Consistent Dynamic Mode Decomposition*

Omri Azencot , Wotao Yin , and Andrea Bertozzi †

Abstract. We propose a new method for computing Dynamic Mode Decomposition (DMD) evolution matrices, which we use to analyze dynamical systems. Unlike the majority of existing methods, our approach is based on a variational formulation consisting of data alignment penalty terms and constitutive orthogonality constraints. Our method does not make any assumptions on the structure of the data or their size, and thus it is applicable to a wide range of problems including non-linear scenarios or extremely small observation sets. In addition, our technique is robust to noise that is independent of the dynamics and it does not require input data to be sequential. Our key idea is to introduce a regularization term for the forward and *backward* dynamics. The obtained minimization problem is solved efficiently using the Alternating Method of Multipliers (ADMM) which requires two Sylvester equation solves per iteration. Our numerical scheme converges empirically and is similar to a *provably* convergent ADMM scheme. We compare our approach to various state-of-the-art methods on several benchmark dynamical systems.

Key words. Dynamic Mode Decomposition, Dynamical Systems, ADMM, variational formulation

AMS subject classifications. 37N30, 65K10, 90C26

1. Introduction.

Over the last few years, data-driven approaches became prevalent in analyzing dynamical systems [22]. In the common scenario, a collection of system *observations* is provided and a *linear* object that encodes the dynamics is generated based solely on the data. These data-driven approaches are advantageous in that they make minimal assumptions on the governing equations of the system, and in particular, these techniques are applicable even to non-linear dynamics. In this context, Dynamic Mode Decomposition (DMD) [30] methods gained a lot of attention lately, in part due to their computational efficiency as well as their analysis capabilities of the system at hand. DMD-based methods were successfully applied to various flows including detonation waves, cavity flows and jets [25, 32, 31]. In short, DMD computes a matrix whose spectrum, represented by the eigenvalues and eigenvectors, provides meaningful information such as growth and decay rates of the system or dominant coherent structures in the flow. The goal of this paper is to propose a new method for computing DMD matrices that is based on interpreting the problem in a variational form, taking into account the forward and backward dynamics and solving it efficiently via splitting.

Developing data-driven methodologies for the analysis of non-linear dynamical systems is an active research domain with DMD being one of its main avenues. In particular, DMD was recently generalized and extended in several works having the objective of alleviating some of the shortcomings in the original technique. For instance, a limiting assumption

*Submitted to the editors DATE.

Funding: This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 793800, a Zuckerman STEM Leadership Postdoctoral Fellowship, NSF Grant DMS-1720237, ONR Grant N000141712162, NSF grant DMS-1737770 and the City of Los Angeles, Gang Reduction Youth Development (GRYD) Analysis Program.

†Department of Mathematics, University of California, Los Angeles CA 90095, (azencot@math.ucla.edu, wotaoyin@math.ucla.edu, bertozzi@math.ucla.edu).

in [29, 30] requires that the data is given in a sequential form, namely, the input snapshots represent an equally spaced time series of observations. In Tu et al. [34] and other works, this limitation is relaxed and pairs of equispaced observations are used instead, whereas in [15, 24, 1], no assumption is made on the regularity of the temporal sampling. Another drawback of several DMD methods is the bias they exhibit in the presence of noise and whether the noise interacts with the dynamics [2] or not [7]. To address this challenge, variants of DMD were proposed in the literature based on solving jointly for the basis and the evolution operator [36], formulating the problem as a total least squares minimization [17], and fitting an exponential model [1]. Other methods cope with noise by utilizing Kalman filters [26, 27], adapting DMD to online data [18, 16], and developing a Rayleigh–Ritz modal decomposition [8], among other approaches [7]. Under this classification, our method is applicable to non-sequential data and it performs extremely well when sensor noise corrupts the data, as we show in Section 5.

Perhaps closest to our approach is the work of Dawson et al. [7] where the idea of making DMD more robust to noise by considering the forward and *backward* evolution is investigated. More specifically, in forward-backward DMD (fbDMD) [7], the DMD matrix is estimated via the square root of the product of the forward model with the inverse of the backward DMD matrix. The backward estimate is generated by switching the "before" and "after" roles of the snapshots. Our machinery is based on the same observation of exploiting the forward and backward dynamics, but in a completely different way. Inspired by ideas from *Computer Graphics* [28, 19], we formulate the task of computing the DMD matrix in a variational form that includes penalties for both directions. The obtained minimization is unfortunately highly non-linear and non-convex, and thus we introduce an auxiliary variable that represents the backward dynamics, arriving at an optimization problem with quadratic objective terms and bilinear constraints. This problem can be solved efficiently using splitting techniques such as the Alternating Direction Method of Multipliers (ADMM) [5]. The obtained scheme is iterative, where at each step we solve two Sylvester equations and perform a trivial update. In addition, we show that our problem can be modified such that a provably convergent scheme can be devised. Overall, we obtain an efficient algorithm that exhibits fast convergence rates in practice and provides improved estimates of various properties of the dynamical system.

The rest of the paper is organized as follows. In Section 2 we provide background details related to dynamic mode decomposition techniques and the alternating method of multipliers. Section 3 details our approach for generating consistent DMD evolution matrices where we derive the variational formulation, and we propose an effective ADMM splitting scheme to solve it in practice. In Section 4, we prove that the problem we consider can be changed so that it admits an ADMM-type algorithm which is provably converging. Section 5 provides a quantitative and qualitative evaluation of our method with respect to several DMD algorithms. Section 6 concludes our work, discusses limitations, and offers a few potential directions for future work.

2. Background. In what follows, we briefly present the most relevant details regarding DMD algorithms. We refer to [22] for a more comprehensive text on the recent developments and applications of DMD-based techniques. In addition, we describe the essential components of ADMM and their link to our work, where we point to the paper by Boyd et al. [5] for additional information.

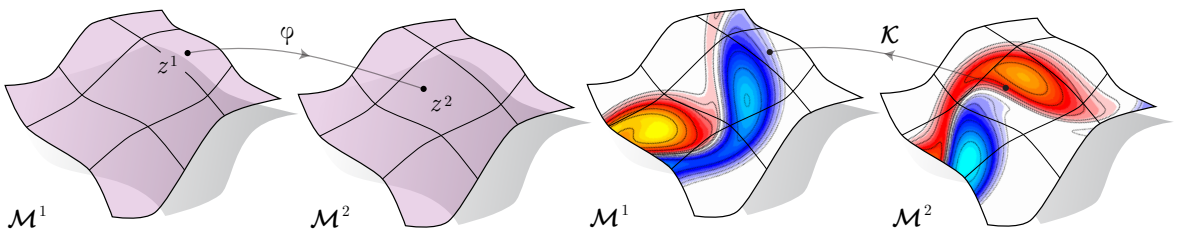

Figure 1. *The dynamical system φ maps from the manifold $\mathcal{M}$ at time 1 to time 2 (left), whereas the associated Koopman operator maps between scalar functions defined on $\mathcal{M}$ (right).*

2.1. DMD. Dynamic Mode Decomposition (DMD) emerged in the fluid dynamics field [30] as a data driven approach for analyzing a dynamical system based on observational data. DMD is strongly related to Koopman theory [21], where a non-linear dynamical system φ acting on a finite-dimensional manifold $\mathcal{M}$ is encoded using an infinite-dimensional linear Koopman operator $\mathcal{K}$. In this context, DMD can be viewed as a practical approach to produce a matrix A whose spectrum approximates the spectrum of the operator $\mathcal{K}$. Thus, A is an informative object and its dominant eigenvalues and eigenvectors are directly linked to dynamical features of the system such as growth, decay, frequency and flow modes. These results encourage the community to investigate DMD as an effective tool for analyzing various linear and nonlinear dynamical systems [22].

A common scenario, considered in several DMD-based techniques, is to assume to be given a set of temporally related pairs of observations $\tilde{x}_j$ and $\tilde{y}_j, j = 1, 2, .., n$, such that

$$\tilde{y}_j(z) = \tilde{x}_j(\varphi(z)) \ , \tag{2.1}$$

where $z \in \mathcal{M}$, the dynamical system is $\varphi : \mathcal{M} \to \mathcal{M}$, and $\tilde{x}_j$, $\tilde{y}_j : \mathcal{M} \to \mathbb{R}$. Namely, if $\tilde{x}_j$ represents some quantity at time t, then $\tilde{y}_j$ measures the same quantity at a later time $t + \Delta t$, as it changes due to the dynamics φ, see Fig. 1 for an illustration of this setup. Examples of the input observations could be the spatial coordinates [7] or the scalar vorticity [34], among other system-related data. The time series of observations $\{\tilde{x}_j\}_{j=1}^n$ and $\{\tilde{y}_j\}_{j=1}^n$ is used to construct matrices $\tilde{X}$ and $\tilde{Y}$ such that

$$\tilde{X} = [\tilde{x}_1 \ \tilde{x}_2 \ ... \ \tilde{x}_n] \in \mathbb{R}^{m \times n}, \quad \tilde{Y} = [\tilde{y}_1 \ \tilde{y}_2 \ ... \ \tilde{y}_n] \in \mathbb{R}^{m \times n} \ , \tag{2.2}$$

where the manifold $\mathcal{M}$ is of dimension $|\mathcal{M}| = m$. We note that our data is equispaced in time, i.e., Δt is the same for every j, as is commonly assumed in the DMD literature, although other scenarios were considered, e.g., [33]. Using the above notation, the goal of many DMD algorithms is to find a matrix $\tilde{A} \in \mathbb{R}^{m \times m}$ such that $\tilde{A}\tilde{X} = \tilde{Y}$.

