Chapter 6
DMD with Control

The modeling of complex, high-dimensional systems that exhibit dynamics and require control is permeating not only the traditional engineering and physical sciences, but also modern applications such as eradication efforts of infectious diseases, distribution systems, and the electrical grid. DMD with control (DMDc) utilizes both the measurements of the system and the applied external control to extract the underlying dynamics and the input-output characteristics in an *equation-free* manner [222]. DMDc inherits the advantageous characteristics of DMD: it operates solely on snapshot data, efficiently handles high-dimensional measurement data, and connects measurement data to the analysis of nonlinear dynamical systems via Koopman operator theory. DMDc can also extract input-output characteristics, allowing for the construction of a set of reduced order models (ROMs) that can be used to predict and design controllers for high-dimensional, complex systems.

6.1 ▪ Formulating DMDc

The DMDc method was originally developed to analyze measurement data collected from complex systems where exogenous forcing has been applied during the observation period. The goal of eradicating infectious diseases, such as poliomyelitis, offers a motivating example for the development of DMDc; this is discussed in Chapter 11. As cases of paralyzed children are observed, large numbers of children are also concurrently being vaccinated to fight the spread of this terrible disease. DMDc utilizes both the measurements of the state and the exogenous input to disambiguate the underlying dynamics and the impact of the forcing. Characterizing the underlying dynamics and the input-output properties is critical to the development of predictive models and control strategies to drive the system to a particular state. In this section, we describe the basic formulation of DMDc.

6.1.1 ▪ Data and a dynamical system with inputs

The goal of DMDc is to characterize the relationship between three measurements: the current measurement $\mathbf{x}_k$, the future state $\mathbf{x}_{k+1}$, *and* the current control $\mathbf{u}_k$. The relationship can be approximated by the canonical discrete linear dynamical system

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{6.1}$$

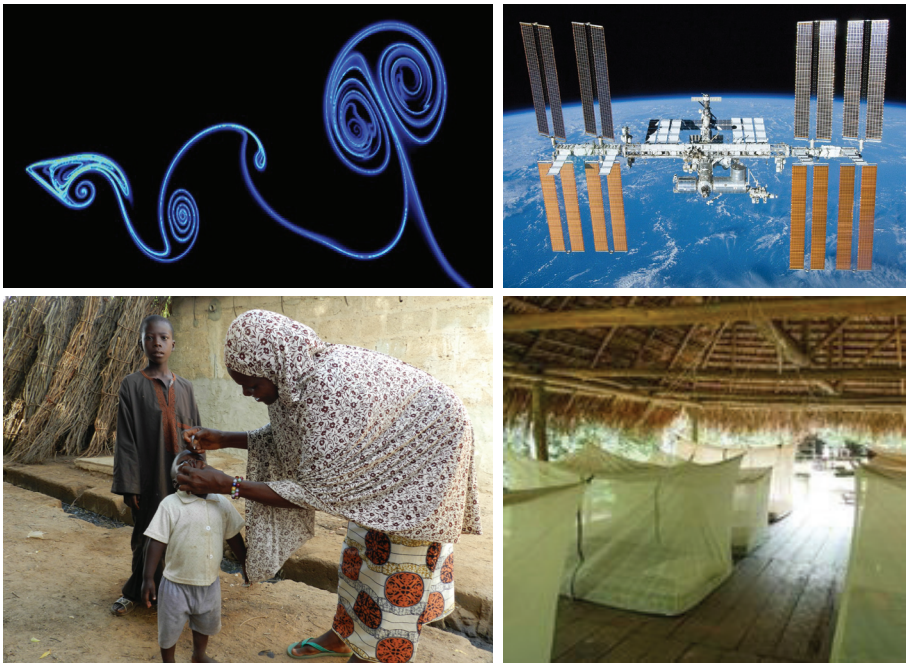

Figure 6.1. *The top left panel shows a pitching airfoil. The top right panel shows the International Space Station. The bottom left image shows a child being vaccinated against polio. The bottom right image shows bednets encompassing a series of beds in a hospital. Each of these complex systems have both state dynamics and the possibility of actuation. For the pitching airfoil, DMDc is able to handle the high dimensionality of the system and discover the input-output characteristics. In the spread of infectious disease, vaccinations and bednets are a key actuation tool for arresting transmission of disease. Top left panel from* [222]*, top right panel reprinted courtesy of NASA, bottom panels reprinted with permission from the Institute for Disease Modeling.*

