Reduced Order Modeling for Stochastic Prediction Onboard Autonomous Platforms at Sea

J. P. Heuss[a], P. J. Haley, Jr.[a], C. Mirabito[a], E. Coelho[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

Global Oceans 2020: Singapore - U.S. Gulf Coast | 978-1-7281-5446-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/IEEECONF38699.2020.9389149

***Abstract*—We describe and investigate several Dynamic Mode Decomposition (DMD) and reduced order projection methods for regional stochastic ocean predictions. We then showcase some of their results as applied to a 300-member set of ensemble forecasts from the POSYDON-POINT sea experiment in the Middle Atlantic–New York Bight region for the period 23–27 August 2018 as well as to a 42-day data-driven reanalysis from the AWACS–SW06 sea experiment in the Middle Atlantic Bight region for the period 14 August to 24 September 2006. We discuss these results for use by autonomous platforms in uncertain scenarios as well the combination of DMD with ideas from large-ensemble forecasting and Dynamically-Orthogonal (DO) differential equations.**

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

I. Introduction

For autonomous unmanned platforms at sea, assessing the regional uncertainties and predicting the likely scenarios for the maritime environment around the platforms is a grand challenge. In light of the high dimensionality of ocean models and of the limited observations, such a probabilistic prediction would indeed be most useful [1], [2]. Due to the operational constraints including onboard power and space limitations, new efficient stochastic reduced order models (ROMs) are needed for onboard predictions. The regional stochastic ROMs could then learn and assimilate information from both remote comprehensive probabilistic ocean forecasts and sparse measurements made by the platforms themselves.

To initiate this research, we first investigated several Dynamic Mode Decomposition (DMD) methods [3] for onboard regional ocean predictions. DMD methods commonly utilize a set of fixed-time snapshots from a single simulation or data set over a period of time, and reduce this set of snapshots to the dynamic modes. The modes are then utilized to forecast further in time from the knowledge of a new initial condition. Two challenges with classic DMD methods are that DMDs are not commonly coupled with a dynamical model, e.g. the original ocean partial differential equations (PDEs), and do not commonly account for uncertainty. The reduced-order dynamically-orthogonal (DO) differential equations approach however directly works with the ocean PDEs and optimizes the instantaneous accuracy of the uncertainty representation [4]–[7]. A long-term research objective is thus to combine DMD ideas of time-space reductions with reduced-order DO ideas, so as to achieve adaptive reduced-order stochastic predictions for onboard autonomous platforms.

In what follows, we first describe each of the DMD methods we have evaluated as well as a few reduced order projection and subspace methods. We then showcase some of their results as applied to a 300-member set of ensemble forecasts from the POSYDON-POINT experiment in the Middle Atlantic - New York Bight region for the period 23–27 August 2018 [8] as well as to a 42-day data-driven reanalysis from the Autonomous Wide Aperture Cluster for Surveillance (AWACS) and Shallow-Water 06 (SW06) sea experiments in the Middle Atlantic Bight region for the period 14 August to 24 September 2006 [9]–[12]. Finally, we discuss these results for use by autonomous platforms in uncertain scenarios, initiating the combination of DMD and DO ideas.

II. Methodology

A. Dynamic Mode Decomposition

Dynamic Mode Decomposition (DMD) was originally developed by Schmid [13], [14]. DMD allows for complicated nonlinear data sets to be decomposed into spatiotemporal coherent structures. The equation-free nature of DMD is attractive because it can be used for a wide range of data and provide useful diagnostic, prediction, and control methods. A major promise of DMD is the ability to synthesize data from simulations, experiments, or environmental measurements into accurate Reduced Order Models (ROMs). There have been many methods of DMD developed in its short history and improvement of DMD methods is an area of ongoing research.

B. DMD Architecture

The DMD architecture normally considers a continuous-time dynamical system

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t; \mu) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the vector representing the state of the system at time t, μ contains the parameters of the system, and $\mathbf{f}(\cdot)$ is the dynamics. We are typically concerned with systems with $n \gg 1$ (n representing the size of the state space), as such large systems correspond to the discretization of PDEs at many discrete locations in space. We can further discretize (1) in time or sample the solution at every Δt. Denoting the discrete time index by subscript k such that $\mathbf{x}_k = \mathbf{x}(k\Delta t)$, measurements (or estimates) of the system are collected at the discrete intervals from $k = 1, 2, ..., m$. Numerical solutions are typically used to predict future states solutions of (1) as analytical solutions cannot often be determined. DMD uses an equation-free idea where the right hand side of (1) does not need to be known. Instead, past discrete solutions or direct measurements are used as inputs by DMD to approximate the dynamics and allow for future state prediction.

DMD constructs the approximate linear representation of (1) as

$$\frac{d\mathbf{x}}{dt} = \mathfrak{B}\mathbf{x}. \tag{2}$$

If the initial condition is $\mathbf{x}(0) = \mathbf{x}_1$, then the solution is [15]

$$\mathbf{x}(t) = \sum_{i=1}^{n} \boldsymbol{\phi}_i \exp(\omega_i\, t) b_i = \boldsymbol{\phi} \exp(\boldsymbol{\Omega}\, \mathbf{t})\mathbf{b}, \tag{3}$$

where $\boldsymbol{\phi}_i$ and ω_i are the eigenvectors and eigenvalues of the matrix $\mathfrak{B}$ and b_k contains the coordinates of $\mathbf{x}_1$ in the eigenvector basis. Methods for computing $\mathbf{b}$ are discussed in descriptions of DMD methods in the following section. $\boldsymbol{\phi}$ is a matrix whose columns are made up by the eigenvectors $\boldsymbol{\phi}_i$ and $\boldsymbol{\Omega}$ is the matrix whose diagonals are ω_i.

From the given continuous dynamics (2), it is possible to construct a discrete-time system given by,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k\,, \tag{4}$$

where

$$\mathbf{A} = \exp(\mathfrak{B}\Delta t)\,. \tag{5}$$

Here, $\mathfrak{B}$ is the continuous-time dynamics matrix in (2) and Δt is the fixed interval between time steps. The eigenvectors and eigenvalues of $\mathbf{A}$ are referred to as the DMD modes ($\boldsymbol{\phi}_j$) and DMD eigenvalues (λ_j) respectively. The solution to (4) can then be given by

$$\mathbf{x}_{k+1} = \sum_{j=1}^{n} \boldsymbol{\phi}_j \lambda_j^k b_j = \boldsymbol{\phi}\boldsymbol{\Lambda}^k\mathbf{b}\,. \tag{6}$$

A DMD method estimates the low-rank eigendecomposition of matrix $\mathbf{A}$ such that

$$\|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k\|_2 \tag{7}$$

is minimized for times $k = 1, 2, ..., m-1$. The optimality holds over the training window in which $\mathbf{A}$ is constructed and can be used for future predictions beyond the window.

To minimize the error (7) using the sample set of snapshots from $k = 1, 2, \cdots, m$, two matrices are formed (they will be the inputs to the DMD algorithms). For a sequential set of column vectors $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ where each $\mathbf{x}_k \in \mathbb{R}^n$, matrices $\mathbf{X}$ and $\mathbf{X}'$ are formed as follows

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix} \tag{8}$$

and

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \tag{9}$$

Considering $\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k)$ where $\mathbf{F}$ is the map corresponding to the evolution of (1), DMD computes the eigenvalues and eigenvectors of the best-fit liner operator $\mathbf{A}$ that relates $\mathbf{X}' \approx \mathbf{A}\mathbf{X}$. If the size of the state space is small, $\mathbf{A}$ could be computed as $\mathbf{A} = \mathbf{X}'\mathbf{X}^\dagger$, where $\dagger$ indicates the Moore-Penrose pseudoinverse. This is not normally practical when the state space is large, so DMD methods provide alternatives to find the eigendecomposition of $\mathbf{A}$ which then allows for future-state predictions.

C. DMD Methods

In this paper, we utilize and compare several methods for reduced-order regional ocean prediction. Although many of the DMD methods can be used with imaginary inputs, we denote the algorithms assuming real inputs since the ocean simulation data we use are real numbers.

1) Projected DMD: For projected DMD the snapshots must be in order (which is not a requirement for some other methods such as exact DMD). The algorithm produces a low-rank matrix projected onto Proper Orthogonal Decomposition (POD) modes to improve efficiency when computing the eigenvectors (DMD modes) and eigenvalues of the time-stepping matrix $\mathbf{A}$ [3]. The DMD modes are projected onto POD modes, hence the name projected DMD.

