AIAA Journal
Vol. 49, No. 8, August 2011

Reduced-Order Modeling of a High-Fidelity Propulsion System Simulation

Kyunghoon Lee,[*] Taewoo Nam,[†] Christopher Perullo,[‡] and Dimitri N. Mavris[§]
Georgia Institute of Technology, Atlanta, Georgia 30332-0150

DOI: 10.2514/1.J050887

Ever stringent aircraft design requirements on simultaneous reduction in fuel consumption, emissions, and noise necessitate innovative, integrated airframe designs that require concurrent engine designs. To fulfill these design challenges, aerospace engineers have relied on a physics-based engine modeling environment, such as the numerical propulsion system simulation. To expedite the use of numerical propulsion system simulation in aircraft design, this research proposes a methodology for the reduced-order modeling of numerical propulsion system simulation by incorporating the following two techniques: probabilistic principal component analysis for basis extraction and neural networks for weighting coefficient prediction. To efficiently achieve an empirical orthogonal basis, this research capitalizes on an expectation-maximization algorithm for probabilistic principal component analysis to handle numerical propulsion system simulation engine decks that typically lack some data due to failed offdesign performance analyses; they result from the numerical instabilities present in the Newton–Raphson method used within numerical propulsion system simulation to find a converged solution at a given flight condition. In addition, to effectively explore a weighting coefficient space, this research uses neural networks to deal with six numerical propulsion system simulation engine modeling parameters. As a proof of concept, the proposed numerical propulsion system simulation reduced-order modeling method is applied to a numerical propulsion system simulation turbofan engine model usually employed for conventional civil transport aircraft. Comprehensive prediction quality investigations reveal that engine performance metrics estimated by the reduced-order numerical propulsion system simulation model show considerably good agreement with those directly obtained by numerical propulsion system simulation. Furthermore, the reduced numerical propulsion system simulation engine model is integrated with the flight optimization system in lieu of directly using numerical propulsion system simulation as an illustration of the utility of numerical propulsion system simulation reduced-order modeling for aircraft design research.

Nomenclature

$\mathbf{C}$	=	model covariance matrix
d	=	dimension of an observed variable
$\mathbf{I}$	=	identity matrix
k	=	number of iterations
$\mathcal{L}$	=	log-likelihood function
$\mathcal{L}_C$	=	complete-data log-likelihood function
N	=	snapshot ensemble size
$\mathcal{N}$	=	Gaussian probability distribution
p	=	probability density function
q	=	dimension of a latent variable, i.e., model selection
$\mathbb{R}^n$	=	n-dimensional real number space
$\mathbf{S}$	=	sample covariance matrix
$\mathbf{T}$	=	collection of error-accounted observed variables
$\mathbf{t}$	=	error-accounted observed variable
$\mathrm{tr}(\cdot)$	=	trace
$\mathbf{V}$	=	eigenvector matrix, i.e., proper orthogonal decomposition basis
$\mathbf{V}_e$	=	guessed $\mathbf{V}_q$ obtained from $\tilde{\mathbf{Y}}^{(0)}$
$\mathbf{V}_q$	=	matrix of first q eigenvectors
$\mathbf{W}$	=	factor-loading matrix
$\mathbf{X}$	=	collection of latent variables
$\mathbf{x}$	=	latent variable
$\mathbf{Y}$	=	collection of observed variables
$\mathbf{y}$	=	observed variable
$\boldsymbol{\epsilon}$	=	error variable
λ	=	eigenvalue
$\boldsymbol{\mu}$	=	mean vector
σ^2	=	variance
$\mathbf{1}_N$	=	vector with N ones
$\langle\cdot\rangle$	=	expectation
$\bar{(\cdot)}$	=	sample mean
$\widetilde{(\cdot)}$	=	estimation

Subscripts

j	=	jth vector
ML	=	maximum likelihood estimate

Superscript

(k)	=	kth iteration

I. Introduction

HISTORICALLY, aircraft engine design has occurred in isolation of airframe design during the design processes. However, as advanced aircraft concepts, such as NASA's N+1, N+2, and N+3 aircraft, aim for radical improvements on fuel burn, noise, and emissions in 10, 20, and 30 years, aircraft design research using the traditional engine design approach results in a limited design

Presented as Paper 2010-9193 at the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, 13–15 September 2010; received 25 August 2010; revision received 18 January 2011; accepted for publication 6 February 2011. Copyright © 2011 by Kyunghoon Lee, Taewoo Nam, Christopher, Perullo, and Dimitri Mavris. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/11 and $10.00 in correspondence with the CCC.

[*] Postdoctoral Fellow, Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering. Member AIAA.

[†] Research Engineer II, Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering.

[‡] Research Engineer I, Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering.

[§] Boeing Professor of Advanced Aerospace Systems Analysis and Director of Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering. Fellow AIAA.

space. To broaden out a feasible design space for the next generation of aircraft, aircraft design must take into account not only airframe but also propulsion system aspects in parallel [1,2]. Since heuristic engine performance correlations traditionally used in aircraft design are inappropriate for such design studies, one requires a sophisticated physics-based engine simulation. For this purpose, the numerical propulsion system simulation (NPSS) [3,4], developed by the joint efforts of NASA and industry, is a de facto standard tool in aerospace engineering. Because of its object-oriented architecture, NPSS provides a generic engine modeling framework that allows one to devise propulsion systems for various vehicles ranging from conventional configurations, such as commercial jet transport, supersonic business jets, and military aircraft, to visionary configurations, such as hybrid wing–body and truss-braced wing aircraft.

Although NPSS itself is not as computationally prohibitive as other high-fidelity physics-based simulations, its capability to interface with high-fidelity codes can drastically increase its computational time on the order of hours [5,6]. The reduced-order modeling (ROM) of NPSS can normally decrease computational time from hours or minutes to seconds, and it is desirable for the following circumstances:

1) Nondeterministic and optimization-driven design studies [7,8] repeatedly invoke NPSS a colossal number of times during a design process.

2) Directly interweaving NPSS into a multidisciplinary design environment may demand arduous integration work.

3) Cooperative research entities that do not own NPSS or possess engine design expertise can reap the benefits of NPSS analysis capabilities through a reduced-order NPSS model.

Particularly with regard to the first case, when an engine design becomes tightly combined with a vehicle design, aircraft system design efforts require that engine cycle parameters be varied in concert with vehicle design variables to arrive at a truly optimized, integrated design. As illustrated in Fig. 1, this airframe- and engine-integrated design approach demands the repeated execution of NPSS, which easily becomes computationally expensive. Therefore, despite up-front computational costs, a reduced-order model of NPSS that can almost instantaneously yield an engine deck is indispensable to every aircraft system design process.

For the modeling of engine performance metrics, semiempirical equation-based approaches fitting measurement data through coefficient adjustments have been used to derive fuel consumption models [9,10]; however, these models are not generally accurate for all mission profiles; thus, they are limitedly acceptable for only certain mission segments. For example, [10] pointed out that the fuel consumption model of the base of aircraft data is mostly accurate for a cruise mode, which inspired the authors to develop their fuel consumption model that is particularly accurate at climbing/landing around terminal areas. In contrast with these restrictively applicable models, this research attempts to devise generic prediction models of engine performance metrics with NPSS through its reduced-order model. Conceptually, a reduced-order model is twofold; it has 1) an empirical orthogonal basis that is normally obtained by proper orthogonal decomposition (POD), referred to as principal component analysis (PCA) [11]; and 2) weighting coefficients that adjust the contributions of basis vectors accordingly with changes in the independent modeling parameters.

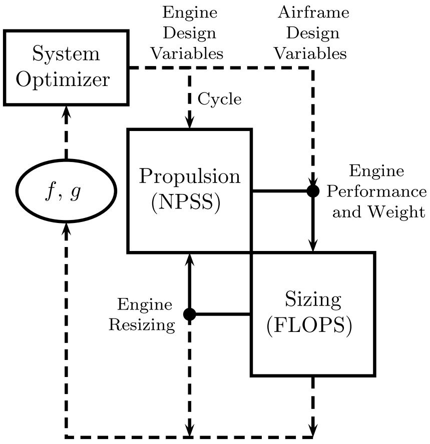


Fig. 1 Schematic of an airframe- and engine-integrated design environment.

To construct a reduced-order model of NPSS, one typically encounters research predicaments in both evaluating empirical orthogonal bases and determining weighting coefficients. Regarding empirical basis extraction, due to successive discontinuity shifts appearing at every throttle setting change, resultant NPSS engine decks contain distinct stationary discontinuities that lead to a sawtooth-type data pattern as a matter of a data format. Moreover, due to numerical instabilities occurring at certain offdesign performance analyses within NPSS, some degree of data absence is inescapable with NPSS. Although an empirical orthogonal basis evaluated by POD can deal with variations in simulation data exhibiting stationary discontinuities [12], POD fails for deficient data. To surmount the evaluation of a POD basis for given incomplete data, such as NPSS engine decks, one can rely on either gappy POD [13] or probabilistic PCA (PPCA) [14]; while the former generates an orthogonal basis by solving a least-squares problem defined with a gappy norm and a POD basis, the latter yields a nonorthogonal basis through probability parameter estimation with the help of an expectation-maximization (EM) algorithm, termed an EM algorithm for PCA (EM-PCA).

Since [15] showed the potential application of gappy POD to aerospace engineering problems, gappy POD has been used for various aerospace engineering applications, such as inverse airfoil design [16] and spurious particle image velocimetry (PIV) data restoration [17]. Similarly, [18] introduced PPCA to the realm of aerospace engineering; furthermore, [19,20] exhaustively investigated PPCA in comparison with gappy POD, demonstrating greater efficiency than gappy POD for PIV data reconstruction. Unlike the previous applications of gappy POD and the EM-PCA, which focused on restoring missing data, this research intends to exploit these methods to retrieve an empirical orthogonal basis from deficient simulation data. According to [20], the EM-PCA is computationally superior to gappy POD for such a sparse missing data pattern of NPSS engine decks caused by occasionally failed offdesign performance analyses. Therefore, this research adopts the EM-PCA instead of gappy POD to efficiently distill the POD bases of engine performance metrics from compiled NPSS engine decks.

The other research impediment to the general use of POD-based ROM for design applications is difficulties associated with predicting weighting coefficients as the number of independent modeling parameters grows. Once weighting coefficients at known independent modeling parameters are evaluated with the method of weighted residuals, such as least-squares methods, one is required to infer weighting coefficients at unknown independent modeling parameters for prediction. In the case of POD-based ROM used for unsteady flow analysis that is time dependent only, the problem of estimating weighting coefficients is tractable; however, as POD-based ROM diffuses into design applications for which the number of independent modeling parameters is typically more than one, weighting coefficient estimation turns into a matter of multivariate scattered data interpolation [21].

For instance, [22] employed cubic spline interpolation in the reduced-order steady aerodynamic model to delineate variations in airfoil surface pressure due to two flow parameters: a Mach number and an angle of attack. Likewise, [23] used a POD-based reduced-order model for parametric studies of weapon aerodynamics, expanding the number of modeling parameters to three: a Mach number, an incidence angle, and a flare base radius. In addition, a couple of design of experiments (DOEs) were tested, such as full factorial and Latin hypercube designs, and several response surface construction schemes were examined, including linear regression and polynomial-augmented multiquadratic radial basis function. Furthering these previous endeavors, this research capitalizes on neural networks in connection with POD-based ROM to handle a large number of independent modeling parameters, namely, six

NPSS engine modeling parameters. Moreover, to ease the coefficient evaluation with neural networks, this research benefits from an augmented DOE by adding the corner points of a parameter space to a maximum-entropy design, i.e., a space-filling DOE specialized for computer simulations [24,25].

Overall, this research aims to create a reduced-order NPSS model with the EM-PCA and neural networks, each of which is adopted to address inherently gappy NPSS engine decks and a large number of engine modeling parameters, respectively. The outline of this paper is as follows. Section II briefly describes the formulations of the two rudimentary components for the proposed NPSS ROM method: PPCA along with the EM-PCA and neural networks. Subsequently, Sec. III presents an NPSS engine model and the process of NPSS engine deck generation. Afterward, Sec. IV illustrates the accuracy of the reduced-order NPSS model compared with genuine NPSS and then exploits the reduced-order NPSS model in the flight optimization system (FLOPS) [26], a NASA vehicle sizing and synthesis code, to examine system level errors propagated from the reduced-order NPSS model. Section V completes this paper with concluding remarks and recommendations for future work. Finally, the Appendix provides supplementary materials that show the EM-PCA yields identical results to those of gappy POD at lower computational costs, which substantiates the use of the EM-PCA over gappy POD.

II. Theories for Reduced-Order Modeling

A. Probabilistic Principal Component Analysis

Without loss of generality, both the error-accounted snapshot $\mathbf{t}_j \in \mathbb{R}^d$ and error-accounted snapshot ensemble $\mathbf{T} = \{\mathbf{t}_j\}_{j=1}^N \in \mathbb{R}^{d\times N}$ are treated as mean centered for the derivation of the PPCA formulation.