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^q$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, and $\mathbf{B} \in \mathbb{R}^{n\times q}$. The operator $\mathbf{A}$ describes the dynamics of the unforced system. The operator $\mathbf{B}$ characterizes the impact of the input $\mathbf{u}_k$ on the state $\mathbf{x}_{k+1}$. The relationship in (6.1) does not need to hold exactly, similar to DMD. The goal is to find approximations of $\mathbf{A}$ and $\mathbf{B}$ from measurements. The state snapshot measurement matrices $\mathbf{X}$ and $\mathbf{X}'$ are constructed in the same manner as for DMD. A new measurement snapshot matrix is constructed for the inputs

$$\Upsilon = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}. \tag{6.2}$$

(6.1) can be rewritten in matrix form to include the new data matrices

$$\mathbf{X}' \approx \mathbf{A}\mathbf{X} + \mathbf{B}\Upsilon. \tag{6.3}$$

A standard perspective from the control community is to include a measurement equation. A linear observation equation of the form $\mathbf{y} = \mathbf{C}\mathbf{x}$ is typically constructed, indicating a hidden dynamic state $\mathbf{x}$ and an output measurement $\mathbf{y}$. For a large number of DMD applications involving numerical simulations, the measurement matrix $\mathbf{C}$ is the

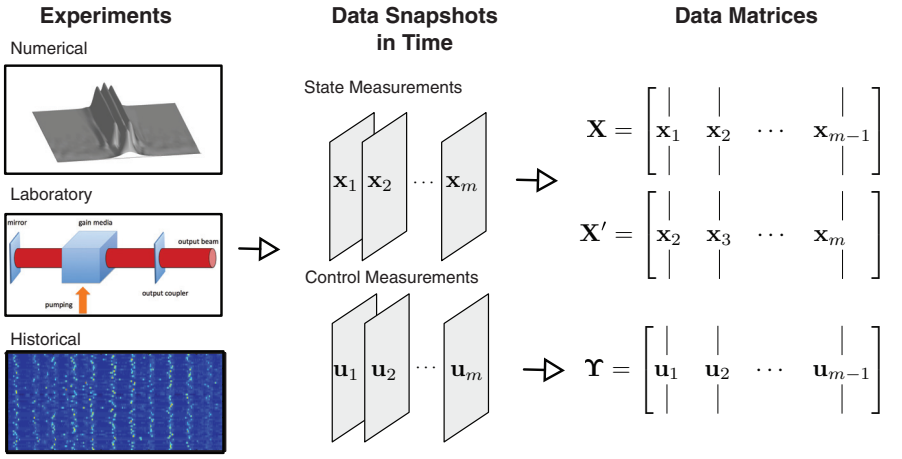

Figure 6.2. *Shaping data matrices for DMDc. Data can be collected from a number of different sources, such as numerical simulations, laboratory experiments, and historical data. For systems with exogenous forcing, measurements of the inputs are also important. The DMDc method utilizes measurements of both the state and the external forcing [modified figure from* [222]*].*

identity. The observation equation is important within the engineering perspective of system identification. For many aerospace applications considered in the 1980s and 1990s, the number of measurements was typically smaller than the number of states required to describe the input-output models [151, 152]. This is discussed in greater detail in Chapter 7. DMDc was developed for the opposite regime, where the number of state measurements is much larger than the rank of the underlying system.

Figure 6.2 illustrates the collection of state and input measurement data from numerical simulations, laboratory experiments, and historical records. DMDc utilizes the three data matrices $\mathbf{X}$, $\mathbf{X}'$, and $\mathbf{\Upsilon}$ to discover a best-fit approximation of the operators $\mathbf{A}$ and $\mathbf{B}$. The next two subsections detail how to solve for these operators. The first subsection outlines the analysis and algorithm if the matrix $\mathbf{B}$ is known or well estimated. The second subsection describes how to fit $\mathbf{A}$ and $\mathbf{B}$.

6.1.2 ▪ The operator $\mathbf{B}$ is known

Let us first consider a case where the operator $\mathbf{B}$ is known or well estimated. This idealistic assumption is not true for most complex systems without well-defined or physics-based governing equations, but it does illustrate the major motivation of this method. Without including the effects of the inputs, standard DMD does not produce the correct dynamic characteristics.

