Adaptive Stochastic Reduced Order Modeling for Autonomous Ocean Platforms

T. Ryu[a], J. P. Heuss[a], P. J. Haley, Jr.[a], C. Mirabito[a], E. Coelho[b], P. Hursky[b], M. C. Schönau[b], K. Heaney[b], P. F. J. Lermusiaux[a,*]

[a] *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA*

[b] *Applied Ocean Sciences, Fairfax Station, VA*

[†]Corresponding Author: pierrel@mit.edu

***Abstract*—Onboard probabilistic forecasting and data assimilation is challenging for unmanned autonomous platforms. Due to the operational constraints, efficient adaptive reduced order models (ROMs) are needed. To extend the duration for which Dynamic Mode Decomposition (DMD) predictions are accurate, we utilize and augment incremental methods that update the reduced order state but also adapt the DMD. Our adaptive ROM methods are dynamic and stochastic. They update the state, parameters, and basis functions, in response to the changing forecasts, possibly computed in remote centers, and to observations made by the autonomous platforms and by other assets. For the latter, to allow learning even when observations are sparse and multivariate, we employ Bayesian data assimilation. Specifically, we extend the Gaussian Mixture Model - Dynamically Orthogonal (GMM-DO) filter to stochastic DMD forecasts and Bayesian GMM updates of the DMD coefficients, state, and parameters, learning from the limited gappy observation data sets.**

***Index Terms*—reduced order model, Dynamic Mode Decomposition, stochastic models, autonomous marine vehicles, data assimilation, autonomy**

I. Introduction

Significant challenges for unmanned autonomous platforms at sea include predicting the likely scenarios for the ocean environment, quantifying regional uncertainties, and updating forecasts of the evolving dynamics using their observations [1], [2]. Due to the operational constraints such as onboard power, memory, bandwidth, and space limitations, efficient adaptive reduced order models (ROMs) are needed for onboard predictions. Dynamic Mode Decomposition (DMD) [3][and references therein], a data-driven dimensionality reduction algorithm, can be used for accurate predictions for short periods in ocean environments [4]. To extend the duration for which DMD predictions are accurate, we utilize incremental methods that update the reduced order state but also adapt the DMD [5], [6]. The resulting adaptive ROM methods become dynamic and stochastic, and they update the state, parameters, and basis functions, in response to the changing forecasts, possibly computed in remote centers.

Once at sea, multivariate ocean observations are made by autonomous platforms as well as by other assets. These data are often limited and gappy, especially for the in situ measurements. To control errors and improve the accuracy of onboard forecasts, the ROMs should assimilate and learn from these observations. To capture all of the information contained in the sparse observations, principled Bayesian data assimilation methods [7]–[10] should be utilized for such onboard assimilation. Bayesian filtering requires probabilistic predictions, hence the need for stochastic ROMs. In this process, these ROM predictions but the ROM subspace itself can be updated [9], [11], [12]. For this adaptive data assimilation, we employ the Gaussian Mixture Model - Dynamically Orthogonal (GMM-DO) filter [13]. To predict prior probabilities for the full dynamical state, the GMM-DO filter uses the stochastic Dynamically Orthogonal (DO) field equations [14] or their ensemble approximation [15], and the corresponding adaptive stochastic subspace, effectively approximating the Fokker–Planck equation. At assimilation times, the DO subspace realizations are fit to semiparametric Gaussian Mixture Models (GMMs) using the Expectation-Maximization algorithm and the Bayesian Information Criterion. Bayes's law is then efficiently carried out analytically within the evolving stochastic subspace. We extend this approach to stochastic DMD forecasts and Bayesian GMM-DO updates of the DMD state and parameters, learning from the limited gappy observation data sets.

A. Adaptive ROM Methods for Forecast Dissemination

Classic DMD uses a set of fixed time snapshots to generate modes and coefficients that can be utilized to provide reduced order model forecasts given some initial conditions. The reduced modes and coefficients are however limited to linearly approximating the underlying dynamics of the time snapshots that they were generated from. Therefore, over a longer time period, in which the nonlinear ocean forecasts computed in remote centers significantly change, the accuracy of the DMD reduced order models onboard the autonomous platforms deteriorates [4].

There are various adaptive methods used to keep the DMD reduced order model relevant over longer periods [5], [6]. These methods allow the update of DMD modes and coefficients with new time snapshots that can be communicated to the unmanned autonomous platforms. However, due to the operational bandwidth limitations, it is infeasible to send the full-dimensional time snapshots to them. Instead, we propose the incremental Low-Rank DMD (iLRDMD) framework in which we: i) project the full-dimensional time snapshots onto a set of Proper Orthogonal Decomposition (POD) basis [4], ii) on a shorter time period, communicate these projected

978-0-692-93559-0 ©2021 MTS

coefficients to the autonomous platforms to perform adaptive DMD, and iii) provide forecasts. To keep the projection onto the POD basis optimal over time, we utilize incremental POD methods [16] which are communicated to the autonomous platforms less frequently. Under this framework, we reduce the communication and computational load by operating in the space of basis projected coefficients.

B. Data Assimilation for Adaptive Stochastic DMD

Data assimilation (DA) is the process of quantitatively estimating dynamically evolving fields by melding information from observations with that predicted by computational models. Many DA algorithms exist, some more efficient than others [17], [18]. Due to operational constraints onboard autonomous platforms, efficient algorithms to predict uncertainty and assimilating data are needed. Extending the GMM-DO filter ideas, we show how forecasts made using DMD can be updated by Bayesian data assimilation. One computational advantage is the linear approximation made in the DMD model; it allows easy forward integration and maintains the GMM properties. We analyze the accuracy of such stochastic DMD predictions by comparison with other nonlinear stochastic methods. Additionally we compared these results to those of the Ensemble Kalman Filter (EnKF) [19].

II. Methodology

A. Adaptive ROM

We now describe the three main components of the iLRDMD framework: i) establishing the relationship between the DMD operator in the full-dimensional state space to the DMD operator in the low-rank representation space, ii) the incremental update algorithm of the DMD operator, and iii) the incremental update algorithm of the set of tailored POD basis that forms the low-rank representation space.

1) Low-rank DMD: Many applications deal with high-dimensional state $\mathbf{x} \in \mathbb{R}^{n_X}$ that have a low-rank representation under some tailored basis. [4] has shown that using projection using POD basis in particular, is effective in inexpensively communicating and reconstructing the high-fidelity ocean forecasts that are often available for specific marine applications. The iLRDMD expands this idea and relates the DMD operator computed in the high-dimensional state space, $\mathbf{A_X}$, to that computed in the low-rank representation space, $\mathbf{A_S}$. We establish the relationship between the DMD operators in the high-dimensional state and low-rank representation space similarly to the result of [20] on compressed sensing DMD. Given a full-dimensional state $\mathbf{x}_j$ and $\mathbf{x}'_j$ for $j = 1, \ldots, k$, where $\mathbf{x}'_j = \mathbf{x}_{j+1}$, we define the matrices

$$\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k], \quad \mathbf{X}'_k = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_k], \tag{1}$$

and the DMD operator as the best least squares fit to

$$\mathbf{X}'_k = \mathbf{A}_{\mathbf{X}_k}\mathbf{X}_k. \tag{2}$$

The DMD operator in the full-dimensional state space A_X can then be computed as

$$\mathbf{A}_{\mathbf{X}_k} = \mathbf{X}'_k\mathbf{X}_k^{\dagger}, \tag{3}$$

where $\mathbf{X}_k^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{X}_k$.

