# Dynamic mode decomposition with control

Joshua L. Proctor<sup>1\*</sup>, Steven L. Brunton<sup>2</sup>, J. Nathan Kutz<sup>2</sup>

<sup>1</sup>Institute of Disease Modeling Bellevue, WA 98004, United States

<sup>2</sup>Applied Mathematics, University of Washington, Seattle, WA 98195, United States

DRAFT: last updated September 24, 2014

Abstract—We develop a new method which extends Dynamic Mode Decomposition (DMD) to incorporate the effect of control to extract low-order models from high-dimensional, complex systems. DMD finds spatial-temporal coherent modes, connects local-linear analysis to nonlinear operator theory, and provides an equation-free architecture which is compatible with compressive sensing. In actuated systems, DMD is incapable of producing an input-output model; moreover, the dynamics and the modes will be corrupted by external forcing. Our new method, Dynamic Mode Decomposition with control (DMDc), capitalizes on all of the advantages of DMD and provides the additional innovation of being able to disambiguate between the underlying dynamics and the effects of actuation, resulting in accurate input-output models. The method is data-driven in that it does not require knowledge of the underlying governing equations, only snapshots of state and actuation data from historical, experimental, or black-box simulations. We demonstrate the method on highdimensional dynamical systems, including a model with relevance to the analysis of infectious disease data with mass vaccination (actuation).

## I. INTRODUCTION

We introduce the method of Dynamic Mode Decomposition with control (DMDc) to analyze observational data arising from complex, high-dimensional systems that exhibit dynamics and require control. By utilizing both measurements of the system and the applied external control, the underlying, unforced dynamics can be extracted and specified in an equation-free manner, i.e. the underlying equations of motion do not have to be known. In addition, a description of how the control inputs affect the system are also discovered and characterized. With a quantitative understanding of the input-output characteristics, a reduced-order-model can be generated for both prediction and design of controllers for high-dimensional, complex systems.

Controlling high-dimensional systems remains an extremely challenging task as many control strategies do not scale well with the dimension of the system. In particular, controllers developed on a full system may be computationally prohibitive to implement, introducing unacceptably large latencies. Moreover, many control laws are determined by solving a large Riccati equation ( $\mathcal{H}_2$ ) or through an iterative procedure ( $\mathcal{H}_\infty$ ), constituting an enormous up-front cost. Thus, practical engineering control strategies for dealing with high-dimensional observational data revolves around dimensionality-reduction techniques. Such methods, often based upon the singular value decomposition of the data, allow one to construct low-dimensional subspaces where computationally tractable con-

\* Corresponding author. Tel.: +1 509 868 5696. E-mail address: joproctor@intven.com (J.L. Proctor). trollers can be designed and implemented [29], [20], [17], [31]. Balanced truncation is a classic method developed to specifically take advantage of underlying low-dimensional observable and controllable subspaces to create a balanced, reduced-order model [29]. Generalizations of this scheme which combine balanced truncation with the SVD on empirical data, such as the balanced proper orthogonal decomposition, have already been shown to overcome some of the computational difficulties associated with the high-dimension of complex systems, but still requires a pernicious linear adjoint calculation [40], [31], [18]. Further innovations around system identification methods, such as the Eigensystem Realization Algorithm (ERA) and the Observer Kalman Filter Identification (OKID), were developed to aid in the discovery of input-output models for systems with control [20], [21]. The dimension of the measurements, though, were assumed to be low and the system

DMDc has a number of advantages for high-dimensional, complex systems. First, it is based upon the DMD algorithm which is a data-driven, equation-free architecture that reconstructs the underlying dynamics of the system from snapshot measurements alone [34], [30], [8], [38]. Substantial success has been achieved in the application of DMD to fields such as fluid dynamics which have been historically difficult to analyze and construct controllers due to the enormous number of spatial states required for simulation [14], [33], [35], [2], [39], [37]. Second, DMD has acquired popularity as a method for systems with nonlinear dynamics, due to a strong connection between DMD and Koopman operator theory [24], [26], [30], [6], [27]. Finally, DMD can be modified to take advantage of sparse, or limited, measurements of the complex system [5], [39], [19]. Sparse measurements have recently been leveraged in a variety of complex systems, some for control [3], [32], [25], [12]. Such a scenario arises in many physical, biological and engineering systems due to limited numbers of sensors. Such advantages, in combination with the control architecture advocated here, warrant serious consideration of the DMDc as an equation-free control strategy in complex systems.

As a motivating example, DMDc can be applied to the field of computational epidemiology focusing on the eradication of diseases. The advent of new monitoring tools and a substantial focus on the quantitative assessment of resource allocation is beginning to generate large sets of data describing the spread of infectious disease. A substantial literature exists focused on mathematically modeling the spread of infectious disease and the effect of external control (e.g. vaccinations for Polio and bed nets for Malaria) [1]. A common challenge in computational epidemiology is deciding how to model the spread of

disease leading to an enormous number of phenomenological models [22]. Equation free techniques such as DMD and DMDc provide a complementary modeling tool for analyzing the spatial-temporal spread of infectious disease. Focusing on *only* the historical data containing state information (i.e. number of infections in a spatial location in a given time) and whether control interventions have been applied (i.e. number of vaccinations in a spatial location in given time), DMDc discovers the dynamical properties of the complex systems.

The outline of the paper is as follows: § II describes the background on the method DMD. § III describes the new method Dynamic Mode Decomposition with control. The following section § IV presents a number of numerical examples including an artificial application based on an epidemiological problem. § V discusses a number of similarities and differences from system identification methods.

#### II. BACKGROUND: DYNAMIC MODE DECOMPOSITION

Dynamic Mode Decomposition (DMD) is a powerful datadriven method for analyzing complex systems. Using measurement data from numerical simulations or laboratory experiments, DMD attempts to extract important dynamic characteristics such as unstable growth modes, resonance, and spectral properties. This section provides the mathematical mathematical background of DMD [34], [30], [38].