1. Probability Model

PPCA [14] is a probabilistic generalization of PCA. The PPCA formulation presumes a linear latent variable model such that

$$\mathbf{t}_j(\mathbf{x}_j; \mathbf{W}, \boldsymbol{\epsilon}) = \mathbf{W}\mathbf{x}_j + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{t}_j \in \mathbb{R}^d$ is an error-accounted observed variable, $\mathbf{x}_j \in \mathbb{R}^q$ is a latent variable, $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is a measurement error independent of $\mathbf{x}_j$, and $\mathbf{W} \in \mathbb{R}^{d\times q}$ is a factor-loading matrix. With the assumption of Gaussian probability models for $\mathbf{x}_j$ and $\boldsymbol{\epsilon}$, such that $\mathbf{x}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, respectively, a Gaussian probability model for $\mathbf{t}_j$ is derived such that $\mathbf{t}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, where $\mathbf{C}$ is a model covariance defined as $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}$. The conditional probability of $\mathbf{t}_j$ given $\mathbf{x}_j$ is

$$p(\mathbf{t}_j|\mathbf{x}_j; \mathbf{W}, \sigma^2) = (2\pi\sigma^2)^{-d/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2\right) \tag{2}$$

from $p(\boldsymbol{\epsilon})$ with $\boldsymbol{\epsilon} = \mathbf{t}_j - \mathbf{W}\mathbf{x}_j$, and the marginal probability of $\mathbf{t}_j$ is evaluated as follows:

$$p(\mathbf{t}_j; \mathbf{W}, \sigma^2) = (2\pi)^{-d/2}|\mathbf{C}|^{-1/2} \exp(-\tfrac{1}{2}\mathbf{t}_j^T\mathbf{C}^{-1}\mathbf{t}_j) \tag{3}$$

By Bayes's rule, the posterior probability of $\mathbf{x}_j$ given $\mathbf{t}_j$ can be determined as

$$\begin{aligned} p(\mathbf{x}_j|\mathbf{t}_j) &= (2\pi)^{-q/2}|\sigma^{-2}\mathbf{M}|^{1/2} \\ &\quad\times \exp[-\tfrac{1}{2}(\mathbf{x}_j - \mathbf{M}^{-1}\mathbf{W}^T\mathbf{t}_j)^T(\sigma^{-2}\mathbf{M})(\mathbf{x}_j - \mathbf{M}^{-1}\mathbf{W}^T\mathbf{t}_j)] \end{aligned} \tag{4}$$

where $\mathbf{M} = \mathbf{W}^T\mathbf{W} + \sigma^2\mathbf{I}$.

2. Expectation-Maximization Algorithm for Probabilistic Principal Component Analysis

An EM algorithm [27] is a statistical inference method that can naturally deal with incomplete data for which the method of maximum likelihood fails. Based on the EM algorithm application to a factor analysis model [28], [14] derived an EM algorithm for PPCA, yielding the EM-PCA. To find the maximum likelihood estimates (MLEs) of PPCA parameters, the EM-PCA iteratively alternates the following two steps: 1) an expectation step (E-step) that estimates unknown variables, i.e., a latent variable and an observed variable containing missing data; and 2) a maximization step (M-step) that evaluates parameter estimates in such a way that they maximize the expectation of a log-likelihood function. To invoke the EM algorithm to PPCA, [14] combined $\mathbf{t}_j$ and $\mathbf{x}_j$ so that a complete data set $(\mathbf{t}_j, \mathbf{x}_j)$ involved inherent missing data due to $\mathbf{x}_j$. The probability model of $(\mathbf{t}_j, \mathbf{x}_j)$ can be evaluated as the joint distribution of $\mathbf{t}_j$ and $\mathbf{x}_j$, such that

$$\begin{aligned} p(\mathbf{t}_j, \mathbf{x}_j) &= (2\pi\sigma^2)^{-d/2} \\ &\quad\times \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2\right)(2\pi)^{-q/2}\exp\left(-\frac{1}{2}\|\mathbf{x}_j\|^2\right) \end{aligned} \tag{5}$$

From $p(\mathbf{t}_j, \mathbf{x}_j)$ in Eq. (5), the complete-data log-likelihood function $\mathcal{L}_C$ is

$$\begin{aligned} \mathcal{L}_C(\mathbf{W}, \sigma^2) &= \sum_{j=1}^N \ln p(\mathbf{t}_j, \mathbf{x}_j; \mathbf{W}, \sigma^2) = -\frac{N}{2}(d+q)\ln(2\pi) \\ &\quad -\frac{Nd}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{j=1}^N \|\mathbf{t}_j - \mathbf{W}\mathbf{x}_j\|^2 - \frac{1}{2}\sum_{j=1}^N \|\mathbf{x}_j\|^2 \end{aligned}$$

and its expectation is

$$\begin{aligned} \langle\mathcal{L}_C\rangle &= -\frac{N}{2}(d+q)\ln(2\pi) - \frac{Nd}{2}\ln(\sigma^2) \\ &\quad -\frac{1}{2\sigma^2}\operatorname{tr}(\mathbf{T}^T\mathbf{T} - 2\langle\mathbf{X}\rangle^T\mathbf{W}^T\mathbf{T} + \mathbf{W}^T\mathbf{W}\langle\mathbf{X}\mathbf{X}^T\rangle) - \frac{1}{2}\operatorname{tr}(\langle\mathbf{X}\mathbf{X}^T\rangle) \end{aligned} \tag{6}$$

where

$$\langle\mathbf{X}\rangle = \mathbf{M}^{-1}\mathbf{W}^T\mathbf{T}, \qquad \langle\mathbf{X}\mathbf{X}^T\rangle = \sigma^2\mathbf{M}^{-1} + \langle\mathbf{X}\rangle\langle\mathbf{X}\rangle^T$$

The first E-step estimates both $\mathbf{x}_j$ and $\mathbf{t}_j$ with missing data, using their expectations such that $\langle\mathbf{x}_j\rangle = \mathbf{M}^{-1}\mathbf{W}^T\mathbf{t}_j$ and $\langle\mathbf{t}_j\rangle = \mathbf{W}\mathbf{x}_j$, evaluated from Eqs. (4) and (2), respectively. In a matrix form, the E-step estimates the collection of each unknown variable as follows:

$$\langle\mathbf{X}\rangle = \mathbf{M}^{-1}\mathbf{W}^T\mathbf{T} \quad \text{and} \quad \langle\mathbf{T}\rangle = \mathbf{W}\mathbf{X} \tag{7}$$

where $\mathbf{X} = \{\mathbf{x}_j\}_{j=1}^N \in \mathbb{R}^{q\times N}$ and $\mathbf{T} = \{\mathbf{t}_j\}_{j=1}^N \in \mathbb{R}^{d\times N}$. The subsequent M-step evaluates parameter estimates maximizing $\langle\mathcal{L}_C\rangle$ in Eq. (6); thus, the PPCA MLEs are derived from stationary equations that annihilate the first derivatives of $\langle\mathcal{L}_C\rangle$ with respect to $\mathbf{W}$ and σ^2:

$$\tilde{W} = \mathbf{T}\langle\mathbf{X}\rangle^T\langle\mathbf{X}\mathbf{X}^T\rangle^{-1} \tag{8a}$$

$$\tilde{\sigma}^2 = \frac{1}{Nd}\operatorname{tr}(\mathbf{T}^T\mathbf{T} - 2\langle\mathbf{X}\rangle^T\mathbf{W}^T\mathbf{T} + \mathbf{W}^T\mathbf{W}\langle\mathbf{X}\mathbf{X}^T\rangle) \tag{8b}$$

Finally, both the E- and M-steps comprise the EM-PCA such that the generalized E-step is

$$\langle\mathbf{X}\rangle = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{T} \tag{9a}$$

$$\langle\mathbf{T}\rangle = \mathbf{W}\mathbf{X} \tag{9b}$$

and the M-step is

$$\tilde{\mathbf{W}} = \mathbf{T}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \tag{9c}$$

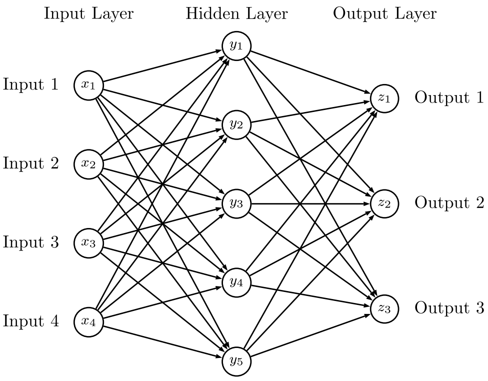


Fig. 2 Single hidden-layer feedforward neural network.

which ignores σ^2 for measurement noise irrelevant to numerical simulation data. Since the EM-PCA in Eq. (9) generates a nonorthogonal basis $\mathbf{W}$, the orthogonalization of $\mathbf{W}$ to achieve $\mathbf{V}_q$ is inevitable.

B. Neural Networks

As an analogy to a biological neural network, an artificial neural network [29] is a computational architecture interconnecting artificial neurons to address various problems through artificial intelligence characterized as learning: in other words, generalization. Since neural networks are trained so that they can adapt to unexperienced situations, they have been used in diverse applications of data processing, such as function approximation and data classification: image processing, engine management, automatic aircraft landing systems, and so on. As an illustration, Fig. 2 depicts a simple feedforward network with one hidden layer consisting of hidden nodes as similar to the neurons of biological neural networks. For a given training set, a neural network adjusts weights and biases between input and hidden layers as well as hidden and outer layers in order to match its outputs to known target values. For weight and bias determination, a backpropagation algorithm is commonly used in connection with a feedforward network to minimize errors, i.e., differences between outputs and target values. As the name of backpropagation implies, it successively propagates errors backward from output to input layers; for example, the weights and biases between the hidden and output layers are first rectified, and then those between hidden and input layers are modified.

For a single hidden-layer feedforward network, depicted in Fig. 2, the mathematical representation of neurons often relies on a sigmoid function

$$\sigma(t) = \frac{1}{1 + e^{-t}} \tag{10}$$

which is one of the most widely used activation functions [30]. Note that the sigmoid function in Eq. (10) represents gradual changes at input boundaries and rapid progression at in-between boundaries, squashing an input within the range of zero and one. Let $\mathbf{x} \in \mathbb{R}^l$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{z} \in \mathbb{R}^n$ be the inputs of neurons, the outputs of neurons, and the outputs of a single hidden-layer feedforward network, respectively, shown in Fig. 2; then, a neuron output y_j determines its value with Eq. (10) as a function of a weighted sum of the input variables and a bias, such that

$$y_j(x_i) = \sigma\left(\sum_{i=1}^{l} w_{ij}^{[x]} x_i + a_j^{[x]}\right)$$

where $w_{ij}^{[x]}$ is the weight of the ith input variable x_i, and $a_j^{[x]}$ is the bias of the jth hidden node. Similarly, the kth output variable z_k is evaluated as

$$z_k(y_j) = \sum_{j=1}^{m} w_{jk}^{[y]} y_j + a_k^{[y]}$$

where $w_{jk}^{[y]}$ is the weight of the jth hidden node, and $a_k^{[y]}$ is the bias of the kth output variable. After all, a single hidden-layer feedforward network assumes a nonlinear relationship between the input and output variables such that

$$z_k(x_i) = \sum_{j=1}^{m} w_{jk}^{[y]} y_j + a_k^{[y]} = \sum_{j=1}^{m} w_{jk}^{[y]} \sigma\left(\sum_{i=1}^{l} w_{ij}^{[x]} x_i + a_j^{[x]}\right) + a_k^{[y]}$$

and $w_{ij}^{[x]}$, $w_{jk}^{[y]}$, $a_j^{[x]}$, and $a_k^{[y]}$ necessitate a training process to find their value minimizing errors.

C. Proper-Orthogonal-Decomposition-Based Reduced-Order Modeling

Given POD basis vectors $\{\mathbf{v}_j\}_{j=1}^{q} \in \mathbb{R}^{d\times q}$, $\mathbf{y} \in \mathbb{R}^d$ can be approximated on a q-dimensional subspace as

$$\mathbf{y}\,(\boldsymbol{\vartheta}, \mathbf{x}) \simeq \sum_{j=1}^{q} \alpha_j(\boldsymbol{\vartheta}) \mathbf{v}_j(\mathbf{x}) + \bar{\mathbf{y}} \tag{11}$$

where $\bar{\mathbf{y}}$ is a sample mean, and α_j is the optimal weighting coefficient of $\mathbf{v}_j$, minimizing a projection error. Note that POD basis vectors account for spatial variations in $\mathbf{y}$ due to changes in $\mathbf{x}$ and, likewise, the coefficients do so for the parametric variations in $\mathbf{y}$ due to changes in $\boldsymbol{\vartheta}$. Provided that POD basis vectors are invariant to $\boldsymbol{\vartheta}$, POD-based ROM can predict $\mathbf{y}$ at unobserved parameters in a form similar to Eq. (11) such that

$$\tilde{\mathbf{y}}(\boldsymbol{\vartheta}, \mathbf{x}) \approx \sum_{j=1}^{q} \tilde{\alpha}_j(\boldsymbol{\vartheta}) \mathbf{v}_j(\mathbf{x}) + \bar{\mathbf{y}} \tag{12}$$

with the help of an appropriate weighting coefficient $\tilde{\alpha}_j$. For the evaluation of POD basis vectors and weighting coefficients, this research employs the EM-PCA and neural networks, respectively, to construct a reduced-order model of NPSS.

III. Generation of Reduced-Order Numerical Propulsion System Simulation Model

A two-spool separate flow turbofan was created as a thermodynamic cycle model within NPSS using the typical flight envelope of a commercial jet aircraft. To generate engine performance information, NPSS normally runs in two stages: the first stage sizes engine components according to desired engine cycle parameters, and the second (offdesign) stage, called a performance analysis, produces engine performance data at each set of engine operating conditions, consisting of a Mach number, an altitude, and a throttle setting. Note that a throttle setting is also known as a power code or a power-level angle [31]. As shown in Table 1, NPSS provides the results of engine performance analyses as an engine deck that adjoins two tables: the

Table 1 NPSS engine deck format

Engine deck inputs			Engine deck responses		
Mach number	Altitude, ft	Throttle setting	Gross thrust, lbf	Ram drag, lbf	Fuel flow, lbm/h
0	0	50	90,000	0	27,000
⋮	⋮	⋮	⋮	⋮	⋮
0.9	43,000	21	14,162	15,150	520.5

engine deck inputs on the left side of the table list various combinations of engine operating conditions, and the engine deck responses on the right side of the table arrange corresponding engine performance metrics, such as gross thrust, ram drag, fuel flow, and so forth.