Under a set of tailored POD basis $\mathbf{\Psi}_{n_S} \in \mathbb{R}^{n_X \times n_S}$, where n_S is the number of retained modes, the full-dimensional state has a low-rank representation,

$$\mathbf{x}_j = \mathbf{\Psi}_{n_S}\mathbf{c}_j, \quad \mathbf{x}'_j = \mathbf{\Psi}_{n_S}\mathbf{c}'_j, \quad j = 1, \ldots, k, \tag{4}$$

and matrices $\mathbf{C}_k$ and $\mathbf{C}'_k \in \mathbb{R}^{n_S \times k}$ are defined similarly to their full-dimensional counterparts in (1). Note that since the POD modes $\mathbf{\Psi}_{n_S}$ are orthonormal, $\mathbf{c}_j$ and $\mathbf{c}'_j$ can be computed from the high-dimensional state $\mathbf{x}_j, \mathbf{x}'_j$ by projecting them onto the POD modes as

$$\mathbf{c}_j = \mathbf{\Psi}_{n_S}^T\mathbf{x}_j, \quad \mathbf{c}'_j = \mathbf{\Psi}_{n_S}^T\mathbf{x}'_j, \quad j = 1, \ldots, k. \tag{5}$$

The DMD operator in the low-rank representation space can then also be defined as

$$\mathbf{A}_{\mathbf{S}_k} = \mathbf{C}'_k\mathbf{C}_k^{\dagger}. \tag{6}$$

From (4) we can rewrite $\mathbf{X}_k$ and $\mathbf{X}'_k$ as,

$$\mathbf{X}_k = \mathbf{\Psi}_{n_S}\mathbf{C}_k, \quad \mathbf{X}'_k = \mathbf{\Psi}_{n_S}\mathbf{C}'_k. \tag{7}$$

Substituting (7) in (2) gives,

$$\mathbf{\Psi}_{n_S}\mathbf{C}'_k = \mathbf{A}_{\mathbf{X}_k}\mathbf{\Psi}_{n_S}\mathbf{C}_k, \tag{8}$$

right multiplying the pseudoinverse of $\mathbf{C}_k$ on both sides gives,

$$\mathbf{\Psi}_{n_S}\mathbf{C}'_k\mathbf{C}_k^{\dagger} = \mathbf{A}_{\mathbf{X}_k}\mathbf{\Psi}_{n_S}. \tag{9}$$

Substituting (6) into (9), we obtain

$$\mathbf{\Psi}_{n_S}\mathbf{A}_{\mathbf{S}_k} = \mathbf{A}_{\mathbf{X}_k}\mathbf{\Psi}_{n_S}. \tag{10}$$

This establishes the relationship between the DMD operators defined in different spaces. The relations in (10) can be further expanded to relate the DMD values and vectors of $\mathbf{A}_{\mathbf{S}_k}$ to that of $\mathbf{A}_{\mathbf{X}_k}$, which are the eigenvalues and eigenvectors of the DMD operators. Right multiplying (10) by the eigenvectors of $\mathbf{A}_{\mathbf{S}_k}$, $\mathbf{\Phi}_{\mathbf{S}_k}$,

$$\mathbf{\Psi}_{n_S}\left(\mathbf{A}_{\mathbf{S}_k}\mathbf{\Phi}_{\mathbf{S}_k}\right) = \mathbf{A}_{\mathbf{X}_k}\mathbf{\Psi}_{n_S}\mathbf{\Phi}_{S_k}, \tag{11}$$

then using the eigenvalue problem formulation of $\mathbf{A}_{\mathbf{S}_k}\mathbf{\Phi}_{\mathbf{S}_k} = \mathbf{\Lambda}_{\mathbf{S}_k}\mathbf{\Phi}_{\mathbf{S}_k}$, where $\mathbf{\Lambda}_{\mathbf{S}_k}$ is the diagonal matrix with entries as the eigenvalues of $\mathbf{A}_{\mathbf{S}_k}$, (11) can be rewritten as,

$$\mathbf{\Lambda}_{\mathbf{S}_k}\left(\mathbf{\Psi}_{n_S}\mathbf{\Phi}_{\mathbf{S}_k}\right) = \mathbf{A}_{\mathbf{X}_k}\left(\mathbf{\Psi}_{n_S}\mathbf{\Phi}_{\mathbf{S}_k}\right). \tag{12}$$

Therefore, from (12), $\mathbf{\Lambda}_{\mathbf{S}_k}$ is the eigenvalue and $\mathbf{\Psi}_{n_S}\mathbf{\Phi}_{\mathbf{S}_k}$ is the eigenvector of $\mathbf{A}_{\mathbf{X}_k}$. This essentially lets one to construct the DMD operator using the low-rank representation snapshots and recover the DMD vectors and values of the full-dimensional DMD operator.

2) Weighted Incremental DMD: In classic DMD methods, the DMD modes and values are static in time. Therefore, as the underlying dynamics change in time, the classic DMD's predictive accuracy diminishes. [5], [6] propose two ways of incrementally updating the DMD operator with new snapshots. In this paper, we chose to utilize the weighted incremental DMD (iDMD) method of [5] as it is applicable even when the Sherman-Morrison identity is unavailable. Both of the aforementioned update methods include a weighting factor $\rho_f \leq 1$, which is used to slowly phase out the old snapshots in the "memory" of the DMD operator as shown in (13). Additionally, we deliberately used $\mathbf{c}_j$ as defined earlier, in the iDMD formulation below, because the iLRDMD framework looks to incrementally build the DMD operator in the low-rank representation space $\mathbf{A}_{\mathbf{S}_k}$. The formulation provided below is modified from [5] for consistency of notations and brevity.

$$\begin{aligned}\mathbf{C}_k^{\rho_f} &= [\rho_f^{k-1}\mathbf{c}_1, \rho_f^{k-2}\mathbf{c}_2, \cdots, \mathbf{c}_k] \\ \mathbf{C}_k^{'\rho_f} &= [\rho_f^{k-1}\mathbf{c}'_1, \rho_f^{k-2}\mathbf{c}'_2, \cdots, \mathbf{c}'_k].\end{aligned} \tag{13}$$

Assume $\mathbf{C}_k^{\rho_f}$ and $\mathbf{C}_k^{'\rho_f}$ as defined in (13) and the DMD operator $\mathbf{A}_{\mathbf{S}_k}$ from $\mathbf{C}_k^{'\rho_f} = \mathbf{A}_{\mathbf{S}_k}\mathbf{C}_k^{\rho_f}$ are known. The weighted incremental DMD algorithm finds $\mathbf{A}_{\mathbf{S}_{k+1}}$ given new snapshots $\mathbf{c}_{k+1}$ and $\mathbf{c}'_{k+1}$. The DMD operator at $k+1$-th snapshot, $\mathbf{A}_{\mathbf{S}_{k+1}}$, would be defined from $\mathbf{C}_{k+1}^{'\rho_f} = \mathbf{A}_{\mathbf{S}_{k+1}}\mathbf{C}_{k+1}^{\rho_f}$ where

$$\begin{aligned}\mathbf{C}_{k+1}^{\rho_f} &= [\rho_f^k\mathbf{c}_1, \rho_f^{k-1}\mathbf{c}_2, \cdots, \rho_f\mathbf{c}_k, \mathbf{c}_{k+1}] = [\rho_f\mathbf{C}_k^{\rho_f}, \mathbf{c}_{k+1}] \\ \mathbf{C}_{k+1}^{'\rho_f} &= [\rho_f^k\mathbf{c}'_1, \rho_f^{k-1}\mathbf{c}'_2, \cdots, \rho_f\mathbf{c}'_k, \mathbf{c}'_{k+1}] = [\rho_f\mathbf{C}_k^{'\rho_f}, \mathbf{c}'_{k+1}].\end{aligned} \tag{14}$$