#### A. Dynamical systems and data

The fundamental assumption that connects the state of a linear dynamical system  $x_k$  to the next  $x_{k+1}$  is

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

where  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . The process under observation is often continuous (whether from a numerical model or experiment), and measurements of the continuous state  $\mathbf{x}(t)$ can be collected at regular time intervals  $\Delta t$  denoted by  $\mathbf{x}_k = \mathbf{x}(k\Delta t)$ . Each measurement in time  $\mathbf{x}_k$  will be referred to as snapshots within this manuscript [36]. We denote the sequence of snapshots collected by the following description:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix},$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}, \tag{2}$$

where m is total number of snapshots and X' is the timeshifted snapshot matrix of X, i.e. X' = AX. For DMD, data is often collected at regular time intervals  $\Delta t$ . The number of snapshots required for DMD varies with the application, but is intimately related to the linearity properties of the Koopman Operator. The solution will converge by decreasing the recording interval  $\Delta t \rightarrow 0$ , thus indicating the number of snapshots required (for an illuminating numerical example see [34]). New directions for DMD have focused on novel paradigms for collecting data in time [38] and across the state of the system [5]. Each are utilizing the concepts of sparsity and compressed sensing techniques [4].

The dynamical system Eq. (1) and data snapshots Eq. (2) can be described more compactly in the following form:

$$X = AX'. (3)$$

Solving for an approximation of the process matrix A given the data matrices X and X' is the primary objective of DMD.

## B. Dynamic Mode Decomposition

The following section describes how to find the dynamic modes and eigenvalues of the underlying system A described in Eq. (3). We can find A by using the following definition:

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger},\tag{4}$$

where † is the Moore-Penrose pseudoinverse [38]. A computationally efficient and accurate method for finding the pseudo inverse is via the singular value decomposition (SVD). The SVD of X results in the well-known decomposition:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^* = \begin{bmatrix} \tilde{\mathbf{U}} & \tilde{\mathbf{U}}_{rem} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & 0 \\ 0 & \boldsymbol{\Sigma}_{rem} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{V}}^* \\ \tilde{\mathbf{V}}_{rem}^* \end{bmatrix}, \quad (5)$$

$$\approx \tilde{\mathbf{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}}^* \qquad (6)$$

where  $\mathbf{U} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{n \times m-1}$ ,  $\tilde{\mathbf{V}}^* \in \mathbb{R}^{m-1 \times m-1}$ ,  $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times r}$ ,  $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r \times r}$ ,  $\tilde{\mathbf{V}}^* \in \mathbb{R}^{r \times m-1}$ , rem indicates the remaining m-1-r singular values, and \* denotes the complex conjugate transpose. Eq. (6) demonstrates how to reduce the dimension of the data matrix X by appropriately choosing a truncation value r of the singular values thus eliminating the remainder (rem) terms and allowing for the psuedo-inverse to be accomplished since  $\hat{\Sigma}$  is square. Choosing the appropriate truncation value r has a rich scientific history; notably, the Eckart-Young theorem provides a rigorous and popular method for choosing r [11], [28], [13]. In addition, there are recent theoretical developments attempting to identify the correct rwhen X may have additive noise [9], [10].

Using the SVD of the snapshot matrix X in Eq. (6), the following approximation of the matrix A can be computed:

$$\mathbf{A} \approx \bar{\mathbf{A}} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}^*, \tag{7}$$

where  $\bar{\bf A}$  is an approximation of the operator  $\bf A$  from Eq. (6). A dynamic model of the process can be constructed given by the following:

$$\mathbf{x}_{k+1} = \bar{\mathbf{A}}\mathbf{x}_k,\tag{8}$$

where x and  $\bar{A}$  have the same dimension as the matrices described earlier in Eq. (1). An eigenvalue analysis of the matrix A would produce the dynamic modes and eigenvalues of the system. The computation, though, can be prohibitively expensive if  $n \gg 1$ .

If  $r \ll n$ , a more compact and computationally efficient model can be found by projecting  $x_k$  on to a linear subspace of dimension r. This basis transformation takes the form  $Px = \tilde{x}$ . As previously shown by DMD, a convenient transformation has already been computed via the SVD of X, given by P =U. The reduced-order model can be derived as follows:

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{U}}^* \bar{\mathbf{A}} \tilde{\mathbf{U}} \tilde{\mathbf{x}}_k$$

$$= \tilde{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{x}}_k$$
(10)

$$= \tilde{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{x}}_k \tag{10}$$

$$= \mathbf{A}\tilde{\mathbf{x}}_k. \tag{11}$$

The reduced-order-model is given by the following:

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1}. \tag{12}$$

The eigendecomposition of  $\tilde{\mathbf{A}}$  defined by  $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda$  yields eigenvalues and eigenvectors that can be investigated for fundamental properties of the underlying system such as growth modes and resonance frequencies. In addition, the computation is efficient since  $\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}$  and  $r \ll n$ .

**Remark** Computing the eigendecomposition of  $\tilde{\mathbf{A}}$  versus  $\bar{\mathbf{A}}$  can be a computationally crucial step for efficiency. For example, the domain discretization of a fluids or epidemiological problem can have an arbitrarily large set of dimensions n. The direct solution of the  $n \times n$  eigenvalue problem might not be feasible, thus solving the  $r \times r$  is substantially more attractive. The observation is reminiscent of the Method of Snapshots by Sirovich [36].

For DMD, the eigenvalues of  $\tilde{\bf A}$  and  $\bar{\bf A}$  are equivalent [34] and the eigenvectors are related via a linear transformation. The eigenvectors of  $\bar{\bf A}$  are called dynamic modes [34], [38]. Note, there is a difference between computing the dynamic modes with the Exact DMD method from Tu et.al. [38] and Schmid [34]. Here we describe the Exact DMD method giving the following relationship between the eigenvectors of  $\tilde{\bf A}$  and the dynamic modes  $\phi$  of  $\bar{\bf A}$ :

$$\phi = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \mathbf{w}. \tag{13}$$

If  $\lambda \neq 0$ , then this is the DMD mode for  $\lambda$ . If the eigenvalue is 0, then the dynamic mode is computed using  $\phi = \mathbf{U}\mathbf{w}$ . The Exact DMD algorithm has a number of advantages over the original procedure; for a detailed discussion, see [38].