In practice, solving directly for $\tilde{A}$ could be challenging, especially when m is extremely large or when $m > n$, leading to an underdetermined system. One way to mitigate these difficulties is to reduce the spatial dimension of the input data. Many dimensionality reduction techniques have been developed in recent years, where the Proper Orthogonal Decomposition

(POD) [4] is typically chosen mostly due to its algorithmic simplicity and computational efficiency. One of the outputs of POD is a set of r orthogonal modes $B \in \mathbb{C}^{m\times r}$ such that the linear subspace spanned by B approximates $\mathbb{R}^m$ well enough. From now on, we denote by X and Y the projection of $\tilde{X}$ and $\tilde{Y}$ onto the first r POD modes. Formally,

$$X = B^*\tilde{X} \in \mathbb{R}^{r\times n} \ , \quad Y = B^*\tilde{Y} \in \mathbb{R}^{r\times n} \ , \tag{2.3}$$

where B^* is the conjugate transpose of B. To compute the matrix B, we facilitate the Singular Value Decomposition (SVD) to obtain the expression $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^*$ and $B = \tilde{U}_r$, i.e., the first r left singular vectors that correspond to the dominant r singular values. In this reduced form, the problem of DMD is to solve the equation $AX = Y$, for which the least squares solution is analytically given by $A = YX^+$, where X^+ is the Moore–Penrose pseudoinverse of X. The DMD algorithms we will present next can be thought of as various approaches for approximating such A matrices.

The following Algorithm 2.1 was introduced in Tu et al. [34] and is known as the `Exact DMD` method. While this approach is not one of the original DMD techniques as was proposed in [29, 30], it is a close variant of these methods and it serves as the baseline algorithm for many extensions and comparisons in the DMD literature. We note that in Step (3), instead of taking the pseudoinverse X^+, the authors took its projection onto the first r modes. Indeed, we have that

$$X^+ = \left(B^*\tilde{X}\right)^+ = \left(\tilde{U}_r^*\tilde{U}\tilde{S}\tilde{V}^*\right)^+ \approx \left(\tilde{U}_r^*\tilde{U}_r\tilde{S}_r{\tilde{V}_r}^*\right)^+ = \tilde{V}_r\tilde{S}_r^{-1} \ .$$

Also, Step (4) involves the eigendecomposition (EIG) of A, typically yielding a complex-valued spectrum since A does not exhibit a special structure in general. Finally, in many DMD-based algorithms, Steps $(1-2)$ and $(4-5)$ are shared, whereas Step (3) is different. This is also the case in our Algorithm 3.1 where the main change is the way we construct the matrix A.

Algorithm 2.1 Exact Dynamic Mode Decomposition (`Exact DMD`)

1: Input matrices $\tilde{X}, \tilde{Y} \in \mathbb{R}^{m\times n}$ and a scalar $r \in \mathbb{R}$

2: Compute the SVD of $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^*$, and generate $X = \tilde{U}_r^*\tilde{X}, Y = \tilde{U}_r^*\tilde{Y}$

3: Denote $A = Y\tilde{V}_r\tilde{S}_r^{-1}$

4: Compute the EIG of A, with $Av_j = \lambda_j v_j$, where $v_j \in \mathbb{C}^r$, $\lambda_j \in \mathbb{C}$

5: The DMD spectrum is defined as the set of eigenvalues λ_j, and vectors $\psi_j = \lambda_j^{-1}\tilde{Y}\tilde{V}_r\tilde{S}_r^{-1}v_j$

2.2. Regularizing DMD. In many scenarios, the time sequence of data is generated using sensory devices. For example, Schmid et al. [31] applied DMD to snapshots of a helium jet, collected using particle-image-velocimetry (PIV) measurements. Naturally, in these settings, the observations are assumed to be corrupted with various types of noise. The existence of process or sensor noise results in a certain bias in traditional DMD algorithms such as `Exact DMD`, as was recently shown in [7, 17, 1]. To address these shortcomings, several extensions to DMD were recently proposed in the literature. From an optimization standpoint, these

modified DMD methods as well as our approach can be viewed as *regularizing* the original minimization problem, introducing algorithms that are more robust in the presence of noise. In our discussion here, we focus on the methods `fbDMD` [7], `tlsDMD` [17] and `Optimized DMD` [1].

The main idea behind the forward-backward DMD (`fbDMD`) technique is to take into account the forward dynamics, i.e., transforming X into Y, as well as the backward system where Y is mapped to X. The motivation is that by considering both directions, much of the bias to noise can be eliminated. In fact, we build on the exact same observation, however, we arrive at a completely different method. The algorithm `fbDMD` follows the same steps of Algorithm 2.1, except for the matrix construction which is given by

$$A = \left(A_f A_b^{-1}\right)^{1/2} \ , \tag{2.4}$$

where $A_f = \tilde{U}_X^* \tilde{Y} \tilde{V}_X \tilde{S}_X^{-1}$ is the forward estimate, and $A_b = \tilde{U}_Y^* \tilde{X} \tilde{V}_Y \tilde{S}_Y^{-1}$ is the backward one. Notice that the SVD of both $\tilde{X} = \tilde{U}_X \tilde{S}_X \tilde{V}_X^*$ and $\tilde{Y} = \tilde{U}_Y \tilde{S}_Y \tilde{V}_Y^*$ are used. Assuming that efficient routines for computing the square root of a matrix such as `sqrtm` of MATLAB are available, the time complexity for this algorithm is $\mathcal{O}\left(\min\{mn^2, m^2n\} + r^3\right)$, and thus it is governed by the SVD part as we typically have $r \ll m, n$.

In a different paper [17], the authors propose another algorithm known as the total least squares DMD (`tlsDMD`). Intuitively, this approach tries to symmetrize the way noise is being handled so that it assumes noise polluted both X and Y, whereas other methods implicitly account only for noise in Y. Similarly to the latter algorithm, `tlsDMD` provides an alternative definition for the A matrix. Specifically,

$$A = U_{br}\, U_{tr}^{-1} \ , \quad \text{with } \begin{pmatrix} X \\ Y \end{pmatrix} = USV^* \text{ and } U = \begin{pmatrix} U_{tr} & \mathcal{U}_{tr} \\ U_{br} & \mathcal{U}_{br} \end{pmatrix} \ . \tag{2.5}$$

Namely, the projected observations X and Y are combined into a matrix of size $2r \times n$, whose r dominant left singular vectors are used to compute A. The matrix $U_{tr} \in \mathbb{C}^{r \times r}$ encodes the top left part of U and $U_{br} \in \mathbb{C}^{r \times r}$ represents the bottom left part of U. The scalar r satisfies $r < n/2$ in this method. Overall, the computational requirements of `tlsDMD` are on the order of $\mathcal{O}\left(\min\{mn^2, m^2n\} + r^3\right)$.

Finally, a recent development for computing DMD matrices was introduced in [1] resulting in the `Optimized DMD` method. Essentially, the authors formulate DMD as a non-linear least squares minimization problem. To this end, the ensemble of observations is put together, e.g., $Z = \begin{pmatrix} X & Y \end{pmatrix} \in \mathbb{R}^{m \times 2n}$, and the goal is to fit Z with a linear combination of non-linear functions $\Phi \in \mathbb{R}^{2n \times l}$. In practice, Φ is taken from a family of exponential functions such as $\Phi(\alpha, t)_j = \exp(\alpha_j t)$, where the set of parameters $\alpha \in \mathbb{C}^k$ is unknown. The optimization problem takes the form of

$$\underset{\alpha, B}{\text{minimize}} \quad |Z^T - \Phi(\alpha) B|_F^2 \ , \tag{2.6}$$

where $B \in \mathbb{C}^{l \times m}$ is the set of unknown coefficients which determine the linear superposition of non-linear functions from Φ. Observing that B can be eliminated from the optimization, problem (2.6) may be efficiently solved using the *variable projection method* [14]. We note that the DMD spectrum and the matrix A could be constructed using the computed outputs Φ and B, and we refer to [1] for further details.

2.3. ADMM. The *Alternating Direction Method of Multipliers* (ADMM) is a numerical optimization approach for efficiently solving separable objective functions. ADMM was first introduced in 1970's in [12, 10], recently popularized by [13, 5], and generalized for nonconvex optimization in [35, 11]). A general scenario for which ADMM is effective involves the following minimization problem,

$$\underset{x,z}{\text{minimize}} \quad f(x) + g(z) \ , \quad \text{s.t.} \quad Ax + Bz = c \ , \tag{2.7}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}$ and $g(z) : \mathbb{R}^m \to \mathbb{R}$ are convex functions, the linear constraints include matrices $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$ and a vector $c \in \mathbb{R}^p$. To solve (2.7), we define the following augmented Lagrangian,

$$\mathcal{L}_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}|Ax + Bz - c|_2^2 \ . \tag{2.8}$$

ADMM exploits the fact that $\mathcal{L}_\rho$ can be decomposed with respect to the variables x and z, leading to a numerical splitting scheme consisting of the iterations

$$\begin{aligned} x^{k+1} &= \arg\min \mathcal{L}_\rho(x, z^k, y^k) \\ z^{k+1} &= \arg\min \mathcal{L}_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \ , \end{aligned} \tag{2.9}$$

where $\rho > 0$ is the penalty parameter in the augmented Lagrangian. The advantage of utilizing ADMM is twofold, solving alternately for x and z typically involves simpler minimization problems compared to a joint optimization, and convergence results require mild assumptions.