Algorithm: Projected DMD:

1. Arrange the inputs into sequential snapshot matrices $\mathbf{X}$ and $\mathbf{X}'$ as in (8) and (9).

2. Compute the compact Singular Value Decomposition (SVD) of $\mathbf{X}$ such that

$$\mathbf{X} \approx \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^T \tag{10}$$

where $\mathbf{U}_r \in \mathbb{R}^{n\times r}$, $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$ and, $\mathbf{V}_r \in \mathbb{R}^{(m-1)\times r}$, r is the reduced rank of $\mathbf{X}$, and $\mathbf{V}_r^T$ denotes the transpose of matrix $\mathbf{V}_r$. The matrix $\mathbf{U}_r$ (the left singular vectors) are the proper orthogonal decomposition (POD) modes.

3. The matrix $\mathbf{A}$ could be computed as follows

$$\mathbf{A} = \mathbf{X}'\mathbf{V}_r\boldsymbol{\Sigma}_r^{-1}\mathbf{U}_r^T. \tag{11}$$

In practice though, computing $\mathbf{A}$ is extremely expensive so instead, it is much more computationally efficient to define a matrix $\tilde{\mathbf{A}}$ which is the $r \times r$ projection of $\mathbf{A}$ onto POD modes as

$$\tilde{\mathbf{A}} \overset{\text{def}}{=} \mathbf{U}_r^T\mathbf{X}'\mathbf{V}_r\boldsymbol{\Sigma}_r^{-1}. \tag{12}$$

where the left/right multiply of (11) by $\mathbf{U}_r^T/\mathbf{U}_r$ projected onto the POD modes.

4. The eigenvalues and eigenvectors of $\tilde{\mathbf{A}}$ are determined by

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\boldsymbol{\Lambda}, \tag{13}$$

where column vectors of $\mathbf{W}$ are the eigenvectors of $\tilde{\mathbf{A}}$ and diagonals of $\boldsymbol{\Lambda}$ are the corresponding eigenvalues $\tilde{\lambda}_j$. The eigenvalues of $\tilde{\mathbf{A}}$ are also the non-zero eigenvalues of the much larger matrix $\mathbf{A}$. The projected DMD modes (which are the estimated non-zero eigenvectors of the full matrix $\mathbf{A}$) are given by the column vectors of the matrix,

$$\boldsymbol{\Phi} = \mathbf{U}_r\mathbf{W}. \tag{14}$$

5. To perform state reconstruction or future-state prediction, for convenience we define a matrix $\mathbf{\Omega}$ whose diagonal entries are made up by $\omega_j = \frac{\ln(\lambda_j)}{\Delta t}$. Then the predicted DMD solution at some time t is given by

$$\mathbf{x}(t) \approx \mathbf{\Phi} \exp(\mathbf{\Omega} t)\, \mathbf{b}. \tag{15}$$

Here $\mathbf{b}$ is a vector of DMD amplitudes. If the initial snapshot $\mathbf{x}_1$ is at time $t_1 = 0$, then $\mathbf{x}_1 = \mathbf{\Phi}\,\mathbf{b}$ and thus

$$\mathbf{b} = \mathbf{\Phi}^{\dagger}\mathbf{x}_1. \tag{16}$$

This can be expensive if the size of $\mathbf{x}_1$ is large. As an alternative, DMD amplitudes can be calculated much more inexpensively using POD projected data [16]. If we consider $\tilde{\mathbf{A}}$ defines the linear model for the dynamics such that $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\,\tilde{\mathbf{x}}_k$, then we can compute the DMD amplitudes using the following:

$$\mathbf{x}_1 = \mathbf{\Phi}\mathbf{b} \tag{17a}$$
$$\mathbf{U}_r\tilde{\mathbf{x}}_1 = \mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{b} \tag{17b}$$
$$\tilde{\mathbf{x}}_1 = \mathbf{U}_r^T\mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}\mathbf{W}\mathbf{b} \tag{17c}$$
$$\tilde{\mathbf{x}}_1 = \tilde{\mathbf{A}}\mathbf{W}\mathbf{b} \tag{17d}$$
$$\tilde{\mathbf{x}}_1 = \mathbf{W}\mathbf{\Lambda}\mathbf{b} \tag{17e}$$
$$\mathbf{b} = (\mathbf{W}\mathbf{\Lambda})^{-1}\tilde{\mathbf{x}}_1. \tag{17f}$$

2) Exact DMD: Exact DMD solves the same problem as projected DMD but computes the exact DMD modes of $\mathbf{A}$ rather than projecting onto POD modes [3]. These exact modes are determined all while still avoiding explicitly computing $\mathbf{A}$. Indeed, as showed by Tu et. al. [17], the dominant r exact eigenvectors of matrix $\mathbf{A}$ can be computed from the eigendecomposition of $\tilde{\mathbf{A}}$. Besides determining the exact DMD modes, as long as input pairs $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are in the same columns of $\mathbf{X}$ and $\mathbf{X}'$ respectively, another benefit is that the calculated eigenvalues and eigenvectors of $\mathbf{A}$ will be the same regardless of the order of the snapshot pairs.

Algorithm: Exact DMD:

1. Arrange the inputs into snapshots matrices $\mathbf{X}$ and $\mathbf{X}'$ as in (8) and (9).

2. Perform SVD as in equation (10), compute $\tilde{\mathbf{A}}$ as in (12), and perform eigendecomposition as in (13).

3. The exact DMD modes are given by the column vectors of the matrix

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}\mathbf{W}. \tag{18}$$

4. The state solution at time t can be predicted using (15).

3) Compressed DMD: Compressed DMD solves the Euclidean norm minimization problem, but it compresses the inputs first [18]. With matrices $\mathbf{X}$ and **X'**, it is possible to compress the inputs, compute DMD on the compressed inputs and reconstruct DMD modes and eigenvalues of the full-state by linearly combining full-state snapshots according to the compressed DMD transformation. This DMD computation is much faster than DMD on full-state data particularly for large data sets. It should be noted that the compressed DMD here refers to the spatial compression.

Compressed DMD relies on two essential conditions. First, the snapshots have to be sparse in some basis given by the columns of $\mathbf{\Psi}$, so that $\mathbf{X} = \mathbf{\Psi}\mathbf{S}$ and $\mathbf{X}' = \mathbf{\Psi}\mathbf{S}'$. Here, $\mathbf{S}$ and $\mathbf{S}'$ have sparse columns and are considered the sparse portions of $\mathbf{X}$ and $\mathbf{X}'$ respectively. The basis $\mathbf{\Psi} \in \mathbb{R}^{n\times n}$ can be Fourier, wavelet, or it can be the first $p-$dominant POD modes found by the initial SVD (a potential implementation strategy would be to re-evaluate the POD modes from time to time to reflect the change in the dynamics) [19]. Next, we consider a (pseudo)-measurement matrix $\mathbf{C} \in \mathbb{R}^{p\times n}$, where $p < n$, that must be incoherent with respect to the sparse basis $\mathbf{\Psi}$, i.e. rows of $\mathbf{C}$ are uncorrelated with columns of $\mathbf{\Psi}$. This will hold true generally as long as $\mathbf{C}$ is a Gaussian random measurement matrix. If we assume each column of $\mathbf{X}$ and $\mathbf{X}'$ is in the same sparse subspace of the basis $\mathbf{\Psi}$, then we can ensure that the POD modes and DMD modes will also be in the same sparse subspace.

Algorithm: Compressed DMD:

1. Arrange the inputs into snapshots matrices $\mathbf{X}$ and $\mathbf{X}'$ as in (8) and (9).

2. Determine the reduced number of (pseudo)-measurements to be used, p, and use a random matrix (or Gaussian random matrix) $\mathbf{C} \in \mathbb{R}^{p\times n}$ to compress the inputs and obtain reduced sets as follows:

$$\mathbf{Y} = \mathbf{C}\mathbf{X} \text{ and } \mathbf{Y}' = \mathbf{C}\mathbf{X}'. \tag{19}$$

3. Compute exact DMD on $(\mathbf{Y}, \mathbf{Y}')$ as in (10), (12), (13), and (18) replacing $\mathbf{X}$ with $\mathbf{Y}$ and $\mathbf{X}'$ with $\mathbf{Y}'$ to obtain $(\mathbf{\Lambda}_{\mathbf{Y}}, \mathbf{W}_{\mathbf{Y}})$ and $\mathbf{\Phi}_{\mathbf{Y}}$.

4. The full-state DMD modes can then be constructed using

$$\mathbf{\Phi} = \mathbf{X}'\mathbf{V}_{\mathbf{Y}}\mathbf{\Sigma}_{\mathbf{Y}}^{-1}\mathbf{W}_{\mathbf{Y}} \tag{20}$$

5. With the DMD modes ($\mathbf{\Phi}$) and the DMD eigenvalues (diagonal entries of $\mathbf{\Lambda}_{\mathbf{Y}}$), the DMD future-state prediction for the full-state uses (15).