#### III. DYNAMIC MODE DECOMPOSITION WITH CONTROL

This section presents the mathematical description of Dynamic Mode Decomposition with control (DMDc). Understanding the dynamic characteristics of complex systems that have both internal dynamics and applied external control is fundamental to controller design and sensor placement. The DMDc method helps discover the underlying dynamics without the confounding effect of external control. In addition, the method also quantifies the effect of control inputs on the state of the system. Fig. 1 illustrates the data collection, the algorithm, and applications of DMDc.

The underlying dynamical system and measured data matrices are redefined to include systems with control inputs in § III-A. The subsequent section § III-B describes how to solve for the dynamic modes if the effect of the inputs on the state is already well-known or well-estimated. The last section § III-C shows how to solve for both the dynamic modes and the input matrix.

## A. Dynamical system with control

The new method modifies the basic assumption of DMD. The linear dynamical system connecting the future state  $\mathbf{x}_{k+1}$ 

now relies on information from both the current state  $\mathbf{x}_k$  and the current control  $\mathbf{u}_k$  given by the following:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k,\tag{14}$$

where  $\mathbf{x}_j \in \mathbb{R}^n$ ,  $\mathbf{u}_j \in \mathbb{R}^l$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{B} \in \mathbb{R}^{n \times l}$ . Data matrices can be constructed with temporal snapshots of the state and control input over time. The state snapshots  $\mathbf{X}$  and  $\mathbf{X}'$  are collected in the same manner as Eq. (2). We denote a new sequence of control input snapshots collected by the following description:

$$\Upsilon = \begin{bmatrix} & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}. \tag{15}$$

Eq. (14) can be rewritten to include the new data matrices:

$$\mathbf{X}' = \mathbf{A}\mathbf{X} + \mathbf{B}\mathbf{\Upsilon}.\tag{16}$$

Utilizing the three data matrices, approximations of the linear mappings **A** and **B** can be found. In the following two sections, we describe how to find the dynamic modes of **A** given the inclusion of control snapshots. The first section outlines the analysis and algorithm if the matrix **B** is known or well estimated. If unknown, the second section describes how to discover both **A** and **B** from the observation matrices.

### B. The map $\mathbf{B}$ is known

The following section describes how to find the dynamic modes and eigenvalues of the underlying system **A** when the matrix **B** is known. The assumption that **B** is known or well-estimated is an idealistic view of most complex systems, but it helps provide one of the major motivations for this work. Finding the underlying dynamics **A** in a complex system where control has been applied is essential for designing controllers and placement of sensors. If external control has been applied to the system, standard DMD would produce incorrect dynamic information. The more general case where **B** is unknown will be described in the following section.

Eq. (16) can be re-arranged by pairing the time-shifted state snapshot matrix with the control snapshot matrix and the known matrix **B**.

$$\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon} = \mathbf{A}\mathbf{X} \tag{17}$$

The mapping A can be solved for similar to Eq. (4). Again, the truncated singular value decomposition of X gives the matrix factorization  $\tilde{U}\tilde{\Sigma}\tilde{V}^*$ . Thus, the approximation of A is given by the following description:

$$\mathbf{A} \approx \bar{\mathbf{A}} = (\mathbf{X}' - \mathbf{B}\Upsilon)\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*. \tag{18}$$

Note, if the control snapshots are  $\mathbf{u}_j = \mathbf{0}$ ,  $\forall j \in [1, m]$ , then the derivation is equivalent to DMD. A dynamic model of both the computed process and the given input matrix can be constructed described by the following:

$$\mathbf{x}_{k+1} = \bar{\mathbf{A}}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \tag{19}$$

where  $\mathbf{x}$ ,  $\bar{\mathbf{A}}$ , and  $\mathbf{B}$  are the same dimensions of the matrices described earlier in Eq. (1). If  $r \ll n$  though, a more compact and computationally efficient model can be found using the

same basis transformation  $\mathbf{P}\mathbf{x} = \tilde{\mathbf{x}}$  as described earlier for DMD. Again, a convenient transformation has already been computed via the SVD of  $\mathbf{X}$ , given by  $\mathbf{P} = \tilde{\mathbf{U}}$ . The reduced-order model can be derived as follows:

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{U}}^* \bar{\mathbf{A}} \tilde{\mathbf{U}} \tilde{\mathbf{x}}_k + \tilde{\mathbf{U}}^* \mathbf{B} \mathbf{u}_k, \tag{20}$$

$$= \tilde{\mathbf{U}}^* (\mathbf{X}' - \mathbf{B} \Upsilon) \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{x}}_k + \tilde{\mathbf{U}}^* \mathbf{B} \mathbf{u}_k$$
 (21)

$$= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k \tag{22}$$

The reduced-order approximation of A is given by the following:

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^* (\mathbf{X}' - \mathbf{B} \Upsilon) \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1}$$
 (23)

The eigendecomposition of  $\tilde{\mathbf{A}}$  defined by  $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$  yields eigenvectors that can be used to find the dynamic modes. Similar to Exact DMD, the dynamic modes can be found with the following description:

$$\phi = (\mathbf{X}' - \mathbf{B}\Upsilon)\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\mathbf{w}$$
 (24)

If  $\lambda \neq 0$ , then this is the DMD mode for  $\lambda$ . If the eigenvalue is 0, then the dynamic mode is computed using  $\phi = \tilde{\mathbf{U}}\mathbf{w}$ .

### C. The map B is unknown

The assumption that  ${\bf B}$  is known indicates a significant amount of knowledge about how control affects the system. This section relaxes that assumption and notably demonstrates that approximations of the matrices  ${\bf A}$  and  ${\bf B}$  can both be found from state and control snapshots. To the experimentalist or analyst, this is by far more interesting since only the snapshots of the control and state are required to find the properties of the underlying process  ${\bf A}$  and how that process is affected by control  ${\bf B}$ .