The DMD operator $\mathbf{A}_{\mathbf{S}_{k+1}}$ is then updated from $\mathbf{A}_{\mathbf{S}_k}$ as

$$\mathbf{A}_{\mathbf{S}_{\mathbf{k+1}}} = \mathbf{A}_{\mathbf{S}_k} + \left(\mathbf{c}'_{k+1} - \mathbf{A}_{\mathbf{S}_k}\mathbf{c}_{k+1}\right)\mathbf{v}_{\mathbf{G}_{k,2}}\mathbf{\Sigma}_{\mathbf{C}_{k+1}^{\rho_f}}\mathbf{U}^*_{\mathbf{C}_{k+1}^{\rho_f}}, \tag{15}$$

where the components of the update equation are given as

$$\begin{aligned}\mathbf{U}_{\mathbf{C}_{k+1}^{\rho_f}} &= \mathbf{U}_{\mathbf{C}_k^{\rho_f}}\mathbf{U}_{\mathbf{G}_k}, \quad \mathbf{\Sigma}_{\mathbf{C}_{k+1}^{\rho_f}} = \rho_f\mathbf{\Sigma}_{\mathbf{G}_k}, \\ \mathbf{V}_{\mathbf{C}_{k+1}^{\rho_f}} &= \begin{bmatrix}\mathbf{V}_{\mathbf{C}_k^{\rho_f}}\mathbf{V}_{\mathbf{G}_{k,1}} \\ \mathbf{v}_{\mathbf{G}_{k,2}}\end{bmatrix},\end{aligned} \tag{16}$$

with SVD of $\mathbf{C}_k^{\rho_f}$ is given by $\mathbf{C}_k^{\rho_f} = \mathbf{U}_{\mathbf{C}_k^{\rho_f}}\mathbf{\Sigma}_{\mathbf{C}_k^{\rho_f}}\mathbf{V}^*_{\mathbf{C}_k^{\rho_f}}$. Finally, the components $\mathbf{U}_{\mathbf{G}_k}, \mathbf{\Sigma}_{\mathbf{G}_k}, [\mathbf{V}_{\mathbf{G}_{k,1}}; \mathbf{v}_{\mathbf{G}_{k,2}}]$ are obtained from

$$\mathbf{G}_k \triangleq [\mathbf{\Sigma}_{\mathbf{C}_k^{\rho_f}}, \quad \rho_f^{-1}\mathbf{U}^*_{\mathbf{C}_k^{\rho_f}}\mathbf{c}_{k+1}] = \mathbf{U}_{\mathbf{G}_k}\mathbf{\Sigma}_{\mathbf{G}_k}\begin{bmatrix}\mathbf{V}_{\mathbf{G}_{k,1}} \\ \mathbf{v}_{\mathbf{G}_{k,2}}\end{bmatrix}^*. \tag{17}$$

3) incremental POD: As the underlying dynamics change significantly, the set of tailored basis that the high-dimensional state is projected on may not be as optimal. Therefore, the amount of information that the POD-projected coefficients contain may not be sufficient to incrementally build an accurate DMD operator. In such a case, it is crucial to be able to update the basis at which the high-dimensional state themselves are projected onto, using methods such as [16]. Below, we provide a simple identity that the algorithm for brevity and refer to [16] for specifics of implementation the algorithm. Note that the left singular vectors of the SVD is equivalent to the POD modes [3]. Letting the rank-n_S SVD of matrix $\mathbf{X}_k$ as defined in (1) be

$$\mathbf{X}_k \approx \mathbf{U}_{\mathbf{X}_k}\mathbf{\Sigma}_{\mathbf{X}_k}\mathbf{V}^T_{\mathbf{X}_k}, \tag{18}$$

the SVD of matrix $\mathbf{X}_{k+1} = \begin{bmatrix}\mathbf{X}_k & \mathbf{x}_{k+1}\end{bmatrix}$ would be

$$\begin{bmatrix}\mathbf{U}_{\mathbf{X}_k}\mathbf{\Sigma}_{\mathbf{X}_k}\mathbf{V}^T_{X_k} & \mathbf{x}_{k+1}\end{bmatrix} =$$

$$\begin{bmatrix}\mathbf{U}_{\mathbf{X}_k} & (\mathbf{I} - \mathbf{U}_{\mathbf{X}_k}\mathbf{U}^T_{\mathbf{X}_k})\mathbf{x}_{k+1}/p\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma}_{\mathbf{X}_k} & \mathbf{U}^T_{\mathbf{X}_k}\mathbf{x}_{k+1} \\ \mathbf{0} & p\end{bmatrix}\begin{bmatrix}\mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}\end{bmatrix}, \tag{19}$$

where p is defined as

$$p = ||\mathbf{x}_{k+1} - \mathbf{U}_{\mathbf{X}_k}\mathbf{U}^T_{\mathbf{X}_k}\mathbf{x}_{k+1}||. \tag{20}$$

4) iLRDMD Framework: The iLRDMD framework integrates the three main components outlined to provide adaptive DMD forecasts with reduced communication bandwidth costs. It achieves this by using the tailored set of POD modes to obtain the low-rank representation of the high-dimensional snapshots, which are communicated in batches to the platforms. The platform utilizes these batches of low-rank representation to incrementally update the DMD operator in the low-rank space $\mathbf{A}_{\mathbf{S}}^k$ with the iDMD algorithm. Then a forecast in the high-dimensional space is made using the previously established relationship between $\mathbf{A}_{\mathbf{S}_k}$ and $\mathbf{A}_{\mathbf{X}_k}$, until the new batch of low-rank representation is communicated to the platform. Finally, as the underlying dynamics change significantly over a longer period of time, the iLRDMD framework utilizes the incremental POD algorithm to keep the tailored set of POD modes relevant in time.

B. Data Assimilation for Adaptive Stochastic DMD

In this paper, for DA, we primarily focus on DMD with the GMM-DO filter but also utilize DMD with EnKF for comparison purposes. The GMM-DO filter combines the use of Gaussian mixture models, the Expectation-Maximization (EM) algorithm and the Bayesian Information Criterion (BIC) to accurately approximate distributions based on Monte Carlo data allowing for fast and effective Bayesian inference [13], [21]. The GMM-DO filter couples these concepts with an efficient representation of the evolving probabilistic description of the uncertain dynamical field: the Dynamically Orthogonal field equations. By limiting attention to a dominant evolving stochastic subspace of the complete state space, we can determine the GMM distributions much more quickly and efficiently compared to working in the state space. Much of the following discussion of GMM-DO and related concepts is derived from from [13], [22].

1) Gaussian Mixture Models: The probability density function (pdf) for a random vector $\mathbf{X} \in \mathbb{R}^n$ that is distributed according to a multivariate GMM is

$$p_{\mathbf{X}}(\mathbf{x}) = \sum_{j=1}^{M} \pi_j \times \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}_j, \mathbf{P}_j) \tag{21}$$

subject to

$$\sum_{j=1}^{M} \pi_j = 1. \tag{22}$$

We define the following terms as the mixture complexity (or number of mixture components) $M \in \mathbb{N}$, the mixture weights $\pi_j \in [0, 1]$, the mixture mean vectors $\bar{\mathbf{x}}_j \in \mathbb{R}^n$, and the mixture covariance matrices $\mathbf{P}_j \in \mathbb{R}^{n \times n}$. The multivariate Gaussian density function can given by:

$$\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) = \frac{1}{(2\pi)^{n/2}|\mathbf{P}|^{1/2}} \exp[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{x}})]. \tag{23}$$

GMMs provide a semi-parametric framework in which to approximate unknown distributions [23]. GMMs can essentially be looked at as a flexible alternative to the fully parametric Gaussian distribution where $M = 1$ and the kernel density estimator [24] where $M = N$ (the number of realization points/ensemble members). The fully parametric Gaussian model often forces the data into a structure that may not be realistic and is incapable of modeling multi-modal or largely skewed inputs. The kernel density estimator requires the retention of all N members for inference, a computationally expensive task. Additionally, due to the over fitting that is associated with fitting a kernel to every member, kernel density estimators often necessitate heuristic choosing of the kernel's shape parameter. For these reasons, GMMs are popular due to their efficiency and accuracy at representing complex distributions. For large complexity and small covariance, as a matter of fact, a GMM model converges uniformly to smooth distributions [25]. An example of the three distribution types described above (from [13]) is shown in Fig. 1.