4) Total DMD: Conventional DMD methods often fail to accurately capture the underlying dynamics when snapshot data contain significant sensor noise. Total DMD projects each input time snapshot and its sequential pair onto a joint subspace. When the sensor noise remains mostly in the orthogonal complement of the joint subspace, this reduces noise in the inputs to the DMD algorithm. This leads to a slight cost increase but allows DMD to capture the dynamical descriptions much more effectively. Conventional DMD methods (such as Exact and Projected) minimize errors with respect to the time-shifted matrix $\mathbf{X}'$ only. This yields a biased analysis when the snapshots exhibit noise. Total DMD, however, minimizes the orthogonal distance between the linear fit involving both initial $\mathbf{X}$ and final $\mathbf{X}'$ states, allowing a de-biased analysis [20].

Mathematically, instead of minimizing the residual $\|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$ as in other methods, Total DMD [20] minimizes $\left\|\begin{bmatrix}\Delta\mathbf{X}\\ \Delta\mathbf{X}'\end{bmatrix}\right\|_F$ subject to $\mathbf{X}'+\Delta\mathbf{X}'=\mathbf{A}(\mathbf{X}+\Delta\mathbf{X})$. This total-least-squares problem can then be solved by projecting $\mathbf{X}$ and $\mathbf{X}'$ onto an augmented subspace created from the matrix $\begin{bmatrix}\mathbf{X}\\ \mathbf{X}'\end{bmatrix}$.

Algorithm: Total DMD:

1. Arrange the inputs into snapshots matrices $\mathbf{X}$ and $\mathbf{X}'$ as in (8) and (9).

2. Determine the best subspace by compact SVD an augmented matrix $\mathbf{Z}$

$$\mathbf{Z}=\begin{bmatrix}\mathbf{X}\\ \mathbf{X}'\end{bmatrix}=\mathbf{U_Z\Sigma_Z V_Z^{\mathit{T}}}. \tag{21}$$

3. Project $\mathbf{X}$ and $\mathbf{X}'$ onto the subspace, resulting in noise-free / reduced inputs $\mathbf{Y}$ and $\mathbf{Y}'$ as follows:

$$\mathbf{Y}=\mathbf{X}\mathbf{V_Z}\mathbf{V_Z}^T \text{ and } \mathbf{Y}'=\mathbf{X}'\mathbf{V_Z}\mathbf{V_Z}^T. \tag{22}$$

4. Perform DMD using (10), (12), (13), and (18) except that wherever $\mathbf{X}$ is used substitute $\mathbf{Y}$ and wherever $\mathbf{X}'$ is used substitute $\mathbf{Y}'$. Instead of (18), (14) could be used if projected DMD modes are preferred over the exact DMD modes.

5. Use (15) to compute the state space prediction at time t.

5) Optimized DMD: A pitfall of many of the DMD methods (e.g. projected, exact, compressed) is that the computed eigenvalues are biased by the presence of sensor noise in the inputs (only residuals of the linear system are minimized). Total DMD is one of the de-biasing methods that tries to overcome this by minimizing both the initial and final errors. Alternatively, Optimized DMD [21] addresses the issues of biases due to sensor noise by creating an optimization problem where the identified linear operator has a fixed rank. The standard DMD methods treat the data pairwise, snapshot to snapshot, rather than as a whole, and favor one direction (forward in time). Optimized DMD allows the reconstruction errors to be distributed throughout, eliminating noise.

Algorithm: Optimized DMD:

1. Arrange inputs into a single matrix made up of all the snapshots:

$$\mathbf{X}=\begin{bmatrix}| & | & & |\\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m\\ | & | & & |\end{bmatrix}. \tag{23}$$

2. Suppose $\mathbf{x}(t)$ is the solution to $\dot{\mathbf{x}}(t)=\mathfrak{B}\mathbf{x}(t)$. With initial condition $\mathbf{x}_1$, the analytical solution is:

$$\mathbf{x}(t)=\exp(\mathfrak{B}t)\mathbf{x}_1. \tag{24}$$

3. Assume that the matrix $\mathfrak{B}$ is diagonalizable such that $\mathfrak{B}=\mathbf{S}\boldsymbol{\Lambda}\mathbf{S}^{-1}$ where $\mathbf{S}\in\mathbb{R}^{n\times r}$ and $\boldsymbol{\Lambda}\in\mathbb{R}^{r\times r}$ for a target rank r. Therefore, $\mathbf{x}(t)$ can be rewritten:

$$\mathbf{x}(t)\approx\mathbf{S}\exp(\boldsymbol{\Lambda}t)\mathbf{S}^\dagger\mathbf{x}_1. \tag{25}$$

4. If we let the diagonals of $\boldsymbol{\Lambda}$ be given by $\alpha_1,\alpha_2,\ldots,\alpha_r$ and define the matrix basis function or time matrix $\boldsymbol{\Omega}(\alpha)$ with entries $\Omega(\alpha)_{k,j}=exp(\alpha_j t_k)$ for m sample times t_k, $\mathbf{X}^T$ can be written:

$$\mathbf{X}^T\approx\boldsymbol{\Omega}(\alpha)\mathbf{B}, \tag{26}$$

where $B_{i,j}=S_{j,i}\left(\mathbf{S}^\dagger\mathbf{x}_1\right)_i$ are the entries of $\mathbf{B}$. Here, $S_{j,i}$ is the j-th row and i-th column of matrix $\mathbf{S}$ and $(\mathbf{S}^\dagger\mathbf{x}_1)_i$ is the i-th entry of the vector $(\mathbf{S}^\dagger\mathbf{x}_1)$.

5. Suppose $\hat{\alpha}$ and $\hat{\mathbf{B}}$ are solutions to minimization of $\left\|\mathbf{X}^T-\boldsymbol{\Omega}(\alpha)\mathbf{B}\right\|_F$ (solved using a variable projection algorithm [22], [23]).

6. Optimized DMD eigenvalues are defined by $\lambda_i=\hat{\alpha}_i$ and the DMD modes are

$$\boldsymbol{\phi}_i=\frac{1}{\left\|\hat{\mathbf{B}}^T(:,i))\right\|_2}\hat{\mathbf{B}}^T(:,i), \tag{27}$$

where $\hat{\mathbf{B}}^T(:,i)$ is the i-th column of $\hat{\mathbf{B}}^T$.

7. If we define $b_i=\left\|\hat{\mathbf{B}}^T(:,i)\right\|_2$, future-state prediction at time t_k can then be made with the following equation:

$$\mathbf{x}_k=\sum_{i=1}^{r}b_i\exp(\lambda_i\,t_k)\,\boldsymbol{\phi}_i. \tag{28}$$

6) Streaming DMD: Most DMD methods view DMD as a post-processing tool, meaning we require a large number of inputs to extract spatial and temporal modes for analysis. There are many instances where an online and incrementally updated method would be advantageous. A streaming method, where a snapshot pair is evaluated as it is received, can minimize data storage [24]. Streaming DMD allows for DMD to be performed incrementally as new snapshot pairs are available. We arrange snapshot pairs $\mathbf{x}_k\in\mathbb{R}^n$ and $\mathbf{x}_{k+1}\in\mathbb{R}^n$ that are spaced a fixed time interval apart (Δt) stored in $\mathbf{X}$ and $\mathbf{X}'$ as in (8) and (9).

First we must compute a matrix $\mathbf{Q_x}\in\mathbb{R}^{n\times r}$ (where r is the reduced rank of $\mathbf{X}$ and $\mathbf{X}'$) whose columns form orthonormal basis for the image of $\mathbf{X}$. The DMD operator is then

$$\mathbf{K}=\mathbf{Q_X}\tilde{\mathbf{K}}\mathbf{Q_X}^T, \tag{29}$$

where $\tilde{\mathbf{K}}$ is a $r\times r$ matrix defined by

$$\tilde{\mathbf{K}}\overset{\text{def}}{=}\mathbf{Q_X}^T\mathbf{X}'\mathbf{X}^\dagger\mathbf{Q_X}. \tag{30}$$

The DMD modes and eigenvalues are the eigenvectors and eigenvalues of $\mathbf{K}$ which may be computed from the much smaller matrix $\tilde{\mathbf{K}}$.

