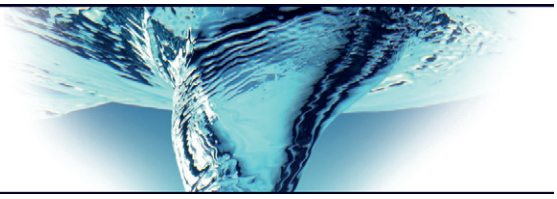
J. Fluid Mech. (2020), *vol.* 897, R1, doi:10.1017/jfm.2020.409

JFM RAPIDS
journals.cambridge.org/rapids



Leveraging reduced-order models for state estimation using deep learning

Nirmal J. Nair[1,†] **and Andres Goza**[1]

[1]Department of Aerospace Engineering, University of Illinois at Urbana–Champaign, Urbana IL 61801, USA

(Received 22 December 2019; revised 25 February 2020; accepted 20 May 2020)

State estimation is key to both analysing physical mechanisms and enabling real-time control of fluid flows. A common estimation approach is to relate sensor measurements to a reduced state governed by a reduced-order model (ROM). (When desired, the full state can be recovered via the ROM.) Current methods in this category nearly always use a linear model to relate the sensor data to the reduced state, which often leads to restrictions on sensor locations and has inherent limitations in representing the generally nonlinear relationship between the measurements and reduced state. We propose an alternative methodology whereby a neural network architecture is used to learn this nonlinear relationship. A neural network is a natural choice for this estimation problem, as a physical interpretation of the reduced state–sensor measurement relationship is rarely obvious. The proposed estimation framework is agnostic to the ROM employed, and can be incorporated into any choice of ROMs derived on a linear subspace (e.g. proper orthogonal decomposition) or a nonlinear manifold. The proposed approach is demonstrated on a two-dimensional model problem of separated flow around a flat plate, and is found to outperform common linear estimation alternatives.

Key words: low-dimensional models, computational methods, vortex shedding

1. Introduction

In fluid dynamics, the goal of state estimation (SE) is to accurately estimate the instantaneous flow field using a set of limited sensor measurements. Achieving this goal can provide insights into key physics and facilitate the prediction and control of flows in various engineering applications. In many problems where SE is of interest, a reduced-order model (ROM) of the high-dimensional system is also typically available. Accordingly, a class of SE strategies that leverage this low-order representation have emerged – that is, estimation is done on a reduced state obtained

† Email address for correspondence: njn2@illinois.edu

© The Author(s), 2020. Published by Cambridge University Press

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

from the ROM (the full state can be recovered via the ROM when desired). In this article, we focus on such methods which are particularly promising for real-time control applications. We note in passing that ROM-based SE approaches are not the only options; many successful SE methods do not rely on a low-order representation of the flow state. Examples include sparse identification using SINDy (Loiseau, Noack & Brunton 2018), identifying problem-specific parameters via an ensemble Kalman filter (Darakananda *et al.* 2018) or convolutional autoencoder (Hou, Darakananda & Eldredge 2019), or SE using a shallow decoder (Erichson *et al.* 2019). However, it may be computationally prohibitive to integrate these SE approaches into ROMs due to an intermediate step that involves the high-dimensional fluid state.

In most model order reduction approaches, the dynamics of the high-dimensional state are projected onto a low-dimensional linear subspace. A number of bases for this subspace have been developed, e.g. proper orthogonal decomposition (POD) (Lumley 1967), dynamic mode decomposition (Schmid 2010) and balanced POD (Willcox & Peraire 2002). More recently, nonlinear ROMs have been developed that utilize local bases instead of a global basis (Amsallem, Zahr & Farhat 2012), or a global nonlinear manifold constructed using autoencoders from deep learning (Lee & Carlberg 2019; Otto & Rowley 2019).

Estimation of the reduced state derived from ROMs can be broadly divided into two categories: intrusive and non-intrusive. Intrusive SE models such as Kalman filtering (Kalman 1960) and particle filtering (Gordon, Salmond & Smith 1993) rely on an observer dynamical system to predict the state (which is later updated based on observed data). These data-assimilation approaches have been coupled with POD-based ROMs on various flow problems (Tu *et al.* 2013; Kikuchi, Misaka & Obayashi 2015). On the other hand, non-intrusive methods are model-free and can be further classified into library- and non-library-based approaches.

In library-based approaches, the sensor measurements are approximated with the same library that is used for the ROM (e.g. obtained from POD modes (Bright, Lin & Kutz 2013) or the training data itself (Callaham, Maeda & Brunton 2019)). The resulting optimization problem can be solved in the ℓ_1 norm to promote sparsity (Candes & Tao 2006). Alternatively, the reduced state can be estimated in the ℓ_2 norm, termed gappy-POD (Everson & Sirovich 1995). To overcome ill-conditioning and overfitting in this ℓ_2 setting, sensor locations can be chosen through greedy (Clark *et al.* 2018), optimal (Brunton *et al.* 2013) or sparse (Sargsyan, Brunton & Kutz 2015) sensor placement algorithms, which can outperform ℓ_1-based approaches (Manohar *et al.* 2018). However, the need for problem-specific sensor locations in this estimation framework limits its flexibility.

Non-library-based approaches, on the other hand, provide an empirically determined map between the measurements and reduced state. This alleviates restrictions on sensor locations and ill-conditioning inherent to library-based methods. One example is linear stochastic estimation (LSE), which provides a linear map through an ℓ_2 minimization of available data (Adrian 1975). Although traditional LSE relates sensor measurements to the high-dimensional state, recent variants estimate the reduced state (Taylor & Glauser 2004; Podvin *et al.* 2018). Quadratic stochastic estimation (Murray & Ukeiley 2007) provides a specific nonlinear extension to LSE. However, for complex fluid flow problems, the nonlinear relationship between the sensor measurements and the reduced state is generally unknown, and a more flexible framework is necessary.

In this work, we model this nonlinear relationship using neural networks. Neural networks are a natural candidate for this aim, as they have been posited to act as high-dimensional interpolators for function approximation (Mallat 2016; Brunton, Noack &

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

Koumoutsakos 2020). This approach allows for a lower number of sensors and greater flexibility in sensor locations compared with its linear counterparts. We demonstrate the efficacy of our approach on a two-dimensional model problem of separated flow past a flat plate, and compare results to those obtained via gappy-POD and LSE. While our results on the model problem are obtained using a POD-based ROM, we emphasize that our formulation is agnostic to the ROM, and can be incorporated into either linear or nonlinear ROMs.