The control snapshot matrix $\mathbf{\Upsilon}$ and operator $\mathbf{B}$ can be subtracted from the time-shifted snapshot matrix $\mathbf{X}'$ in (6.3), giving

$$\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon} \approx \mathbf{A}\mathbf{X}. \tag{6.4}$$

The goal is to solve for the operator $\mathbf{A}$. The procedure is similar to the standard DMD algorithm where we first compute the truncated SVD with rank truncation r of the state snapshot matrix $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$. The best-fit operator $\mathbf{A}$ can then be computed as

$$\mathbf{A} = (\mathbf{X}' - \mathbf{B}\mathbf{\Upsilon})\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*. \tag{6.5}$$

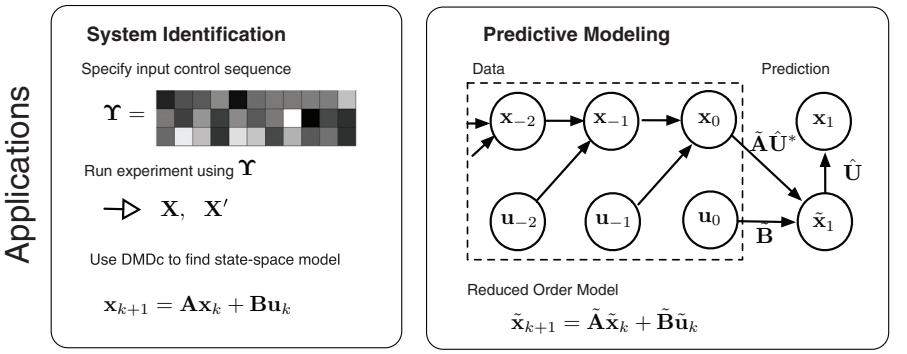

Figure 6.3. *Two possible applications of DMDc: System Identification and predictive modeling* [222].

This derivation is equivalent to DMD if the input snapshots are either $\mathbf{u}_j = 0 \; \forall \; j \in [1, m-1]$ or $\mathbf{B} = 0$. If the truncation value $r \ll n$, a more compact and computationally efficient model can be found using a basis transformation $\tilde{\mathbf{x}} = \mathbf{U}^*\mathbf{x}$ similar to DMD. The reduced-order approximation of $\mathbf{A}$ is

$$\tilde{\mathbf{A}} = \mathbf{U}^*(\mathbf{X}' - \mathbf{B}\Upsilon)\mathbf{V}\Sigma^{-1}. \tag{6.6}$$

Note that a ROM can be constructed using $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}} = \mathbf{U}^*\mathbf{B}$. The eigendecomposition of $\tilde{\mathbf{A}}$, defined by $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda$, yields eigenvectors that can be used to find the dynamic modes of $\mathbf{A}$. The exact dynamic modes can be found with the description

$$\Phi = (\mathbf{X}' - \mathbf{B}\Upsilon)\mathbf{V}\Sigma^{-1}\mathbf{W}. \tag{6.7}$$

If $\lambda \neq 0$, then (6.7) is the exact DMD mode for λ. If the eigenvalue is 0, then the dynamic mode is computed using $\Phi = \mathbf{U}\mathbf{W}$.

6.1.3 ▪ The operator B is unknown

This section describes how to find approximations of the operators $\mathbf{A}$ *and* $\mathbf{B}$ solely from snapshot data. By relaxing the assumption that $\mathbf{B}$ is known, a wider range of complex systems can be investigated with DMDc. Experimentalists and analysts can use DMDc to disambiguate the underlying dynamics and the impact of external inputs. Further, DMDc could be used to help design the inputs to explore the dynamical properties of the underlying system.

(6.3) can be reformulated as

$$\mathbf{X}' \approx [\mathbf{A} \;\; \mathbf{B}] \begin{bmatrix} \mathbf{X} \\ \Upsilon \end{bmatrix} = \mathbf{G}\Omega, \tag{6.8}$$

so that $\mathbf{G} \triangleq [\mathbf{A} \;\; \mathbf{B}]$ is the augmented operator matrix and $\Omega \triangleq \left[\begin{smallmatrix} \mathbf{X} \\ \Upsilon \end{smallmatrix}\right]$ is the augmented data matrix.

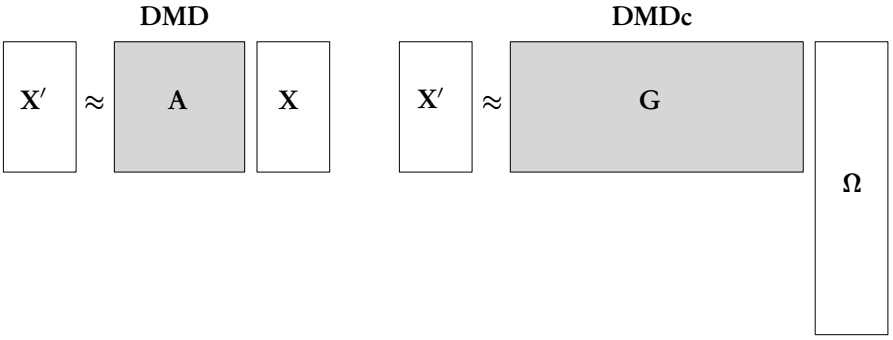

Figure 6.4. *An illustration showing the size of the matrices for DMD and DMDc. The gray boxes are the operators that DMD and DMDc discover.*