It is often useful to facilitate a change of variables and to define a *scaled* version for the dual variable y, denoted by $\rho u = y$. This choice significantly reduces the length of formulas, and thus we will opt for this version throughout the paper. We denote by $r(x, z) = Ax+Bz-c$, and we re-write the scaled augmented Lagrangian in terms of u,

$$\mathcal{L}_\rho(x, z, u) = f(x) + g(z) + \frac{\rho}{2}|r + u|_2^2 - \frac{\rho}{2}|u|_2^2 \ .$$

The associated splitting scheme is similar in the x and z updates where we replace y^k with u^k in (2.9), whereas for the u update we have $u^{k+1} = u^k + r(x^{k+1}, z^{k+1})$.

3. Consistent Dynamic Mode Decomposition. In this section we describe our main algorithm for computing an approximation of the DMD operator that is associated with some known dynamical observations. The key observation in our approach is the consideration of the forward *and* backward dynamics within the same framework. In this context, we propose a variational formulation of the problem where we simultaneously solve for the forward and backward DMD operators. Unfortunately, the formulation we arrive at is highly non-linear and non-convex, and thus challenging to solve in practice. Our main contribution is an effective splitting numerical scheme which is efficient yet easy to code.

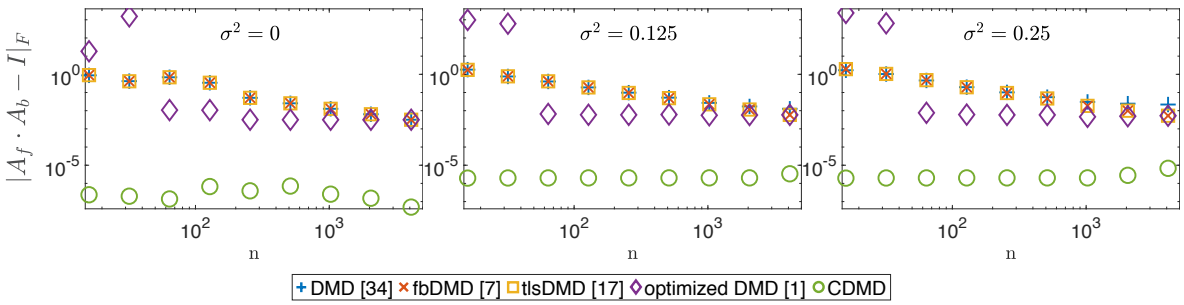

Figure 2. *Our method penalizes the obtained inconsistency of composing the forward and backward evolution operators, A, B. In practice, employing consistency constraints in our optimization regularizes the problem significantly, yielding robust estimates in the presence of noise as we show in Section 5. In the plots above, we demonstrate the consistency error for the sine example (5.2) as achieved by various DMD techniques. Our method yields extremely low error rates, whereas* `Exact DMD`*,* `fbDMD` *and* `tlsDMD` *obtain higher rates that decrease as the number of observations n increases. Finally,* `optimized DMD` *generates the second to best consistency estimates with the exception of low number of observations, where their error rates are the highest.*

3.1. Forward and backward dynamics. Let the two matrices $X, Y \in \mathbb{R}^{r\times n}$ represent our POD-projected data such that each column in X is associated with the corresponding column in Y under the dynamics (see Subsection 2.1). Several Dynamic Mode Decomposition (DMD) algorithms study the forward dynamics, i.e., find A such that $AX \approx Y$. We advocate the consideration of the backward dynamics, namely, we also want that $A^{-1}Y \approx X$. This idea was previously explored in [7, Section 2.4], where the authors proposed the `fbDMD` algorithm which takes into account both directions. However, there are a few key differences between our approach and theirs, as we detail below. Formally, we consider the following variational problem,

$$\underset{A}{\text{minimize}} \quad \frac{1}{2}\left|AX - Y\right|_F^2 + \frac{1}{2}\left|X - A^{-1}Y\right|_F^2 \ , \tag{3.1}$$

where $|\cdot|_F$ is the Frobenius norm. We note that if A is orthogonal, i.e., $A^{-1} = A^T$, then the above addends are equal, however in the general case we have

$$\begin{aligned}\left|AX - Y\right|_F^2 &= \mathrm{Tr}(X^TA^TAX - 2X^TA^TY + Y^TY) \\ &\neq \mathrm{Tr}(X^TX - 2X^TA^{-1}Y + Y^TA^{-T}A^{-1}Y) = \left|X - A^{-1}Y\right|_F^2 \ .\end{aligned}$$

3.2. Change of variables. The optimization problem (3.1) is highly non-linear and non-convex due to the A^{-1} term. Therefore, instead of directly solving this challenging problem, we introduce the auxiliary variable $B = A^{-1}$, and we re-formulate to arrive at,

$$\underset{A,B}{\text{minimize}} \quad \frac{1}{2}\left|AX - Y\right|_F^2 + \frac{1}{2}\left|X - BY\right|_F^2 \ , \quad \text{s.t.} \quad AB = I, BA = I \ , \tag{3.2}$$

where the constitutive constraints $AB = I$ and $BA = I$ guarantee that minimizers of (3.2) are inverse of each other. From an optimization point of a view, if one of the constraints is satisfied then the second constraint holds as well. However, in practice, adding both constraints is a reasonable choice as they symmetrize the approximate invertible relations of A and B. We

refer to the above re-formulation as the *Consistent Dynamic Mode Decomposition (CDMD)* problem. To motivate our methodology, we quantify the consistency error $|AB-I|_F$ obtained by several existing methods including ours, and we plot the results in Figure 2. Indeed, our technique is highly consistent compared to the other approaches, almost independently of the number of observations n. We note that when the consistency error is large, it may hint of overfitting to data, since the forward and backward estimations represent systems that are far from being inverse of each other.

The CDMD functional (3.2) appeared previously in Computer Graphics applications where a discrete map between two dimensional surfaces is being sought. Namely, given two geometric shapes such as two different poses of the same person, the goal is to determine where each point on one shape is mapped to its corresponding point on the second shape. DMD operators (also known as *functional maps* [28]) arise in this application as they allow to align features in the spectral domain and to extract a point to point map as a post processing step. With respect to CDMD, Eynard et al. [9] investigate a close variant of our CDMD problem, and solved it directly using a non-linear conjugate gradients approach. An alternative formulation was studied in [19], based on the observation that the matrix

$$Z = \begin{pmatrix} I & A \\ B & I \end{pmatrix}$$

is low-rank when $AB = I$. Instead of minimizing the rank of Z, Huang et al. [19] replace the low-rank constraint with its convex relaxation expressed via the nuclear norm [6].

Our approach depends on the following straightforward insight. Under the change of variables B, the energy functional in (3.2) becomes fully *separable*. Namely, if we denote

$$\hat{f}(A) = \frac{1}{2}|AX - Y|_F^2 \ , \quad \tilde{f}(B) = \frac{1}{2}|X - BY|_F^2 \ , \tag{3.3}$$

then we seek to minimize $\hat{f}(A)+\tilde{f}(B)$ subject to the constitutive invertibility constraints. This understanding calls for the development of an Alternating Direction Method of Multipliers (ADMM)-type approach [5]. ADMM is advantageous in effectively solving separable optimization problems, since it systematically leads to splitting schemes composed of potentially simpler minimization tasks. Moreover, the theory associated with ADMM-based techniques is well-developed with several general results related to convergence, optimality conditions and stopping criteria. Unfortunately, the constraints associated with our problem are *non-linear*, and thus while one can employ an ADMM approach, the theoretical guarantees of standard ADMM do not apply. Recently, Gao et al. [11] showed that under mild assumptions, ADMM with *multiaffine* constraints converges if the penalty parameter in the augmented Lagrangian is sufficiently large. In Section 4, we show that CDMD can be modified to fit a family of optimization problems that are considered in [11] for which converging ADMM schemes can be devised.

3.3. A splitting scheme. We now turn to present the main algorithm in this work. Our starting point is to define the augmented Lagrangian for problem (3.2) given by,

$$\mathcal{L}(A, B, Q) = \hat{f}(A) + \tilde{f}(B) + \frac{\rho}{2}|R(A, B) + Q|_F^2 - \frac{\rho}{2}|Q|_F^2 \ , \tag{3.4}$$

where $\rho \in \mathbb{R}^+$ is a scalar penalty parameter, the matrix $R(A,B)$ combines the constitutive constraints into a single matrix, and the matrix Q is the scaled dual variable (see e.g., [5, Section 3.1.1]). Specifically, the matrices R and Q are given by

$$R(A,B) = \begin{pmatrix} AB - I \\ BA - I \end{pmatrix} \in \mathbb{R}^{2r\times r} \ , \quad Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \in \mathbb{R}^{2r\times r} \ .$$

We note that if one adopts the method of multipliers approach, the augmented Lagrangian $\mathcal{L}(A,B,Q)$ could be directly minimized, as was done in [9]. However, the term $|R(A,B)+Q|_F^2$ includes a quartic combination of unknowns, and thus the optimization problem (3.4) is highly non-linear. Instead, our numerical scheme splits the updates so that A and B are not updated jointly but in an alternate fashion. Specifically, given initial A^0, B^0, Q^0 and ρ, ADMM takes the form of

1. $A^{k+1} = \arg\min_A \hat{f}(A) + \frac{\rho}{2}|R(A,B^k) + Q^k|_F^2$
2. $B^{k+1} = \arg\min_B \tilde{f}(B) + \frac{\rho}{2}|R(A^{k+1},B) + Q^k|_F^2$
3. $Q^{k+1} = Q^k + R(A^{k+1},B^{k+1})$

Below, we show that minimizing Steps (1) and (2) lead in both cases to a Sylvester Equation which can be efficiently solved using the QR decomposition, see [3] for further details. The update in Step (3) is trivial and requires a single evaluation of R. Overall, we obtain an efficient algorithm with time complexity of $\mathcal{O}(Kr^3)$, where K is the total number of iterations.