2. State estimation: ROM-based framework and prior work

2.1. *ROM-based state estimation framework*

Consider the dynamical system resulting from the semi-discretization of partial differential equations such as the Navier–Stokes equations:

$$\dot{\boldsymbol{w}} = \boldsymbol{f}(\boldsymbol{w}, t; \boldsymbol{\mu}), \quad \boldsymbol{w}(t_n; \boldsymbol{\mu}) = \boldsymbol{w}^n(\boldsymbol{\mu}), \tag{2.1a,b}$$

where $\boldsymbol{w}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_w}$ represents the high-dimensional state that depends on time $t \in [0, t_{max}]$ and a vector of N_d parameters $\boldsymbol{\mu} \in \mathbb{R}^{N_d}$ consisting of, for instance, Reynolds number, angle of attack, Mach number, etc. The nonlinear function $\boldsymbol{f}(\boldsymbol{w}, t; \boldsymbol{\mu})$ governs the dynamics of the state $\boldsymbol{w}(t; \boldsymbol{\mu})$. Provided that one has an estimate of an instantaneous state $\boldsymbol{w}^n(\boldsymbol{\mu})$ at an arbitrary time t_n, the initial value problem (2.1) can be used to determine $\boldsymbol{w}^{n+i}(\boldsymbol{\mu})$ for $i = 1, 2, \ldots$. We refer to (2.1) as the full-order model (FOM).

We consider the scenario where a reduced-order model (ROM) of (2.1) is available. In this case, the high-dimensional state $\boldsymbol{w}(t; \boldsymbol{\mu})$ is approximated on a low-dimensional manifold as

$$\boldsymbol{w}(t; \boldsymbol{\mu}) \approx \boldsymbol{w}_r(t; \boldsymbol{\mu}) = \boldsymbol{\Phi}(\boldsymbol{a}(t; \boldsymbol{\mu})), \tag{2.2}$$

where $\boldsymbol{\Phi} : \mathbb{R}^{N_k} \mapsto \mathbb{R}^{N_w}$ denotes the nonlinear manifold, $\boldsymbol{a}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_k}$ is the reduced state on this manifold, and $N_k \ll N_w$ is the dimension of the reduced state. To facilitate a clean presentation of the ROM, we assume that $\boldsymbol{\Phi}(\boldsymbol{a})$ is continuously differentiable such that $\boldsymbol{\Upsilon}(\dot{\boldsymbol{a}}) = \dot{\boldsymbol{\Phi}}(\boldsymbol{a})$ for some $\boldsymbol{\Upsilon} : \mathbb{R}^{N_k} \mapsto \mathbb{R}^{N_w}$. Substituting the ROM approximation (2.2) in (2.1), and projecting the resulting equation onto a test manifold $\boldsymbol{\Lambda} : \mathbb{R}^{N_w} \mapsto \mathbb{R}^{N_k}$ such that $\boldsymbol{\Lambda}\boldsymbol{\Upsilon}$ is injective, yields

$$\dot{\boldsymbol{a}} = \boldsymbol{\Psi}(\boldsymbol{f}(\boldsymbol{\Phi}(\boldsymbol{a}), t; \boldsymbol{\mu})), \quad \boldsymbol{a}(t_n; \boldsymbol{\mu}) = \boldsymbol{a}^n(\boldsymbol{\mu}), \tag{2.3a,b}$$

where $\boldsymbol{\Psi}(\cdot) = (\boldsymbol{\Lambda}\boldsymbol{\Upsilon})^{-1} \circ \boldsymbol{\Lambda}(\cdot)$ and $\boldsymbol{a}^n(\boldsymbol{\mu})$ is the initial condition at time instant t_n for the new initial value problem (2.3). In the case of Galerkin projection, where $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda} = \boldsymbol{\Phi}^{\mathrm{T}}$ are linear and orthogonal, $\boldsymbol{\Psi} = \boldsymbol{\Phi}^{\mathrm{T}}$.

Now, the original SE goal of estimating the instantaneous high-dimensional state $\boldsymbol{w}^n(\boldsymbol{\mu})$ reduces to estimating the lower-dimensional state $\boldsymbol{a}^n(\boldsymbol{\mu})$. That is, the SE problem amounts to identifying a map $\mathcal{G} : \mathbb{R}^{N_s} \mapsto \mathbb{R}^{N_k}$ between the sensor measurements and the reduced state such that $\boldsymbol{a}^n = \mathcal{G}(\boldsymbol{s}^n)$, where $\boldsymbol{s}^n(\boldsymbol{\mu}) \in \mathbb{R}^{N_s}$ denotes the sensor measurements at time instant n, and N_s is the number of sensors in the flow field. A schematic of the ROM-based SE framework described here is displayed in figure 1.

2.2. *Prior work: linear estimation models*

The traditional approach of identifying the map $\mathcal{G}$ is given by gappy-POD (Everson & Sirovich 1995). In this approach, $\boldsymbol{\Phi}$ is restricted to be linear and the sensors

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

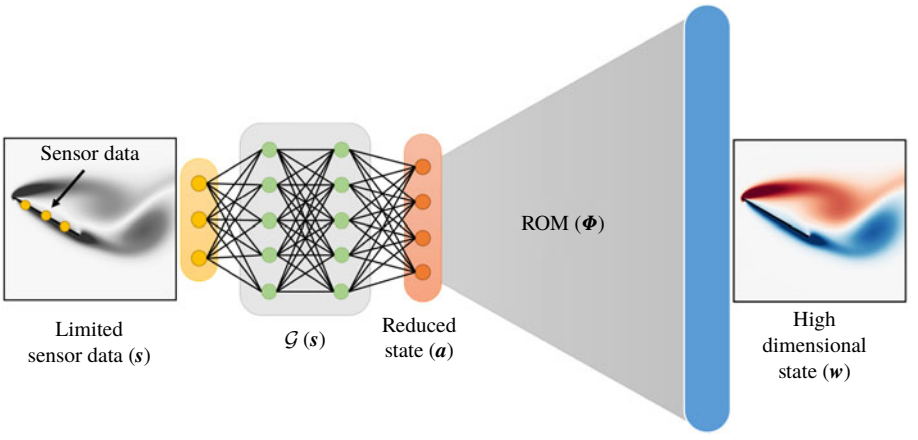


FIGURE 1. Schematic of the ROM-based SE framework, which employs a low-dimensional representation $\boldsymbol{\Phi}$ of the high-dimensional state $\boldsymbol{w}^n$. Here $\mathcal{G}$ maps sensor measurements $\boldsymbol{s}^n$ to the reduced state $\boldsymbol{a}^n$. In the proposed approach described in § 3, $\mathcal{G}$ is nonlinear and constructed from a neural network (as pictured). For traditional linear estimation models described in § 2.2, $\mathcal{G}$ represents a real-valued matrix and the pictured neural network is no longer valid.