In general, a functional relationship between engine deck inputs and responses hinges on the following factors: an engine cycle, an engine architecture, and the underlying fidelity of engine component models. Among these three elements, the architecture of an engine is often chosen first in conceptual aircraft design based on the mission requirements of an aircraft, as well as the financial wherewithal and competitive assessment of an engine manufacturer. Once an engine architecture is determined, one manipulates the engine cycle parameters so that a designed engine can perform to the level of the mission requirements imposed on a vehicle of interest. As indicated by [32], engine cycle changes give rise to many secondary effects, such as the influence of component pressure ratios on component efficiencies, and most of these subsidiary effects are included in the NPSS model, implemented for this research; conventional empirical models often lack these effects.

A. Numerical Propulsion System Simulation Reduce-Order Modeling Procedure

The following steps briefly describe the construction of a reduced-order NPSS model with the help of EM-PCA and neural networks. More details regarding each step will be presented accordingly after this subsection.

1) The first step is to generate engine deck snapshots using NPSS at various engine cycle parameters. Step 1 populates sample engine decks, not only for extracting empirical orthogonal bases with the EM-PCA but also for training weighting coefficient models with neural networks. One can benefit from DOE techniques specialized for computer simulations to effectively capture an entire engine modeling parameter space. Note that a DOE is required to encompass the corners of the parameter space in order for a reduced-order model to properly account for all variations within the parameter space.

2) The second step is to compile each engine deck response of interest. Step 2 separately collects engine responses from populated engine decks, producing snapshot ensembles of engine deck responses for basis extraction with the EM-PCA.

3) The third step is to use the EM-PCA to obtain the empirical orthogonal bases of engine deck responses. Step 3 capitalizes on the EM-PCA to distill empirical orthogonal bases from the snapshot ensembles of engine deck responses. Based on eigenvalues representing the relative importance of corresponding basis vectors, one is required to choose a proper number of basis vectors that sufficiently delineate all variations in the snapshot ensembles. Note that the EM-PCA technically yields a nonorthogonal basis that necessitates orthogonalization for an orthogonal basis.

4) The fourth step is to evaluate optimal weighting coefficients by least-squares methods. Step 4 determines the best coefficients: those that produce minimum projection errors for given empirical orthogonal bases. Owing to the orthogonality of the empirical bases, the evaluation of weighting coefficients reduces to mere matrix multiplication dispensing with matrix inversion. The optimal coefficients achieved in this step will be targets for neural network models in step 5.

5) The fifth step is to use neural networks to build the prediction models of weighting coefficients. Step 5 constructs coefficient prediction models based on function-approximation neural networks to estimate weighting coefficients at unseen engine modeling parameters. For the prediction models of weighting coefficients, the engine modeling parameters in the DOE table in step 1 are taken as inputs, and the optimal weighting coefficients evaluated in step 4 are fed as targets. Note that the coefficient prediction models significantly sway the goodness of a reduced-order NPSS model, since weighting coefficients are the only factors that can account for changes in engine modeling parameters; empirical bases are assumed to be invariant to the parameter changes in the formulation of POD-based ROM.

The sixth step is to examine model-fit errors with training data. Step 6 carries out model-fit-error tests for evaluating the accuracy of the reduced-order NPSS model built upon the empirical bases in step 3 and the neural network models in step 5. Although the optimal weighting coefficients achieved in step 4 can replicate observed engine deck responses, weighting coefficients estimated by the neural network models in step 5 do not necessarily yield the exact weighting coefficients at known engine modeling parameters. Therefore, model-fit-error tests are useful for verifying the quality of the prediction models of weighting coefficients.

The seventh step is to examine model prediction errors with randomly generated test data. Step 7, the final step, repeats the same error analysis in step 6 with random test data that are new to the reduced-order NPSS model. This prediction test with random data is conducive to examining the overall credibility of the reduced-order model at unobserved engine modeling parameters.

B. Numerical Propulsion System Simulation Engine Deck Generation

To develop a reduced-order NPSS model, this research varied the following six engine modeling parameters: extraction ratio, fan pressure ratio (FPR), high-pressure compressor pressure ratio (HPCPR), low-pressure compressor pressure ratio (LPCPR), maximum combustor exit temperature (MaxT4) at takeoff, and sea-level static thrust T_{SLS}. These engine cycle and scaling parameters are commonly used in conceptual engine design [33] and specified at cruise and maximum power settings otherwise stated. Each parameter has its own unique impact, not only on engine performance but also on the closeness of engine performance to thrust and fuel flow requirements. While mass flow is the direct input that affects engine thrust, it is not convenient to use mass flow as an input in the traditional aircraft sizing process. For this reason, the propulsion simulation has been set up to accept an input from the vehicle sizing analysis; NPSS then iterates on engine mass flow during the engine sizing process to achieve a specified T_{SLS}.

Overall, different combinations of the six parameters, for which the ranges are presented in Table 2, populated a total of 500 sample engine decks. For the strategic exploration of an engine modeling parameter space, the generation of sample engine decks adopted a maximum-entropy space-filling design [34] augmented with a total of 76 corner points using JMP software. At each set of the engine modeling parameters, NPSS produced engine performance data with respect to 924 engine operating conditions composed of 11 altitudes, $5 \sim 10$ Mach numbers, and 11 throttle settings. As an illustration, Fig. 3 shows various combinations of engine operating altitudes and Mach numbers employed for the NPSS analysis in this research. Note that each altitude is associated with dissimilar sets of Mach numbers, and every Mach and altitude combination contains data for the same throttle settings, ranging from full power to idle [31].

As a result, a total of 500 snapshots of an engine deck were generated and compiled, resulting in four 924×500 snapshot ensembles for four different engine deck responses of interest in conceptual aircraft design: gross thrust, ram drag, fuel flow, and emission index NO_x (EINOx). These snapshot ensembles inevitably lack some offdesign performance analyses at certain flight conditions, since the Newton–Raphson method in NPSS sporadically causes convergence failures leading to absent performance information, particularly at low throttle settings. As an illustration, Fig. 4 depicts failed performance analyses observed in the collected

Table 2 Ranges of NPSS engine cycle and scaling parameters

Parameter	Range
Extraction ratio	$1 \sim 1.2$
FPR	$1.5 \sim 1.7$
HPCPR	$18 \sim 22$
LPCPR	$1.2 \sim 1.6$
MaxT4, °R	$3200 \sim 3600$
T_{SLS}, lbf	$80,000 \sim 100,000$

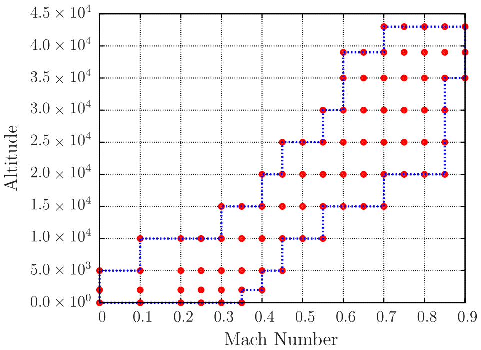


Fig. 3 Flight envelope of NPSS engine model.

snapshots of engine deck responses; a training data set is deficient of 0.143% data in Fig. 4a, and a test data set is absent of 0.12% data in Fig. 4b. In addition, Fig. 5 shows the number of failed performance analyses per engine deck for both the training and test data sets; the maximum numbers of failed performance analyses for the training and test data sets are five and six out of 924 performance analyses, respectively. After all, despite a relatively minute percentage of data deficiency, POD is incapable of dealing with the snapshot ensembles of engine deck responses containing missing data; hence, the EM-PCA is required to extract empirical orthogonal bases from the training data set.

C. Implementation of a Reduced-Order Numerical Propulsion System Simulation Model

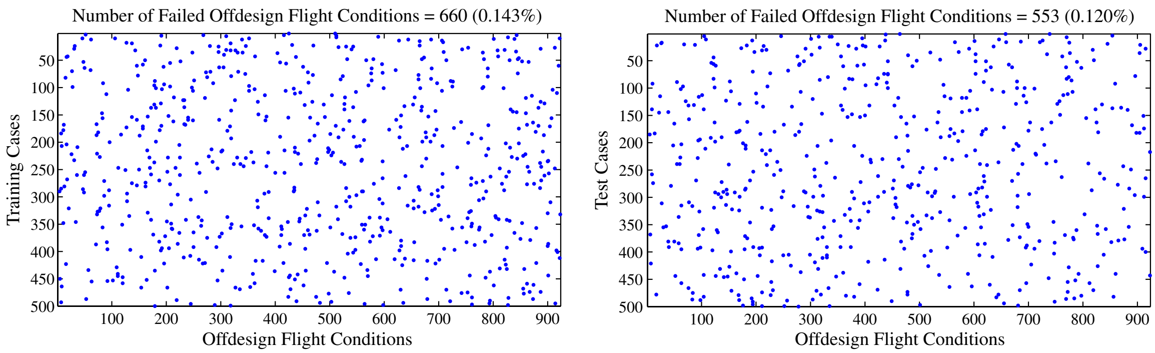
1. Expectation-Maximization Principal Component Analysis for Basis Extraction

The EM-PCA in Eq. (9) is implemented in MATLAB with two algorithmic variations: 1) whether to evaluate a sample mean and update a mean-centered snapshot ensemble at each iteration, and 2) how to initialize a nonorthogonal basis $\mathbf{W}$ before the onset of iterations. For notational convenience, the former is denoted by "$\boldsymbol{\mu}$ inv."/"$\boldsymbol{\mu}$ var." and the latter by "`rand`"/"$\mathbf{V}_e$" in the names of EM-PCA implementations. For instance, $\boldsymbol{\mu}$ inv. indicates that an EM-PCA implementation treats both a sample mean and a mean-centered snapshot ensemble as constant during iterations. In contrast, $\boldsymbol{\mu}$ var. implies the other case, which allows an EM-PCA implementation to vary both a sample mean and a mean-centered snapshot per iteration. Similarly, `rand` shows that an EM-PCA implementation initializes $\mathbf{W}$ with a random matrix, whereas $\mathbf{V}_e$ expresses that an EM-PCA implementation initializes $\mathbf{W}$ with a guessed POD basis $\mathbf{V}_e$ obtained from an estimated snapshot ensemble for which the missing data are filled with a sample mean beforehand to initiate iterations. As a result, a total of four EM-PCA implementations are generated, and a root-mean-square residual (RMSR), such that

$$\mathrm{RMSR}^{(k)} = \sqrt{\frac{1}{dN}\sum_{j=1}^{N}\|\tilde{\mathbf{y}}_j^{(k)} - \tilde{\mathbf{y}}_j^{(k-1)}\|_{L^2}^2} \tag{13}$$

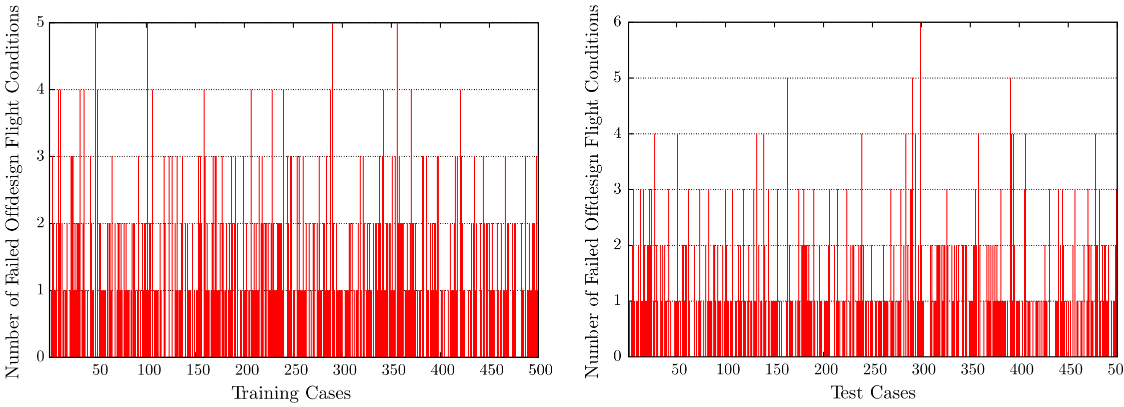
is employed to determine their convergence in terms of a normalized RMSR with respect to the first RMSR value. For a convergence threshold, a normalized RMSR is set to 10^{-6} for all EM-PCA implementations.

For the given snapshot ensembles of engine deck responses, all four EM-PCA implementations yield virtually identical eigenspectra, modes, and restored failed performance analyses. To begin with, Table 3 shows the eigenspectra of engine deck responses, evaluated by the EM-PCA $\boldsymbol{\mu}$ inv.:rand, and these eigenvalues generally indicate relative variations associated with corresponding modes in engine deck responses. Overall, just four modes are found to be suffi-



a) Training data **b) Test data**

Fig. 4 Distributions of failed NPSS offdesign performance analyses.



a) Training data **b) Test data**

Fig. 5 Number of failed NPSS offdesign flight analyses.