For streaming DMD situations we assume that we only have access to a pair of snapshhots $(\mathbf{x}_k,\mathbf{x}_{k+1})$ at time k. There is then an alternate way to compute $\tilde{\mathbf{K}}$ to allow for it to be incrementally updated as new snapshots are available. To do this, we need to determine the orthonomal basis for $\mathbf{X}$ and $\mathbf{X}'$ creating column matrices $\mathbf{Q_x}\in\mathbb{R}^{n\times r}$ and $\mathbf{Q}'_\mathbf{x}\in\mathbb{R}^{n\times r}$. Next, we project $\mathbf{X}$ and $\mathbf{X}'$ onto the respective orthogonal base $\mathbf{Y}\overset{\text{def}}{=}\mathbf{Q_X}^T\mathbf{X}$ and $\mathbf{Y}'\overset{\text{def}}{=}\mathbf{Q_{X'}}^T\mathbf{X}'$. Then, we can define new matrices $\mathbf{C} = \mathbf{Y}'\mathbf{Y}^T$, $\mathbf{G_X} = \mathbf{Y}\mathbf{Y}^T$, and $\mathbf{G_{X'}} = \mathbf{Y}'\mathbf{Y}'^T$ which allows us to rewrite $\tilde{\mathbf{K}}$ as

$$\tilde{\mathbf{K}} = \mathbf{Q}_\mathbf{X}^T \mathbf{Q}_{\mathbf{X}'} \mathbf{C} \mathbf{G}_\mathbf{X}^\dagger. \tag{31}$$

Advantages of this formulation is that $\tilde{\mathbf{K}}$ can be updated incrementally and much less storage is required particularly for streams where m is already extremely large and increasing. Methods that learn subspaces and complete rank updates can be used [6], [25]–[29].

Algorithm: Streaming DMD:
As a new snapshot pair is acquired, an iteration to update the DMD is performed as follows:

1. For the new snapshot pair $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, compute the residuals

$$e_\mathbf{X} = (\mathbf{I} - \mathbf{Q_X}\mathbf{Q}_\mathbf{X}^T)\mathbf{x}_k \tag{32}$$

and

$$e_{\mathbf{X}'} = (\mathbf{I} - \mathbf{Q}_{\mathbf{X}'}\mathbf{Q}_{\mathbf{X}'}^T)\mathbf{x}_{k+1} \tag{33}$$

where $\mathbf{I}$ is the appropriate dimension Identity matrix.

2. If $\|e_\mathbf{X}\| > \epsilon$ or $\|e_{\mathbf{X}'}\| > \epsilon$ (where ϵ is a user defined tolerance), the dimension of the corresponding basis, $\mathbf{Q_X}$ or $\mathbf{Q}_{\mathbf{X}'}$, is increased by appending an additional column $e_\mathbf{X}/\|e_\mathbf{X}\|$ or $e_{\mathbf{X}'}/\|e_{\mathbf{X}'}\|$ respectively. $\mathbf{G_X}$, $\mathbf{G}_{\mathbf{X}'}$, and $\mathbf{C}$ must be zero-padded to maintain dimensional consistency.

3. If either basis, $\mathbf{Q_X}$ or $\mathbf{Q}_{\mathbf{X}'}$, becomes too large ($r > r_0$), compute leading eigenvectors of $\mathbf{G_X}$ and $\mathbf{G}_{\mathbf{X}'}$ ($\mathbf{W_X}$ and $\mathbf{W}_{\mathbf{X}'}$ respectively), then set

$$\mathbf{G_X} \leftarrow \mathbf{W}_\mathbf{X}^T \mathbf{G_X} \mathbf{W_X},$$

$$\mathbf{G}_{\mathbf{X}'} \leftarrow \mathbf{W}_{\mathbf{X}'}^T \mathbf{G}_{\mathbf{X}'} \mathbf{W}_{\mathbf{X}'},$$

$$\mathbf{C} \leftarrow \mathbf{W}_{\mathbf{X}'}^T \mathbf{C} \mathbf{W_X},$$

$$\mathbf{Q_X} \leftarrow \mathbf{Q_X} \mathbf{W_X},$$

and

$$\mathbf{Q}_{\mathbf{X}'} \leftarrow \mathbf{Q}_{\mathbf{X}'} \mathbf{W}_{\mathbf{X}'}.$$

Here, r_0 is a pre-specfied maximum allowable matrix rank at which the truncation step occurs.

4. Next, set

$$\mathbf{y}_k = \mathbf{Q}_\mathbf{X}^T \mathbf{x}_k \tag{34}$$

and

$$\mathbf{y}_{k+1} = \mathbf{Q}_{\mathbf{X}'}^T \mathbf{x}_{k+1}. \tag{35}$$

Then, let

$$\mathbf{G_X} \leftarrow \mathbf{G_X} + \mathbf{y}_k \mathbf{y}_k^T,$$

$$\mathbf{G}_{\mathbf{X}'} \leftarrow \mathbf{G}_{\mathbf{X}'} + \mathbf{y}_{k+1} \mathbf{y}_{k+1}^T,$$

and

$$\mathbf{C} \leftarrow \mathbf{C} + \mathbf{y}_{k+1} \mathbf{y}_k^T.$$

5. If DMD modes and/or eigenvalues are needed at the end of an iteration, compute the eigenvectors and eigenvalues of $\mathbf{C}\mathbf{G}_\mathbf{X}^\dagger$. If $\mathbf{v}_j$ is the j-th eigenvector of $\mathbf{C}\mathbf{G}_\mathbf{X}^\dagger$ then $\mathbf{Q_X}\mathbf{v}_j$ is the j-th DMD mode.

6. Now with the DMD modes and eigenvalues, DMD future-state prediction or reconstruction can take place with equations (15) and (16).

D. Reduced Order Projection and Subspace Methods

The above DMD methods are commonly utilized when high-fidelity model predictions are not available or too expensive to compute, and when enough past observations are available to directly build reduced order models. These reduced order models are then integrated forward in time to compute the forecasts. In specific marine applications, high-fidelity forecasts computed on land are however often available. In this case, sending these forecasts efficiently to communication-limited autonomous platforms is one of the challenges. One approach to achieve this is to project the forecasts onto a suitable subspace and to only transmit the projected coefficients to the autonomous platforms. There are many options for such projections [3], [30], several of which based on singular value decomposition (SVD), including adaptive SVD [31]–[33]. In the stochastic case, the adaptive SVD methods become related to DO decompositions [6], [34], [35]. We briefly review next some of the classic projection and subspace methods.

E. POD, PCA, and EOF Projections

Proper Orthogonal Decomposition (POD) is a method of analysis that identifies the dominant structures in a dataset [36]. In the above DMD notation, the dominant POD modes are the columns of $\mathbf{U}_r$ that are determined from the SVD of the matrix $\mathbf{X}$, as in (10). These modes, or basis functions, capture as much energy of the system as possible [37]. POD is a commonly used model reduction technique, often based on snapshot data only. In the POD literature, Principal Component Analysis (PCA), and Empirical Orthogonal Functions [38] are often used to indicate similar basis functions, but POD modes are not necessarily mean subtracted as PCA models are (which is also refereed to as EOF analysis when based on empirical data only [39]).

Due to the limited storage and power available on autonomous vehicles, projection methods such as POD can prove extremely useful for sending reduced forecasts to an autonomous vehicle. Sending the full-state data is indeed often impractical or impossible. To reduce the state space at time k using POD modes, one option is to project the state-space onto POD modes

$$\tilde{\mathbf{x}_k} = \mathbf{U}_r^T \mathbf{x}_k, \tag{36}$$

where $\tilde{\mathbf{x}_k} \in \mathbb{R}^r$ and $\mathbf{U}_r \in \mathbb{R}^{n \times r}$. The state space can be reconstructed by $\mathbf{x}_k = \mathbf{U}_r \tilde{\mathbf{x}}_k$. If the POD modes from a set of past training data (columns of $\mathbf{U}_r$) were pre-loaded onto a vehicle, this subspace could be used to reconstruct full-state data with future predictions that were inexpensively sent to the vehicle. This is advantageous because in the ocean the full-state dimensions are often $\mathcal{O}(10^5 - 10^9)$ where as the reduced rank r is often $\mathcal{O}(10^1 - 10^3)$.