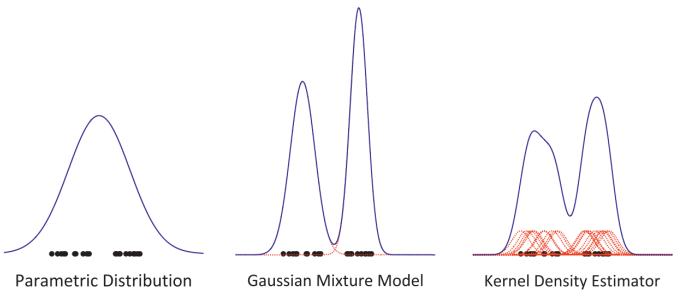

Fig. 1. Parametric (Gaussian) distribution, GMM, and Kernel Density approximation of 20 samples generated from mixture of uniform distributions: $p_X(x) = \frac{1}{2}\,\mathcal{U}(x; -8, -1) + \frac{1}{2}\,\mathcal{U}(x; 1, 8)$ where $\mathcal{U}(x; a, b) = \frac{1}{b-a}$ is the continuous unifrm probability density function for random variable X. Source [13].

2) Expectation-Maximization Algorithm: The EM algorithm describes an iterative procedure for estimating the parameters of a target distribution that maximize the probability of obtaining the available inputs, $\{\mathbf{x}\} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, thus arriving at the Maximum Likelihood (ML) estimate for the unknown set of parameters. The nice thing about ML estimators are that they are consistent and asymptotically efficient [26].

For many realistic cases, obtaining the ML estimate by differentiating the parametric probability distribution with respect to a parameter of interest and equating to zero, gives a nonlinear result that lacks closed form solution [27]. These cases mean that the use of numerical optimization methods is best.

The EM algorithm is often introduced, in the literature, as a method to estimate ML parameters when incomplete inputs or data is present. One crucial step is the *completion* of the data (by imputing from the data that is known like techniques presented in [28]). This data completion allows for the ML solution to be much more tractable and computationally efficient. The competed data problem normally allows for a closed form solution to the estimation problem and allows us to obtain the ML estimation parameters by a simple partial differential. For more detailed description see [29], [30], and [23].

3) Bayesian Inference Criterion: For simplicity, when describing GMMs, the mixture complexity, M, is often assumed to be known and constant. In reality the optimal mixture complexity is often unknown. Determining the best complexity of a GMM can be complicated and there are several methods that exist [23], [31], [32]. For our purposes we utilize the Bayesian Inference Criterion (BIC) that, when minimized, allows us to choose the best mixture complexity, M, that optimally represents the distribution.

4) Dynamically Orthogonal Field Equations: To employ the discussed GMM with EM and BIC efficiently we now discuss the DO equations. The Dynamically Orthogonal (DO) Field Equations [33], [34] are a closed reduced set of evolution equations for general stochastic continuation fields described by stochastic partial differential equations (SPDEs).

The DO equations allow for the stochastic subspace and the stochastic coefficients to be dynamically evolved in time after being initialized based on the initial pdf and evolved. This presents a significant advantage compared to the proper orthogonal decomposition (POD) [35], [36] and polynomial chaos [37], which both fix in time parts of their truncated expansion. For the purpose of this paper, we replace this evolution in time with a different evolution in time, namely the DMD forecast.

5) DMD with GMM-DO Filter: The following is a brief description of DMD with the GMM-DO filter. For a more detailed description of GMM-DO including comparisons and discussions of similar methods see [22]. Note that the discussion of the initial conditions and forecast step differs from [22] and [13] because we use the DMD of the ensemble forecast for our forecast. GMM-DO is an efficient technique that preserves non-Gaussian statistics and respects non-linear dynamics [13].

a) Forecast: The forecast that we use is based on DMD of the ensemble [4]. The approach allows for efficient extraction and forecasting which is advantageous for use where computing power and storage is limited. With an ensemble of N simulations used as training inputs, projected DMD is applied to extract the POD modes $\mathbf{U}_r$, DMD modes $\mathbf{\Phi}$, and the DMD eigenvalues λ_i for $i = 1, \cdots, r$ where r is the reduced rank of the training inputs determined by the DMD algorithm. The DMD coefficients b_q for q ensemble

members, are determined based on initial conditions of each realization/ensemble member. Forecast at time k is then made by the following:

$$\mathbf{x}_{q,k}^f = \mathbf{x}_q^f(k) \approx \mathbf{\Phi} \exp(\mathbf{\Omega} k)\, \mathbf{b}_q \quad q = 1, \cdots, N, \tag{24}$$

where $\mathbf{\Omega}$ is a diagonal matrix of discrete time eigenvalues where the diagonal entries are $\frac{\lambda_i}{\Delta t_{\text{snap}}}$ for $i = \{1, \cdots, r\}$ and Δt_{snap} is the time between snapshots. If no observations are made at time k then the DMD forecast $\mathbf{x}_{q,k}^f$ is used by the vehicle as the forecast at time k. If, however, observations are available (either made by the platform or communicated to the platform) then GMM-DO is used for data assimilation. We start by placing the DMD forecasts into a matrix $\mathbf{X}_k^f$;

$$\mathbf{X}_k = \begin{bmatrix} | & | & & | \\ \mathbf{x}_k^1 & \mathbf{x}_k^2 & \cdots & \mathbf{x}_k^N \\ | & | & & | \end{bmatrix}. \tag{25}$$

We then determine the modes, $\boldsymbol{\chi}_k$, and the subspace coefficients $\{\boldsymbol{\gamma}_k\} = \{\boldsymbol{\gamma}_{1,k}, \cdots, \boldsymbol{\gamma}_{N,k}\}$:

$$\mathbf{x}_k^f = \bar{\mathbf{x}}_k^f + \boldsymbol{\chi}_k^f \boldsymbol{\gamma}_k^f, \tag{26}$$

via a mean subtraction and SVD giving us:

$$\boldsymbol{\chi}_k \mathbf{\Sigma}_k^f {\mathbf{V}_\mathbf{k}^\mathbf{f}}^T = \text{SVD}(\mathbf{x}_k^f - \bar{\mathbf{x}}_k^f), \tag{27}$$

and define the subspace coefficients as $\{\boldsymbol{\gamma}_k\} \equiv \mathbf{\Sigma}_k^f {\mathbf{V}_\mathbf{k}^\mathbf{f}}^T$. Here, $\boldsymbol{\chi}_k \in \mathbb{R}^{n \times s}$ is the matrix of modes forming an orthonormal basis for the subspace at time k.

b) Observation: Presently, we employ a linear observation model:

$$\mathbf{Y}_k = \mathbf{H}\mathbf{X}_k + \mathbf{\Upsilon}_k, \quad \mathbf{\Upsilon} \sim \mathcal{N}(\boldsymbol{\nu}_k; 0, \mathbf{R}), \tag{28}$$

where $\mathbf{Y}_k \in \mathbb{R}^p$ is the observation random vector at time k, $\mathbf{H} \in \mathbb{R}^{p \times n}$ is the linear observation model, and $\mathbf{\Upsilon} \in \mathbb{R}^p$ is the noise vector (assumed to be Gaussian). Note that the observation vector is denoted $\mathbf{y}_k \in \mathbb{R}^p$ and the realized noise vector is denoted $\boldsymbol{\nu}_k \in \mathbb{R}^p$.