directly measure the high-dimensional state at $N_s < N_w$ flow locations, such that $\boldsymbol{s}^n(\boldsymbol{\mu}) = \boldsymbol{C}\boldsymbol{w}^n(\boldsymbol{\mu}) \approx \boldsymbol{C}\boldsymbol{\Phi}\boldsymbol{a}^n(\boldsymbol{\mu})$. The matrix $\boldsymbol{C} \in \mathbb{R}^{N_s \times N_w}$ contains 1 at measurement locations and 0 at all other locations. The reduced state $\boldsymbol{a}^n$ is obtained by the minimization problem:

$$\boldsymbol{a}^n = \underset{\hat{\boldsymbol{a}}^n}{\arg\min} \|\boldsymbol{s}^n - \boldsymbol{C}\boldsymbol{\Phi}\hat{\boldsymbol{a}}^n\|_2^2. \tag{2.4}$$

The solution to (2.4) is analytically provided by the Moore–Penrose pseudo-inverse of $\boldsymbol{C}\boldsymbol{\Phi}$, resulting in the linear map $\mathcal{G}$ being defined as $\boldsymbol{a}^n = \mathcal{G}(\boldsymbol{s}^n) = (\boldsymbol{C}\boldsymbol{\Phi})^+\boldsymbol{s}^n$.

The LSE can be considered as a generalization of gappy-POD where the sensor measurements are not restricted to lie in the span of the basis of the high-dimensional state. In other words, the linear operator $(\boldsymbol{C}\boldsymbol{\Phi})^+$ can be replaced by a more general matrix $\boldsymbol{G} \in \mathbb{R}^{N_k \times N_s}$; that is, $\mathcal{G}$ is represented via $\boldsymbol{a}^n = \mathcal{G}(\boldsymbol{s}^n) = \boldsymbol{G}\boldsymbol{s}^n$. In LSE, $\boldsymbol{G}$ is determined from data via the optimization problem,

$$\boldsymbol{G} = \underset{\hat{\boldsymbol{G}}}{\arg\min} \|\boldsymbol{S} - \hat{\boldsymbol{G}}\boldsymbol{A}\|_2^2, \tag{2.5}$$

where $\boldsymbol{S} \in \mathbb{R}^{N_s \times M}$ and $\boldsymbol{A} \in \mathbb{R}^{N_k \times M}$ are snapshot matrices of sensor measurements and reduced states, respectively, consisting of M snapshots. The solution to (2.5) is analytically obtained as $\boldsymbol{G} = \boldsymbol{S}\boldsymbol{A}^+$.

2.2.1. *Drawbacks of linear estimation models*

In gappy-POD, the sensor locations encoded in $\boldsymbol{C}$ can significantly influence the condition number of $\boldsymbol{C}\boldsymbol{\Phi}$. In particular, sensor locations are required to coincide with regions where the columns of $\boldsymbol{\Phi}$ have significant non-zero and distinct values. Sensor locations that do not satisfy this property lead to an inaccurate estimation of the reduced state, $\boldsymbol{a}^n = (\boldsymbol{C}\boldsymbol{\Phi})^+\boldsymbol{s}^n$. Furthermore, choosing more library elements than sensors, $N_k > N_s$, can result in overfitting. These limitations can be resolved

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

by selecting optimal sensor locations that improve the condition number of $\boldsymbol{C\Phi}$ (Manohar *et al.* 2018) and/or incorporating regularization in (2.4) to mitigate overfitting. However, the need to select specific sensor locations reduces the flexibility of gappy-POD.

Unlike gappy-POD, LSE is significantly more robust to sensor locations. However, it linearly models a (generally non-trivial) nonlinear relationship between the sensor measurements and the reduced state. Therefore, this approach is limited by the rank of $\boldsymbol{G}$, which is at best $\text{rank}(\boldsymbol{G}) \leqslant \min(N_k, N_s)$. The estimation performance of LSE can be improved by increasing the number of sensors, N_s, though this is not always possible depending on the given application. We propose a method for more robustly recovering the reduced state by learning a nonlinear relationship using a neural network.

3. Proposed approach: deep state estimation

In our proposed approach, the map $\mathcal{G}$ is further generalized to nonlinearly relate sensor measurements to the reduced state as

$$\boldsymbol{a}^n = \mathcal{G}(\boldsymbol{s}^n) = \boldsymbol{g}(\boldsymbol{s}^n, \boldsymbol{\theta}), \tag{3.1}$$

where $\boldsymbol{g}(\cdot, \boldsymbol{\theta})$ with $\boldsymbol{g} : \mathbb{R}^{N_s} \to \mathbb{R}^{N_k}$ denotes the nonlinear function parametrized by a set of parameters $\boldsymbol{\theta}$. For various complex fluid flow problems, the nonlinear relationship between $\boldsymbol{s}^n$ and $\boldsymbol{a}^n$ is rarely obvious. Therefore, in this work, we propose to model $\boldsymbol{g}(\boldsymbol{s}^n, \boldsymbol{\theta})$ via a more general approach using neural networks. We refer to this proposed approach as deep state estimation (DSE).