The dynamical system from Eq. (16) can be manipulated giving the following representation:

$$\mathbf{X}' = [\mathbf{A} \ \mathbf{B}] \begin{bmatrix} \mathbf{X} \\ \mathbf{\Upsilon} \end{bmatrix} = \mathbf{G}\mathbf{\Omega},$$
 (25)

where  $\Omega$  contains both the state and control snapshot information. Here, we again seek a best-fit solution of the operator G which now contains the process dynamics A and input matrix B. A SVD is performed on the augmented data matrix giving  $\Omega = U\Sigma V^* \approx \tilde{U}\tilde{\Sigma}\tilde{V}^*$ . The truncation value of the SVD for  $\Omega$  will be defined as p. Note, the truncation value of  $\Omega$  should be larger than of  $\Omega$ . The following computation provides an approximation of G:

$$\mathbf{G} \approx \bar{\mathbf{G}} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}^*, \tag{26}$$

where  $\mathbf{G} \in \mathbb{R}^{n \times (n+l)}$ . We can now find approximations of the matrices  $\mathbf{A}$  and  $\mathbf{B}$  by breaking the linear operator  $\tilde{\mathbf{U}}$  in to two separate components given by the following:

$$[\mathbf{A}, \ \mathbf{B}] \approx [\bar{\mathbf{A}}, \ \bar{\mathbf{B}}]$$
 (27)

$$\approx [\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_{1}^{*}, \ \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_{2}^{*}]$$
 (28)

where  $\tilde{\mathbf{U}}_1 \in \mathbb{R}^{n \times p}$ ,  $\tilde{\mathbf{U}}_2 \in \mathbb{R}^{l \times p}$ , and  $\tilde{\mathbf{U}} = [\tilde{\mathbf{U}}_1^* \ \tilde{\mathbf{U}}_2^*]^T$ . Similar to Eq. (28), a dynamic model using the matrices  $\bar{\mathbf{A}}$  and  $\bar{\mathbf{B}}$ , but for a large dimensional system where  $n \gg 1$ , this is computationally prohibitive. Here, we again seek a reduced

order model of rank  $r \ll n$  where a transformation is required such that  $\mathbf{x} = \mathbf{P}\tilde{\mathbf{x}}$  and  $\tilde{\mathbf{x}} \in \mathbb{R}^r$ .

Unlike DMD, the truncated left singular vectors  $\tilde{\mathbf{U}}$  can not be used to define the subspace on which the state evolves. For Eq. (28), the truncated left singular vectors of  $\Omega$  define the **input** space. To find a linear transformation  $\mathbf{P}$  for the state  $\mathbf{x}$ , we utilize a reduced-order subspace of the **output** subspace. This fundamental observation allows for DMDc to discover a reduced-order representation of the dynamics  $\mathbf{A}$  and input matrix  $\mathbf{B}$ .

To find the reduced-order subspace of the output space, a second singular value decomposition is required. The data matrix of the **output** space  $\mathbf{X}'$  can be approximated by the familiar SVD:  $\hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$  where the truncation value is r and  $\hat{\mathbf{U}} \in \mathbb{R}^{n \times r}$ ,  $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{r \times r}$ , and  $\hat{\mathbf{V}}^* \in \mathbb{R}^{r \times m-1}$ . Note, the two SVDs will likely have different truncation values of the input and output matrices p and r and p > r. Using the transformation  $\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{x}}$ , the following reduced-order approximations of  $\mathbf{A}$  and  $\mathbf{B}$  can be computed:

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}^* \bar{\mathbf{A}} \hat{\mathbf{U}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_1^* \hat{\mathbf{U}}$$
(29)

$$\tilde{\mathbf{B}} = \hat{\mathbf{U}}^* \bar{\mathbf{B}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_2^*$$
 (30)

where  $\tilde{\mathbf{A}} \in \mathbb{R}^{r \times r}$  and  $\tilde{\mathbf{B}} \in \mathbb{R}^{r \times l}$ . We can then form the reduced order equation as Eq. (22) given by the following

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k \tag{31}$$

Similar to DMD, the dynamic modes of  $\mathbf{A}$  can be found by first solving the eigenvalue decomposition  $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$ . The transformation from eigenvectors to dynamic modes of  $\mathbf{A}$  is slightly modified and is given by the following:

$$\phi = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_{1}^{*} \hat{\mathbf{U}} \mathbf{w}. \tag{32}$$

where the relationship between  $\phi$  and  $\mathbf{w}$  is similar to Exact DMD.

#### D. The algorithm

The following section outlines the algorithm.

- 1) Collect and construct the snapshot matrices:

  Collect the state and control snapshots and form the matrices X, X', and  $\Upsilon$  as described in Eq. (2) and Eq. (15). Stack the data matrices X and  $\Upsilon$  to construct the matrix  $\Omega$ .
- 2) Compute the SVD of the input space Ω. Compute the singular value decomposition of Ω as described in Eq. (6) thereby obtaining the decomposition Ω ≈ ŨΣŨ\* with truncation value p.
- 3) Compute the SVD of the output space X'. Compute the singular value decomposition of X' as described in Eq. (6) thereby obtaining the decomposition  $X' \approx \hat{U} \hat{\Sigma} \hat{V}^*$  with truncation value r.
- 4) Compute the approximation of the operators  $\mathbf{G} = [\mathbf{A} \ \mathbf{B}]$

Compute the following:

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_1^* \hat{\mathbf{U}}$$
 (33)

$$\tilde{\mathbf{B}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_2^* \tag{34}$$