c) GMM-DO Update: The update occurs at a discrete time instant and for simplicity we omit the time index k in the following. During the update, the modes are presently unchanged by the observations and thus no prior/forecast or posterior/analysis notation is used on the modes $\boldsymbol{\chi}$. The goal is to update the mean state $\bar{\mathbf{x}}^f$ and the ensemble members $\{\boldsymbol{\gamma}^f\} = \{\boldsymbol{\gamma}_1^f, \cdots, \boldsymbol{\gamma}_N^f\}$ according to (28) and the observations $\mathbf{y}$ to get the GMM-DO posterior estimate:

$$\mathbf{x}_q^a = \bar{\mathbf{x}}^a + \boldsymbol{\chi}\boldsymbol{\gamma}_q^a, \quad q = \{1, \cdots, N\}. \tag{29}$$

GMM representations of prior ensemble set: at discrete time when new set of observations or measurements are obtained, the EM algorithm and BIC allow us to conclude the GMM that best expresses the ensemble set within the stochastic subspace which is significantly less expensive than doing this in the full state space. The parameters of the GMM are denoted:

$$\pi_j^f, \boldsymbol{\mu}_j^f, \mathbf{\Sigma}_j^f, \quad j = 1, \cdots, M, \tag{30}$$

where $\pi_j^f \in [0, 1]$ are the component weights, $\boldsymbol{\mu}_j^f \in \mathbb{R}^s$ is the mean vector of mixture component j in stochastic subspace, and $\mathbf{\Sigma}_j^f \in \mathbb{R}^{s \times s}$ is the covariance matrix of mixture component j in stochastic subspace.

The best mixture complexity is decided by the BIC successively fitting GMMs of increasing complexity using the EM algorithm until a minimum BIC is determined. The end result is a GMM that is best fit to the ensemble set in the stochastic subspace. The resulting prior pdf can be written as follows:

$$p_{\mathbf{\Gamma}^f}(\boldsymbol{\gamma}^f) = \sum_{j=1}^{M} \pi_j^f \times \mathcal{N}(\boldsymbol{\gamma}^f; \boldsymbol{\mu}_j^f, \mathbf{\Sigma}_j^f). \tag{31}$$

A very important property of the GMM-DO filter is that the GMM can be expanded into the state space by:

$$\bar{\mathbf{x}}_j^f = \bar{\mathbf{x}}^f + \boldsymbol{\chi}\boldsymbol{\mu}_j^f \tag{32}$$

and

$$\mathbf{P}_j^f = \boldsymbol{\chi}\mathbf{\Sigma}_j^f\boldsymbol{\chi}^T. \tag{33}$$

The mixture weights remain unchanged when expanding to the state space. The GMM prior distribution in the state space then takes on the form:

$$p_{\mathbf{X}^f}(\mathbf{x}^f) = \sum_{j=1}^{M} \pi \times \mathcal{N}(\mathbf{x}^f; \bar{\mathbf{x}}_j^f, \mathbf{P}_j^f). \tag{34}$$

A crucial property of GMM-DO is that the same distribution would have been obtained if we had performed the fitting to the state space (but the fitting in the subspace is done at a fraction of the cost).

Bayesian Update: Since the dimension of the subspace is much less than the state space ($s \ll n$), for realistic problems performing a GMM-DO filter update step is computationally feasible in the subspace. The following is the update step that determines the posterior distribution in the stochastic subspace $p_{\mathbf{\Gamma}^a}(\boldsymbol{\gamma}^a)$ but would be equivalent to evolving the distribution in the full state space because the modes $\boldsymbol{\chi}$ are unchanged (proof is shown in [13], [22]).

Using the GMM-DO filter we utilize our observation to determine the posterior analysis GMM parameters:

$$\pi_j^a, \boldsymbol{\mu}_j^a, \mathbf{\Sigma}_j^a \quad j = 1, \cdots, M.$$

Generating the posterior ensemble set: We complete the update by utilizing the same idea as in the ESSE Scheme A [38] by generating a posterior ensemble set in the stochastic subspace $\{\boldsymbol{\gamma}^a\} = \{\boldsymbol{\gamma}_1^a, \cdots, \boldsymbol{\gamma}_N^a\}$ in accordance with the a posteriori GMM, $p_{\mathbf{\Gamma}^a}(\boldsymbol{\gamma}^a)$, that has the parameters π_j^a, $\boldsymbol{\mu}_j^a$, $\mathbf{\Sigma}_j^a$ for $j = 1, \cdots, M$.

Now we have arrived at the posterior DO representation for the state vector based on Bayesian data assimilation of the observation $\mathbf{y}$ at time k as:

$$\mathbf{x}_{q,k}^a = \bar{\mathbf{x}}_k^a + \boldsymbol{\chi}_k \boldsymbol{\gamma}_{q,k}^a, \quad q = \{1, \cdots, N\}. \tag{35}$$

d) DMD Coefficients Updates: Now that the update is complete and a posterior ensemble set is obtained, we can then utilize the DMD modes, $\mathbf{\Phi}$ and the new ensemble set to obtain new DMD coefficients, $\mathbf{b}_q$. If time k is set to 0 for simplicity of notation, the coefficients are:

$$\mathbf{b}_q = \mathbf{\Phi}^{\dagger}\mathbf{x}^a_{q,k}, \quad q = 1, \cdots, N, \tag{36}$$

and the forecast at some time t in the future is:

$$\mathbf{x}^f_{q,t} = \mathbf{x}^f_q(t) \approx \mathbf{\Phi}\exp(\mathbf{\Omega}t)\,\mathbf{b}_q \quad q = 1, \cdots, N. \tag{37}$$

III. Application Results

We use a 2D flow behind an island/cylinder as a benchmark test case to validate and develop the iLRDMD adaptive ROM algorithm as well as the DMD with DA schemes. The benchmark test case was run with a Reynolds number of 200 that was determined by a free stream velocity of 2 m/s, the diameter of the island at 1 m, simulation time step of 0.01 s and domain of 20 $m \times 3$ m, leading to 171×21 grid points. The results from this simulation were saved every 20 simulation time-steps as columns of the snapshot matrix and regarded as the 'true' snapshots. The underlying dynamics undergoes significant change in this test case as the laminar flow develops into flow with vortex shedding. A snapshot of the vorticity field is shown in Fig. 2 to exemplify a sample vorticity field (top plot) and two sensors (bottom plot) for flow behind an island/cylinder at $\mathrm{Re} = 200$. Sensor one, labeled s_1, is plotted in a blue solid line and sensor 2, labeled s_2, is plotted in an orange dotted. After the simulations converge to steady-state vortex shedding, snapshots are collected at regular intervals of $20\Delta t$ or $0.2s$.

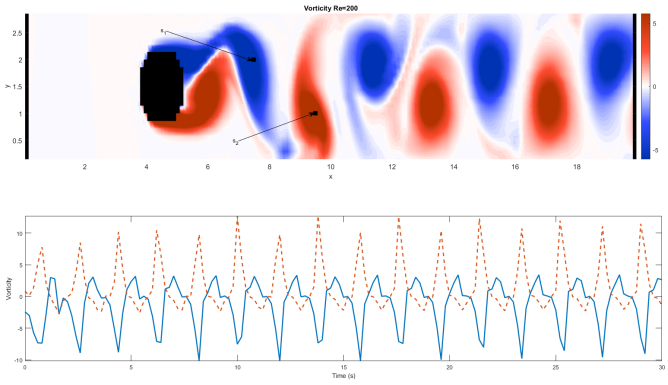

Fig. 2. Example of vorticity field (top plot) and two sensors (bottom plot) for flow behind an island/cylinder at $\mathrm{Re} = 200$. Sensor one, labeled s_1, is plotted in a blue solid line and sensor 2, labeled s_2, is plotted in an orange dotted line.