3.1. *Neural network architecture*

In this work, we employ a neural network architecture consisting of N_l fully connected layers represented as a composition of N_l functions,

$$\boldsymbol{g}(\boldsymbol{\xi}; \boldsymbol{\theta}) = \boldsymbol{h}_{N_l}(\cdot; \boldsymbol{\Theta}_{N_l}) \circ \cdots \circ \boldsymbol{h}_2(\cdot; \boldsymbol{\Theta}_2) \circ \boldsymbol{h}_1(\boldsymbol{\xi}; \boldsymbol{\Theta}_1), \tag{3.2}$$

where the output vector at the ith layer ($i = 1, 2, \ldots, N_l$) is given by $\boldsymbol{h}_i(\boldsymbol{\xi}; \boldsymbol{\Theta}_i) = \sigma(\boldsymbol{\Theta}_i[\boldsymbol{\xi}^{\mathrm{T}}, 1]^{\mathrm{T}})$ with $\boldsymbol{h}_i(\cdot; \boldsymbol{\Theta}_i) : \mathbb{R}^{l_{i-1}} \to \mathbb{R}^{l_i}$. Essentially, an affine transformation of the input vector followed by a pointwise evaluation of a nonlinear activation function, $\sigma(\cdot) : \mathbb{R} \to \mathbb{R}$, is performed. Here, l_i is the size of the output at layer i and $\boldsymbol{\Theta}_i \in \mathbb{R}^{l_i \times l_{i-1}+1}$ comprises the weights and biases corresponding to layer i.

A schematic of our proposed DSE approach exhibiting this neural network architecture with $N_l = 3$ fully connected layers is shown in figure 1. Sensor measurements of dimensions $l_0 = N_s$ are fed as inputs while the output layer comprising the reduced state has dimensions $l_{N_l} = N_k$. We note that other architectures such as graph convolutional or recurrent neural networks for exploiting spatial or temporal locality of sensor measurements, respectively, could be utilized to construct the neural network. However, we choose fully connected layers owing to its simplicity and small dimensions of input and output. The weights $\boldsymbol{\theta} \equiv (\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_{N_l})$ are evaluated by training the neural network, the details of which are provided in the next section.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

3.2. *Training the neural network*

The first step in training is to collect snapshots of high-dimensional states, sensor measurements and reduced states. The FOM (2.1) is solved at N_p sampled parameters $\boldsymbol{\mu}^s$ and N_t time instants to obtain the snapshots $\boldsymbol{w}^n(\boldsymbol{\mu}_i^s)$ for $i = 1, \ldots, N_p$ and $n = 1, \ldots, N_t$, resulting in a total of $M = N_t N_p$ snapshots. Then, $\boldsymbol{s}^n(\boldsymbol{\mu}_i^s)$ is evaluated via some known transformation of $\boldsymbol{w}^n(\boldsymbol{\mu}_i^s)$ and $\boldsymbol{a}^n(\boldsymbol{\mu}_i^s)$ is derived by solving

$$\boldsymbol{a}^n(\boldsymbol{\mu}_i^s) = \underset{\hat{\boldsymbol{a}}^n}{\arg\min} \|\boldsymbol{w}^n - \boldsymbol{\Phi}(\hat{\boldsymbol{a}}^n)\|_2^2. \tag{3.3}$$

When $\boldsymbol{\Phi}$ is linear with orthogonal columns, for instance POD modes, the solution to (3.3) is obtained by $\boldsymbol{a}^n(\boldsymbol{\mu}_i^s) = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{w}^n(\boldsymbol{\mu}_i^s)$. Next, the snapshots of $\boldsymbol{s}^n(\boldsymbol{\mu}_i^s)$ and $\boldsymbol{a}^n(\boldsymbol{\mu}_i^s)$ for $i = 1, \ldots, N_p$ and $n = 1, \ldots, N_t$ are standardized via z-score normalization (Goodfellow, Bengio & Courville 2016) to enable faster convergence while training the neural network.

Typically, prior to training, the data are divided into training and validation sets, which are used to evaluate/update the weights $\boldsymbol{\theta}$ and test the accuracy of the network, respectively. Accordingly, for each sampled parameter, $\boldsymbol{\mu}_i^s$, snapshots at N_{train} and $N_{valid} = N_t - N_{train}$ random time instants are chosen for training and validation, respectively, resulting in a total of $M_{train} = N_{train}N_p$ training and $M_{valid} = N_{valid}N_p$ validation snapshots. Once the neural network is trained, it is tested on a set of testing snapshots, which were utilized in neither training nor validation. Accordingly, we collect $M_{test} = N_{test}N_p^*$ testing snapshots of $\boldsymbol{w}^n(\boldsymbol{\mu}_i^*)$, $\boldsymbol{s}^n(\boldsymbol{\mu}_i^*)$ and $\boldsymbol{a}^n(\boldsymbol{\mu}_i^*)$ for $i = 1, \ldots, N_p^*$ and $n = 1, \ldots, N_{test}$ time instants evaluated at N_p^* unsampled parameters such that $\boldsymbol{\mu}^* \nsubseteq \boldsymbol{\mu}^s$.

We train the neural network $\boldsymbol{g}(\cdot;\, \boldsymbol{\theta})$ to evaluate the trainable parameters $\boldsymbol{\theta}$ by minimizing the ℓ_2 error between the reduced state and its approximation, given by

$$\boldsymbol{\theta} = \underset{\hat{\boldsymbol{\theta}}}{\arg\min} \sum_{i=1}^{M_{train}} \|\boldsymbol{a}^{(i)} - \boldsymbol{g}(\boldsymbol{s}^{(i)}, \hat{\boldsymbol{\theta}})\|_2^2, \tag{3.4}$$

where the superscript (i) denotes the ith training snapshot. The problem (3.4) is solved using the stochastic gradient descent (SGD) method with mini-batching and early stopping (which acts as a regularizer to avoid overfitting) (Goodfellow *et al.* 2016).

4. Numerical experiments: flow over a flat plate

In this section, our proposed DSE approach is applied on a test case of a two-dimensional flow over a flat plate. We choose $\boldsymbol{\Phi}$ to be linear containing the first N_k POD modes of the snapshot matrix, whose columns are given by $\boldsymbol{w}^n(\boldsymbol{\mu}_i^s) - \bar{\boldsymbol{w}}$ for $i = 1, \ldots, N_p$ and $n = 1, \ldots, N_t$, where $\bar{\boldsymbol{w}} = (1/M) \sum_{i=1,n=1}^{i=N_p,n=N_t} \boldsymbol{w}^n(\boldsymbol{\mu}_i^s) \in \mathbb{R}^{N_w}$ is the mean. We again emphasize that POD is chosen for its ubiquity in practice and ease of presentation; the estimation framework described above can be incorporated into a range of ROMs.