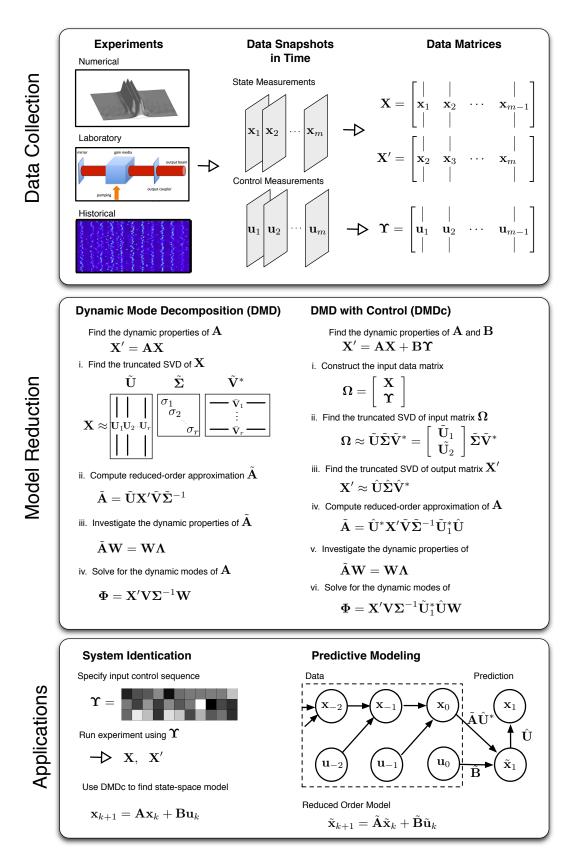


Fig. 1: The illustration outlines the three major components of applying DMDc. The top panel describes the collection of data from either a numerical, laboratory, or historical data and the curation of the data in to matrices for the methods. Note, the figure in the historical plot is the data representing pre-vaccination Measles cases in the UK normalized similar to that found in [23]. The middle panel outlines the procedure for DMD and DMDc for comparison. The bottom panel illustrates two practical applications of DMDc.

# 5) Perform the eigenvalue decomposition of $\tilde{\mathbf{A}}$ Perform the eigenvalue decomposition given by the following:

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda} \tag{35}$$

# 6) Compute the dynamic Modes of the operator A

$$\mathbf{\Phi} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_1^* \hat{\mathbf{U}} \mathbf{W}$$
 (36)

## IV. APPLICATIONS

The following section describes a number of numerical examples for the application of this method. The examples increase in complexity as the section progresses. The emphasis for each of these examples is the benefit of including control snapshot information to the analysis.

## A. Example 1 - Unstable linear system with proportional controller

DMDc can help discover the underlying dynamics of a system through measurements of both the state and external inputs. Here, we demonstrate the idea on a simple twodimensional unstable linear system with a stabilizing controller. Despite the simplicity of the mathematical problem, the example is illustrative for the general concept of DMDc. Consider the following dynamical system:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k \quad (37)$$

where  $u_k = K[x_1]_k$  and K = -1. The proportional controller clearly stabilizes the system by moving the unstable eigenvalue within the unit circle. If we have access to the input data and the B matrix as described in §III-C, we can collect state and control snapshots to perform the DMDc computation. For an initial condition  $\begin{bmatrix} 4 \\ 7 \end{bmatrix}^T$ , the following are the data matrices constructed from computing the first five temporal snapshots of Eq. (37):

$$\mathbf{X} = \begin{bmatrix} 4 & 2 & 1 & 0.5 \\ 7 & 0.7 & 0.07 & 0.007 \end{bmatrix}$$
 (38)

$$\mathbf{X} = \begin{bmatrix} 4 & 2 & 1 & 0.5 \\ 7 & 0.7 & 0.07 & 0.007 \end{bmatrix}$$

$$\mathbf{X}' = \begin{bmatrix} 2 & 1 & 0.5 & 0.25 \\ 0.7 & 0.07 & 0.007 & 0.0007 \end{bmatrix}$$
(38)

$$\Upsilon = \begin{bmatrix} -4 & -2 & -1 & -0.5 \end{bmatrix} \tag{40}$$

Following the description in §III-B, we compute the singular value decomposition of X. Here, we use MATLAB's economy sized singular value decomposition algorithm to give the following matrix factorization of X.

$$\tilde{\mathbf{U}} = \begin{bmatrix} -0.5239 & -0.8462 \\ -0.8462 & 0.5329 \end{bmatrix} \tag{41}$$

$$\tilde{\mathbf{\Sigma}} = \begin{bmatrix} 8.2495 & 0\\ 0 & 1.6402 \end{bmatrix} \tag{42}$$

$$\tilde{\mathbf{U}} = \begin{bmatrix}
-0.5239 & -0.8462 \\
-0.8462 & 0.5329
\end{bmatrix}$$

$$\tilde{\mathbf{\Sigma}} = \begin{bmatrix}
8.2495 & 0 \\
0 & 1.6402
\end{bmatrix}$$

$$\tilde{\mathbf{V}} = \begin{bmatrix}
-0.9764 & 0.2105 \\
-0.2010 & -0.8044 \\
-0.0718 & -0.4932 \\
-0.0330 & -0.2557
\end{bmatrix}$$
(41)

Now, we can compute Eq. (18) using the data matrices in Eq. (40), the SVD matrices in Eq. (43), and the matrix B in Eq. (37) giving the following approximation to **A**:

$$\bar{\mathbf{A}} = \begin{bmatrix} 1.5 & 0\\ 0 & 0.1 \end{bmatrix} \tag{44}$$

where we recover the unstable linear dynamics from data of the state and control snapshots. This example demonstrates the utility of DMDc with recovering unstable dynamics from a system that would otherwise appear to be stable.

## B. Example 2 - Large-scale, stable Linear Systems

In this section, we investigate stable linear systems where the number of measurements are significantly greater than the dimensionality of the underlying system. The previous example demonstrated the utility of the method on a lowdimensional unstable model. Here, the method is applied to large-scale dynamical systems that have an underlying lowdimensional attractor.