F. K-SVD and other sparse Projections

K-SVD allows for dictionary learning to create a dictionary for sparse representation of a signal [40] (note our notation differs from the K-SVD literature for consistency with the prior notation). There continues to be a growing interest in the study of sparse representation of signals based on an over-complete dictionary. Let $\mathbf{x} \in \mathbb{R}^n$ be an observed signal, and the dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$, and $\mathbf{y} \in \mathbb{R}^K$ are the representation coefficients. We assume that without noise, $\mathbf{x} = \mathbf{D}\mathbf{y}$, where the vector $\mathbf{y}$ is sparse. The goal of dictionary learning is to learn an overcomplete dictionary, $\mathbf{D}$, that contains K signal-atoms. The sparse representation is the solution to

$$\min_{y} \|\mathbf{y}\|_0 \text{ subject to } \mathbf{x} = \mathbf{D}\mathbf{y} \tag{37}$$

or

$$\min_{y} \|\mathbf{y}\|_0 \text{ subject to } \|\mathbf{x} - \mathbf{D}\mathbf{y}\|_2 \leq \epsilon \tag{38}$$

where ϵ is some small tolerance. To choose the best possible codebook, $\mathbf{D}$, we solve

$$\min_{D,Y} \|\mathbf{D}\mathbf{Y} - \mathbf{X}\|_F^2 \text{ subject to } \|\mathbf{y_i}\|_0 \leq T \tag{39}$$

where $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1, & \ldots, & \mathbf{x}_m \end{bmatrix}$ is the collection of m observations, $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1, & \ldots, & \mathbf{y}_m \end{bmatrix}$ is the collection of m representation coefficient vectors, and T is a constraint on the number of non-zero entries allowed in a column of $\mathbf{D}$. Now the problem can be solved by alternating minimization

$$\mathbf{Y}^{(j+1)} = \min_{y} \left\| \mathbf{D}^{(j)}\mathbf{Y} - \mathbf{X} \right\|_F^2 \text{ subject to } \|\mathbf{y_i}\|_0 \leq T. \tag{40}$$

and

$$\mathbf{D}^{(j+1)} = \min_{D} \left\| \mathbf{D}\mathbf{Y}^{(\mathbf{j+1})} - \mathbf{X} \right\|_F^2. \tag{41}$$

In words, at each iteration the sparse representation $\mathbf{Y}$ and the dictionary $\mathbf{D}$ are updated. The K-SVD algorithm may be used for compression of large ocean data [41], [42] and enable efficient transmission of ocean forecast and acoustic data to remote vehicles with bandwidth-limited, disadvantaged communications links. This is the subject of ongoing work.

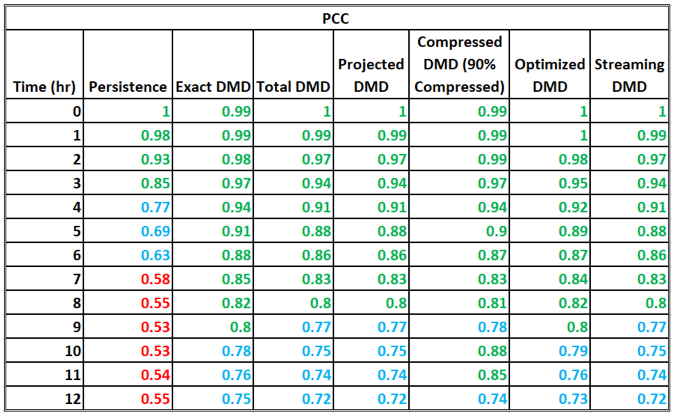
III. Results and Discussions

We now showcase the application of reduced order methods to two regional stochastic ocean forecasting experiments. Specifically, we apply several of the above DMD and SVD-based reduction methods, and discuss these results for use by simulated underwater vehicles in uncertain scenarios.

A. DMD Method Predictions

First, we consider a 12-hr forecast period (August 27, 2018 00Z) in the POSYDON-POINT experiment [8], and compare the performance of the SST forecasts of the above DMD methods with respect to the persistence forecast. All of the DMD forecasts were made from the first 85 hourly ocean simulation snapshots (00Z August 23 to 12Z August 26, 2018). The true forecast is assumed to be that of the full ocean modeling system [9].

The pattern correlation coefficients (PCCs) [43] for the Sea Surface Temperature (SST) forecasts of DMD methods are given in Table I. From forecast hour 2 and beyond, all DMD methods beat persistence. The DMD algorithms were also compared for salinity and velocity, and at different times and depths, with similar results (not shown here). For its low cost, Compressed DMD performed well. Overall, Exact DMD performed best, followed by Compressed DMD, Optimized DMD, and Total DMD.



PCC							
Time (hr)	Persistence	Exact DMD	Total DMD	Projected DMD	Compressed DMD (90% Compressed)	Optimized DMD	Streaming DMD
0	1	0.99	1	1	0.99	1	1
1	0.98	0.99	0.99	0.99	0.99	1	0.99
2	0.93	0.98	0.97	0.97	0.99	0.98	0.97
3	0.85	0.97	0.94	0.94	0.97	0.95	0.94
4	0.77	0.94	0.91	0.91	0.94	0.92	0.91
5	0.69	0.91	0.88	0.88	0.9	0.89	0.88
6	0.63	0.88	0.86	0.86	0.87	0.87	0.86
7	0.58	0.85	0.83	0.83	0.83	0.84	0.83
8	0.55	0.82	0.8	0.8	0.81	0.82	0.8
9	0.53	0.8	0.77	0.77	0.78	0.8	0.77
10	0.53	0.78	0.75	0.75	0.88	0.79	0.75
11	0.54	0.76	0.74	0.74	0.85	0.76	0.74
12	0.55	0.75	0.72	0.72	0.74	0.73	0.72

TABLE I: Pattern correlation coefficients (PCC) of SST errors vs. forecast time for the ensemble member 100 of the POSYDON-POINT experiment, for seven DMD methods. Green indicates PCC values above 0.8, blue PCC values between 0.6 and 0.8, and red PCC values below 0.6.

B. Splitting the Domain into Regions

To improve future-state prediction using DMD, we found that in most instances, splitting the domain into separate regions where relatively distinct or independent dynamics occurred was a good idea. In the following example, we separate the domain in two: the shelf region and the slope and deep water region. DMD methods were employed on SST from each domain on individual ensemble members to compute 12 hour forecasts using 85 hour training snapshots. The results are shown in Fig. 1. The 12 hour future SST predictions had PCC values of 0.54 for persistence, 0.61 for Total DMD on the entire domain simultaneously, and 0.75 for Total DMD on the shelf/deep water separately. The smaller domain for each of the regions means that the cost of DMD on the split domain and DMD on the entire domain were similar. We found similar results on other tests cases. In general, they imply that regional DMD forecasts may be improved by using a multi-domain approach. This would prove beneficial when dealing with autonomous vehicles operating in these specific regions, without capabilities for much inter-vehicle communication.

C. Training Size

We evaluated the effect of the size of the training set on the performance of DMD methods. To illustrate results, we show in Fig. 2 the zonal velocity errors from the persistence

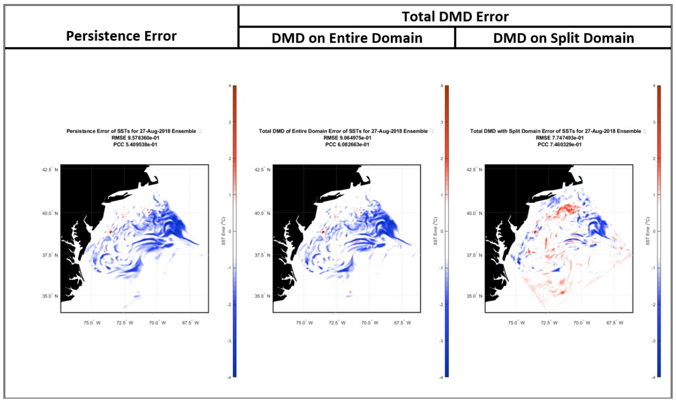

Persistence Error
Total DMD Error
DMD on Entire Domain
DMD on Split Domain
Persistance Error of SSTs for 27-Aug-2018 Ensemble
Total DMD of Entire Domain Error of SSTs for 27-Aug-2018 Ensemble
Total DMD with Split Domain Error of SSTs for 27-Aug-2018 Ensemble

Fig. 1: SST errors of 12 hour forecast for persistence (left), Total DMD applied to entire domain simultaneously (center), Total DMD on shelf/deep water separately (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment.

and exact DMD forecasts on September 16, 2006 19Z, for the AWACS/SW06 experiment. Here, for a 12-hr state forecast, persistence results in a PCC of 0.83; for 300 snapshot training, exact DMD results in a PCC of 0.87, and for 800 snapshot training, exact DMD results in a PCC of 0.91. In general, we find that the DMD methods provide better forecasts when the number of training snapshots increases at the expense of longer computation times.

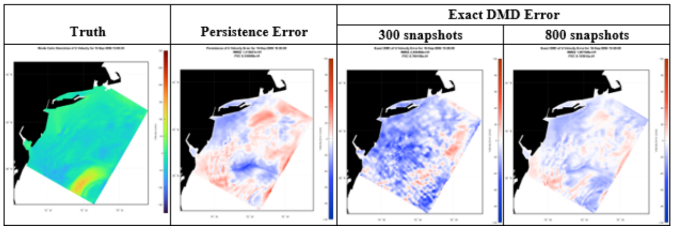


Fig. 2: Zonal velocity truth (left), errors for the persistence (second from left) and exact DMD forecasts on Sep 16, 2006 19Z, for the AWACS/SW06 experiment, training with either 300 snapshots (second from right) or 800 snapshots (right).