All results reported in this section are predictive. That is, the estimated states all lie in parameter regions $\boldsymbol{\mu}^*$ not sampled for training the neural network. The results generated by DSE are compared with gappy-POD and LSE, described in § 2.2. The performance of these approaches is analysed by computing the relative error

$$\text{error}\,(\%) = \frac{\|\boldsymbol{w}^n(\boldsymbol{\mu}^*) - \boldsymbol{w}_r^n(\mu^*)\|_2}{\|\boldsymbol{w}^n(\boldsymbol{\mu}^*)\|_2} \times 100, \tag{4.1}$$

where $\boldsymbol{w}^n(\boldsymbol{\mu}^*)$ and $\boldsymbol{w}_r^n(\boldsymbol{\mu}^*)$ are the FOM and estimated solutions, respectively.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

4.1. *Problem description*

We consider flow over a flat plate of length 1 unit at $Re = 200$. The parameter of interest μ is the angle of attack (AoA) of the flat plate. Two sets of AoA are considered: (a) $\mu \in [25°, 27°]$ and (b) $\mu \in [70°, 71°]$. Both parameter sets lead to separated flow and vortex shedding.

The problem is simulated using the incompressible Navier–Stokes equations in the immersed boundary framework of Colonius & Taira (2008), which utilizes a discrete vorticity–streamfunction formulation. The solver employs a fast multi-domain approach for handling far-field Dirichlet boundary conditions of zero vorticity. Accordingly, for the two sets of AoA considered, we utilize five grid levels of increasing coarseness. The grid spacing of the finest domain (and of the flat plate) is $\Delta x = 0.01$, and the time step is $\Delta t = 0.0008$. All the snapshots are collected after roughly five shedding cycles, by which time the system approximately reaches limit cycle oscillations of vortex shedding (and lift and drag). The domain sizes of the finest and coarsest grid levels are (a) $[-1, 4] \times [-2, 2]$ and $[-37.5, 42.5] \times [-32, 32]$, and (b) $[-1, 3] \times [-1, 5]$ and $[-30, 34] \times [-45, 51]$. The total number of grid points in the finest domain is thus (a) 500×400 and (b) 400×600, respectively. All collected snapshots and SE results correspond to the finest domain only.

Note that all simulations are conducted by placing the flat plate at 0° and aligning the flow at angle μ with respect to the plate. This is done to obtain POD modes that are a good low-dimensional representation of the high-dimensional flow field for the range of AoAs considered. However, while displaying the results, the flow fields are rotated back to align the plate at angle μ for readability.

For SE via DSE, the neural network consists of $N_l = 3$ layers with dimensions $l_i = 500$ for $i = 1, 2$, $l_0 = N_s$ and $l_{N_l} = N_k$. For the nonlinear activation function σ, we use rectified linear units (ReLU) (Goodfellow *et al.* 2016) at the hidden layers and an identity function at the output layer. Following the data segregation strategy explained in § 3.2, training data are split into 80 % and 20 % for training and validation, respectively. For SGD, learning rate is set to 0.1 during the first 500 epochs (for faster convergence), which is then reduced to 0.01 for the remainder of training, momentum is set to 0.9 and mini-batch size is set to 80. Training is terminated when the error on the validation dataset does not reduce over 100 epochs, which is chosen as the early stopping criterion. Overall, the network is trained for approximately 2000 epochs on Pytorch.

4.2. *Flow at* $\mu \in [25°, 27°]$

Here we compare the predictive capabilities of our proposed DSE approach to several linear SE approaches for AoA $\mu \in [25°, 27°]$, and where the sensors measure vorticity at $N_s = 5$ locations on the body. The matrix $\boldsymbol{\Phi}$ used for the ROM is constructed using $N_k = 25$ POD modes of the vorticity snapshot matrix.

The AoAs used for training the SE methods are $\mu^s = \{25°, 25.2°, \ldots, 27°\}$, and those used for testing are $\mu^* = \{25.5°, 26.25°, 26.75°\} \nsubseteq \mu^s$. For each AoA, $N_t = 250$ training snapshots and $N_{test} = 50$ test snapshots are sampled between $t = 20$ and 28 convective time units. Thus, a total of $M = 2750$ training and $M_{test} = 150$ testing snapshots are used, respectively.

Figure 2 shows the vorticity contours produced by the high-fidelity model (FOM), gappy-POD, LSE and DSE at $\mu^* = 26.75°$, $t = 27.17$. The flow field constructed by DSE more accurately matches the FOM solution as compared to gappy-POD and LSE. Moreover, the average relative error of the states estimated at all 150 testing instances by DSE is only 1.03 % as compared to 34.74 % and 16.84 % due to gappy-POD and LSE, respectively.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

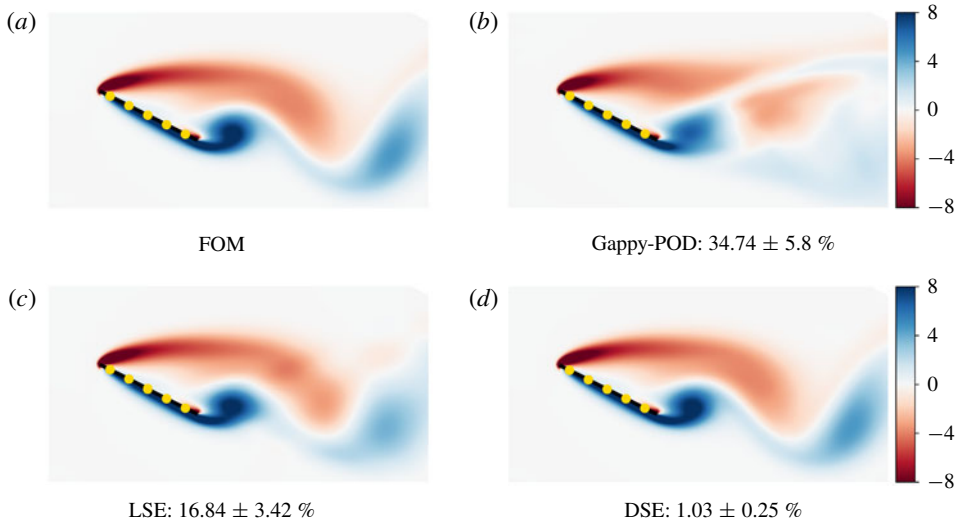


FIGURE 2. Comparison of vorticity contours estimated by the high-fidelity model (FOM), gappy-POD, LSE and DSE at $\mu^* = 26.75°$, $t = 27.17$ and corresponding relative error (mean and standard deviation of state estimations at all testing instances) with $N_k = 25$ modes and $N_s = 5$ sensors that measure vorticity. The contour values are limited between -8 and $+8$ for visualizing vortical structures in the wake.