To construct these large-scale systems, a low-dimensional stable model is generated and subsequently embedded in to a higher dimensional subspace. There are three steps for generating the model and data matrices to compare the output of DMDc and the generated model:

# 1) Generate a low-dimensional stable state-space model, A and B

Generate discrete random state-space systems using MATLAB's command Discrete Random State Space Method. These stable-discrete state-space models can be used as numerical experiments for DMDc. Here, we have chosen a 5 dimensional model, 2 input variables, and 100 measurement variables. The output is a state space model A. B. and C.

## 2) Generate random input data Y

Using MATLAB's randn command, generate a matrix of random inputs,  $\Upsilon \in \mathbb{R}^{2 \times m-1}$ .

# 3) Use the model and input vector to generate the data matrices X and X'

Using the model and the input matrix, generate output data for the snapshot matrix.

Using the data matrices X, X' and  $\Upsilon$ , the DMDc computation can be performed to find an approximation of  $\tilde{\mathbf{A}}$  and B. To compare the generated model and the model produced by DMDc, we assign C = U. The assignment allows for the comparison of state-space models.

The singular values of the frequency response, a multi-input multi-output (MIMO) generalization of a BODE plot, is used to compare the two models. The MATLAB command sigma will generate the frequency response for both systems. Fig. 2 illustrates one such comparison arising from a single numerical realization from the ensemble. Note, there is no distinction between the generated model (in red) and the model from DMDc (in blue) for both control inputs (both lines).

## C. Example 3 – A sparse linear system in the Fourier domain

The final example for DMDc is a large-scale dynamical system on a spatial grid. The system consists of high-dimensional

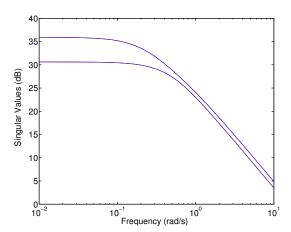


Fig. 2: The singular values of the frequency response for a large scale, stable linear systems. The blue line is from the model from DMDc and the red is from the real model. Note, an equivalent frequency response can not be constructed from DMD alone since it does not consider input-output systems.

full-state measurements, although the dynamics are governed by a low-dimensional dynamical system in the Fourier domain. The motivation for this example comes from epidemiology and infectious disease spread where the measurements can be high-dimensional in both space and time. For example, consider the number possible states of a dynamical system to represent flu infections across the world over a decade, including both spatial discretization and disease heterogeneity factors. In this example, the underlying attractor could be quite low dimensional. To complicate this picture, actuation in the form of a spatial delivery of vaccinations is also occurring each year, which can directly affect the dynamics of an infectious disease.

Here, we construct a sparse dynamical system in a twodimensional Fourier domain as an abstraction of the problem described above. Only 5 modes are allowed to be non-zero. The dynamical system on these spatial modes is constructed in the following way: for each mode, a temporal oscillation frequency is chosen randomly and a small, stable damping rate is similarly chosen. The boundary conditions are periodic, thus restricting the dynamics to a torus. This system was previously constructed in [5] to demonstrate compressive DMD. Here though, the example is extended to allow for actuation in the spatial domain. The spatial actuation is then Fourier transformed in order to compute the effect on the underlying dynamical system. The spatial grid used is  $128 \times 128$ .

Similar to the previous examples, the underlying dynamics of the system can be discovered soley from state and control snapshots in the spatial domain using DMDc. The top left plot of Fig. 3 shows the evolution of one such unforced system in space. The right plot shows the effect of actuation on the same system. The actuation is a localized negative control input applied in the spatial domain, shown in the lower left plot. The eigenvalue plot shows that DMDc discovers the underlying eigenvalues more accurately than DMD. In addition, the zero-

valued Fourier modes can be contaminated with Gaussian noise without a qualitatively change in the behavior of DMDc.

#### V. CONNECTIONS TO SYSTEM IDENTIFICATION METHODS

This section explores the connection of DMDc to two system identification methods: the Eigensystem Realization Algorithm (ERA) and the Observer Kalman Filter Identification (OKID). These system identification methods were developed to derive a state-space model for control in aerospace applications involving flexible structures [20], [21]. The identification process involves applying control and observing system behavior. DMDc and other modal decomposition methods can be used similarly, but may also be applied to historical data records from many other fields such as epidemiological modeling. Here, we briefly describe the similarities and differences between the methods.

System identification methods such as ERA/OKID were developed for input-output systems which typically have a higher rank/dimensionality than the number of observables r>n [16], [20]. To contrast, modal decomposition methods such as DMD, DMDc, Proper Orthogonal Decomposition (POD), and Balanced Proper Orthogonal Decomposition (BPOD) are typically applied to complex systems where the number of measurements are significantly larger than the rank of the underlying attractor  $n\gg r$ , e.g. fluid dynamics problems. Fig. 4 illustrates the regime of applications where each of these methods are typically applied. In addition, DMD and POD have been previously established as analysis methods for *nonlinear* complex systems [24], [30], [27]

Previous work by Tu et al. [38] has established a number of connections between DMD and ERA. The similarities and differences between DMDc and ERA listed in this section, though, are more readily compared since both algorithms assume input-output systems. The following list offers a brief comparison between how DMDc and ERA differ for the construction of a typical input-output model:

#### • The data matrix construction:

The data from the ERA procedure is fundamentally impulse-response data, whereas DMDc can have arbitrary input histories. The input histories are fundamental to the DMDc procedure. Despite this difference, the construction of the state data matrices are similar between DMDc and ERA. Both the matrix **X** from DMDc and the Hankel matrix **H** of ERA assume snapshots of the state at regular intervals. The data matrix **H** is also vertically stacked with time shifted versions of the snapshot. DMDc does not require shift-stacking the matrix since there is little risk of column-rank deficiency due to the typically large number of observables. The two state matrices are equivalent on the condition that the **H** is not vertically stacked with state snapshots [38].