Table 3 Normalized eigenspectra of engine deck responses

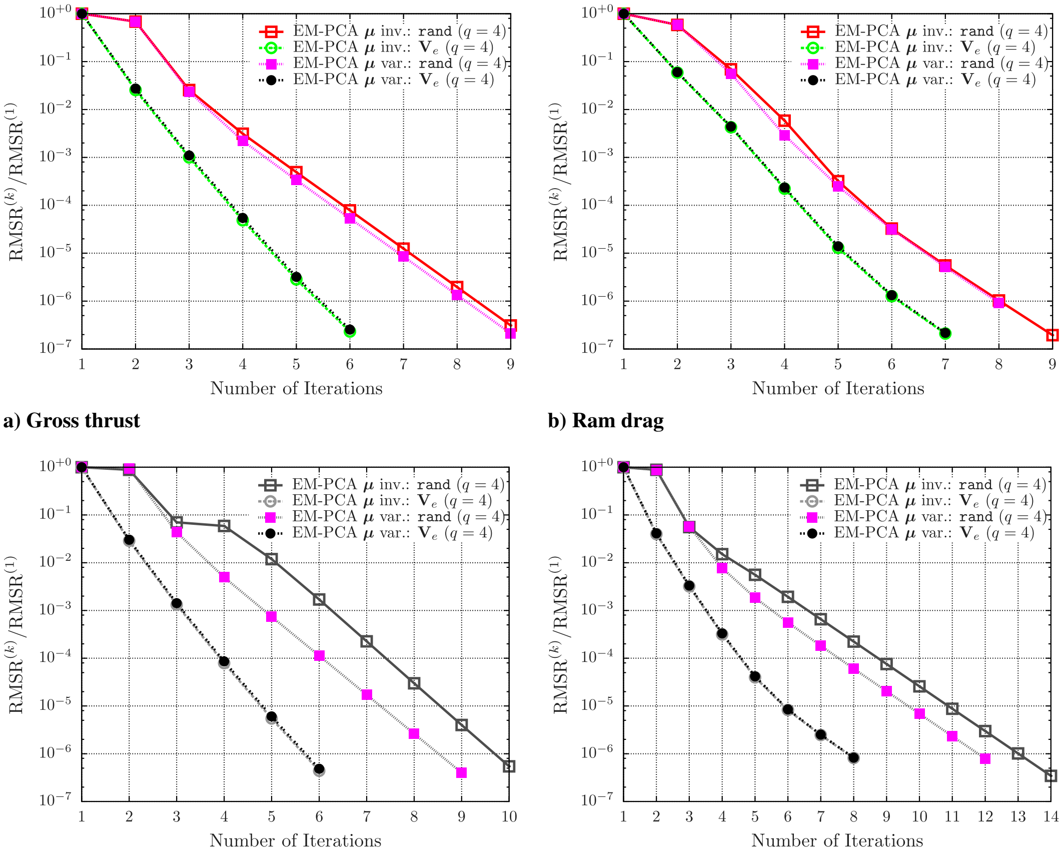
	Gross thrust	Ram drag	Fuel flow	EINO_X
λ_1	9.016077e − 01	9.884888e − 01	9.187537e − 01	9.702735e − 01
λ_2	7.471218e − 02	8.252409e − 03	7.289498e − 02	2.574894e − 02
λ_3	2.179765e − 02	3.052206e − 03	4.010828e − 03	1.988086e − 03
λ_4	1.453996e − 03	1.425448e − 04	3.268362e − 03	1.155463e − 03
$\sum_{j=1}^{4} \lambda_j$	9.995715e − 01	9.999360e − 01	9.989278e − 01	9.991660e − 01

cient enough to account for 99.9% of variations in all four snapshot ensembles of the engine deck responses because they change a little for the ranges of the engine modeling parameters, delineated in Table 2. Therefore, significant dimensionality reduction from 924 to 4 is achieved for the four engine deck responses by virtue of the EM-PCA implementations. After all, this research used four modes to develop a reduced-order model of the four engine deck responses based on the eigenspectrum analysis in Table 3.

Figure 6 delineates the convergence histories of the four EM-PCA implementations, using four modes for the given snapshot ensembles of engine deck responses. In Fig. 6, all EM-PCA implementations quickly reach their convergence thresholds, owing to the insignificant amount of missing data in the snapshot ensembles of the engine deck responses. Note that dissimilar **W** initializations clearly differentiate the convergence behavior of the EM-PCA implementations in Fig. 6; the EM-PCA implementations with $\mathbf{V}_e$ require fewer iterations than those with `rand`, regardless of the $\boldsymbol{\mu}$ var. and $\boldsymbol{\mu}$ inv. implementations. This convergence advantage of the **W** initialization with $\mathbf{V}_e$ mainly results from the low percentage of missing data and the uncomplicated variations of compiled engine deck responses, both of which are conducive to obtaining $\mathbf{V}_e$ close to the true $\mathbf{V}_q$. Unlike the **W** initialization difference in terms of `rand` and $\mathbf{V}_e$, the $\boldsymbol{\mu}$ inv. and $\boldsymbol{\mu}$ var. difference produces almost no influence on the convergence histories of the EM-PCA implementations, which implies that a sample mean fluctuates very little during iterations.

2. *Neural Networks for Coefficient Fitting*

To build coefficient prediction models through neural networks, this research used `newff` in MATLAB, producing feedforward network models consisting of a single hidden layer with 40 neurons. For a neuron activation function, the neural network models adopt a typical tan-sigmoid transfer function in Eq. (10) implemented by `tansig`, and for a backpropagation algorithm, the Levenberg–Marquardt method realized by `trainlm` is invoked with the following stopping rules: the maximum number of epochs is 100 and the convergence threshold is 10^{-6} in terms of a mean square error (MSE). Since the magnitude of weighting coefficients is considerably large because of normalized empirical bases, training processes were mostly terminated by reaching the maximum number of epochs. In the training process, this research randomly chose each 20% of training data for model verification and testing, and then it fitted the final coefficient prediction models from scratch by fully using the training data. Overall, Table 4 presents the quality of weighting coefficient models in terms of the coefficient of determination R^2, and the values of R^2 are almost close to one, which demonstrates fairly exceptional fitting results.



a) Gross thrust

b) Ram drag

c) Fuel flow

d) EINO_X

Fig. 6 Convergence histories of EM-PCA implementations.

Table 4 R^2 of weighting coefficients of engine deck responses for training data

	Gross thrust	Ram drag	Fuel flow	EINO$_X$
α_1	9.999979e − 01	9.999985e − 01	9.999984e − 01	9.999989e − 01
α_2	9.999865e − 01	9.999432e − 01	9.999871e − 01	9.999603e − 01
α_3	9.999564e − 01	9.998302e − 01	9.998875e − 01	9.998023e − 01
α_4	9.986888e − 01	9.952442e − 01	9.995231e − 01	9.990423e − 01

IV. Verification of Reduced-Order Numerical Propulsion System Simulation Model

A. Goodness-of-Fit Analysis

1. *Model-Fit-Error Test*

As the first step toward verifying the previously developed reduced-order NPSS model, this research examined the estimation performance of the reduced-order model with the training data. Since the empirical bases of engine deck responses are presumed to be invariant to known training data, the prediction quality of the reduced-order model hinges solely on neural network models for weighting coefficient evaluations. For the quantification of the prediction capability of the reduced-order model, the following three metrics are employed: an R^2, a normalized root square error (NRSE), such that

$$\text{NRSE} = \sqrt{\left\| \frac{n_{ij}y_{ij} - n_{ij}\tilde{y}_{ij}}{n_{ij}y_{ij}} \right\|_{L^2}^2} \tag{14}$$

and a normalized root MSE (NRMSE), defined by

$$\text{NRMSE} = \sqrt{\frac{1}{\sum_{i=1}^{d} n_{ij}} \sum_{j=1}^{N} \left\| \frac{\mathbf{n}_j \circ \mathbf{y}_j - \mathbf{n}_j \circ \tilde{\mathbf{y}}_j}{\mathbf{n}_j \circ \mathbf{y}_j} \right\|_{L^2}^2} \tag{15}$$

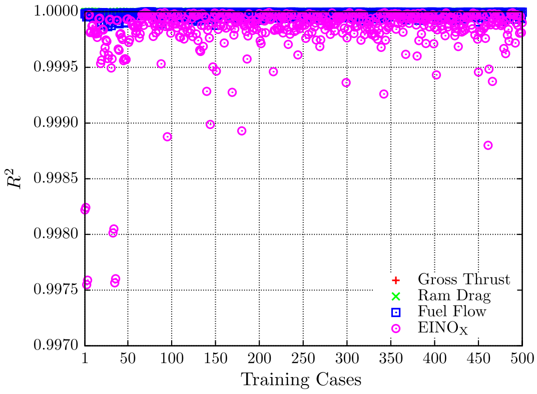


Fig. 7 R^2 for training data.

In Eqs. (14) and (15), ∘ denotes pointwise multiplication, and $\mathbf{n}_j \in \mathbb{R}^d$ indicates the availability of the elements of $\mathbf{y}_j$ as follows:

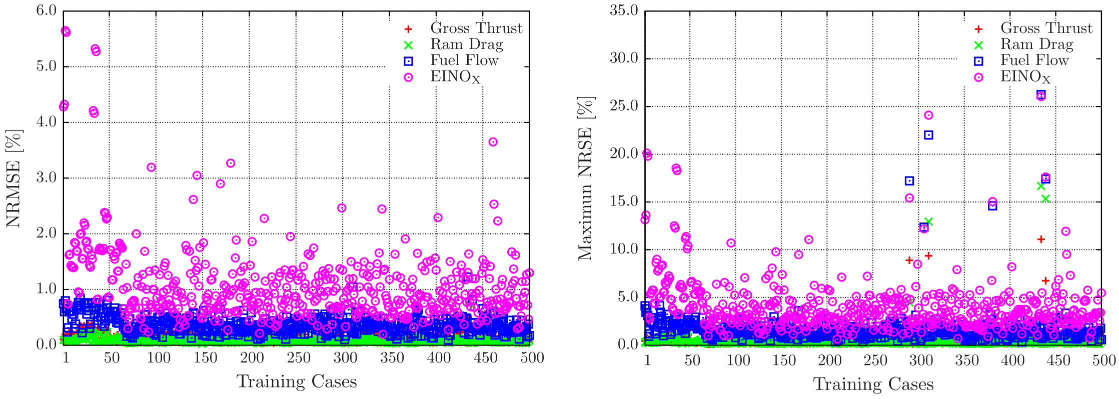
$$n_{ij} = \begin{cases} 0 & \text{if } y_{ij} \text{ is missing,} \\ 1 & \text{if } y_{ij} \text{ is known,} \end{cases} \quad \text{for } i = 1, \ldots, d$$

Because of occasionally absent engine deck responses in failed performance analyses, both NRSE and NRMSE require $\mathbf{n}_j$ in their evaluations to ignore unavailable data. Note that NRSE and NRMSE are normalized with respect to a true value, implying a relative error of a predicted value compared with the true value. In view of error analysis, an R^2 represents the overall fitness of predicted values to exact values; however, both NRSE and NRMSE quantify the deviations of predicted values from true values. Specifically, an NRSE measures the relative difference between a predicted and an exact value, and an NRMSE is the average of NRSEs for an entire engine deck response.

For the training data, R^2 values are at least over 0.9975 in Fig. 7, demonstrating the exceptional prediction capability of the reduced-order NPSS model. In Fig. 8, NRMSEs are mostly less than 2% and increase to around 6% in Fig. 8a; similarly, maximum NRSEs are mostly less than 5% and rise up to around 26% in Fig. 8b. Although Fig. 8b contains some training cases exhibiting large maximum NRSEs, these maximum NRSEs occur only in a single particular offdesign flight condition among the 924 offdesign flight conditions. Therefore, certain training cases with high maximum NRSEs in Fig. 8b do not necessarily entail worst-prediction results in terms of R^2 and NRMSE. For instance, although the highest NRSE for EINOx is 26.05% in the 434th training case in Fig. 8b, the overall prediction quality of the case is not inferior at all, as indicated by its R^2 and NRMSE, 0.9999162 and 1.31%, respectively. Moreover, relatively high maximum NRSEs in Fig. 8b are mostly caused by the numerical instability of NPSS, which will be discussed in detail later in Sec. IV.A.3. Overall, the predicted accuracy of the four engine deck responses, from most to least accurate, were gross thrust, ram drag, fuel flow, and EINOx. Note that one could exactly reproduce engine deck responses in the training data with available optimal least-squares coefficients instead of using estimated weighting coefficients via the neural network models.

2. *Quality of Empirical Bases and Coefficients*

To delve into the source of prediction errors, this research examined the quality of both the empirical bases and the weighting



a) NRMSE

b) Maximum NRSE

Fig. 8 NRMSE and maximum NRSE for training data.