The minimization tasks in Steps (1) and (2) are relatively simple as they comprise of energy functionals that are quadratic in A and in B, respectively. Thus, the associated first order optimality conditions are *linear*. For instance, the Jacobian of the energy in Step (1) is

$$\begin{aligned} \nabla_A \left[\mathcal{L}(A,B^k,Q^k)\right] &= \nabla_A \hat{f}(A) + \frac{\rho}{2}\nabla_A \left(|R(A,B^k)+Q^k|_F^2\right) \\ &= (AX-Y)X^T + \rho\left(A\,B^k - I + Q_1^k\right)(B^k)^T + \rho(B^k)^T\left(B^kA - I + Q_2^k\right) \ . \end{aligned}$$

After re-arrangement and equating to zero, we arrive at the following *Sylvester Equation*, $C_1A + A\,C_2 = C_3$, which is linear in A. The matrices C_1, C_2 and C_3 are given by

$$\begin{aligned} C_1 &= \rho(B^k)^TB^k \ , \\ C_2 &= XX^T + \rho B^k(B^k)^T \ , \\ C_3 &= YX^T + 2\rho(B^k)^T - \rho Q_1^k(B^k)^T - \rho(B^k)^TQ_2^k \ . \end{aligned} \tag{3.5}$$

The derivation for Step (2) follows along the same lines, yielding a different Sylvester Equation $D_1B + BD_2 = D_3$ with coefficient matrices given by

$$\begin{aligned} D_1 &= \rho(A^{k+1})^TA^{k+1} \ , \\ D_2 &= YY^T + \rho A^{k+1}(A^{k+1})^T \ , \\ D_3 &= XY^T + 2\rho(A^{k+1})^T - \rho(A^{k+1})^TQ_1^k - \rho Q_2^k(A^{k+1})^T \ . \end{aligned} \tag{3.6}$$

3.4. The numerical algorithm. We summarize our technique for computing consistent dynamic mode decomposition in Algorithm 3.1. Note that Steps $1-2$ and $10-11$ are shared with Algorithm 2.1, whereas our main contribution is provided in Steps $3-9$ where the construction of the DMD matrix A is described. We note that the algorithm below describes how to compute an approximation of the forward dynamics A and its associated decomposition, however, an estimate of the backward dynamics can be extracted as well by defining $B = B^k$, where k is the last iteration index.

Algorithm 3.1 Consistent Dynamic Mode Decomposition (`CDMD`)

1: Input matrices $\tilde{X}, \tilde{Y} \in \mathbb{R}^{m \times n}$ and scalars $r, \rho \in \mathbb{R}$
2: Compute the SVD of $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^*$, and generate $X = \tilde{U}_r^* \tilde{X}, Y = \tilde{U}_r^* \tilde{Y}$
3: Initialize $A^0 = YX^+, B^0 = XY^+, Q^0 = 0$
4: **for** $k = 0, 1, 2, \ldots$ **do**
5: Solve $A^{k+1} = \texttt{sylvester}(C_1, C_2, C_3)$, using Eq. (3.5)
6: Solve $B^{k+1} = \texttt{sylvester}(D_1, D_2, D_3)$, using Eq. (3.6)
7: Update $Q^{k+1} = Q^k + R(A^{k+1}, B^{k+1})$
8: Update ρ following Eq. (3.8)
9: **end for**
10: Compute the EIG of the last A, with $Av_j = \lambda_j v_j$, where $v_j \in \mathbb{C}^r$, $\lambda_j \in \mathbb{C}$
11: The DMD spectrum is defined as the set of eigenvalues λ_j, and vectors $\psi_j = \lambda_j^{-1} \tilde{Y} \tilde{V}_r \tilde{S}_r^{-1} v_j$

3.5. Stopping criteria. To establish a practical stopping condition, we keep track of two residual quantities that are related to the primal and dual problems. A similar termination approach is described in [5]. We define the following *primal residual* and *dual residual*,

$$r^k = R(A^k, B^k) \ , \quad s^k = \rho \begin{pmatrix} A^k - A^{k-1} \\ B^k - B^{k-1} \end{pmatrix} \ , \tag{3.7}$$

where the termination rule we employ is given by $|r^k|_F \leq \epsilon^{\text{pri}}$ and $|s^k|_F \leq \epsilon^{\text{dual}}$. The tolerances ϵ^{pri} and ϵ^{dual} can be computed using absolute and relative thresholds, such as

$$\epsilon^{\text{pri}} = \sqrt{r}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max \left\{ |A^k B^k|_F, |B^k A^k|_F \right\} \ ,$$
$$\epsilon^{\text{dual}} = \sqrt{2r}\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \rho |Q^k|_F \ .$$

3.6. Dynamic update of the penalty parameter ρ. In general, varying ρ based on the current estimates of the primal and dual residuals may lead to faster convergence rates. We implement a simple scheme that was proposed in e.g., [5] and is given by

$$\rho^{k+1} := \begin{cases} \tau \rho^k & \text{if } |r^k|_F > \mu |s^k|_F \\ \rho^k / \tau & \text{if } |s^k|_F > \mu |r^k|_F \\ \rho^k & \text{otherwise,} \end{cases} \tag{3.8}$$

where we take $\tau = 2$ and $\mu = 5$ in practice.

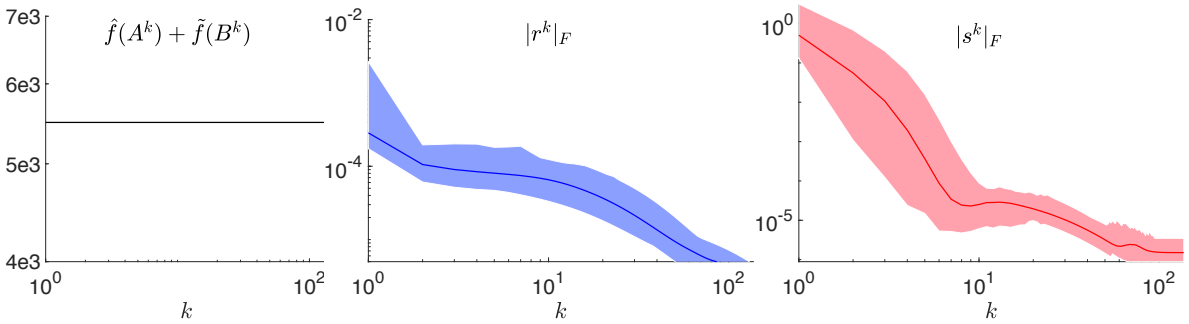

Figure 3. *The empirical convergence of our Algorithm 3.1 when applied to non-linear data with high levels of noise is demonstrated in the above plot. Our method terminates in* ≈ 100 *steps, where the objective function is stabilized on an optimal value (left) and the primal and dual residuals converge rapidly (middle and right). We repeat this test* $N = 1000$ *times, and we show the variance in convergence via the shaded areas where the average is represented by bold curves.*

4. Provably Convergent CDMD Scheme. Unfortunately, while the above Algorithm 3.1 is effective and behaves well in practice as we show in Section 5 and in Fig. 3, it is not *provably* convergent. In what follows, we address this shortcoming and propose an alternative converging scheme, which requires only an additional negligible amount of computations. To this end, we follow the recent work of Gao et al. [11] which showed that under certain conditions, ADMM and its convergence can be extended to include multiaffine constraints. In particular, we show that by introducing additional variables to the CDMD problem (3.2), the obtained minimization problem is of the required form, while satisfying all the necessary conditions in [11].

Gao et al. investigate the convergence of ADMM for problems taking the form,

$$\underset{\mathcal{A},\mathcal{B},\mathcal{C}}{\text{minimize}} \quad h(\mathcal{A},\mathcal{B},\mathcal{C}) \ , \quad \text{s.t.} \quad \mathcal{P}(\mathcal{A},\mathcal{B}) + \mathcal{Q}(\mathcal{C}) = 0 \ , \tag{4.1}$$

where $\mathcal{A} = (A_0, A_1, ..., A_{n_a})$, $\mathcal{B} = (B_0, B_1, ..., B_{n_b})$, and a variable block $\mathcal{C}$. In addition, we have that $h(\mathcal{A},\mathcal{B},\mathcal{C}) = f(\mathcal{A},\mathcal{B})+g(\mathcal{C})$. Finally, $\mathcal{Q}$ is a linear map and, in contrast to "standard" ADMM problems, $\mathcal{P}$ is a *multiaffine* map. Namely, the transformation obtained from fixing all variables A_i and B_j but one, is affine. It is shown in [11] that when several assumptions on $h, \mathcal{P}, \mathcal{Q}$ are met, an ADMM scheme converges to a constrained stationary point, i.e., the sequence $\{\mathcal{A}^k, \mathcal{B}^k, \mathcal{C}^k\}_{k=0}^{\infty}$ is bounded, and that every limit point $(\mathcal{A}^*, \mathcal{B}^*, \mathcal{C}^*)$ is a constrained stationary point. While various configurations of assumptions are considered in [11], we list here a more restrictive set of conditions that hold in our case.