#### • The A matrix:

It was previously shown that if the data matrices described in the first bullet are the same, the matrices A produced by DMD and ERA are equivalent up to a similarity transformation [38]. The A matrix constructed by DMDc is different from DMD and ERA since the

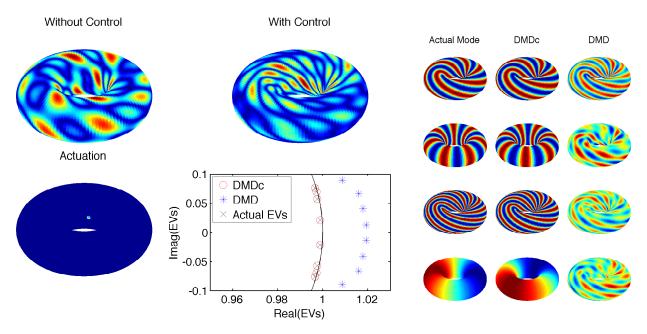


Fig. 3: The top left panel illustrates one realization of Example 3 without actuation over time. The right panel illustrates the same dynamical system, but with actuation. The bottom left panel illustrates the actuation applied in the spatial domain. The bottom middle panel shows a comparison between the actual eigenvalues and the eigenvalues found from DMD and DMDc. On the right the first four dynamic modes of DMD and DMDc are compared to the actual underlying spatial modes.

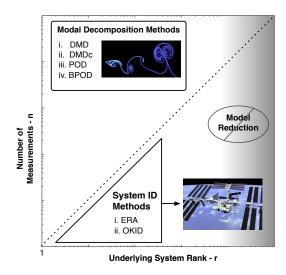


Fig. 4: An illustration depicting the different regimes, with respect to the rank of the system and the number of measurements, of the modal decomposition methods and the system identification methods.

 $\it input$  space and thus the subspace  $\tilde{U_1^*}$  contains added information from the control snapshots.

## • The B matrix

To compute the matrix  ${\bf B}$  using ERA, only the first data snapshot after the impulse is utilized, which translates to the following discrete dynamical system relationship  ${\bf x_1} = {\bf CBu_0}$  where  ${\bf u_0}$  is an impulse and  ${\bf C}$  is the

standard linear map for the observable equation. Note, ERA also requires a projection of the single data snapshot on to the left singular vectors of the data matrix  $\mathbf{H}$  and the same similarity transform  $\mathbf{\Sigma}^{1/2}$  described for the matrix  $\mathbf{A}$ . The ERA formulation can be contrasted with DMDc through the illustration of the difference in the data matrix construction for DMDc given by the following:

The computation to find the matrix  ${\bf B}$  is quite different for DMDc. Arbitrary control histories can be included in  $\Omega$  to compute  ${\bf B}$  whereas ERA is primarily impulse response focused. Further, finding the matrix  ${\bf B}$  with ERA will not be as robust to noise compared with using DMDc and a longer input history.

• The C matrix For DMDc, ERA, and DMD, a linear transformation matrix maps the model state to the observables. Each of these methods utilize the left singular vectors of their data matrices for the mapping. There is an important distinction between the role of the left singular vectors for DMDc and ERA. The mapping for DMDc projects a high-dimensional set of observables on to a lower-dimensional subspace. In ERA, the left singular vectors often lifts the dimension of the observables, see Fig. 4 for an illustration of the rank of the model versus dimension of the observables.

The observer/Kalman filter identification method allows minimal realization algorithms such as ERA to be generalized from impulse response data to data that is driven by rich input signals [21]. The calculation of the above matrices A, B, and C is typically considered more robust when combining OKID with ERA. An often cited computational challenge confronting ERA is the analysis of lightly damped systems. The magnitude of data (number of snapshots) may be prohibitively large for lightly damped systems, and factoring the Hankel matrix using the SVD is computationally prohibitive. A major similarity between OKID and DMDc is the construction of the data matrix; OKID constructs an augmented data matrix that also stacks the control with the state. Similar to DMD and ERA, in the limit of only evaluating the first row of the augmented Hankel matrix, the data matrices between DMDc and OKID are equivalent.

#### VI. DISCUSSION

Complex, high-dimensional data has become ubiquitous in traditional scientific and engineering applications as well as modern data-rich fields such as internet traffic, distribution systems, and transportation networks. Machine-learning and statistical methods have been successfully applied to characterize many of these so-called *big-data* problems. Similarly, scientific and engineering fields, exemplified by control theoretic community, have focused on the development of quantitative and automatic dimensionality reduction methods to both characterize and *control* complex systems. In order to construct effective controllers, the underlying system needs to be well-understood. Accurately describing the underlying system is a challenge when the system is complex, high-dimensional, and without well-characterized governing equations.

Dynamic Mode Decomposition (DMD) is a data-driven, equation-free method that helps meet a number of these modern-day challenges. The method has strong connection to nonlinear operator theory and discovers spatial-temporal coherent modes from data. DMD, though, does not produce accurate reduced-order-models from complex systems with exogenous forcing. Dynamic mode decomposition with control (DMDc) inherits the advantages of DMD, but also provides accurate input-output models for complex systems with actuation. The method can be applied to data from a variety of sources including historical, experimental, and black-box simulations.

Methods such as DMDc will play an increasing role in the analysis of large-scale datasets from complex systems. DMD has already been applied to a significant number of applications in the fluid dynamics community [33], [35], [14], [2], [39], [37] and is expanding to a variety of other applications like background subtraction in video processing [15]. We believe DMDc is poised to similarly excel as a tool for a diverse set of engineering applied science applications where control of the complex system is important. Further, the DMDc method is well-suited to couple with innovative sparsity-promoting sampling and control strategies [7], [32], [3], [12]. This connection has already been demonstrated for DMD both in time and space [5], [19], [39]. DMDc is therefore positioned to have a dramatic effect on the analysis and control of large-scale complex systems.

#### ACKNOWLEDGEMENTS

The authors would like to thank Bill and Melinda Gates for their active support of the Institute for Disease Modeling and their sponsorship through the Global Good Fund. J. N. Kutz acknowledges support from the U.S. Air Force Office of Scientific Research (FA9550-09-0174). Productive discussions about Dynamic Mode Decomposition with Bing Brunton, Clancy Rowley, and Jonathan Tu are likewise greatly appreciated.

