., and Conover, W. J., "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, Vol. 21, 1979, pp. 239–245.

[56]Iman, R. L. and Conover, W. J., "Small sample sensitivity analysis techniques for computer models, with an application to risk assessment," *Communications in Statistics, Part A–Theory and Methods*, Vol. 9, No. 17, 1980, pp. 1749–1842.

[57]Viana, F. A. C., "Things you wanted to know about the Latin hypercube design and were afraid to ask," *Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, May 19-24 2013.