Assumption 4.1. *Solving problem (4.1), the following hold.*

1. *The update order is* $A_0, A_1, ..., A_{n_a}, B_0, B_1, ..., B_{n_b}$ *and a single block* $\mathcal{C}$.
2. $\mathrm{Im}(\mathcal{Q}) \supseteq \mathrm{Im}(\mathcal{P})$.
3. *The objective* $h(\mathcal{A},\mathcal{B},\mathcal{C})$ *is coercive on the feasible set*

$$\Omega = \{(\mathcal{A},\mathcal{B},\mathcal{C}) : \mathcal{P}(\mathcal{A},\mathcal{B}) + \mathcal{Q}(\mathcal{C}) = 0\} \ .$$

4. *The function* $f(\mathcal{A}, \mathcal{B})$ *can be written as*

$$f(\mathcal{A}, \mathcal{B}) = \sum_{i}^{n_a} \hat{f}_i(A_i) + \sum_{j}^{n_b} \tilde{f}_j(B_j) \ ,$$

where every $\hat{f}_i$ *and* $\tilde{f}_j$ *are* (m_i, M_i)*- and* (m_j, M_j)*-strongly convex functions.*
5. *The function* $g(\mathcal{C})$ *is a* (m, M)*-strongly convex function.*
6. *For sufficiently large penalty* ρ*, every ADMM subproblem attains its optimal value.*

To motivate our discussion, we present an illustrative example related to Nonnegative Matrix Factorization (NMF). As we show below, this problem is similar to ours with respect to the biaffine constraints, and thus it provides a natural starting point for our case. Given a matrix Z, its NMF involves the task of finding a pair of nonnegative matrices $A \geq 0$ and $B \geq 0$ such that $Z = AB$ [23]. An ADMM formulation to NMF was originally proposed in [5], yielding the following problem,

$$\underset{A,B,C}{\text{minimize}} \quad \imath(A) + \imath(B) + \frac{1}{2}|C - Z|_F^2 \ , \quad \text{s.t.} \quad C = AB \ , \tag{4.2}$$

where $\imath$ is the indicator function, i.e., $\imath(A) = 0$ if $A \geq 0$ and $\imath(A) = \infty$ otherwise. Gao and colleagues reformulate (4.2) to arrive at an optimization problem whose subproblems are easy to solve while meeting the assumptions required for convergence. The modified version is given by

$$\begin{aligned} \underset{A,A',B,B',C,A'',B''}{\text{minimize}} \quad & \imath(A') + \imath(B') + \frac{1}{2}|C - Z|_F^2 + \frac{\mu}{2}|A''|_F^2 + \frac{\mu}{2}|B''|_F^2 \ , \\ \text{subject to} \quad & C = AB, \ A = A' + A'', \ B = B' + B'' \ . \end{aligned} \tag{4.3}$$

The update order of the variables is B, B', A, A' and (C, A'', B''). We stress that problem (4.3) satisfies a different set of assumptions than those appear in Assumption 4.1, but it is well within the family of problems considered in [11]. We refer to their paper for additional details of the NMF problem considered in relation to converging ADMM schemes.

We now turn to modify the CDMD problem (3.2) to a form which fits all the conditions in Assumption 4.1 and thus its ADMM is provably convergent, due to [11]. We observe that our invertibility constraints $AB = I$ and $BA = I$ are reminiscent of the NMF constraints, and, in particular, they are biaffine with respect to (A, B). Moreover, our objective function consists of highly smooth Frobenius norm terms. Encouraged by these similarities, we introduce the auxiliary variables C, A', A'', B', B'', and we modify the above (3.2) to arrive at the following minimization,

$$\begin{aligned} \underset{A,A',B,B',C,A'',B''}{\text{minimize}} \quad & \frac{1}{2}|A'X - Y|_F^2 + \frac{1}{2}|X - B'Y|_F^2 + \frac{\nu}{2}|C - I|_F^2 + \frac{\mu}{2}|A''|_F^2 + \frac{\mu}{2}|B''|_F^2 \ , \\ \text{subject to} \quad & C = AB, \ C = BA, \ A = A' + A'', \ B = B' + B'' \ , \end{aligned} \tag{4.4}$$

where $\nu, \mu \in \mathbb{R}^+$ are penalty parameters for the C, A'' and B'' variables.

Algorithm 4.1 Provably Convergent CDMD (`CDMD2`)

Input matrices $\tilde{X}, \tilde{Y} \in \mathbb{R}^{m\times n}$ and scalars $r, \rho, \mu \in \mathbb{R}$

Compute the SVD of $\tilde{X} = \tilde{U}\tilde{S}\tilde{V}^*$, and generate $X = \tilde{U}_r^*\tilde{X}, Y = \tilde{U}_r^*\tilde{Y}$

Initialize $A^0 = A'^0 = YX^+, B^0 = B'^0 = XY^+, A''^0 = B''^0 = 0, C^0 = I, Q^0 = 0$

for $k = 0, 1, 2, \ldots$ **do**

Solve $A^{k+1} = \texttt{sylvester}(A_1, A_2, A_3)$, where

$$\begin{aligned} A_1 &= I + (B^k)^T B^k \ , \\ A_2 &= B^k (B^k)^T \ , \\ A_3 &= (C^k - Q_1^k)(B^k)^T + (B^k)^T (C^k - Q_2^k) + A'^k + A''^k - Q_3^k \ . \end{aligned}$$

Solve $A'^{k+1} = \texttt{linsolve}\left(\rho I + XX^T, YX^T + \rho(A^{k+1} - A''^k + Q_3^k)\right)$

Solve $B^{k+1} = \texttt{sylvester}(B_1, B_2, B_3)$, where

$$\begin{aligned} B_1 &= I + (A^{k+1})^T A^{k+1} \ , \\ B_2 &= A^{k+1} (A^{k+1})^T \ , \\ B_3 &= (A^{k+1})^T (C^k - Q_1^k) + (C^k - Q_2^k)(A^{k+1})^T + B'^k + B''^k - Q_4^k \ . \end{aligned}$$

Solve $B'^{k+1} = \texttt{linsolve}\left(\rho I + YY^T, XY^T + \rho(B^{k+1} - B''^k + Q_4^k)\right)$

Solve $C^{k+1} = \frac{\rho}{2\rho+\nu}(A^{k+1}B^{k+1} + B^{k+1}A^{k+1} + Q_1^k + Q_2^k) + \frac{\nu}{2\rho+\nu} I$

Solve $A''^{k+1} = \frac{\rho}{\mu+\rho}(A^{k+1} - A'^{k+1} + Q_3^k)$

Solve $B''^{k+1} = \frac{\rho}{\mu+\rho}(B^{k+1} - B'^{k+1} + Q_4^k)$

Update $Q^{k+1} = Q^k + \mathcal{R}(\mathcal{A}^{k+1}, \mathcal{B}^{k+1}, \mathcal{C}^{k+1})$

Update ρ following Eq. (3.8)

end for

Execute steps (10) – (11) of Algorithm 3.1

To verify that (4.4) meets all the required conditions, we denote $\hat{f}(A') = \frac{1}{2}|A'X - Y|_F^2$, $\tilde{f}(B') = \frac{1}{2}|X - B'Y|_F^2$, and $g(C, A'', B'') = \frac{\nu}{2}|C - I|_F^2 + \frac{\mu}{2}|A''|_F^2 + \frac{\mu}{2}|B''|_F^2$. Also, we define the following residual

$$\mathcal{R}(\mathcal{A}, \mathcal{B}, \mathcal{C}) = \mathcal{P}(A, A', B, B') + \mathcal{Q}(C, A'', B'') = \begin{pmatrix} AB \\ BA \\ A - A' \\ B - B' \end{pmatrix} + \begin{pmatrix} -C \\ -C \\ -A'' \\ -B'' \end{pmatrix} \ .$$

The conditions in Assumption 4.1 hold because the update order is A, A', B, B' and (C, A'', B'') as we show below in Algorithm 4.1. The image of $\mathcal{Q}$ is indeed a superset of $\mathcal{P}$'s image, since it is the (minus) identity transformation in each of its entries, and thus span the entire space.

The objective function h is coercive on the feasible set, because its terms behave as $|x|_F^2$, and therefore whenever $|x|_F \to \infty$ so does $|x|_F^2$. Under some mild conditions, namely, that X and Y are full rank matrices, the function f is composed of (m, M)-strongly convex functions as we show in Appendix A. Similarly, g is a strongly convex function because the Hessian of its terms is positive definite. Finally, the subproblems in our formulation are trivial, linear or a Sylvester-type equation and thus attain their optimal value when ρ is sufficiently large.

We conclude this section with presenting our convergent ADMM scheme along with the specification of its subproblems. The derivation of the matrix expressions that take part in lines 5 and 7 could be carried over in a fashion similar to Eqs. (3.5) and (3.6). We note that lines 6 and 8 of Algorithm 4.1 involve a call to $X = \texttt{linsolve}(A, B)$ which numerically solves the system $XA = B$.

5. Results. In this section, we evaluate the proposed `CDMD` approach and compare it to several state-of-the-art techniques for computing DMD matrices. In particular, we compare against `Exact DMD` [34], `fbDMD` [7], `tlsDMD` [17] and `optimized DMD` [1]. The dynamical systems we consider appeared previously e.g., in [7, 1], and thus can be considered as "benchmark" examples for quantitative and qualitative study of DMD algorithms.