A. Adaptive ROM Results

In the example shown in Fig. 3, we have used the u velocity to examine key characteristics of iLRDMD and draw comparisons with classic DMD. We used the Pattern Correlation Coefficients (PCC) as an aggregate value that indicates how 'good' our predictions are when compared to the 'true' snapshots from the 2D flow behind an island/cylinder simulation. The PCC is a time mean subtracted pattern coefficient [39]. In general terms, a PCC of 1 is a perfect correlation, 0 no correlation, and -1 a perfect anti-correlation. Both the classic DMD and iLRDMD are initialized with laminar flow before any eddy has occurred. It should be noted that while the classic DMD is computed in the high-dimensional space, the iLRDMD was computed with low-rank representations of that. The low-rank representation was initially computed by projecting the high-dimensional snapshots onto the set of POD modes computed from laminar flow snapshots. In Fig. 3, the classic DMD and iLRDMD both exhibit high PCC in the beginning when the flow is still laminar. However, once the eddies start to form after around the $90-$th snapshot, the PCC values for both methods start to rapidly decay. However, while PCC of the classic DMD continues to decay further, the PCC of iLRDMD experiences some improvements after every 20 snapshots. This is because unlike the classic DMD operator, the iLRDMD operator is being updated with new POD-projected, low-rank snapshots. It should be noted that the improvements are only limited to PCC of around 0.4, because the set of POD modes computed from laminar flow snapshots are no longer as optimal when computing the low-rank representations of the high-dimensional state. Therefore, the amount of information that the DMD operator is being updated with is limited as well. In the iLRDMD framework, this would correspond to a situation in which the underlying dynamics have changed significantly and a new set of tailored set of POD modes needs to be used. In Fig. 3, at 200-th snapshot, the new set of POD modes that now also encompasses snapshots with eddies via the iPOD algorithm is made available. The PCC value sees a significant improvement because the DMD operator is now being updated with low-rank, POD-projected snapshots that are projected onto a more optimal set of POD modes.

The significant improvement at 200-th snapshot can be better seen from Fig. 4 and Fig. 5, which show vorticity of the 'truth' compared with the iLRDMD predictions at 199-th and 200-th snapshot respectively. To compute vorticity, the iLRDMD was computed separately for u and v. In Fig. 4, the set of POD modes that the snapshots are being projected on is solely based on laminar flow snapshots. Thus the iLRDMD predicted vorticity does not resemble the 'truth' field well. However, in Fig. 5, the new set of POD modes is communicated to the platform and the DMD operator is updated with a new batch of snapshots projected onto this set. This means that the DMD operator has now been updated with POD-projected snapshots that contain information about the eddies. Therefore, the iLRDMD predicted vorticity field in Fig. 5 correctly resembles that of the 'truth' field.

Importantly, in the examples shown, we used 60 modes for the tailored set of POD modes that the high-dimensional state was projected on. Therefore, excluding the bandwidth cost of communicating the new set of POD modes, this amounts to around 1.7% in bandwith cost compared to having to communicate high-dimensional state for a typical iDMD. Further demonstrating the effectiveness of iLRDMD in addressing

both the predictive accuracy of the ROMs over time and the communication load.

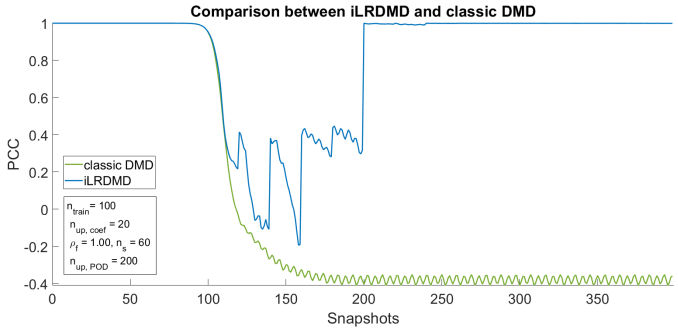

Fig. 3. The pattern correlation coefficients (PCC) for classic DMD and iLRDMD compared with the 'true' snapshots. The DMD operator is updated at every 20 snapshots while the set of POD basis that the low-rank representation is computed on is updated every 200 snapshots.

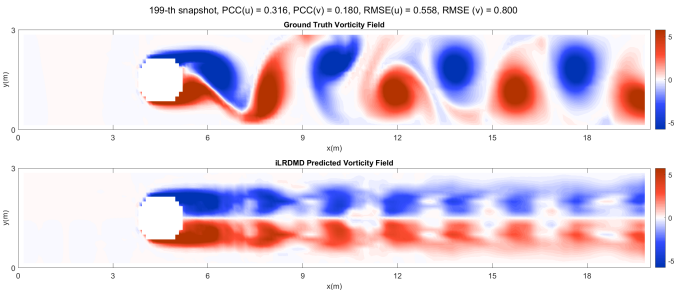

Fig. 4. Vorticity plot of the 'truth' and iLRDMD prediction at 199-th snapshot, before the new set of POD modes are communicated to the platform. PCC and root mean squared error (RMSE) of each u and v prediction on top.

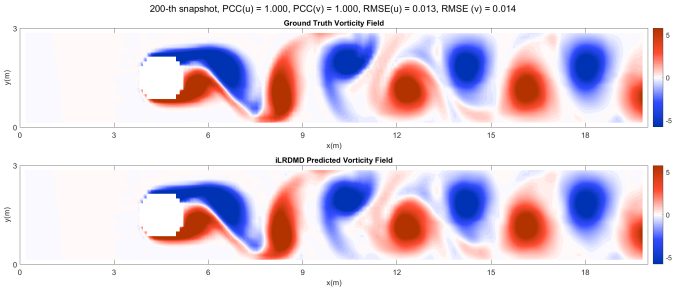

Fig. 5. Vorticity plot of the 'truth' and iLRDMD prediction at 200-th snapshot, after the new set of POD modes are communicated to the platform. PCC and root mean squared error (RMSE) of each u and v prediction on top.

B. DMD with DA Results

With our previously mentioned test case of flow behind an island/cylinder, we generated 'true' snapshots of the flow with a free stream velocity of 2 m/s. To obtain a 30 member ensemble of inputs for testing our DMD with DA schemes, we generated an ensemble with a mean free stream velocity of 2 m/s with perturbations added where the ensemble free stream velocity has a standard deviation of 0.2 m/s. The 'true' field has a Reynolds number of 200 while some of the perturbed ensemble members have Reynolds numbers slightly lower and some slightly higher based on the free stream velocity. Although with the above framework we solve for u, v, density and pressure at each grid-point, in the following example we have used u and v to calculate the vorticity at each point and use the 'true' vorticity as our observed quantity. After vortex shedding begins, we utilize our DMD of the ensemble method to predict the future ensemble set. The test case uses the 30 ensemble members, 25 training snapshots, and makes predictions of the ensemble for 125 snapshots past the last training snapshot.

For visual comparison we calculated the ensemble mean forecast and compared it to the 'truth' at the given time in the future. The mean error field is calculated and displayed as well as the PCC. Fig. 6, Fig. 7, and Fig. 8 show the true field, DMD error field, and PCC values for 0.2 s, 5 s, and 20 s forecast, respectively. As we can see the DMD forecast starts off with a relatively high PCC and low error fields at short forecast times, but within 5 s the error fields are much larger, and by 20s the DMD forecast is very poor. We can see that due to the somewhat cyclical nature of the wake behind the island/cylinder, persistence does a poor job but gets lucky as the pattern repeats itself. These results are similar to what we see in ocean modeling for DMD. This is why utilizing efficient Bayesian data assimilation schemes that extract the information contained in the data prove very beneficial for use with DMD onboard autonomous platforms.