4.3. *Flow at* $\mu \in [70°, 71°]$

We now consider a more strenuous test case of flow at large AoA, which exhibits richer dynamics associated with more complex vortex shedding behaviour; cf. figure 4(*a*). For added complexity and application relevance, we consider sensors that measure the magnitude of surface stress (instead of vorticity) at locations on the body. The basis $\boldsymbol{\Phi}$ is constructed from POD modes of the snapshot matrix containing vorticity and surface stress. Details of the first six POD modes are provided in appendix A. The AoAs for training and testing are $\mu^s = \{70°, 70.2°, \ldots, 71°\}$ and $\mu^* = \{70.25°, 70.5°, 70.75°\} \nsubseteq \mu^s$, respectively. For each AoA, $N_t = 400$ training snapshots and $N_{test} = 80$ test snapshots are sampled between $t = 25$ and 52.2 convective time units, resulting in a total of $M = 2400$ and $M_{test} = 240$ snapshots for training and SE, respectively.

In figure 3, we compare various methods through vorticity contours at $\mu^* = 70.75°$, $t = 38.76$ obtained by using $N_k = 25$ POD modes and $N_s = 5$ sensors. It can be observed that DSE significantly outperforms gappy-POD and LSE. Moreover, the average relative error of the states estimated at all 240 testing instances by DSE is only 2.07 % as compared to 61.75 % and 35.75 % in gappy-POD and LSE approaches, respectively. Additional details of these results are provided in appendix B, where we have plotted the error in estimated vorticity.

Finally, we compare the performances of these SE approaches for different numbers of POD modes, $N_k = \{15, 25, 35\}$, and sensors, $N_s = \{5, 10, 15\}$. The average relative errors of the estimated vorticity for these nine permutations are plotted as markers in figure 4(*b*). The solid lines connect the markers with lowest errors corresponding to each N_k, therefore highlighting the best performance among $N_s = \{5, 10, 15\}$. We also compare the results with the optimal reduced states obtained by projection, $\boldsymbol{a}^n = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{w}^n$. These are called optimal because the POD coefficients are exact, and all error is

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

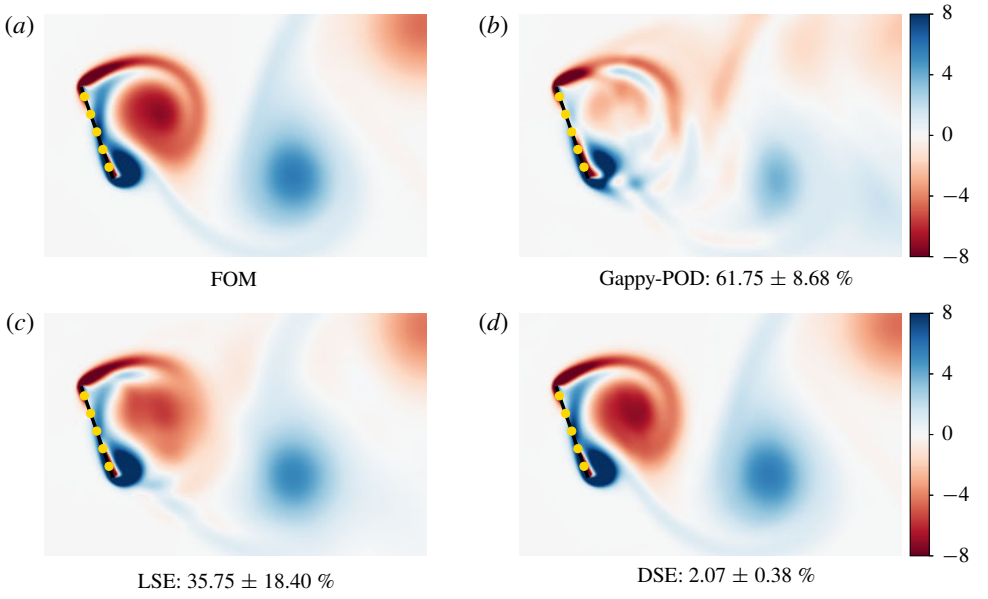


FIGURE 3. Analogue of figure 2 for $\mu^* = 70.75°$, $t = 38.76$, $N_k = 25$ modes, and $N_s = 5$ sensors that measure the magnitude of surface stress.

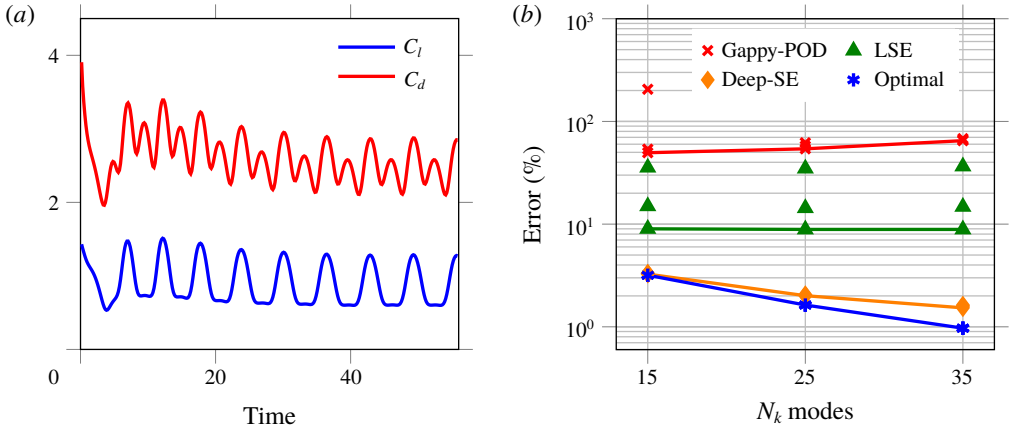


FIGURE 4. (*a*) Demonstration of vortex shedding phenomena computed by FOM: plot of C_l and C_d versus time at $\mu = 70°$. (*b*) Performance comparison of gappy-POD, LSE, DSE and optimal reconstruction when $\mu^* \in [70°, 71°]$: relative error for varying number of surface stress sensors in $N_s = \{5, 10, 15\}$.