Table 5 NRMSE of bases of engine deck responses between training and test data

	Gross thrust	Ram drag	Fuel flow	EINO$_X$
$\mathbf{v}_1$	3.980957e − 03	1.911276e − 03	1.569202e − 03	3.438867e − 03
$\mathbf{v}_2$	1.335774e − 02	1.261081e − 02	1.608686e − 02	1.797702e − 02
$\mathbf{v}_3$	2.060794e − 02	2.217635e − 02	1.591716e − 01	8.656342e − 02
$\mathbf{v}_4$	2.969393e − 02	3.934067e − 02	8.206984e − 02	4.129359e − 02

Table 6 R^2 of coefficients of engine deck responses for test data

	Gross thrust	Ram drag	Fuel flow	EINO$_X$
α_1	9.999939e − 01	9.999968e − 01	9.999952e − 01	9.999953e − 01
α_2	9.995829e − 01	9.991308e − 01	9.998484e − 01	9.972685e − 01
α_3	9.993541e − 01	9.967606e − 01	8.574791e − 01	9.617943e − 01
α_4	9.308675e − 01	9.102620e − 01	7.595319e − 01	8.369770e − 01

coefficient models with the random test data. First, with regard to the empirical bases, this research measured the differences between the empirical bases of the training data and those of the test data in terms of an NRMSE. Table 5 shows that the changes in the empirical bases with respect to the two disparate data sets are insignificant, and the order of magnitude is quite similar across the four engine deck responses; the most dominant first modes exhibit variations one order smaller than the other three subordinate modes. Note that the empirical bases of the test data do not necessarily represent the true bases of the engine deck responses, because the test data are randomly sampled over the space of the engine modeling parameters. Overall, Table 5 sufficiently corroborates the assumed invariancy of the empirical bases for the sample data space of the engine modeling parameters.

Regarding the validity of the weighting coefficient models, this research assessed the fitness of the coefficient prediction models with respect to the random test data in terms of an R^2. As shown in Table 6, the weighting coefficient models of gross thrust and ram drag are considerably reliable, since their R^2 values are over 0.9 for all four of their coefficients. Unlike the weighting coefficient models of gross thrust and ram drag, not all coefficient models have high R^2 values in the cases of fuel flow and EINOx; for fuel flow, the R^2 values of the third and fourth coefficients are relatively low, and so is the fourth coefficient for EINOx. In Sec. IV.A.3, the effects of these low R^2-valued minor coefficients will be discussed in connection with the prediction results of fuel flow and EINOx.

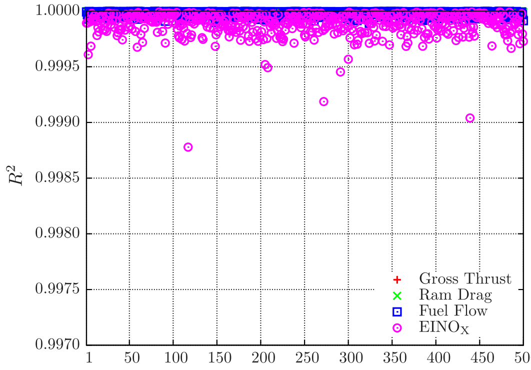
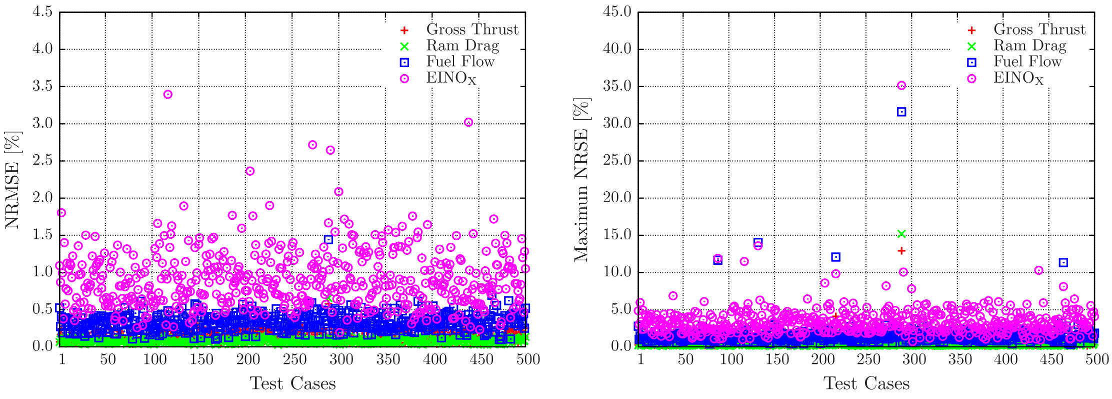


Fig. 9 R^2 for random test data.

3. Model Prediction Error Test

As with the previous prediction quality investigation in Sec. IV.A.1, the same examination process was repeated for 500 randomly populated snapshots of engine deck responses not used for creating the reduced-order NPSS model. Since both the empirical bases and the weighting coefficient models are found to be considerably reliable in Sec. IV.A.2, the reduced-order model yields superb prediction results in Figs. 9 and 10 despite its ignorance about the random test data.

For instance, Fig. 9 depicts that R^2 values are at least over 0.9985. Figure 10 delineates that NRMSEs are overall less than 2% and increase to around 3.5% in Fig. 10a; likewise, maximum NRSEs are mostly less than 5% and skyrocket to around 35% in the 289th test case in Fig. 10b. Again, despite the highest NRSE of EINOx (i.e., 35.16% in the 289th test case), corresponding R^2 and NRMSE values are satisfactory such that 0.9999305 and 1.67%, respectively. Similar to the prediction test results for the training data in Figs. 7 and 8, both gross thrust and ram drag show prediction accuracy superior to the other engine deck responses, namely, fuel flow and EINOx. Although the minor coefficients of fuel flow and EINOx had relatively low R^2 values in Sec. IV.A.2, the predicted fuel flow and EINOx are quite acceptable, as delineated in Figs. 9 and 10. Overall,



a) NRMSE

b) Maximum NRSE

Fig. 10 NRMSE and maximum NRSE for random test data.

Table 7 Comparison of NRSEs at a Mach number of 0.40 and an altitude of 20,000 ft

	Throttle setting	Gross thrust, lbf	Ram drag, lbf	Fuel flow, lbm/h	EINO$_X$, g/kg
	24	15,583.1	11,590.4	2,083.5	7.5379
Old	22	14,618.6	12,337.1	2,292.1	8.5443
	21	12,670.7	11,242.3	1,795.8	6.5093
	24	15,582.9	11,590.4	2,083.0	7.5638
New	22	12,750.4	10,458.5	1,557.5	5.4649
	21	11,250.7	9,809.7	1,287.1	4.4680
	24	15,563.4	11,598.5	2,092.4	7.6300
Predicted	22	12,730.9	10,463.5	1,567.6	5.5400
	21	11,231.5	9,811.6	1,300.4	4.5400
Old NRSE		12.91273%	15.18704%	31.61031%	35.16145%
New NRSE		0.152936%	0.047808%	0.648475%	1.374225%

Table 8 Mission rules for performance analysis of a Boeing 777-like aircraft model

Mission rules	Analysis
Block	5 min. taxi out; 1 min. takeoff; typical climb schedule; 35,000–39,000–41,000 ft step cruise at Mach 0.84; max lift-to-drag ratio descent; 4 min. approach
Reserve	5% contingency fuel; 2 min. missed approach; 200 n mile divert; 30 min. hold at 15,000 ft

the prediction test results for the random test data in Figs. 9 and 10 do not significantly differ from those for the training data in Figs. 7 and 8. Therefore, comprehensive prediction investigations with both the training and test data sets substantiate that the empirical bases as well as the weighting coefficient models are quite dependable, resulting in a credible reduced-order NPSS model for the given ranges of engine modeling parameters.

As previously illustrated in Figs. 8b and 10b, the maximum NRSEs in certain test cases are unacceptably high for all the engine deck responses. To further investigate this irregular behavior of engine deck responses, this research reanalyzed the 289th test case, which exhibited the largest maximum NRSE in Fig. 10b. In Table 7, the formerly obtained engine deck responses are compared with those newly achieved around throttle value 22, at which the largest maximum NRSE occurred. Despite decreasing throttle values, NPSS originally resulted in engine deck responses exhibiting unusual increases at throttle value 22. The unrealistic engine deck responses at throttle value 22 arise when the NPSS produces an unconverged, thereby physically invalid, solution as a numerical artifact. After the Newton–Raphson solver within NPSS was adjusted so that it could generate converged solutions, the new NPSS analysis yielded engine deck responses showing concomitant decreases as the throttle value decreases. Numerically, these inconsistent NPSS results stem from the Newton–Raphson solver of NPSS for solving the equations of conservation of mass and energy in the process of determining engine performance in a given flight condition. Within NPSS, the solution of a flight condition was used as an initial guess for the subsequent flight condition. Should a flight condition fail to converge, then a solution for the maximum throttle at a given Mach number and altitude is substituted as a guess for the subsequent flight condition. Therefore, atypical engine performance data sometimes appear, especially at lower throttle settings. Thanks to the new converged NPSS results, the predicted engine deck responses obtained by the reduced-order NPSS model are now remarkably close to the newly evaluated engine deck responses. Although deviant NPSS results do deteriorate the basis evaluation of engine deck responses, their effects on the empirical bases are minuscule, since the test data set contains few outlying cases. After all, maximum NRSE values drastically drop, as delineated at the bottom of Table 7; for example, in the case of EINO$_X$, the maximum NRSE of 35.16145% is now 1.374225%.

B. Model Error Propagation into Aircraft Mission Performance Analysis

As mentioned in Sec. I, the primary purpose of creating a reduced-order engine model is to facilitate the integration of a high-fidelity propulsion simulation with aircraft systems design and analysis. Therefore, it is also crucial to assess how errors of the reduced-order propulsion model observed in terms of engine performance metrics in Sec. IV.A propagate into aircraft mission performance metrics. To this end, this research performed two batches of aircraft mission performance analyses with FLOPS [26], a NASA-developed tool for aircraft sizing and performance analysis, by using two different types of engine decks: 1) those directly generated by NPSS and 2) those indirectly produced by a reduced-order NPSS model. As illustrated in Fig. 11, this research employed a Boeing 777-like large twin-aisle vehicle model and examined block fuel and total NO$_X$ emissions as two examples of aircraft performance metrics.

To investigate NPSS ROM errors propagated into an aircraft system level, this research evaluated differentials between the exact and estimated values of block fuel and total NO$_X$ emissions. For the simplicity of analysis, the FLOPS aircraft model was treated such that its geometry and zero fuel weight were constant. Since operating empty weight was fixed to 299,939 lb, zero fuel weight was computed as 363,149 lb for both design and economic missions. As to payload, the aircraft model was set to carry a total of 301 passengers, each of whose weight is assumed to be 210 lb, including 30 lb of baggage. Regarding the mission analysis, the design and economic missions of the aircraft model were defined as follows: 1) 8048 n mile range of a design mission with full cabin loading, and 2) 3914 n mile range of an economic mission with the same payload. Table 8 gives more details about requirements for block and reserve fuel.

As a result of FLOPS performance analyses, the effects of a reduced-order NPSS model on aircraft performance analysis were measured as depicted in Fig. 12. The evaluation of relative errors compiles the results of 500 FLOPS runs with 500 engine decks generated based on the 500 random engine parameter settings, which were used for the model prediction error test in Sec. IV.A.3. Note that FLOPS mission analysis failed for certain combinations of engine parameters at which an engine model cannot provide sufficient thrust

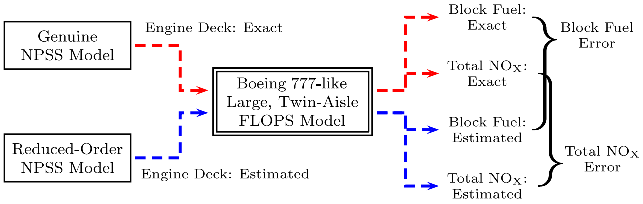
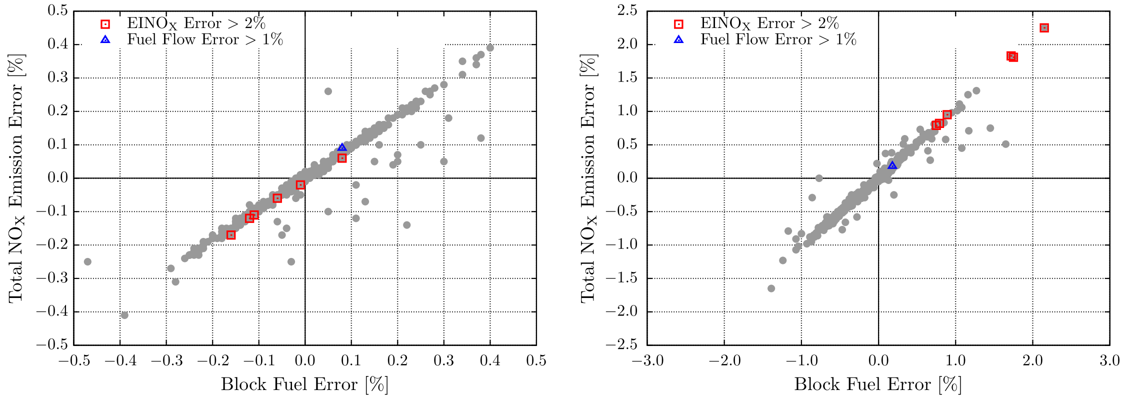


Fig. 11 Evaluation of reduced-order-model error propagated into an aircraft system level.



a) Design mission **b) Economic mission**

Fig. 12 Relative errors of block fuel and total NO$_X$ emissions due to reduced-order NPSS model.

for climb. To begin with, the relative errors observed at the design mission in Fig. 12a are less than 0.5%, conveying that errors of the reduced-order NPSS model result in insignificant impact on aircraft performance metrics. Similarly, the relative errors at the economic mission in Fig. 12b are mostly less than 2.0%; although the relative errors in Fig. 12b are greater than those in Fig. 12a, they still substantiate that the degradation of aircraft performance metrics due to the use of the reduced-order NPSS model is inconsequential.