## REFERENCES

- [1] R. Anderson, R. May, and B. Anderson. *Infectious Diseases of Humans: Dynamics and Control.* Oxford University Press, 1992.
- [2] S. Bagheri. Koopman-mode decomposition of the cylinder wake. Journal of Fluid Mechanics, 726:596–623, 2013.
- [3] Z. Bai, T. Wimalajeewa, Z. Berger, G. Wang, M. Glauser, and P. K. Varshney. Physics based compressive sensing approach applied to airfoil data collection and analysis. AIAA Paper 2013-0772, 51st Aerospace Sciences Meeting, January 2013.
- [4] R. G. Baraniuk. Compressive sensing. IEEE Signal Processing Magazine, 24(4):118–120, 2007.
- [5] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Compressive sampling dynamic mode decomposition. submitted for publication, arXiv, 2013.
- [6] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism a). Chaos: An Interdisciplinary Journal of Nonlinear Science, 22(4):047510, 2012.
- [7] E. J. Candès. Compressive sensing. Proceedings of the International Congress of Mathematics, 2006.
- [8] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.
- [9] D.L. Donoho and M. Gavish. Minimax risk of matrix denoising by singular value thresholding. *arXiv*, (1304:2085v3), 2013.
- [10] D.L. Donoho and M. Gavish. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . arXiv, (1305:5870v2), 2013.
- [11] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [12] Makan Fardad, Fu Lin, and M Jovanovic. Design of optimal sparse interconnection graphs for synchronization of oscillator networks. *IEEE Transactions on Automatic Control*, 29(5):2281–2291, September 2014.
- [13] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerical Mathematics*, 14:403–420, 1970.
- [14] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams. Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *Journal of Fluid Mechanics*, 700:16–28, 2012.
- [15] J. Grosek and J. N. Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. arXiv, 2014.
- [16] B. L. Ho and R. E. Kalman. Effective construction of linear statevariable models from input/output data. In *Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory*, pages 449–459, 1965.
- [17] P. J. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge Monographs in Mechanics. Cambridge University Press, Cambridge, England, 2nd edition, 2012.
- [18] M. Ilak and C.W. Rowley. Modeling of transitional channel flow using balanced proper orthogonal decomposition. *Physics of Fluids*, 20(034103), 2008.
- [19] Mihailo R Jovanović, Peter J Schmid, and Joseph W Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids* (1994-present), 26(2):024103, 2014.
- [20] J. N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.
- [21] J. N. Juang, M. Phan, L. G. Horta, and R. W. Longman. Identification of observer/kalman filter markov parameters: Theory and experiments. *Technical Memorandum 104069 NASA*, 1991.
- [22] M. Keeling and P. Rohani. Modeling Infectious Diseases in Humans and Animals. Princeton University Press, 2008.
- [23] M. J. Keeling and B. T. Grenfell. Disease extinction and community size: Modeling the persistence of measles. *Science*, 275(5296):65–67, 1997

- [24] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. Proceedings of the National Academy of Sciences, 17(5):315– 318, 1931.
- [25] Fu Lin, Makan Fardad, and Mihailo R Jovanovic. Design of optimal sparse feedback gains via the alternating direction method of multipliers. Automatic Control, IEEE Transactions on, 58(9):2426–2431, 2013.
- [26] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. J. Nonlin. Dyn., 41:309–325, 2005.
- [27] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. Annual Review of Fluid Mechanics, 45:357–378, 2013.
- [28] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. Quart. J. Math. Oxford, (11):50–59, 1960.
- [29] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26(1), 1985.
- [30] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.
- [31] C.W. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Internation Journal of Bifurcation and Chaos*, 15(3):997–1013, 2005.
- [32] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy* of Sciences USA, 110(17):6634–6639, 2013.
- [33] P. J. Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in Fluids*, 50:1123–1130, 2011.
- [34] P. J. Schmid, K. E. Meyer, and O. Pust. Dynamic mode decomposition and proper orthogonal decomposition of flow in a lid-driven cylindrical cavity. PIV09-0186, 8th International Symposium on Particle Image Velocimetry, August 2009.
- [35] P. J. Schmid, D. Violato, and F. Scarano. Decomposition of timeresolved tomographic PIV. Experiments in Fluids, 52:1567–1579, 2012.
- [36] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I-III. Q. Appl. Math., XLV(3):561–590, 1987.
- [37] Gilles Tissot, Laurent Cordier, Nicolas Benard, and Bernd R Noack. Model reduction using dynamic mode decomposition. *Comptes Rendus Mécanique*, 2014.
- [38] J. H. Tu, D. M. Luchtenburg, C. W. Rowley, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. submitted for publication, arXiv, 2013.
- [39] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang. Spectral analysis of fluid flows using sub-Nyquist rate PIV data. to appear in Experiments in Fluids, 2014.
- [40] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. AIAA Journal, 40(11):2323–2330, 2002.



**Joshua L. Proctor** received the B.S. degrees in aeronautics and astronautics, the B.A. degree in english literature from the University of Washington, Seattle, WA, in 2006, and the Ph.D. degree in mechanical and aerospace engineering from Princeton University,

Princeton, NJ, in 2011. He is currently an Associate Principal Investigator at the Institute for Disease Modeling.



Steven L. Brunton received the B.S. degree in mathematics with a minor in control and dynamical systems from the California Institute of Technology, Pasadena, CA, in 2006, and the Ph.D. degree in mechanical and aerospace engineering from Princeton University, Princeton, NJ, in 2012. He is currently an Acting Assistant Professor of applied

mathematics at the University of Washington.



**J. Nathan Kutz** received the B.S. degrees in physics and mathematics from the University of Washington, Seattle, WA, in 1990, and the Ph.D. degree in applied mathematics from Northwestern University, Evanston, IL, in 1994. He is currently a Professor and Chair of applied mathematics at the University of Washington.