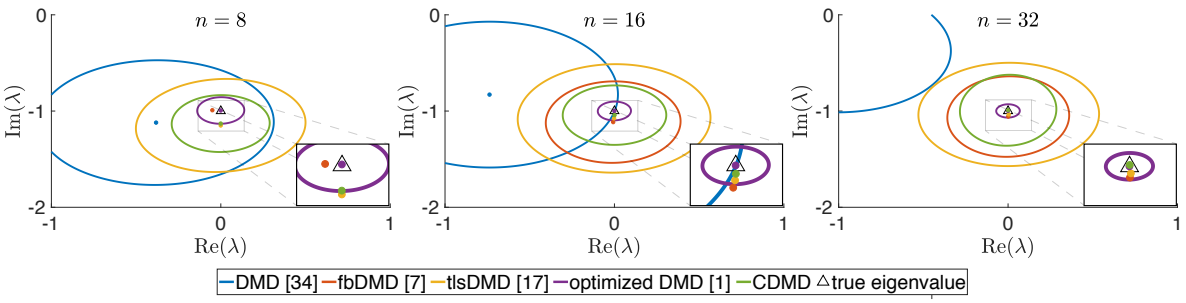

Figure 4. *We plot 95% confidence ellipses (see [7]) for estimating one of the eigenvalues of a periodic linear system (5.1) when varying number of observations $n = 8, 16, 32$ are given. The zoom in boxes show the average estimation for each method. The results above indicate that `CDMD` is second to best in terms of accuracy and variance for all values of n.*

5.1. A periodic linear system. In this example, we use the following linear and non-normal system

$$\dot{z} = \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix} z \ , \tag{5.1}$$

where the system has purely imaginary eigenvalues that are given by $\lambda = \pm i$. Eq. (5.1) is integrated over the $[0, 2\pi]$ temporal segment, starting from the initial point $z_0 = [1\ 0.1]^T$. To stress test our method, we investigate this system when relatively low number of observations is given and high levels of white Gaussian noise affect the data. Specifically, we show in Figure 4 the performance of various methods for computing the eigenvalue $-i$ when noise with variance $\sigma^2 = 0.1$ and Signal-to-Noise (SNR) ratio of 8.6 dB is introduced. We repeat our experiment $N = 10^4$ times, and the average of each of the methods is marked by a dot with a corresponding color. Additionally, we plot the ellipses which enclose the region of 95% of the estimates that

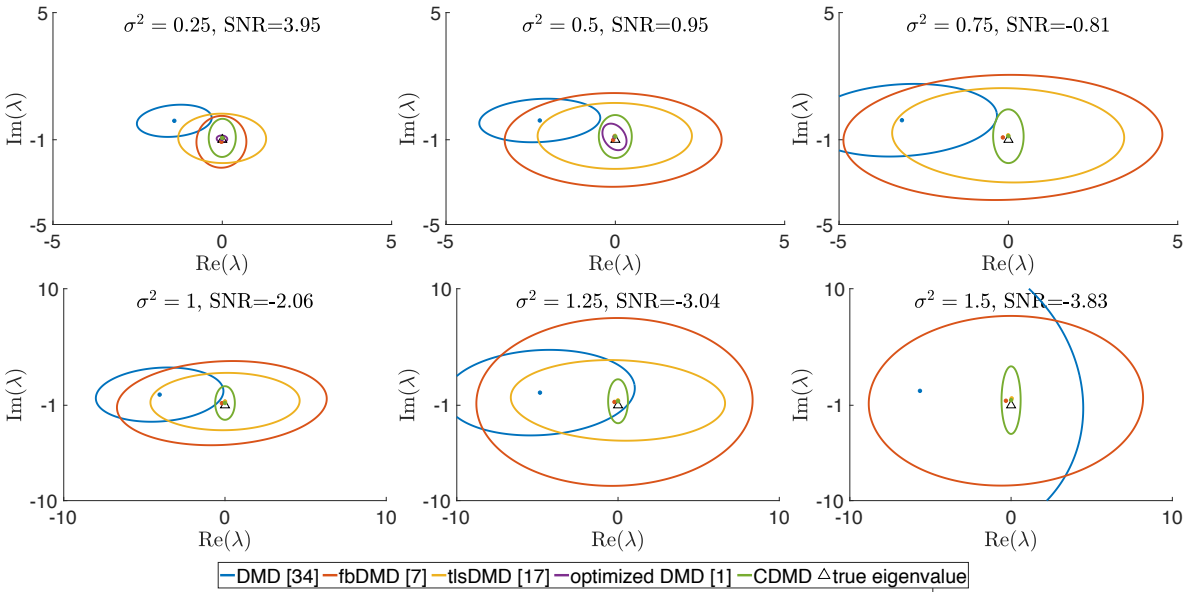

Figure 5. *We compare the above methods based on their approximation for the eigenvalue of System (5.1) when various levels of noise are introduced, $-4 \leq SNR \leq 4$. Interestingly, while* `optimized DMD` *is extremely accurate when $SNR > 0$, it fails for higher levels of noise, and thus it does not appear on these graphs. The methods* `fbDMD` *and* `tlsDMD` *perform well in terms of average, but their spread is much larger than our results which maintain relatively small spread as well as accurate average.*

are closest to the true eigenvalue for each of the techniques. We use the values $n = 8, 16, 32$ for the number of observations, which make the system overdetermined as it is two-dimensional. Nevertheless, these values are relatively small in comparison to related work on this example, see e.g., [7].

Overall, `optimized DMD` achieves excellent results in terms of spread and average values, across all values of n. On the other end, `exact DMD` struggles both in accuracy and spread. `fbDMD` and `tlsDMD` exhibit comparable performance, except for $n = 8$ where `fbDMD` produces a correct mean, but with an extremely large deviation. Finally, our approach outputs consistent deviation and averages, regardless of the value of n. We additionally experiment with various high level of noise $-4 \leq \text{SNR} \leq 4$ and present the results in Figure 5. Note that the bottom row axes are twice as large as the axes in the top row. As can be seen in the graphs, `optimized DMD` is very accurate as long as $\text{SNR} > 0$, but fails when the signal-to-noise ratio drops below zero, and therefore it is omitted from the other graphs. In most cases, `Exact DMD` produces poor approximations when compared to the other methods. In comparison, `fbDMD` and `tlsDMD` generate estimates that are centered around the eigenvalue in general, with growing spread as the SNR decreases. Remarkably, our approach exhibits the least increase in deviation when compared to all other techniques, while producing a relatively accurate average.

In addition, we reconstruct the trajectory using the approximations of the dynamics provided by each of the methods, and we plot the results in Fig. 6 separated to y-coordinate (top row) and x-coordinate (bottom row) over time. It is evident that `Exact DMD` yields a

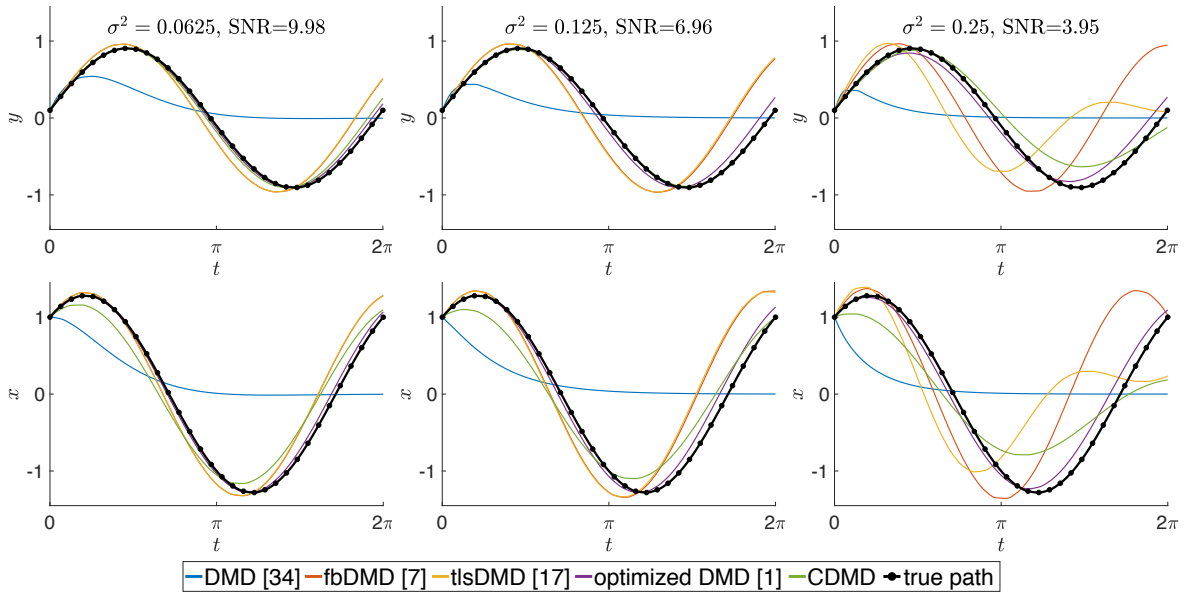

Figure 6. *We reconstruct the trajectory of the periodic system (5.1) using the computed DMD matrices for various noise variances with 32 observations. Most methods yield paths that are close to the true trajectory, where* `optimized DMD` *and our method obtain the best results.*

highly distorted path, whereas the other methods are generally close to the true trajectory. As the amount of noise increases, `fbDMD` and `tlsDMD` develop a significant shift in phase. We measure the distance between the computed paths to the desired curve and we observe that our method achieves second to best results after `optimized DMD`. Specifically, for $\sigma^2 = 0.125$, the L_2 error between the computed path to the ground-truth trajectory divided by the length of the latter is 0.0837 and 0.2611 for `optimized DMD` and `CDMD`, respectively. When $\sigma^2 = 0.25$, the error is 0.1403 and 0.7844 for `optimized DMD` and `CDMD`. In comparison, the other methods yield errors that are five times larger or more.