Furthermore, to easily discern the propagation of engine model errors caused by NPSS ROM to aircraft system performance, this research examined particular cases that exhibited large errors in engine performance metrics, i.e., EINOx and engine fuel flow. In Fig. 12, a square indicates an engine parameter setting resulting in an error more than 2% for EINOx and, likewise, a triangle denotes an engine parameter setting yielding an error more than 1% for engine fuel flow. The circles in Fig. 12 correspond to engine parameter settings for which the errors of EINOx and engine fuel flow are less than 2 and 1%, respectively. Note that the cases of engine parameters, producing large errors in engine performance, do not necessarily lead to considerable errors in aircraft performance. For instance, for the cases in which a reduced-order NPSS model generated the largest error in engine fuel flow, denoted as triangles in both Figs. 12a and 12b, it turned out that they actually resulted in less errors in block fuel and NO$_X$ emissions than other cases. By contrast, large EINOx error-producing cases, marked with squares, show dissimilar error patterns, depending on mission profiles; those cases tended to yield more errors in total NO$_X$ emissions than other cases for the economic mission, but they did not yield more errors in total NO$_X$ emissions than other cases for the design mission.

V. Conclusions

To facilitate the use of high-fidelity propulsion system simulations in aircraft design, this research developed a reduced-order model for NPSS by adopting a POD-based ROM approach with the help of the EM-PCA and neural networks. Because of inevitable data absence observed in the results of NPSS for failed analyses, the EM-PCA is indispensable to achieving POD bases from NPSS engine deck responses. In addition, given the large number of engine modeling parameters in engine design, neural network models are effective at evaluating weighting coefficients in accordance with changes in the engine modeling parameters. As a demonstration, this research created a reduced-order NPSS model for a two-spool separate flow turbofan in order to relate six engine modeling parameters with four engine deck responses for which the dimensionality was 924. For the construction of the reduced-order model, a total of 500 sample data were populated as training data through a maximum-entropy design augmented with corner points, and the same number of data were randomly generated as test data.

By virtue of the EM-PCA and single-layer feedforward networks, the reduced-order NPSS model built upon the first four dominant modes showed considerably low model-fit errors in terms of an NRMSE and an R^2. Moreover, for the given random test data, the reduced-order model generated engine deck responses as accurate and reliable as they were for the training data. The reduced-order NPSS model yielded substantially dependable engine deck responses in general, and the prediction qualities of gross thrust and ram drag were relatively higher than those of fuel flow and EINOx due to their superior weighting coefficient models. Finally, this research compared aircraft performance metrics obtained with FLOPS using engine decks evaluated both directly by NPSS and indirectly by its reduced-order model. Because of the superb prediction accuracy of the reduced-order model, the relative errors of aircraft performance metrics, such as block fuel and total NO$_X$ emissions, are mostly less than 2.5%, encouraging one to capitalize on a reduced-order NPSS model for computationally intensive preliminary design studies.

Since the reduced-order NPSS model was prone to prediction errors, mostly at low altitudes, the following two research tasks are recommended for future work. First, a domain decomposition approach could result in a locally refined reduced-order model that separately accounts for the different degrees of variations in engine deck responses with respect to altitude. Second, a weighted least-squares method could provide biased weighting coefficients such that they reduce more prediction errors at low altitudes than those at high altitudes. Furthermore, the synergy of these two proposed research ideas could also be beneficial to enhance the accuracy of the reduced-order NPSS model. Last, with regard to the irregular increases of engine deck responses at low throttle settings, discussed in Sec. IV.A.3, robust PPCA [35] could be useful for mitigating the outlier effects on empirical basis evaluations.

Appendix

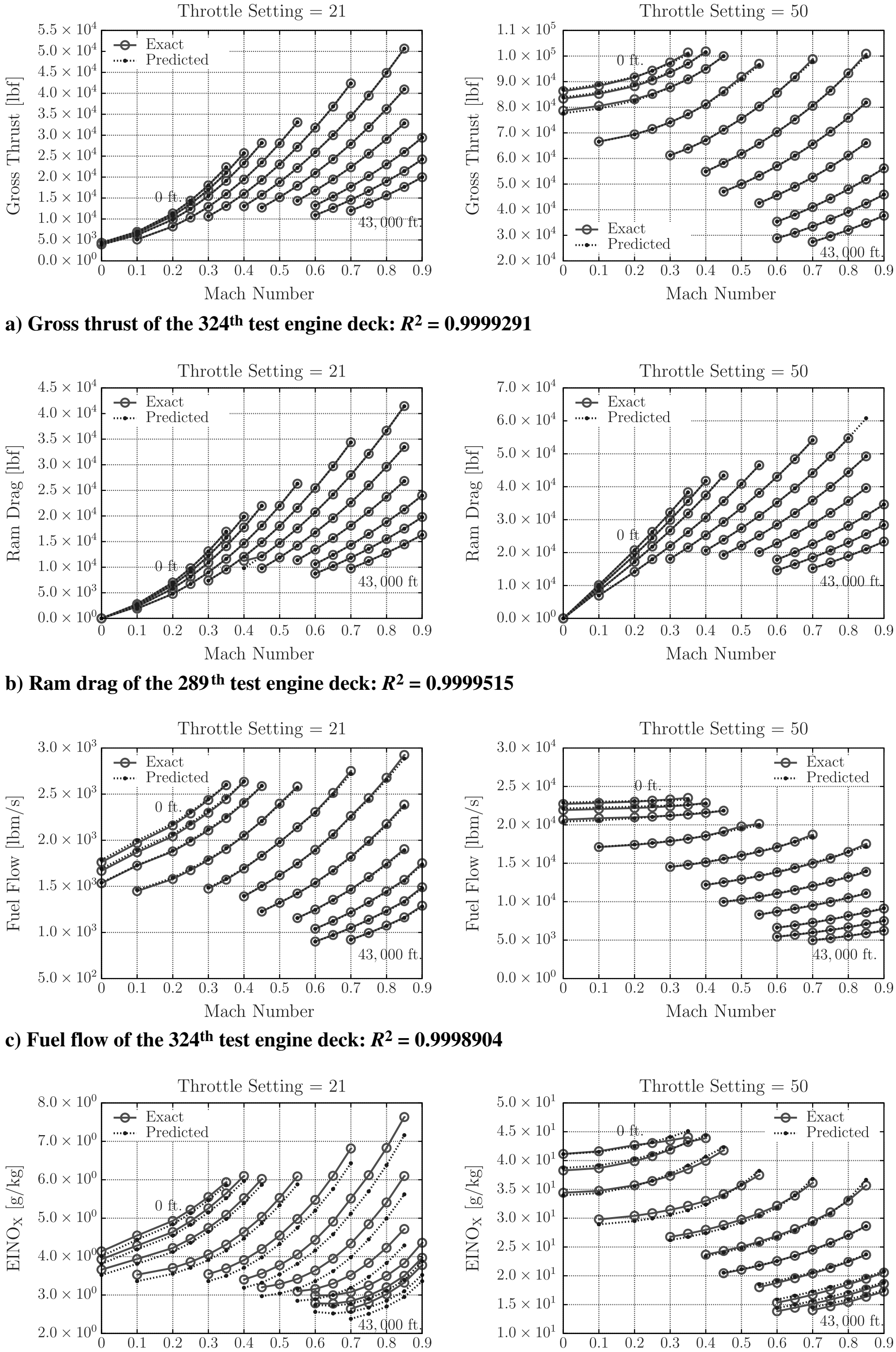
I. Worst Prediction Results of Engine Deck Responses

As an illustration of predicted engine deck responses, the worst-prediction results at the minimum and maximum throttle settings are presented in Figs. A1 and A2 in terms of an R^2 and a maximum NRSE, respectively. Since even the lowest R^2 values are around 0.999, estimated engine deck responses align well with the true values in Fig. A1 and, likewise, despite the highest NRSEs being around 35.16% at most, predicted engine deck responses containing maximum NRSEs show exceptional agreement with the exact values in Fig. A2 because of their considerably high R^2 values. For a detailed view of the regions where the highest NRSE occurs, Fig. A3 provides zoomed-in plots of Fig. A2.

II. Comparison of Results of Gappy Proper Orthogonal Decomposition and Expectation-Maximization–Principal Component Analysis

Although the verification of the EM-PCA against gappy POD is not of primary interest, this research compared the results of the EM-PCA with those of gappy POD to justify the use of the EM-PCA over gappy POD for computational efficiency. As similar to the notational convention for the EM-PCA implementations in Sec. III.C.1, gappy POD implementations (GPOD) are denoted with $\boldsymbol{\mu}$ inv./$\boldsymbol{\mu}$ var. to represent whether a sample mean is evaluated at each iteration. In addition, for only computational performance investigation, “Lanczos” is appended to the names of gappy POD implementations if the Lanczos algorithm is exploited to expedite a POD process. The Lanczos algorithm is realized in MATLAB with `eigs`, which invokes the Fortran library ARPACK [36].

First, the eigenvalues, eigenvectors, and restored failed performance analyses evaluated by the EM-PCA are compared with those evaluated by gappy POD in Figs. A4–A6, respectively. The eigenspectra of the four engine deck responses in Fig. A4 show that both the EM-PCA and gappy POD result in identical eigenvalues at the same number of modes. Similarly, the first modes of the four engine deck responses, illustrated in Fig. A5, also substantiate that the EM-PCA is capable of yielding the same POD bases as gappy POD. Last, as an example of restored failed performance analyses, Fig. A6 delineates the 101th training engine deck, which failed in a total of five offdesign performance analyses. In Fig. A6, all the estimated performance analyses are perfectly aligned, regardless of the EM-PCA and gappy POD implementations. Note that failed analyses located at the lowest throttle value are properly approximated as they follow the overall trend of engine deck responses, because a POD basis can handle stationary discontinuities. Also note that it is hard to



a) Gross thrust of the 324th test engine deck: R^2 = 0.9999291

b) Ram drag of the 289th test engine deck: R^2 = 0.9999515

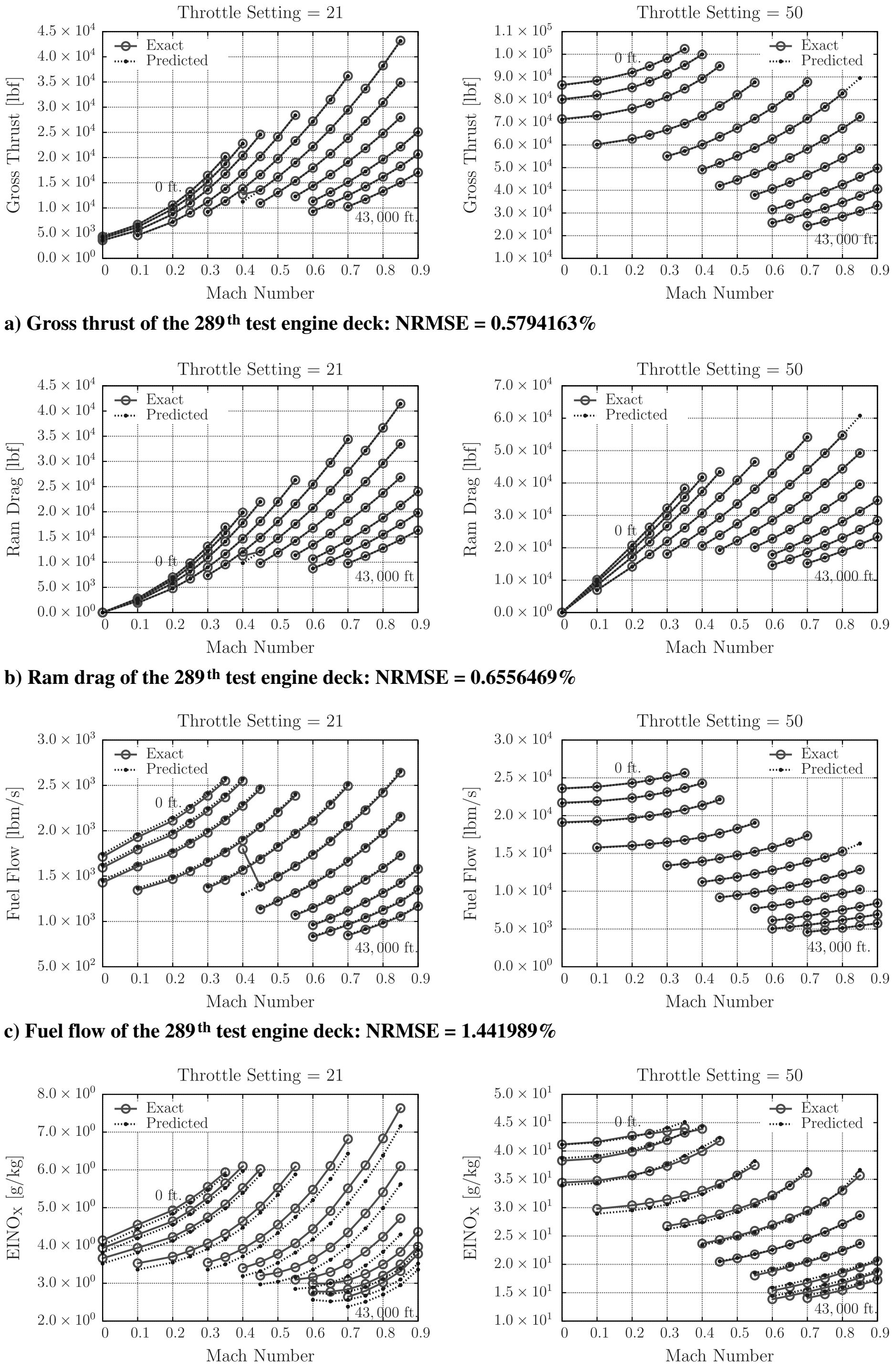
c) Fuel flow of the 324th test engine deck: R^2 = 0.9998904

d) EINO$_X$ of the 117th test engine deck: R^2 = 0.9987783

Fig. A1 Actual and predicted engine deck responses with Mach number: worst R^2.

accurately recover missing engine performance data with simple interpolation techniques if the data occur at extreme throttle values. For example, an absent engine response at the lowest throttle value 21 estimated by the average of engine responses at throttle values 50 and 22 is higher than the engine response at throttle value 22, which is physically invalid.