Even though increasing number of training snapshots usually improves prediction skill, in most instances for the AWACS/SW06 experiment, around 100 to 200 hourly training snapshots allowed for reasonable predictions of about 3 days into the future for SST, Zonal Velocity, and Salinity. An 84 hour SST prediction error alongside persistence error using 199 training snapshots is shown in Fig. 3 to demonstrate this skill. In the case of the 84 hour forecast on Aug 25, 2006 19Z, the persistence forecast has a PCC of just 0.15 and the exact DMD forecast has a PCC 0.81. As would normally be expected, the root mean squared error is also lower for the DMD prediction (0.80 °C) compared to persistence (1.54 °C).

D. 3D Forecasting

The DMD architecture allows for three-dimensional (3D) prediction provided that the 3D snapshots are arranged into column vectors $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$. In the following example, we used 100 training snapshots from the AWACS/SW06 experiment in order to predict zonal velocity for the next 48 hours using Streaming DMD. The PCC values at PE levels 1, 5, 15, and 20 of predictions every 4 hours are shown from Aug 22, 2006 07Z to Aug 24, 2006 07Z in Table II. The DMD prediction significantly outperform the persistence for nearly all times and layers after time 0 (except for around 36 hours). 3D predictions of other variables (SST and Salinity) also outperformed the persistence forecast in the vast majority of instances over the same time period.

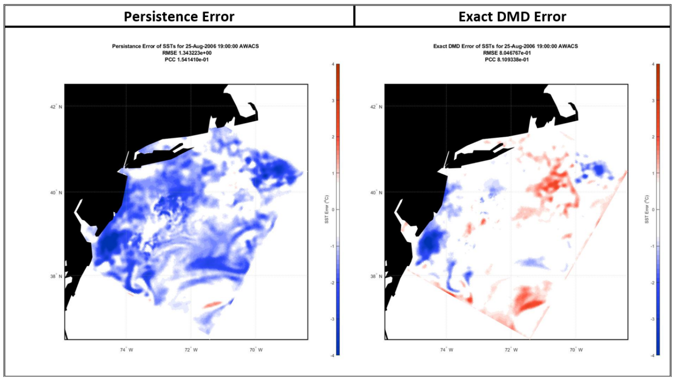

Persistence Error
Exact DMD Error
Persistance Error of SSTs for 25-Aug-2006 19:00:00 AWACS
Exact DMD Error of SSTs for 25-Aug-2006 19:00:00 AWACS

Fig. 3: SST errors of 84 hour forecast for persistence (left) and Exact DMD (right) on Aug 25, 2006 19Z using 199 training snapshots (with 108 DMD modes used) from the AWACS/SW06 experiment.

PCC of Zonal Velocity Forecast at Level Indicated										
	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD	Persistence	Streaming DMD
Time(hr)	Sea Surface		Level 5		Level 10		Level 15		Level 20	
0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.07	0.93	-0.02	0.93	-0.15	0.92	-0.27	0.92	-0.39	0.93
8	-0.25	0.79	-0.32	0.82	-0.40	0.86	-0.45	0.89	-0.46	0.90
12	0.66	0.84	0.65	0.85	0.64	0.87	0.65	0.88	0.72	0.89
16	0.30	0.79	0.25	0.78	0.19	0.75	0.13	0.74	0.09	0.77
20	0.38	0.80	0.34	0.78	0.29	0.74	0.23	0.72	0.15	0.75
24	0.72	0.90	0.71	0.91	0.70	0.91	0.69	0.91	0.68	0.93
28	-0.14	0.54	-0.20	0.55	-0.27	0.58	-0.35	0.64	-0.36	0.73
32	0.13	0.76	0.08	0.71	0.01	0.66	-0.05	0.68	-0.06	0.68
36	0.88	0.82	0.87	0.80	0.87	0.80	0.89	0.81	0.91	0.83
40	0.14	0.66	0.08	0.64	0.00	0.65	-0.08	0.67	-0.18	0.72
44	-0.35	0.61	-0.40	0.62	-0.46	0.64	-0.54	0.67	-0.59	0.70
48	0.71	0.78	0.70	0.79	0.69	0.81	0.67	0.83	0.70	0.85

TABLE II: Pattern correlation coefficients (PCC) of zonal velocity prediction vs. forecast time (from Aug 22, 2006 07Z to Aug 24, 2006 07Z) for the AWACS/SW06 experiment, at PE levels 1, 5, 15, and 20. DMD prediction was made using 3D zonal velocities and Streaming DMD method with 100 training snapshots.

E. Ensemble Mean and Variance

We have shown some results from deterministic predictions using DMD methods. We now consider stochastic PE forecasts (consisting of 300 ensemble members) from the POSYDON-POINT experiment. Variations among the ensemble members are due to perturbations applied to the initial conditions (ICs) but also due to different tidal forcing parameters and atmospheric forcing fields used to force the different ensemble members [8].

We compare three DMD approaches to make probabilistic forecasts. For evaluation, we use the true mean and standard

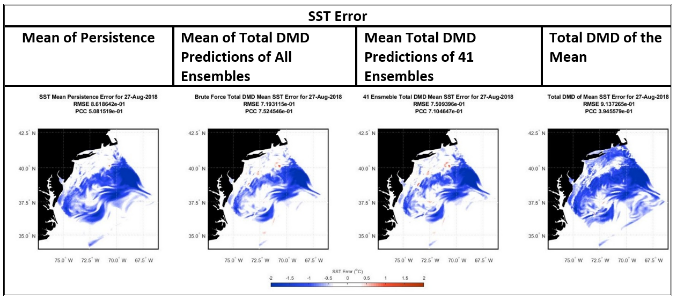

SST Error
Mean of Persistence
Mean of Total DMD Predictions of All Ensembles
Mean Total DMD Predictions of 41 Ensembles
Total DMD of the Mean
SST Mean Persistence Error for 27-Aug-2018
RMSE 8.618642e-01
PCC 5.081519e-01
Brute Force Total DMD Mean SST Error for 27-Aug-2018
RMSE 7.193115e-01
PCC 7.524546e-01
41 Ensmeble Total DMD Mean SST Error for 27-Aug-2018
RMSE 7.509396e-01
PCC 7.104647e-01
Total DMD of Mean SST Error for 27-Aug-2018
RMSE 9.137265e-01
PCC 3.945579e-01
42.5° N
40.0° N
37.5° N
35.0° N
75.0° W
72.5° W
70.0° W
67.5° W
SST Error (°C)
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Fig. 4: SST error from the true SST mean of 12 hour forecast for persistence (left), mean of Total DMD forecasts of all ensembles (second from left), mean of Total DMD forecasts of 41 ensembles (second from right), and Total DMD forecast using mean field (right) on Aug 27, 2018 00Z, for the POSYDON-POINT experiment.

deviation of all POSYDON-POINT experiment PE ensembles at each grid-point at the specified times.

The first approach is a brute force Monte-Carlo method where we perform DMD on each ensemble member individually to make DMD predictions. The mean and standard deviation of the DMD forecasts is then computed at each grid point. With this approach, our probabilistic forecast is the statistics (mean, variance, etc.) of the ensemble of DMD forecasts, i.e. the statistics of the DMD forecasts.

In order to improve over the cost of performing DMD on each ensemble, the second approach is a batch ensemble forecast [26], [44]. Here, we first compute the DMD prediction of ensemble member 1 at time t and the DMD prediction of member 2 at the same time. We then take the mean of these two predictions at each grid point. We then compute the DMD prediction of member 3 at the same time and determined the mean of the predictions of members 1, 2 , and 3. We repeat this for member 4 and so on, until the running mean of the DMD prediction converged. This resulted in an estimate of the mean DMD prediction at each grid point. For the variance field forecast, we used Welford's online variance algorithm [45] to compute the sample variance at each grid point. The cost of this method is much smaller than performing DMD on each ensemble individually (for the POSYDON-POINT SST PE ensemble this resulted in 41 of 300 ensemble members used).

The third approach first determines the mean and variance of the ensemble members at each grid-point and training time t_k. We then employ the DMD algorithm twice, once for the mean field and once for the variance fields. In this approach, our probabilistic forecast is the DMD forecast of the mean, variance, etc., i.e. the DMD forecast of the statistics.

We illustrate the above three approaches using 85 training snapshots to predict the statistics of the SST field. We made a 12 hour forecast of the SST mean and SST variance. The SST mean error results are shown for August 27, 2018 00Z on Fig. 4. The resultant SST PCC values when compared to the true SST field mean are 0.51 for persistence and i) 0.75 for the mean of Total DMD forecasts of all members, ii) 0.71 for the mean of Total DMD forecasts using 41 members, and iii) 0.39 for the Total DMD forecast using the mean field as the input. Due to nonlinearities, these results show that the DMD prediction using the mean field as input performed the worse followed by the mean persistence field. The mean SST Total DMD forecasts using all members and 41 members, however, performed similarly well (with the more expensive forecast using all members logically having a slight edge).