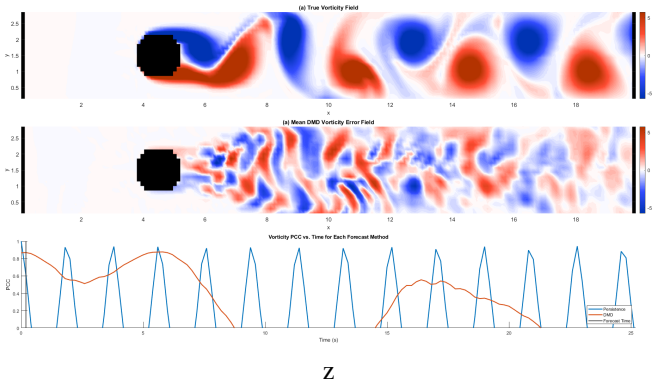

Fig. 6. True vorticity field 0.2 s past the last training snapshot (top plot), DMD error field at 0.2 s past the last training snapshot, and the pattern correlation coefficients (PCC) for persistence and DMD.

The learned dynamics are not necessarily incorrect, but rather the initial conditions used to determine our DMD coefficients, are not close to the 'truth'. When we assimilate data and use this data to calculate new coefficients to make predictions, we can drastically improve our field predictions. In the following example we have observed the vorticity at 25 grid-points at every 0.2 s. These observations are drawn at random from the domain (but do not necessarily have to if the observations were made only by the vehicle itself). Fig. 9, Fig. 10, and Fig. 11 show the true field, DMD with EnKF error field, DMD with GMM-DO error field, and PCC values for 0.2 s, 5 s, and 20 s forecast, respectively. Here we can see that in general the EnKF performs slightly better than DMD

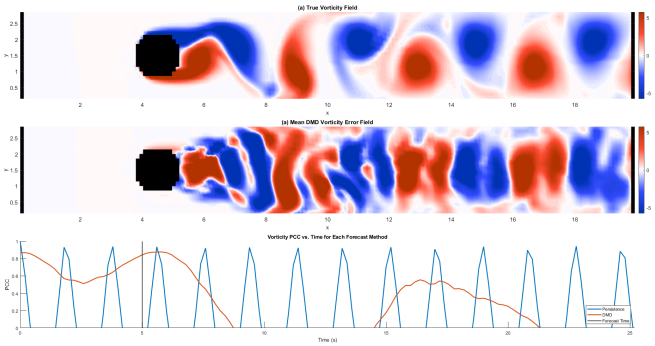

Fig. 7. True vorticity field at 5 s past the last training snapshot (top plot), DMD error field at 5 s past the last training snapshot, and the pattern correlation coefficients (PCC) for persistence and DMD.

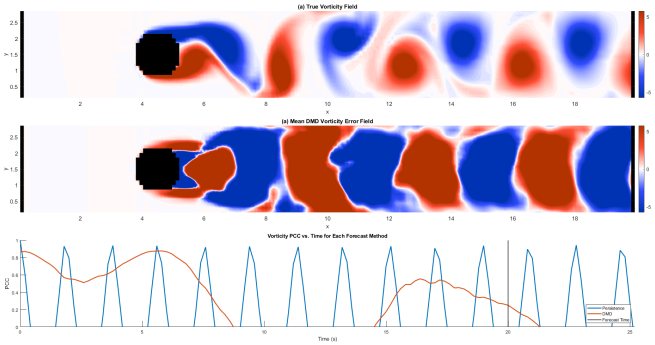

Fig. 8. True vorticity field at 20 s past the last training snapshot (top plot), DMD error field at 20 s past the last training snapshot, and the pattern correlation coefficients (PCC) for persistence and DMD.

forecasts in most instances. On the other hand, the GMM-DO filter performs much better than the EnKF for the duration of our test.

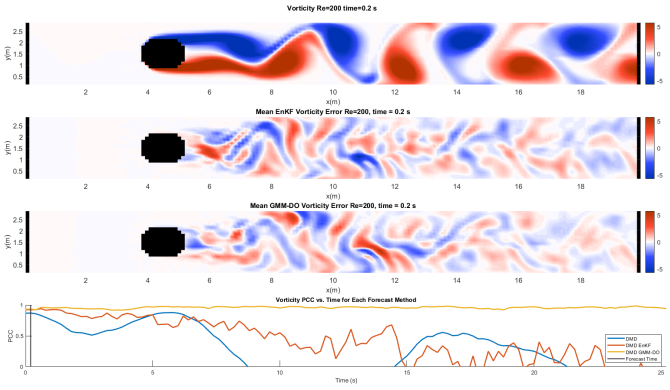

Fig. 9. True velocity field $0.2s$ past the last training snapshot (top plot), DMD with EnKF error field at $0.2s$ past the last training snapshot (second from top), DMD with EnKF error field at $0.2s$ past the last training snapshot (second from bottom), and the pattern correlation coefficients (PCC) for DMD, DMD with EnKF, and DMD with GMM-DO.

IV. Conclusion and Future Work

We explained and demonstrated our adaptive ROM methods for forecast dissemination and our data assimilation for adaptive stochastic DMD technique. We utilized a benchmark

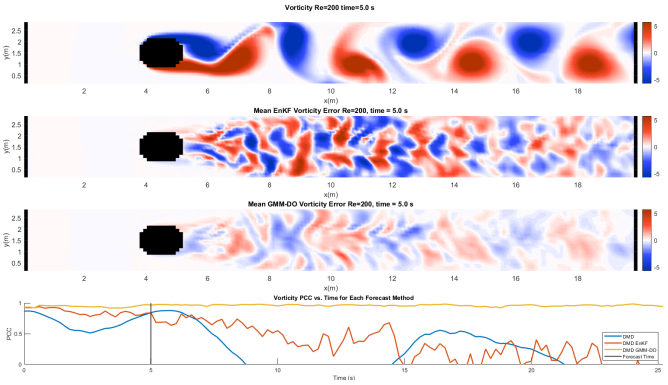

Fig. 10. True velocity field $5s$ past the last training snapshot (top plot), DMD with EnKF error field at $5s$ past the last training snapshot (second from top), DMD with EnKF error field at $5s$ past the last training snapshot (second from bottom), and the pattern correlation coefficients (PCC) for DMD, DMD with EnKF, and DMD with GMM-DO.

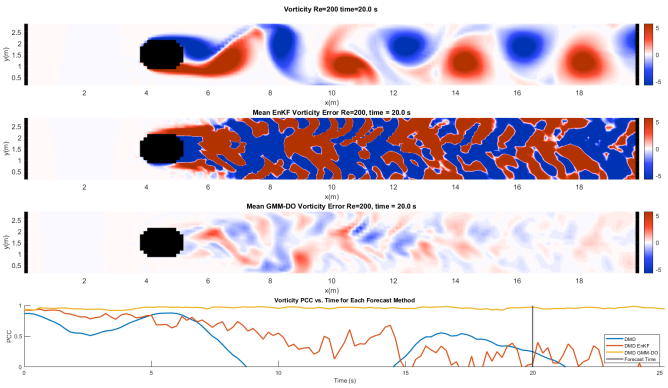

Fig. 11. True velocity field $20s$ past the last training snapshot (top plot), DMD with EnKF error field at $20s$ past the last training snapshot (second from top), DMD with EnKF error field at $20s$ past the last training snapshot (second from bottom), and the pattern correlation coefficients (PCC) for DMD, DMD with EnKF, and DMD with GMM-DO.

test case of flow behind an island/cylinder. In the future we plan to further develop the techniques and demonstrate their effectiveness on more realistic ocean test cases. Another goal is to develop methods to utilize DA to update the DMD models themselves based on data-model misfits and subspace augmentation [9], [10]. Finally, we plan to integrate the adaptive ROM methods we have discussed with our data assimilation for adaptive stochastic DMD. This would enable the autonomous platforms that are unable to receive forecasts computed in remote centers or platforms to make local observations that update their ROMs onboard. This could also be used to augment and update the remote center models that are making forecasts that are periodically sent to the autonomous platforms.