Last, but not least, Fig. A7 and Table A1 present the results of computational performance tests measured in terms of computational time and the number of iterations for the four different snapshot ensembles of engine deck responses. For the minimal effect of random initialization on time measurements, the computational time of the EM-PCA implementations with `rand` was averaged over 100 runs. As shown in Fig. A7, despite their higher numbers of iterations, the EM-PCA implementations are computationally more efficient than the gappy POD implementations. The gappy POD implementations spent most of their computational time on basis evaluations, since the size of a snapshot ensemble cubically affects the POD process. Although the Lanczos algorithm is particularly conducive to accelerating basis evaluations (e.g., a situation in which the number of modes to extract is small, such as four), it was not effective enough for gappy POD implementations to outperform the EM-PCA implementations. Note that $\mathbf{V}_e$ for the EM-PCA implementations was also evaluated with the Lanczos algorithm for computational efficiency due to the large number of snapshots: 500. After all, even though the



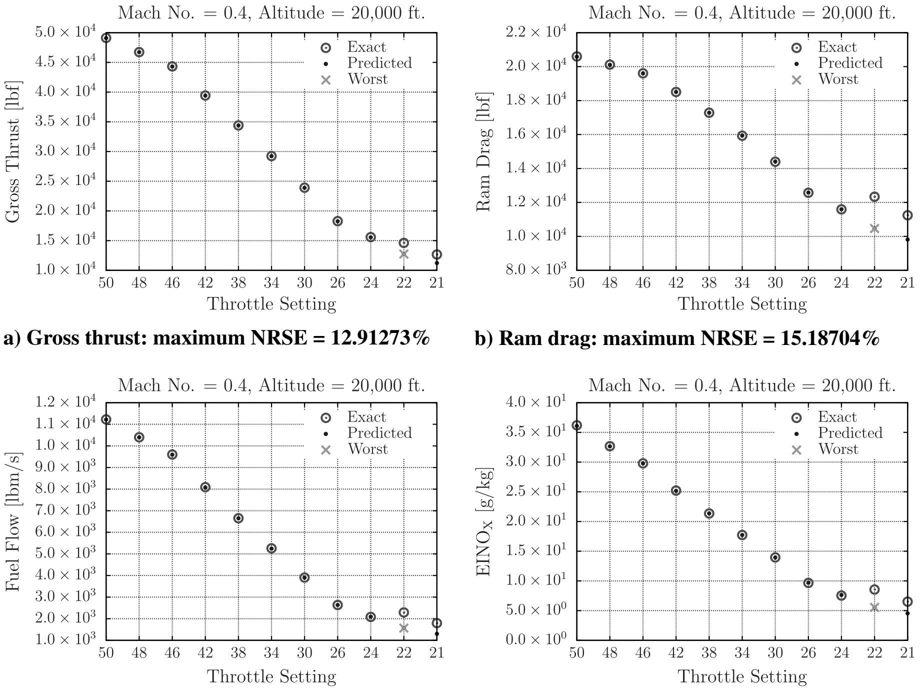
a) Gross thrust of the 289th test engine deck: NRMSE = 0.5794163%

b) Ram drag of the 289th test engine deck: NRMSE = 0.6556469%

c) Fuel flow of the 289th test engine deck: NRMSE = 1.441989%

d) EINO$_X$ of the 117th test engine deck: NRMSE = 3.396025%

Fig. A2 Actual and predicted engine deck responses with Mach number: worst NRMSE.



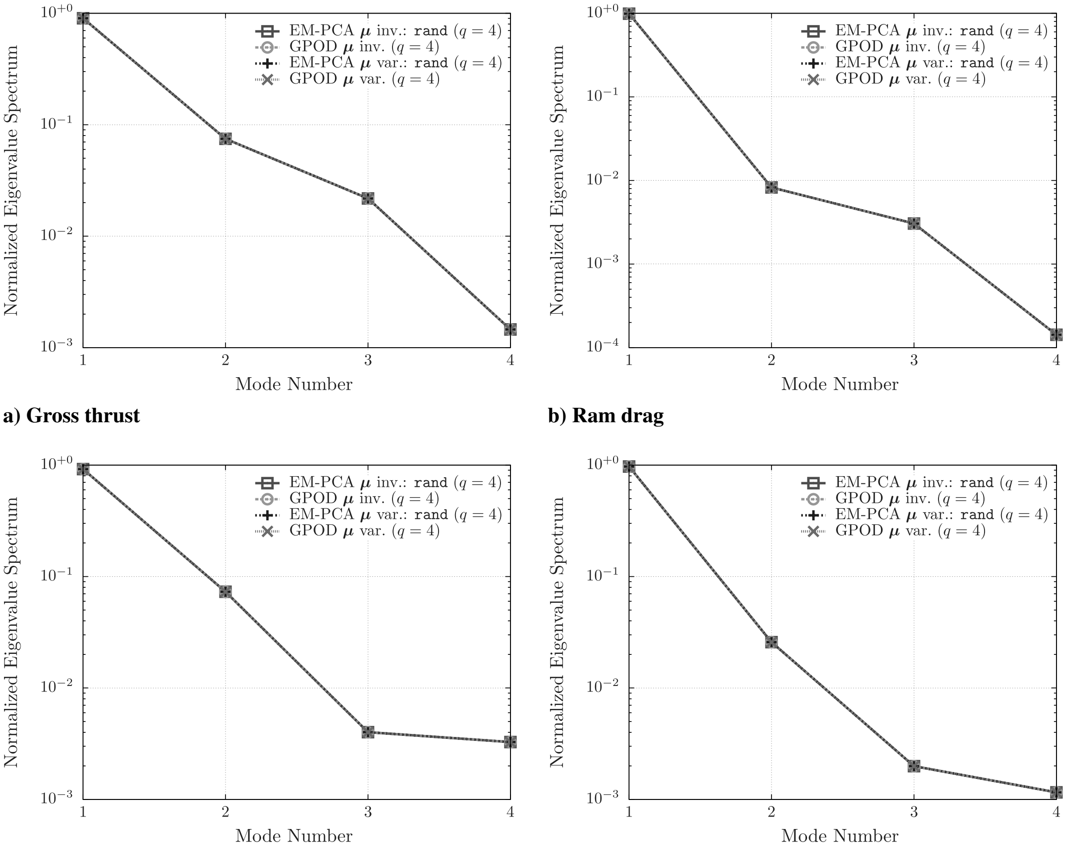
a) Gross thrust: maximum NRSE = 12.91273%

b) Ram drag: maximum NRSE = 15.18704%

c) Fuel flow: maximum NRSE = 31.61031%

d) $\mathbf{EINO_X}$: maximum NRSE = 35.16145%

Fig. A3 Zoomed-in actual and predicted engine deck responses: maximum NRSE.



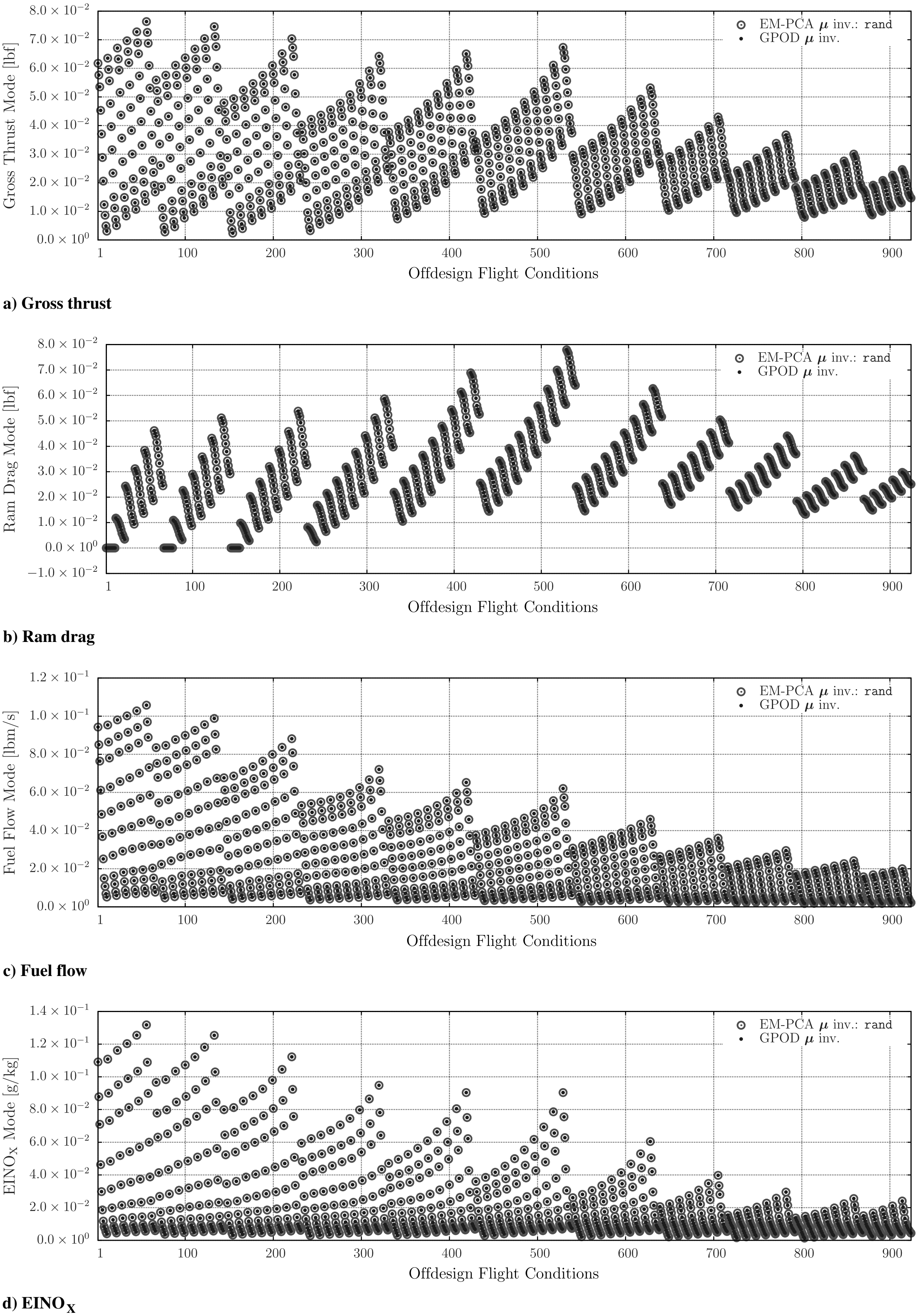
a) Gross thrust

b) Ram drag

c) Fuel flow

d) $\mathbf{EINO_X}$

Fig. A4 Comparison of normalized eigenspectra of engine deck responses.



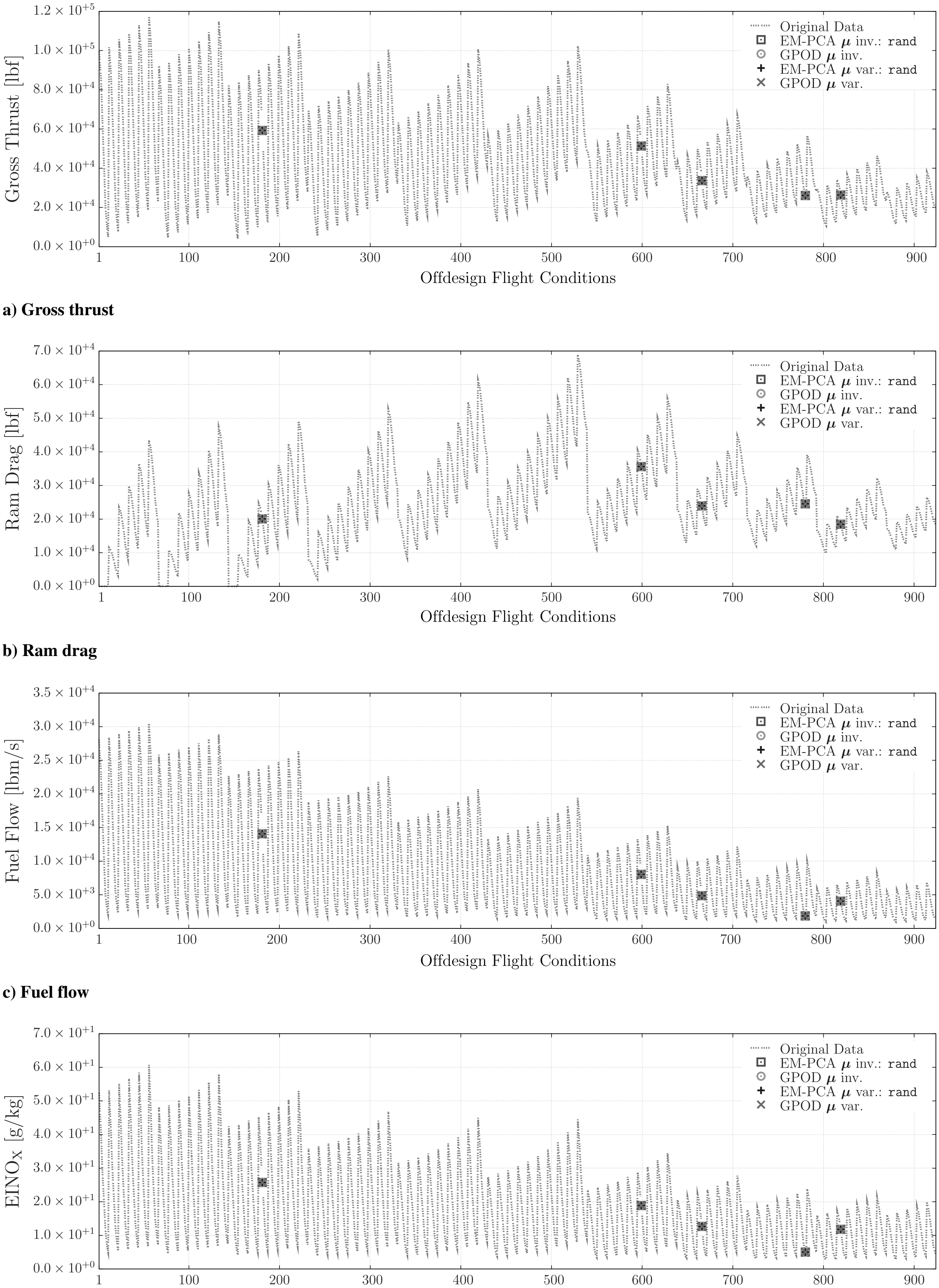
a) Gross thrust

b) Ram drag

c) Fuel flow

d) EINO$_X$

Fig. A5 Comparison of the first modes of the engine deck responses.