5.2. Dominant and hidden dynamics. The next system is a superposition of a growing sine function and a decaying sine function given by

$$z(x,t) = \sin(k_1 x - \omega_1 t)\exp(\gamma_1 t) + \sin(k_2 x - \omega_2 t)\exp(\gamma_2 t)\ , \tag{5.2}$$

where in our experiments we used $k_1 = 1$, $\omega_1 = 1$, $\gamma_1 = 1$ and $k_2 = 0.4$, $\omega_2 = 3.7$, $\gamma_2 = -0.2$. This example is more challenging than the previous one since it involves dynamical features which are of lower magnitude alongside dominant structures. The eigenvalues of this system are of the form $\gamma_i \pm \omega_i, i = 1, 2$, where the "dominant" mode is associated with $i = 1$ and the "hidden" mode is linked to $i = 2$. In Figure 7, we compute $N = 10^4$ times the eigenvalues of the system while employing a noise level of $\sigma^2 = 0.25$, SNR $= 30$ dB over the observations. The results show that for the dominant dynamics, most methods perform well where `optimized DMD` obtains improved estimates as n increases (top row). For the hidden mode, similar results are obtained for $n = 16, 32$, whereas for the lowest $n = 8$, `fbDMD` does not appear in the plot and `tlsDMD` is shifted differently than the other approaches (bottom row).

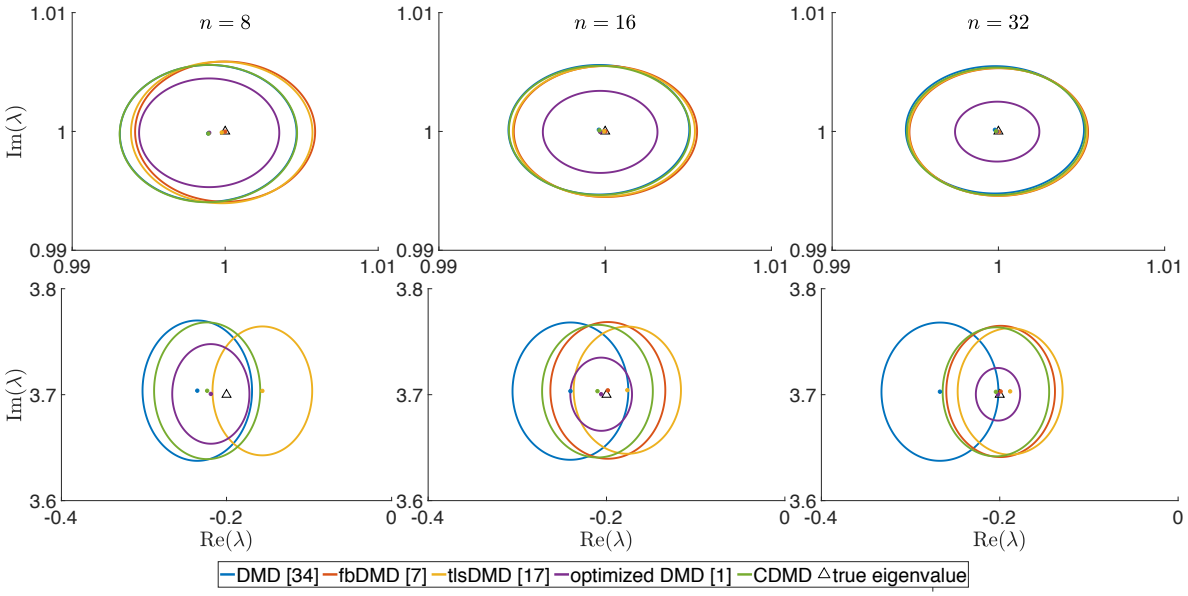

Figure 7. *Given a noisy superposition of sine functions, we estimate the system's eigenvalues for various number of observations n and noise with variance $\sigma^2 = 0.25$ and SNR = 30 dB. We observe that the dominant eigenvalue is approximated well (top row), whereas the hidden dynamics is achieved by most methods with decreasing error as n grows (bottom row).*

In addition, we investigate this system across different levels of noise. In particular, we set $\sigma^2 = 2^{-2},\ 2^{-1}, ..., 2^{10}$ corresponding to SNR in the range $[-10, 30]$. Each noise level is used $N = 10^3$ times, for which we compute both the dominant and hidden DMD eigenvalues. We show the error results of the different methods in Fig. Figure 8, where the error is a linear combination of the average error between the computed eigenvalue and the ground-truth and the minimum radius of the deviation ellipse. Formally,

$$\mathcal{E} = a|\lambda_{\text{avg}} - \lambda_{\text{gt}}| + (1 - a)r_{min}\ , \tag{5.3}$$

where λ_{avg} is the average taken over all eigenvalue estimates, λ_{gt} is the analytic eigenvalue, and r_{min} is the minimum radius. In our experiments, we used $a = 0.9$. Similar to Fig. Figure 5, when SNR approaches zero, `optimized DMD` fails and thus its graphs are shorter. Interestingly, up to a certain SNR, all methods present similar error behavior, where at SNR ≈ 17 there is an exponential increase in the error estimates. When inspecting the individual results, it seems like this high level of noise leads to an extremely large deviation in results, which further affects our error measure.

5.3. Cylinder wake. The last example we consider in this work is of a fluid flow past a cylinder simulated using a numerical solver. We obtain a time series of fluid vorticity fields consisting of $n = 150$ snapshots regularly sampled in time with $\Delta t = 0.2$. We refer to [22] for additional details regarding this dataset such as the chosen physical parameters and other numerical considerations. It is important to note that this particular flow is inherently *non-linear* and thus the underlying assumptions of methods such as `optimized DMD` may not hold. Specifically, it is unclear which functions to fit and whether exponential functions are a good

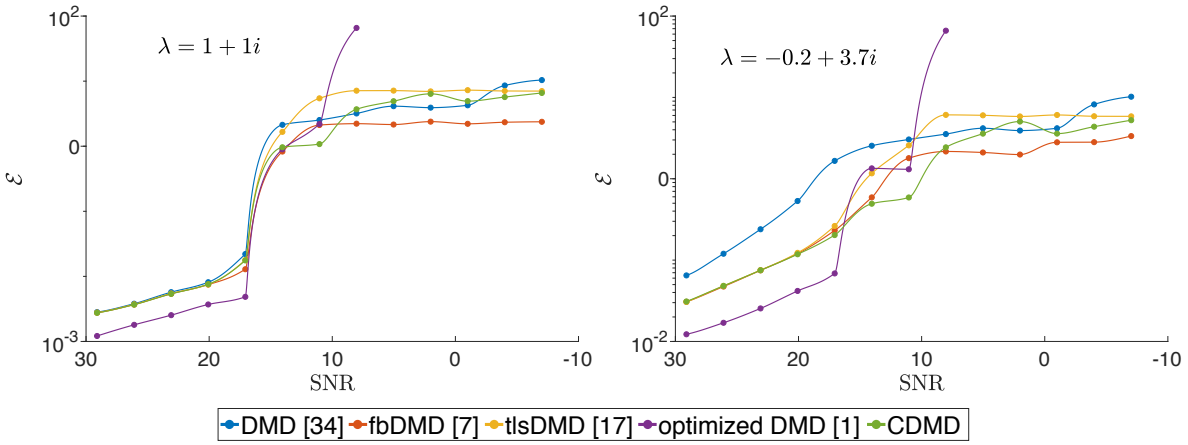

Figure 8. *We compute approximations of the dominant and hidden DMD eigenvalues for various levels of noise, $-10 \leq SNR \leq 30$ and we compute the error for each method using Eq. (5.3). Naturally, for low levels of noise, most methods perform reasonably well, whereas when $SNR \leq 17$ the spread becomes orders of magnitude larger. See the text for further details.*

choice in this scenario. In contrast, our approach (as well as other DMD techniques) does not impose restricting conditions on the input data, making it applicable in such challenging scenarios. In Figure 9, we repeatedly compute the eigenvalues associated with a noisy version of the input data for various noise levels, and we plot the average results as compared to the estimates obtained from the clean observations. Specifically, we repeat this experiment $N = 1000$ times for noise with variance $\sigma^2 = 0.001, 0.01, 0.1$ and SNR $= 30, 20, 10$ dB, respectively. Clearly, `Exact DMD` exhibits a bias in its estimations which is consistent with previous reports such as [7]. On the other hand, `fbDMD` and `tlsDMD` generate improved approximations of the eigenvalues with less accuracy as the noise increases. Our approach is successful in measuring nearly zero growth for all eigenvalues and noise levels with a bias in frequencies for the least dominant eigenvalues. In Figure 10, we demonstrate the averaged dominant DMD modes obtained for $\sigma^2 = 0.1$. In this case, all methods perform comparably well in the noiseless case, where the averaged modes associated with less dominant eigenvalues are clearly noisier.