In Fig. 5, we show 12 hour future SST standard deviation fields for the truth (left), the Total DMD prediction using all members individually with a PCC of 0.80 (second from left), the 41 ensemble members Total DMD SST forecast with a PCC of 0.62 (second from right), and the Total DMD prediction using SST variance as the input with PCC of 0.82 (right). At first glance, the standard deviation fields appear similar. Upon further inspection, we see some areas (e.g. near the eastern most corner) where the 41 ensemble members Total DMD SST forecast underestimates the standard deviation while the other two forecasts tend to slightly overestimate.

DMD with the mean of the ensemble as input performed poorly for the mean prediction compared to persistence but performed reasonably well for predicting the variance. The mean of DMD forecasts performed considerably better than the persistence mean but would require a high number of DMD forecasts to accurately predict the variance of the members.

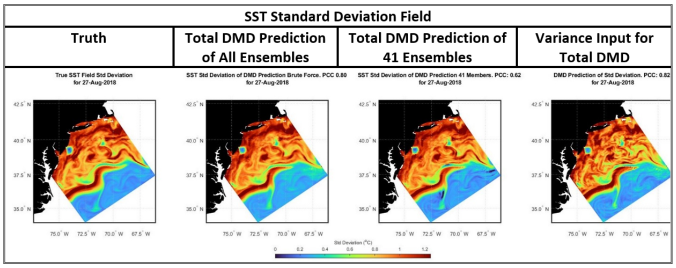

SST Standard Deviation Field
Truth
Total DMD Prediction of All Ensembles
Total DMD Prediction of 41 Ensembles
Variance Input for Total DMD
True SST Field Std Deviation
for 27-Aug-2018
SST Std Deviation of DMD Prediction Brute Force. PCC 0.80
for 27-Aug-2018
SST Std Deviation of DMD Prediction 41 Members. PCC: 0.62
for 27-Aug-2018
DMD Prediction of Std Deviation. PCC: 0.82
for 27-Aug-2018
42.5° N
40.0° N
37.5° N
35.0° N
75.0° W
72.5° W
70.0° W
67.5° W
Std Deviation (°C)
0
0.2
0.4
0.6
0.8
1
1.2

Fig. 5: Standard deviation fields for true SST, the Total DMD SST forecasts of all ensemble members (PCC of 0.80), the 41 ensemble members Total DMD SST forecasts (PCC of 0.62), and the Total DMD forecast using variance as the input (PCC of 0.82) on August 27, 2018 00Z for the POSYDON-POINT experiment.

F. Ensemble DMD Modes

We utilized another approach to combine ensemble members to make probabilistic DMD predictions [17]. With a total of J ensemble members and m times, each snapshot can be represented as $\mathbf{x}_k^j$ where k is the snapshot time and j is the j-th ensemble member. DMD can then be applied to matrices

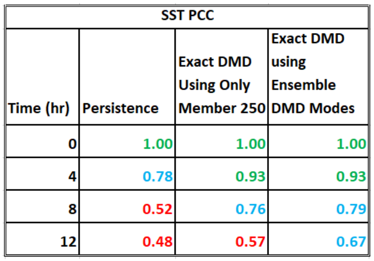
$$\mathbf{X} = \begin{bmatrix} | & & | & | & & | & & | & & | \\ \mathbf{x}_1^1 & \cdots & \mathbf{x}_{m-1}^1 & \mathbf{x}_1^2 & \cdots & \mathbf{x}_{m-1}^2 & \cdots & \mathbf{x}_1^J & \cdots & \mathbf{x}_{m-1}^J \\ | & & | & | & & | & & | & & | \end{bmatrix} \tag{42}$$

and

$$\mathbf{X}' = \begin{bmatrix} | & & | & | & & | & & | & & | \\ \mathbf{x}_2^1 & \cdots & \mathbf{x}_m^1 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_m^2 & \cdots & \mathbf{x}_2^J & \cdots & \mathbf{x}_m^J \\ | & & | & | & & | & & | & & | \end{bmatrix}. \tag{43}$$

This approach allows for extracting the DMD modes and eigenvalues from the entire ensemble set (or a specified number of ensemble members) simultaneously. With the overall DMD modes and eigenvalues, we can make prediction of a particular ensemble member using an initial condition (used to calculate the DMD amplitudes).

Using 60 ensemble members and 85 hour training data from the POSYDON-POINT experiment, we performed DMD on SST values as discussed above. We then predicted SST values for ensemble member 250 using the initial condition from that ensemble. The PCC results are listed in Table III. The PCC values are generally better for the SSTs predicted using ensemble DMD modes compared to using exact DMD on the individual ensemble member. For the 12 hour forecast, we can see that persistence forecast of member 250 has a PCC of 0.48, Exact DMD forecast of member 250 has a PCC of 0.57, and Exact DMD forecast for member 250 using modes extracted from 60 members is 0.67.



SST PCC			
Time (hr)	Persistence	Exact DMD Using Only Member 250	Exact DMD using Ensemble DMD Modes
0	1.00	1.00	1.00
4	0.78	0.93	0.93
8	0.52	0.76	0.79
12	0.48	0.57	0.67

TABLE III: PCCs of SST prediction vs. forecast time (Aug 26, 2018 12Z to Aug 27, 2018 00Z) for the POSYDON-POINT experiment ensemble member 250. Persistence forecast PCCs are consistently lower than Exact DMD forecasts. The DMD forecasts were made using DMD only on member 250 (center column) and using DMD modes extracted using 60 members (right column). Using the DMD modes extracted from multiple members provides better forecasts.

These results indicate that using multiple ensemble members can capture the underlying DMD better than just using a single member. Having the modes extracted using a portion the ensemble reduces storage and computation costs compared to using the full ensemble. This could prove advantageous for unmanned vehicles. The DMD mode could be computed off the vehicle, then a reduced number of DMD modes could be sent to the vehicle for relatively inexpensive predictions with reduced communication and storage needs.

G. Subspace Projections

We now illustrate the projections and compression of large ocean forecasts into pre-defined or adaptive subspaces. The projected forecasts, due to their much reduced dimension, can be readily transmitted to platforms with limited bandwidth. However, this reduction could affect accuracy. To show the effectiveness of POD modes for such reduction, we used 85 hour sea surface zonal velocity training data from ensemble member 150 of the POSYDON-POINT experiment for the period 23–27 August 2018 to determine the "past" POD modes. We then projected the subsequent 12 hour PE forecast onto these past POD modes. The initial PE forecast was a vector with 54960 entries (440 kB), the projected PE forecast had just 22 entries (0.176 kB) using 90% variance-explained criterion. The remote platform, pre-loaded with these 22 past POD modes (less than 10 MB storage), would be able to reconstruct the forecast. In Fig. 6, we show the error between the truth full PE forecast and the 12 hour reconstructed POD-projected forecast next to 12 hour persistence forecast error for surface zonal velocity. PCC for the reconstructed projection is 0.81, while for persistence, it is 0.26. Results for a 24 hour forecast using 73 hour training data (not shown), had a PCC of 0.70 for reconstructed PE forecast onto past POD modes and a PCC of 0.48 for persistence forecast.

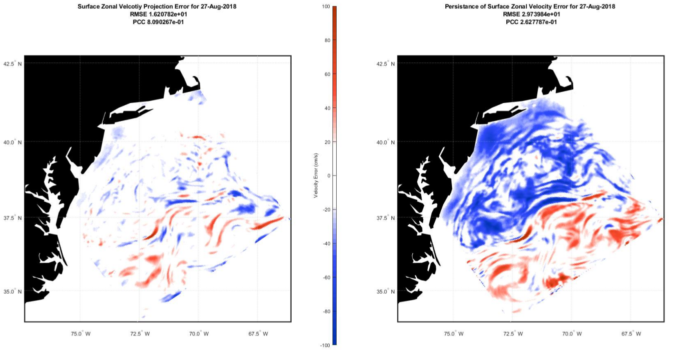


Fig. 6: Surface Zonal Velocity errors for August 27, 2018 00Z for the POSYDON-POINT experiment. The reconstructed 12 hour PE forecast projection on past POD modes (left) with a PCC of 0.81 and the persistence forecast (right) with a PCC of 0.26.