incurred from the ROM approximation (2.2). By contrast, ROM-based SE approaches incur error due to both the ROM approximation and the model error associated with the map $\mathcal{G}$. Therefore, none of the above-mentioned SE methods can be expected to estimate a more accurate state than the optimal reconstruction obtained via these optimal reduced states. The error in optimal reconstruction is plotted in blue and represents a lower bound of the error, not an SE approach. From the plot, it can be observed that DSE produces an error of only 1 %–3 % as compared to 10 %–40 % due

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

to LSE and 50 %–200 % due to gappy-POD. For all numbers of sensors considered, DSE produces errors that are comparable to the lower bound. Estimates by LSE do not improve as the number of modes N_k is increased, due to its rank-related limitations as described in § 2.2.1. Similar error trends were also observed for the previous simpler test case of $\mu \in [25°, 27°]$, though for conciseness these results are not shown in this article.

5. Conclusions

In this article, a deep state estimation (DSE) approach was introduced that exploits a low-order representation of the flow field to seamlessly integrate sensor measurements into reduced-order models (ROMs). In this method, the sensor data and reduced state are nonlinearly related using neural networks. The estimated reduced state can be used as an initial condition to efficiently predict future states or recover the instantaneous full flow field via the ROM approximation. Numerical experiments consisted of two-dimensional flow over a flat plate at high angles of attack, resulting in separated flow and associated vortex shedding processes. At parameter instances not observed during training, DSE was demonstrated to significantly outperform traditional linear estimation approaches such as gappy-POD and linear stochastic estimation (LSE). The robustness of the approach to sensor locations and the physical quantities measured was demonstrated by placing varying numbers of vorticity- and surface-stress-measuring sensors on the body of the flat plate. Finally, it is emphasized that the proposed approach is agnostic to the ROM employed; i.e. while a POD-based ROM was utilized for the numerical experiments, the general DSE framework allows for any choice of linear or nonlinear low-dimensional representation.

Acknowledgement

The authors gratefully acknowledge H. Tran for useful discussions on neural networks and snapshot sampling strategies for training.

Declaration of interests

The authors report no conflict of interest.

Appendix A. POD modes

This section briefly provides a physical intuition of the POD modes utilized in § 4.3 for the $\mu \in [70°, 71°]$ test case. The first six POD modes along with the relative energy content corresponding to vorticity is provided in figure 5. We can clearly observe the vortical structures in the wake due to von Kármán vortex shedding. The first two dominant modes capture the primary vortex near the upper surface of the plate shed from the leading and trailing edges. For instance, this structure corresponds to the leading-edge vortex in figure 3. The remaining modes represent subharmonic spatial structures of the first two modes. A linear combination of all these modes provides a temporal flow-field representation of the alternating vortex shedding phenomena.

Appendix B. Additional results: vorticity error plots

In this section we provide the contour plots of the relative error between the FOM vorticity and vorticity estimated by gappy-POD, LSE and DSE corresponding

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

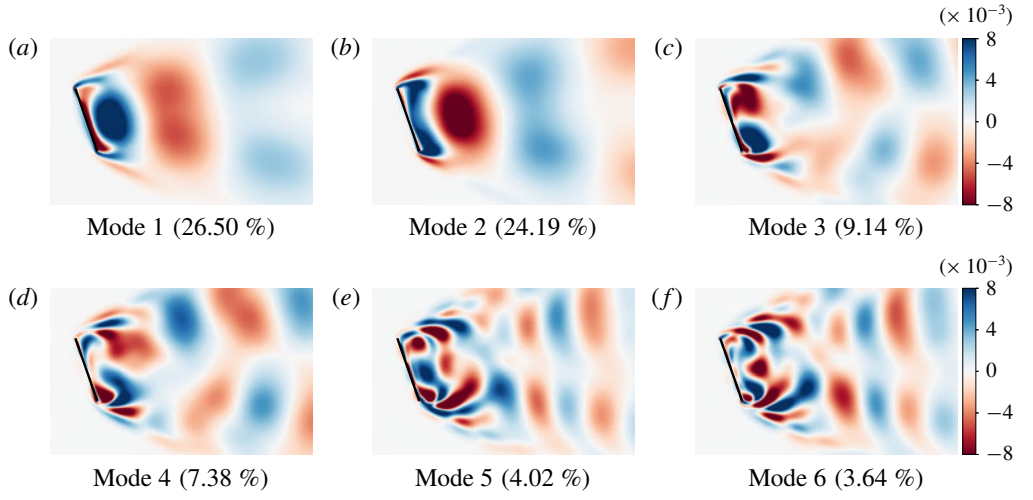


FIGURE 5. POD modes and their relative energy content (provided in parenthesis) corresponding to vorticity of snapshot matrix containing vorticity and surface stress for $\mu \in [70°, 71°]$. The contour values are limited between -0.008 and $+0.008$ for visualizing vortical structures in the wake.

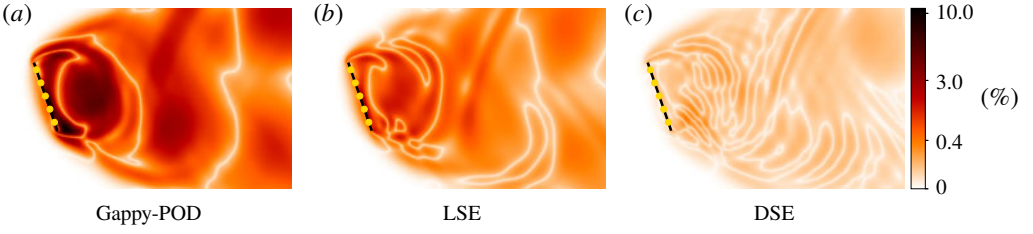


FIGURE 6. Comparison of contours of relative error (in %) between FOM vorticity and vorticity estimated by gappy-POD, LSE and DSE at $\mu^* = 26.75°$, $t = 27.17$ with $N_k = 25$ modes and $N_s = 5$ sensors that measure surface stress.