Acknowledgments

We thank all members of the MSEAS group, past and present. We are grateful to the Office of Naval Research for support under STTR grant N6833519C0348 (ROMs) to the Massachusetts Institute of Technology.

REFERENCES

[1] P. F. J. Lermusiaux, T. Lolla, P. J. Haley, Jr., K. Yigit, M. P. Ueckermann, T. Sondergaard, and W. G. Leslie, "Science of autonomy: Time-optimal path planning and adaptive sampling for swarms of ocean vehicles," in *Springer Handbook of Ocean Engineering: Autonomous Ocean Vehicles, Subsystems and Control*, T. Curtin, Ed. Springer, 2016, ch. 21, pp. 481–498.

[2] P. F. J. Lermusiaux, D. N. Subramani, J. Lin, C. S. Kulkarni, A. Gupta, A. Dutt, T. Lolla, P. J. Haley, Jr., W. H. Ali, C. Mirabito, and S. Jana, "A future for intelligent autonomous ocean observing systems," *Journal of Marine Research*, vol. 75, no. 6, pp. 765–813, Nov. 2017, the Sea. Volume 17, The Science of Ocean Prediction, Part 2.

[3] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dyanmic Mode Decomposition: Data-Driven Modeling of Complex Systems*. Philadelphia, Pennsylvania: SIAM, 2016.

[4] J. P. Heuss, P. J. Haley, Jr., C. Mirabito, E. Coelho, M. C. Schönau, K. Heaney, and P. F. J. Lermusiaux, "Reduced order modeling for stochastic prediction onboard autonomous platforms at sea," in *OCEANS 2020 IEEE/MTS*. IEEE, Oct. 2020, pp. 1–10.

[5] M. Alfatlawi and V. Srivastava, "An incremental approach to online dynamic mode decomposition for time-varying systems with applications to eeg data modeling," *Journal of Computational Dynamics*, vol. 7, no. 2, pp. 209–241, Jan. 2020.

[6] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, "Online dynamic mode decomposition for time-varying systems," 2017.

[7] P. F. J. Lermusiaux, "Adaptive modeling, adaptive data assimilation and adaptive sampling," *Physica D: Nonlinear Phenomena*, vol. 230, no. 1, pp. 172–196, 2007.

[8] P. G. Y. Lu and P. F. J. Lermusiaux, "Pde-based bayesian inference of high-dimensional dynamical models," Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, MSEAS Report 19, 2014.

[9] J. Lin and P. F. J. Lermusiaux, "Minimum-correction second-moment matching: Theory, algorithms and applications," *Numerische Mathematik*, vol. 147, no. 3, pp. 611–650, Mar. 2021.

[10] J. Lin, "Bayesian learning for high-dimensional nonlinear systems: Methodologies, numerics and applications to fluid flows," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Sep. 2020.

[11] P. F. J. Lermusiaux, "Evolving the subspace of the three-dimensional multiscale ocean variability: Massachusetts Bay," *Journal of Marine Systems*, vol. 29, no. 1, pp. 385–422, 2001.

[12] ——, "On the mapping of multivariate geophysical fields: Sensitivities to size, scales, and dynamics," *Journal of Atmospheric and Oceanic Technology*, vol. 19, no. 10, pp. 1602–1637, 2002.

[13] T. Sondergaard and P. F. J. Lermusiaux, "Data assimilation with Gaussian Mixture Models using the Dynamically Orthogonal field equations. Part I: Theory and scheme." *Monthly Weather Review*, vol. 141, no. 6, pp. 1737–1760, 2013.

[14] F. Feppon and P. F. J. Lermusiaux, "A geometric approach to dynamical model-order reduction," *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 1, pp. 510–538, 2018.

[15] P. F. J. Lermusiaux, A. R. Robinson, P. J. Haley, and W. G. Leslie, "Advanced interdisciplinary data assimilation: Filtering and smoothing via error subspace statistical estimation," in *Proceedings of The OCEANS 2002 MTS/IEEE conference*. Holland Publications, 2002, pp. 795–802.

[16] G. M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, and C. K., "Limited-memory adaptive snapshot selection for proper orthogonal decomposition," *International Journal for Numerical Methods in Engineering*, vol. 109, pp. 198–217, Jul. 2017.

[17] A. R. Robinson, P. F. J. Lermusiaux, and N. Q. Sloan III, "Data assimilation," in *The Global Coastal Ocean-Processes and Methods*, ser. The Sea, K. H. Brink and A. R. Robinson, Eds. New York: John Wiley and Sons, 1998, vol. 10, chapter 20, pp. 541–594.

[18] A. R. Robinson and P. F. J. Lermusiaux, "Data assimilation for modeling and predicting coupled physical–biological interactions in the sea," in *Biological-Physical Interactions in the Sea*, ser. The Sea, A. R. Robinson, J. J. McCarthy, and B. J. Rothschild, Eds. New York: John Wiley and Sons, 2002, vol. 12, chapter 12, pp. 475–536.

[19] G. Evensen, *Data Assimilation : The Ensemble Kalman Filter*, 2nd ed. Springer, 2009.

[20] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz, "Compressed sensing and dynamic mode decomposition," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 165–191, Jun. 2015.

[21] T. Sondergaard and P. F. J. Lermusiaux, "Data assimilation with Gaussian Mixture Models using the Dynamically Orthogonal field equations. Part II: Applications." *Monthly Weather Review*, vol. 141, no. 6, pp. 1761–1785, 2013.

[22] T. Sondergaard, "Data Assimilation with Gaussian Mixture Models using the Dynamically Orthogonal Field Equation," Master's thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, September 2011.

[23] G. J. McLachlan and D. Peel, *Finite Mixture Models*. Wiley-Interscience, 2000.

[24] B. Silverman, *Density Estimation for Statistics and Data Anlysis*. Chapman and Hall, 1992.

[25] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[26] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*, 2nd ed. Athena Scientific, 2008.

[27] D. Shah, "Algorithms for inference," MIT OpenCourseWare, Dec. 2014. [Online]. Available: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-438-algorithms-for-inference-fall-2014/lecture-notes/

[28] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends in Machine Learning*, vol. 9, no. 1, pp. 1–118, 2016.

[29] G. J. McLachlan and K. E. Basfor, *Mixture Models: Inference and applications to clustering*. Macel Dekker, Inc., 1988.

[30] G. J. McLachlan and T. Krishman, *The EM algorithm and extensions*. Wiley-Interscience, 1997.

[31] I. Eisenberger, "Genesis of bimodal distributions," *Technometrics*, vol. 6, pp. 357–363, 1964.

[32] E. O. Duda, H. P. E., and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2001.

[33] T. P. Sapsis and P. F. J. Lermusiaux, "Dynamically orthogonal field equations for continuous stochastic dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 238, no. 23–24, pp. 2347–2360, Dec. 2009.

[34] T. P. Sapsis, "Dynamically Orthogonal Field Equations for Stochastic Fluid Flows and Particle Dynamics," PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, February 2011.

[35] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.

[36] P. Holmes, J. Lumley, and G. Berkooz, *Turbulence, Coherent Structures, Dynamical Systems, and Symmetry*. Cambridge University Press, 1996.

[37] R. Ghanem and P. Spanos, *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag, 1991.

[38] P. F. J. Lermusiaux and A. R. Robinson, "Data assimilation via Error Subspace Statistical Estimation, part I: Theory and schemes," *Monthly Weather Review*, vol. 127, no. 7, pp. 1385–1407, 1999.

[39] P. F. J. Lermusiaux, "Data assimilation via Error Subspace Statistical Estimation, part II: Mid-Atlantic Bight shelfbreak front simulations, and ESSE validation," *Monthly Weather Review*, vol. 127, no. 7, pp. 1408–1432, Jul. 1999.