IV. Conclusions

Several reduced order modeling schemes for regional ocean forecasting onboard autonomous platforms at sea were described, investigated, and evaluated. We evaluated Dynamic Mode Decomposition (DMD) [3] methods for ocean PE simulations by comparing several schemes including domain splitting, adjusting training size, and utilizing 3D inputs. Three different approaches that combine uncertainty with DMD were also investigated and found to be practical, especially if we employ either an ensemble of DMD forecasts or the DMD of an ensemble of forecasts. Projecting / compressing high-fidelity forecasts using schemes such as POD projection and K-SVD for sparse representation also showed promise for distributing forecasts to remote vehicles. Future work includes improving and applying the discussed methods onboard ocean platforms. The results indicates that augmenting the DO-PE capability (that provides a best instantaneous stochastic reduction) to a new DO-space-time dynamic reduction should be useful, so as to provide the best reduction over a period of time or for a specific forecast lead time.

Acknowledgments

We thank all members of the MSEAS group, past and present. We especially thank Tony Ryu for his review. We are grateful to the Office of Naval Research for support under STTR grant N6833519C0348 (ROMs) to the Massachusetts Institute of Technology. We thank our colleagues in SW06, AWACS-06, and POSYDON-POINT for their collaboration.

References

[1] P. F. J. Lermusiaux, “Uncertainty estimation and prediction for interdisciplinary ocean dynamics,” *Journal of Computational Physics*, vol. 217, no. 1, pp. 176–199, 2006.

[2] P. F. J. Lermusiaux, C.-S. Chiu, G. G. Gawarkiewicz, P. Abbot, A. R. Robinson, R. N. Miller, P. J. Haley, Jr, W. G. Leslie, S. J. Majumdar, A. Pang, and F. Lekien, “Quantifying uncertainties in ocean predictions,” *Oceanography*, vol. 19, no. 1, pp. 92–105, 2006.

[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] 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.

[5] M. P. Ueckermann, P. F. J. Lermusiaux, and T. P. Sapsis, “Numerical schemes for dynamically orthogonal equations of stochastic fluid and ocean flows,” *Journal of Computational Physics*, vol. 233, pp. 272–294, Jan. 2013.

[6] F. Feppon and P. F. J. Lermusiaux, “Dynamically orthogonal numerical schemes for efficient stochastic advection and Lagrangian transport,” *SIAM Review*, vol. 60, no. 3, pp. 595–625, 2018.

[7] ——, “A geometric approach to dynamical model-order reduction,” *SIAM Journal on Matrix Analysis and Applications*, vol. 39, no. 1, pp. 510–538, 2018.

[8] P. F. J. Lermusiaux, C. Mirabito, P. J. Haley, Jr., W. H. Ali, A. Gupta, S. Jana, E. Dorfman, A. Laferriere, A. Kofford, G. Shepard, M. Goldsmith, K. Heaney, E. Coelho, J. Boyle, J. Murray, L. Freitag, and A. Morozov, “Real-time probabilistic coupled ocean physics-acoustics forecasting and data assimilation for underwater GPS,” in *OCEANS 2020 IEEE/MTS*. IEEE, Oct. 2020, in press.

[9] P. J. Haley, Jr. and P. F. J. Lermusiaux, “Multiscale two-way embedding schemes for free-surface primitive equations in the “Multidisciplinary Simulation, Estimation and Assimilation System”,” *Ocean Dynamics*, vol. 60, no. 6, pp. 1497–1537, Dec. 2010.

[10] N. R. Chapman and J. F. Lynch, “Editorial: Special issue on the 2006 shallow water experiment,” *IEEE Journal of Oceanic Engineering*, vol. 35, no. 1, pp. 1–2, 2010.

[11] M. E. G. D. Colin, T. F. Duda, L. A. te Raa, T. van Zon, P. J. Haley, Jr., P. F. J. Lermusiaux, W. G. Leslie, C. Mirabito, F. P. A. Lam, A. E. Newhall, Y.-T. Lin, and J. F. Lynch, “Time-evolving acoustic propagation modeling in a complex ocean environment,” in *OCEANS - Bergen, 2013 MTS/IEEE*, 2013, pp. 1–9.

[12] K. D. Heaney, P. F. J. Lermusiaux, T. F. Duda, and P. J. Haley, Jr., “Validation of genetic algorithm based optimal sampling for ocean data assimilation,” *Ocean Dynamics*, vol. 66, pp. 1209–1229, 2016.

[13] P. J. Schmid and J. Sesterhenn, “Dynamic mode decomposition of numerical and experimental data,” in *61st Annual Meeting of the APS Division of Fluid Dynamics*. San Antonio, Texas: American Physical Society, Nov. 2008.

[14] P. J. Schmid, “Dynamic mode decompostion and of numerical and experimental data,” *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, Aug. 2010.

[15] W. E. Boyce and R. C. DiPrima, *Elementary Differential Equations*, 9th ed. Wiley, 2008.

[16] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.

[17] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, “On dynamic mode decomposition: Theory and applications,” *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, Jan. 2014.

[18] 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.

[19] I. Bright, G. Lin, and J. N. Kutz, “Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements,” *Physics of Fluids*, vol. 25, no. 12, p. 127102, 2013. [Online]. Available: https://doi.org/10.1063/1.4836815

[20] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, “De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy datasets,” *Theoretical and Computational Fluid Dynamics*, vol. 31, no. 4, pp. 349—-368, Apr. 2017.

[21] T. Ashkam and J. N. Kutz, “Variable projection methods for an optimized dynamic mode decomposition,” *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 380–416, Feb. 2018.

[22] G. Golub and V. Pereyra, “Separable nonlinear least squares: the variable projection method and its applications,” in *Institute of Physics, Inverse Problems*, 2002, pp. 1–26.

[23] L. Kaufman, “A variable projection method for solving separable nonlinear least squares problems,” *BIT Numerical Mathematics*, vol. 15, pp. 49–57, Mar. 1975.

[24] M. S. Hemati, M. O. Williams, and C. W. Rowley, “Dynamic mode decomposition for large and streaming datasets,” *Physics of Fluids*, vol. 26, no. 11, Jun. 2014.

[25] R. W. Brockett, “Dynamical systems that learn subspaces,” in *Mathematical System Theory*. Springer, 1991, pp. 579–592.

[26] 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.

[27] T. P. Sapsis and P. F. J. Lermusiaux, “Dynamical criteria for the evolution of the stochastic dimensionality in flows with uncertainty,” *Physica D: Nonlinear Phenomena*, vol. 241, no. 1, pp. 60–76, 2012.

[28] B. Peherstorfer and K. Willcox, “Online adaptive model reduction for nonlinear systems via low-rank updates,” *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. A2123–A2150, 2015.

[29] ——, “Dynamic data-driven reduced-order models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 21–41, 2015.

[30] J. N. Kutz, *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[31] R. Badeau, B. David, and G. Richard, “Fast approximated power iteration subspace tracking,” *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2931–2941, 2005.

[32] O. Koch and C. Lubich, “Dynamical low-rank approximation,” *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 2, pp. 434–454, 2007.

[33] J. Dehaene, “Continuous-time matrix algorithms systolic algorithms and adaptive neural networks,” 1995.

[34] D. N. Subramani, P. J. Haley, Jr., and P. F. J. Lermusiaux, “Energy-optimal path planning in the coastal ocean,” *Journal of Geophysical Research: Oceans*, vol. 122, pp. 3981–4003, 2017.

[35] D. N. Subramani, “Probabilistic regional ocean predictions: Stochastic fields and optimal planning,” Ph.D. dissertation, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Massachusetts, Feb. 2018.

[36] L. Surovich, “Turbulence and the dynamics of coherent structures parts i-iii,” *Quarterly of Applied Mathematics*, vol. 45, no. 3, pp. 561–571, Oct. 1987.

[37] K. Taira, S. L. Brunton, D. M. Scott, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, “Modal analysis of fluid flows: An overview,” *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, Dec. 2017.

[38] I. Jollife, “T.(2002) principal component analysis.”

[39] A. Navarra and V. Simoncini, *A guide to empirical orthogonal functions for climate data analysis*. Springer Science & Business Media, 2010.

[40] M. Aharaon, M. Elad, and A. Bruckstein, “K-svd: An algortihm for designing overcomplete dictionaries for sparse representation,” *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[41] M. Bianco and P. Gerstoft, “Dictionary learning of sound speed profiles,” *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 1749–1758, 2017.

[42] P. Gerstoft, C. F. Mecklenbräuker, W. Seong, and M. Bianco, “Introduction to compressive sensing in acoustics,” 2018.

[43] 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.

[44] ——, “Estimation and study of mesoscale variability in the Strait of Sicily,” *Dynamics of Atmospheres and Oceans*, vol. 29, no. 2, pp. 255–303, 1999.

[45] B. P. Welford, “Note on a method for calculating corrected sums of squares and products,” *Technometrics*, vol. 4, no. 3, pp. 419–420, Aug. 1962.