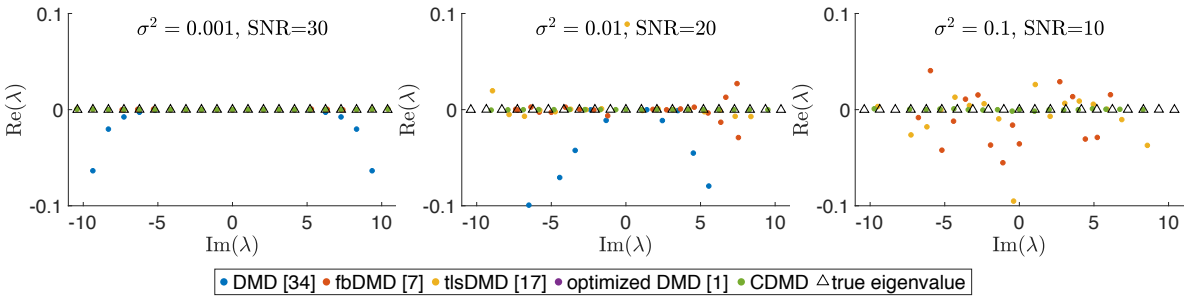

Figure 9. *We compute $r = 21$ eigenvalues using simulation snapshots of a cylinder wake for noise levels $\sigma^2 = 0.001, 0.01, 0.1$ respectively $SNR = 30, 20, 10$ dB. As the noise increases, our method maintains its zero growth estimate (notice that the x-axis represents the real part, cf. Fig. Figure 7), whereas the other methods produce significant erroneous growth/decay estimates.*

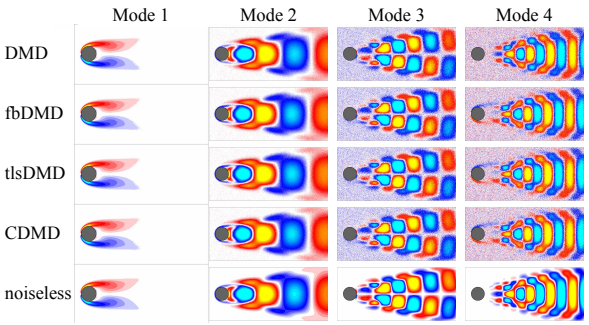

Figure 10. *The averaged DMD modes associated with the cylinder wake are shown for data consisting of noise with variance $\sigma^2 = 0.1$. In spite of the large amounts of white Gaussian noise, all methods produce relatively good estimations when compared to the noiseless scenario (bottom row).*

6. Discussion and Future Work. In this work, we presented a new method for computing Dynamic Mode Decomposition operators that is based on a variational formulation of the underlying problem, while taking into account the forward and backward dynamics. The obtained minimization is solved using an effective splitting ADMM scheme, which performs well in practice in terms of computational requirements and achieved accuracy. Moreover, it is shown that CDMD could be modified to a provably convergent ADMM scheme at the cost of insignificant additional computations. We demonstrate the performance of our method on a few benchmark dynamical systems, compared to several state-of-the-art approaches. Our conclusion is that the generality of our model, along with its improved accuracy for high levels of noise and low number of observations, makes it an interesting alternative among current existing techniques.

One limitation of our approach is related to the non-linearity and non-convexity of the problem we aim to solve. In particular, it is not clear at this point whether the obtained minimizers are local or global, which is a general challenge in these type of problems, as was also noted in [1]. Another difficulty associated with our work involves the interplay between the chosen value of the penalty parameter ρ and the obtained solutions. While in general our technique is robust to the initial value of ρ due to scheme (3.8), it still affects our results to some extent, as can be seen in Figure 2, where for large values of n, our consistency error increases. Finally, our algorithm is more computationally demanding compared to the alternatives. However, this is highly dependent on the particular implementation and choice of parameters such as convergence thresholds and thus can be reduced, depending on the particular application at hand.

We believe that formulating DMD in a variational form is important as other regularizers may be considered along with our consistency constraints such as sparsity promoting penalty terms [20]. We leave this consideration for future work. Moreover, we would like to explore the relation of our approach to existing techniques such as `tlsDMD`. Another interesting direction is to combine the current work with methods that numerically compute an optimal basis [36]. The associated problem is extremely challenging as it is of high dimension, non-linear and typically non-convex. We believe that some of the ideas that we presented in this work could be generalized to this case and we plan on pursuing this direction in the future.

Appendix A. Convexity of $f(\mathcal{A}, \mathcal{B})$.

The function $f(\mathcal{A}, \mathcal{B})$ is (m, M)-strongly convex if each of its terms is strongly convex. Thus, we show it for the first term $\hat{f}(A') = \frac{1}{2}|A'X - Y|_F^2$, and we note that a similar derivation could be carried for the other term. We recall the gradient of $\hat{f}(A')$ and we vectorize it to arrive at the following formulation

$$\nabla \hat{f}(A') = (A'X - Y)X^T = A'XX^T - YX^T \equiv (XX^T \otimes I)\,\mathrm{vec}(A') - \mathrm{vec}(YX^T)\ .$$

Therefore, when viewed as a vectorized function, the Hessian of $\hat{f}$ is given by $\nabla^2 \hat{f} = XX^T \otimes I$. The matrix $X \in \mathbb{R}^{r \times n}$ can be assumed to have full rank, since $r \ll n$, and thus XX^T is positive definite (PD). It is known that the product of two PD matrices is also PD, which means that there exists a scalar $m > 0$ such that the Hessian $\nabla^2 \hat{f} - mI$ is positive semi-definite, and we conclude that $\hat{f}$ is an m-strongly convex function. Finally, $\hat{f}$ is also M-Lipschitz differentiable since $|(A'_1 - A'_2)XX^T|_F \leq |XX^T|_F \cdot |(A'_1 - A'_2)|_F$ and $|XX^T|_F$ is positive and bounded.

REFERENCES

[1] T. Askham and J. N. Kutz, *Variable projection methods for an optimized dynamic mode decomposition*, SIAM Journal on Applied Dynamical Systems, 17 (2018), pp. 380–416.

[2] S. Bagheri, *Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes, and Koopman spectrum*, Physics of Fluids, 26 (2014).

[3] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation ax+ xb= c [f4]*, Communications of the ACM, 15 (1972), pp. 820–826.

[4] G. Berkooz, P. Holmes, and J. L. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annual Review of Fluid Mechanics, 25 (1993), pp. 539–575.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 1–122.

[6] E. J. Candès, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, Journal of the ACM (JACM), 58 (2011), p. 11.

[7] S. T. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, *Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition*, Experiments in Fluids, 57 (2016), p. 42.

[8] Z. Drmac, I. Mezic, and R. Mohr, *Data driven modal decompositions: analysis and enhancements*, SIAM Journal on Scientific Computing, 40 (2018), pp. A2253–A2285.

[9] D. Eynard, E. Rodola, K. Glashoff, and M. M. Bronstein, *Coupled functional maps*, in 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp. 399–407.

[10] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.

[11] W. Gao, D. Goldfarb, and F. E. Curtis, *ADMM for multiaffine constrained optimization*, arXiv preprint arXiv:1802.09592, (2018).

[12] R. Glowinski and A. Marroco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, ESAIM: Mathematical Modelling and Numerical Analysis, 9 (1975), pp. 41–76.

[13] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 323–343.

[14] G. H. Golub and V. Pereyra, *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM Journal on numerical analysis, 10 (1973), pp. 413–432.

[15] F. Guéniat, L. Mathelin, and L. R. Pastur, *A dynamic mode decomposition approach for large and arbitrarily sampled systems*, Physics of Fluids, 27 (2015).

[16] M. Hemati, E. Deem, M. Williams, C. W. Rowley, and L. N. Cattafesta, *Improving separation control with noise-robust variants of dynamic mode decomposition*, in 54th AIAA Aerospace Sciences Meeting, 2016, p. 1103.

[17] 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, 31 (2017), pp. 349–368.

[18] M. S. Hemati, M. O. Williams, and C. W. Rowley, *Dynamic mode decomposition for large and streaming datasets*, Physics of Fluids, 26 (2014), p. 111701.

[19] Q. Huang, F. Wang, and L. Guibas, *Functional map networks for analyzing and exploring large shape collections*, ACM Transactions on Graphics (TOG), 33 (2014).

[20] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, *Sparsity-promoting dynamic mode decomposition*, Physics of Fluids, 26 (2014).

[21] B. O. Koopman, *Hamiltonian systems and transformation in Hilbert space*, Proceedings of the National Academy of Sciences, 17 (1931), pp. 315–318.

[22] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM, 2016.

[23] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing Systems, 2001, pp. 556–562.

[24] R. Leroux and L. Cordier, *Dynamic mode decomposition for non-uniformly sampled data*, Experiments in Fluids, 57 (2016).

[25] L. Massa, R. Kumar, and P. Ravindran, *Dynamic mode decomposition analysis of detonation waves*, Physics of Fluids, 24 (2012).

[26] T. Nonomura, H. Shibata, and R. Takaki, *Dynamic mode decomposition using a kalman filter for parameter estimation*, AIP Advances, 8 (2018), p. 105106.

[27] T. Nonomura, H. Shibata, and R. Takaki, *Extended-kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising*, PloS one, 14 (2019), p. e0209836.

[28] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas, *Functional maps: a flexible representation of maps between shapes*, ACM Transactions on Graphics (TOG), 31 (2012).

[29] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, Journal of Fluid Mechanics, 641 (2009), pp. 115–127.

[30] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of Fluid Mechanics, 656 (2010), pp. 5–28.

[31] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, *Applications of the dynamic mode decomposition*, Theoretical and Computational Fluid Dynamics, 25 (2011), pp. 249–259.

[32] A. Seena and H. J. Sung, *Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations*, International Journal of Heat and Fluid Flow, 32 (2011), pp. 1098–1110.

[33] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang, *Spectral analysis of fluid flows using sub-Nyquist-rate PIV data*, Experiments in Fluids, 55 (2014).

[34] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: theory and applications*, arXiv preprint arXiv:1312.0041, (2013).

[35] Y. Wang, W. Yin, and J. Zeng, *Global convergence of ADMM in nonconvex nonsmooth optimization*. Journal of Scientific Computing, https://doi.org/10.1007/s10915-018-0757-z, 2018.

[36] A. Wynn, D. Pearson, B. Ganapathisubramani, and P. J. Goulart, *Optimal mode decomposition for unsteady flows*, Journal of Fluid Mechanics, 733 (2013), pp. 473–503.