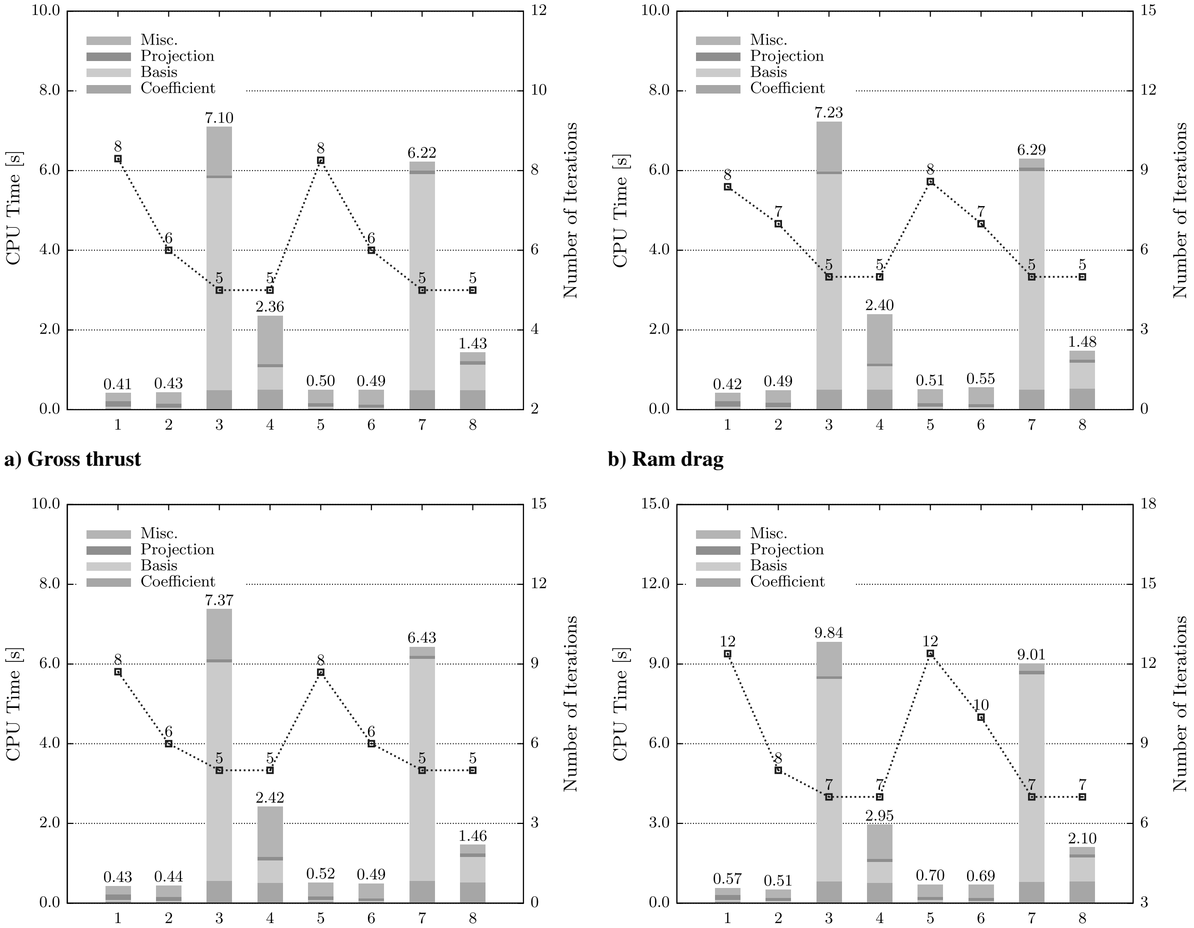
a) Gross thrust

b) Ram drag

c) Fuel flow

d) EINO$_X$

Fig. A6 Comparison of restored engine deck responses: 101th training engine deck.



a) Gross thrust

b) Ram drag

c) Fuel flow

d) EINO$_X$

Fig. A7 Computational time decomposition along with numbers of iterations.

Table A1 Computational performance test parameters ($q = 4$)

	Implementation	Algorithmic variations
1	EM-PCA	$\boldsymbol{\mu}$ inv.: rand
2	EM-PCA	$\boldsymbol{\mu}$ inv.: V_e
3	GPOD	$\boldsymbol{\mu}$ inv.
4	GPOD	$\boldsymbol{\mu}$ inv.: Lanczos
5	EM-PCA	$\boldsymbol{\mu}$ var.: rand
6	EM-PCA	$\boldsymbol{\mu}$ var.: V_e
7	GPOD	$\boldsymbol{\mu}$ var.
8	GPOD	$\boldsymbol{\mu}$ var.: Lanczos

EM-PCA necessitates an extra step for the orthogonalization of its nonorthogonal basis, it generally surpasses gappy POD for the given snapshots of engine deck responses, characterized by the absence of randomly scattered data.

References

[1] Lavelle, T. M., Plencner, R. M., and Seidel, J. A., "Concurrent Optimization of Airframe and Engine Design Parameters," NASA TM 105908, Sept. 1992; also Fourth Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 1992-4713, Sept. 1992.

[2] Drela, M., "Transport Aircraft Optimization," Aerospace Engineering Fall 2009 Seminar, Georgia Inst. of Technology, Atlanta, GA, Dec. 2009.

[3] Lytle, J. K., "The Numerical Propulsion System Simulation: An Overview," NASA TM 2000-209915, June 2000.

[4] Jones, S. M., "An Introduction to Thermodynamic Performance Analysis of Aircraft Gas Turbine Engine Cycles Using the Numerical Propulsion System Simulation Code," NASA TM 2007-214690, March 2007.

[5] Follen, G., and auBuchon, M., "Numerical Zooming Between a NPSS Engine System Simulation and a One-Dimensional High Compressor Analysis Code," NASA TM 2000-209913, April 2000; also *Computational Aerosciences Workshop*, High Performance Computing and Communications Program, Moffett Field, CA, Feb. 2000.

[6] Veres, J. P., "Overview of High-Fidelity Modeling Activities in the Numerical Propulsion System Simulations (NPSS) Project," NASA TM 2002-211351, June 2002.

[7] Mavris, D. N., Bandte, O., and DeLaurentis, D. A., "Robust Design Simulation: A Probabilistic Approach to Multidisciplinary Design," *Journal of Aircraft*, Vol. 36, No. 1, Jan.–Feb. 1999, pp. 298–307. doi:10.2514/2.2437

[8] Mavris, D. N., and DeLaurentis, D. A., "A Probabilistic Approach for Examining Aircraft Concept Feasibility and Viability," *Aircraft Design*, Vol. 3, No. 2, June 2000, pp. 79–101. doi:10.1016/S1369-8869(00)00008-2

[9] Yoder, T., "Development of Aircraft Fuel Burn Modeling Techniques with Applications to Global Emissions Modeling and Assessment of the Benefits of Reduced Vertical Separation Minimums," M.S. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, May 2007.

[10] Senzig, D. A., Fleming, G. G., and Iovinelli, R. J., "Modeling of Terminal-Area Airplane Fuel Consumption," *Journal of Aircraft*, Vol. 46, No. 4, July–Aug. 2009, pp. 1089–1093. doi:10.2514/1.42025

[11] Jolliffe, I., *Principal Component Analysis*, 2nd ed., Springer Series in Statistics, Springer, New York, Oct. 2002.

[12] Lucia, D. J., King, P. I., and Beran, P. S., "Domain Decomposition for Reduced-Order Modeling of a Flow with Moving Shocks," *AIAA Journal*, Vol. 40, No. 11, Nov. 2002, pp. 2360–2362. doi:10.2514/2.1576

[13] Everson, R., and Sirovich, L., "Karhunen–Loeve Procedure for Gappy Data," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, Vol. 12, No. 8, Aug. 1995, pp. 1657–1664.

doi:10.1364/JOSAA.12.001657

[14] Tipping, M. E., and Bishop, C. M., "Probabilistic Principal Component Analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 61, No. 3, 1999, pp. 611–622.
doi:10.1111/1467-9868.00196

[15] Bui-Thanh, T., "Proper Orthogonal Decomposition Extensions and Their Applications in Steady Aerodynamics," M.S. Thesis, Department of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, 2003.

[16] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, Aug. 2004, pp. 1505–1516.
doi:10.2514/1.2159

[17] Murray, N. E., and Ukeiley, L. S., "An Application of Gappy POD: For Subsonic Cavity Flow PIV Data," *Experiments in Fluids*, Vol. 42, No. 1, 2007, pp. 79–91.
doi:10.1007/s00348-006-0221-y

[18] Lee, K., Rallabhandi, S. K., and Mavris, D. N., "Aerodynamic Data Reconstruction via Probabilistic Principal Component Analysis," 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2008-899, Jan. 2008.

[19] Lee, K., and Mavris, D. N., "Unifying Perspective for Gappy Proper Orthogonal Decomposition and Probabilistic Principal Component Analysis," *AIAA Journal*, Vol. 48, No. 6, June 2010, pp. 1117–1129.
doi:10.2514/1.45750

[20] Lee, K., "Investigation of Probabilistic Principal Component Analysis Compared to Proper Orthogonal Decomposition Methods for Basis Extraction and Missing Data Estimation," Ph.D. Thesis, School of Aerospace Engineering, Georgia Inst. of Technology, Atlanta, GA, Aug. 2010.

[21] Alfeld, P., "Scattered data interpolation in three or more variables," *Mathematical Methods in Computer Aided Geometric Design*, edited by T. Lyche, and L. Schumaker, Academic Press, New York, 1989, pp. 1–34.

[22] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics," 21st AIAA Applied Aerodynamics Conference, Orlando, FL, AIAA Paper 2003-4213, June 2003.

[23] Mifsud, M. J., Shaw, S. T., and MacManus, D. G., "A High-Fidelity Low-Cost Aerodynamic Model Using Proper Orthogonal Decomposition," *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 4June 2010, pp. 468–494.
doi:10.1002/fld.2085

[24] Sacks, J., Welch, W., Mitchell, T., and Wynn, H., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, Nov. 1989, pp. 409–423.
doi:10.1214/ss/1177012413

[25] Santner, T. J., Williams, B. J., and Notz, W., *The Design and Analysis of Computer Experiments*, 1st ed., Springer Series in Statistics, Springer, New York, 2003.

[26] McCullers, L., "Flight Optimization System Release 8.11 User's Guide," NASA Langley Research Center, Hampton, VA, Oct. 2009.

[27] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 39, No. 1, 1977, pp. 1–38.

[28] Rubin, D. B., and Thayer, D. T., "EM Algorithms for ML Factor Analysis," *Psychometrika*, Vol. 47, No. 1, March 1982, pp. 69–76.
doi:10.1007/BF02293851

[29] Du, K. L., and Swamy, M. N. S., *Neural Networks in a Softcomputing Framework*, Springer, New York, 2006.

[30] Sjöberg, J., *Mathematica Neural Networks: Train and Analyze Neural Networks to Fit Your Data*, 1st ed., Wolfram Research, Champaign, IL, Sept. 2005.

[31] *Gas Turbine Engine Steady State and Transient Performance Presentation for Digital Computer Programs*, SAE International, Warrendale, PA, 1999.

[32] Schobeiri, M., *Turbomachinery Flow Physics and Dynamic Performance*, Springer, New York, Dec. 2004.

[33] Mattingly, J. D., Heiser, W. H., and Pratt, D. T., *Aircraft Engine Design*, AIAA Education Series, AIAA, Reston, VA, Dec. 2002.

[34] *JMP Design of Experiments*, 6th ed., SAS Institute Inc., Cary, NC, 2005.

[35] Chen, T., Martin, E., and Montague, G., "Robust Probabilistic PCA with Missing Data and Contribution Analysis for Outlier Detection," *Computational Statistics and Data Analysis*, Vol. 53, No. 10, Aug. 2009, pp. 3706–3716.
doi:10.1016/j.csda.2009.03.014

[36] Lehoucq, R. B., Sorensen, D. C., and Yang, C., *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM Publ., Philadelphia, 1998.

T. Zang
Associate Editor