to the results demonstrated in figure 3. Here we define relative error as $|\boldsymbol{w}^n(\boldsymbol{\mu}^*) - \boldsymbol{w}_r^n(\mu^*)|/\|\boldsymbol{w}^n(\boldsymbol{\mu}^*)\|_\infty$, where $\boldsymbol{w}^n(\boldsymbol{\mu}^*)$ and $\boldsymbol{w}_r^n(\boldsymbol{\mu}^*)$ are the FOM and estimated solutions, respectively. We can observe that, while DSE produces approximately uniform but low errors throughout the wake region, gappy-POD and LSE generate very high error, specifically in the estimation of the most dominant structure, namely the leading-edge vortex. Since the vortices in the nearest vicinity of the plate contribute most to its aerodynamic performance, it is crucial to estimate such structures accurately, which is indeed accomplished by DSE.

References

ADRIAN, R. J. 1975 On the role of conditional averages in turbulence theory. In *Proceedings of the 4th Biennial Symposium on Turbulence in Liquids*, pp. 323–332. Science Press.

AMSALLEM, D., ZAHR, M. J. & FARHAT, C. 2012 Nonlinear model order reduction based on local reduced-order bases. *Intl J. Numer. Meth. Engng* **92** (10), 891–916.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

BRIGHT, I., LIN, G. & KUTZ, J. N. 2013 Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Phys. Fluids* **25** (12), 127102.

BRUNTON, B. W., BRUNTON, S. L., PROCTOR, J. L. & KUTZ, J. N. 2013 Optimal sensor placement and enhanced sparsity for classification. arXiv:1310.4217.

BRUNTON, S. L., NOACK, B. R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.

CALLAHAM, J. L., MAEDA, K. & BRUNTON, S. L. 2019 Robust flow reconstruction from limited measurements via sparse representation. *Phys. Rev. Fluids* **4** (10), 103907.

CANDES, E. J. & TAO, T. 2006 Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans. Inf. Theory* **52** (12), 5406–5425.

CLARK, E., ASKHAM, T., BRUNTON, S. L. & KUTZ, J. N. 2018 Greedy sensor placement with cost constraints. *IEEE Sensors J.* **19** (7), 2642–2656.

COLONIUS, T. & TAIRA, K. 2008 A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Comput. Meth. Appl. Mech. Engng* **197** (25–28), 2131–2146.

DARAKANANDA, D., DA SILVA, A., COLONIUS, T. & ELDREDGE, J. D. 2018 Data-assimilated low-order vortex modeling of separated flows. *Phys. Rev. Fluids* **3** (12), 124701.

ERICHSON, N. B., MATHELIN, L., YAO, Z., BRUNTON, S. L., MAHONEY, M. W. & KUTZ, J. N. 2019 Shallow learning for fluid flow reconstruction with limited sensors and limited data. arXiv:1902.07358.

EVERSON, R. & SIROVICH, L. 1995 Karhunen–Loève procedure for gappy data. *J. Opt. Soc. Am.* A **12** (8), 1657–1664.

GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. 2016 *Deep Learning*. MIT Press.

GORDON, N. J., SALMOND, D. J. & SMITH, A. F. M. 1993 Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (Radar and Signal Processing)*, vol. 140, pp. 107–113.

HOU, W., DARAKANANDA, D. & ELDREDGE, J. D. 2019 Machine-learning-based detection of aerodynamic disturbances using surface pressure measurements. *AIAA J.* **57** (12), 5079–5093.

KALMAN, R. E. 1960 A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Engng* **82** (1), 35–45.

KIKUCHI, R., MISAKA, T. & OBAYASHI, S. 2015 Assessment of probability density function based on POD reduced-order model for ensemble-based data assimilation. *Fluid Dyn. Res.* **47** (5), 051403.

LEE, K. & CARLBERG, K. T. 2019 Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.* **404**, 108973.

LOISEAU, J., NOACK, B. R. & BRUNTON, S. L. 2018 Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490.

LUMLEY, J. L. 1967 The structure of inhomogeneous turbulence. In *Atmospheric Turbulence and Wave Propagation* (ed. A. M. Yaglom & V. I. Tatarsky), pp. 166–176. Nauka.

MALLAT, S. 2016 Understanding deep convolutional networks. *Phil. Trans. R. Soc.* A **374** (2065), 20150203.

MANOHAR, K., BRUNTON, B. W., KUTZ, J. N. & BRUNTON, S. L. 2018 Data-driven sparse sensor placement for reconstruction: demonstrating the benefits of exploiting known patterns. *IEEE Control. Syst. Mag.* **38** (3), 63–86.

MURRAY, N. E. & UKEILEY, L. S. 2007 Modified quadratic stochastic estimation of resonating subsonic cavity flow. *J. Turbul.* **8**, N53.

OTTO, S. E. & ROWLEY, C. W. 2019 Linearly recurrent autoencoder networks for learning dynamics. *SIAM J. Appl. Dyn. Syst.* **18** (1), 558–593.

PODVIN, B., NGUIMATSIA, S., FOUCAUT, J., CUVIER, C. & FRAIGNEAU, Y. 2018 On combining linear stochastic estimation and proper orthogonal decomposition for flow reconstruction. *Exp. Fluids* **59** (3), 58.

SARGSYAN, S., BRUNTON, S. L. & KUTZ, J. N. 2015 Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Phys. Rev.* E **92** (3), 033304.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press

SCHMID, P. J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.

TAYLOR, J. A. & GLAUSER, M. N. 2004 Towards practical flow sensing and control via POD and LSE based low-dimensional tools. *Trans. ASME J. Fluids Engng* **126** (3), 337–345.

TU, J. H., GRIFFIN, J., HART, A., ROWLEY, C. W., CATTAFESTA, L. N. & UKEILEY, L. S. 2013 Integration of non-time-resolved PIV and time-resolved velocity point sensors for dynamic estimation of velocity fields. *Exp. Fluids* **54** (2), 1429.

WILLCOX, K. & PERAIRE, J. 2002 Balanced model reduction via the proper orthogonal decomposition. *AIAA J.* **40** (11), 2323–2330.

https://doi.org/10.1017/jfm.2020.409 Published online by Cambridge University Press