DMDc of the measurement trio $\mathbf{X}$, $\mathbf{\Upsilon}$, and $\mathbf{X}'$ is the eigendecomposition of the operator $\mathbf{A}$ defined by

$$\mathbf{G} = \mathbf{X}'\mathbf{\Omega}^{\dagger}, \tag{6.9a}$$

$$[\mathbf{A}\ \ \mathbf{B}] = \mathbf{X}'\begin{bmatrix} \mathbf{X} \\ \mathbf{\Upsilon} \end{bmatrix}^{\dagger}, \tag{6.9b}$$

where $\mathbf{\Omega}$ contains both the measurement and the control snapshot information. We seek a best-fit solution of the operator $\mathbf{G}$. Note that the size of the operator is no longer square, as in the case of $\mathbf{A}$ for DMD; see Figure 6.4 for an illustration.

To solve for $\mathbf{G}$, the SVD is performed on the augmented data matrix, giving $\mathbf{\Omega} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$. SVD is one method for finding a solution that minimizes the Frobenius norm of $||\mathbf{X}' - \mathbf{G}\mathbf{\Omega}||_F$. The truncation value of the SVD for $\mathbf{\Omega}$ is $\tilde{r}$. The truncation value of $\mathbf{\Omega}$ should be equal to or larger than for $\mathbf{X}$. An approximation of $\mathbf{G}$ can be found

$$\mathbf{G} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}^*, \tag{6.10}$$

where $\mathbf{G} \in \mathbb{R}^{n\times(n+q)}$. Approximations of the operators $\mathbf{A}$ and $\mathbf{B}$ can be found by splitting the left singular vectors $\tilde{\mathbf{U}}$ into two separate components:

$$[\mathbf{A},\ \mathbf{B}] \approx [\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_1^*,\ \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_2^*], \tag{6.11}$$

where $\tilde{\mathbf{U}}_1 \in \mathbb{R}^{n\times\tilde{r}}$, $\tilde{\mathbf{U}}_2 \in \mathbb{R}^{q\times\tilde{r}}$, and $\tilde{\mathbf{U}}^* = [\tilde{\mathbf{U}}_1^*\ \ \tilde{\mathbf{U}}_2^*]$. For high-dimensional systems where $n \gg 1$, solving for and storing the operators can be computationally infeasible. A reduced-order approximation can be solved for instead, leading to a more tractable computational model. We seek a transformation to a lower-dimensional subspace on which the dynamics evolve.

DMD utilizes the truncated singular vectors $\mathbf{U}$ to define the subspace for the reduced-order approximation. DMDc cannot use $\tilde{\mathbf{U}}$ to find the low-rank model of the dynamics and input matrix since $\tilde{\mathbf{U}}$ is defined for the *input* space, which now includes both the state measurements and the exogenous inputs. Instead of $\tilde{\mathbf{U}}$, we construct a reduced-order subspace from the *output* measurements $\mathbf{X}'$. This observation is fundamental to understanding DMDc as a method for finding a best-fit operator between input and output data.

The data matrix of the *output* space $\mathbf{X}'$ can be used to find the reduced-order subspace. A second SVD on $\mathbf{X}'$ gives the factorization $\mathbf{X}' = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$, 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 $\hat{\mathbf{V}}^* \in \mathbb{R}^{r\times m-1}$. The two SVDs will likely have different truncation values of the input and output matrices $\tilde{r}$ and r, where $\tilde{r} > r$. Using the transformation $\mathbf{x} = \hat{\mathbf{U}}\tilde{\mathbf{x}}$, the following reduced-order approximations of $\mathbf{A}$ and $\mathbf{B}$ can be computed:

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}^*\bar{\mathbf{A}}\hat{\mathbf{U}} = \hat{\mathbf{U}}^*\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_1^*\hat{\mathbf{U}}, \tag{6.12}$$

$$\tilde{\mathbf{B}} = \hat{\mathbf{U}}^*\bar{\mathbf{B}} \quad = \hat{\mathbf{U}}^*\mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_2^*, \tag{6.13}$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{r\times r}$ and $\tilde{\mathbf{B}} \in \mathbb{R}^{r\times q}$. We can then form the ROM

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{B}}\mathbf{u}_k. \tag{6.14}$$

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

$$\mathbf{\Phi} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{U}}_1^*\hat{\mathbf{U}}\mathbf{W}, \tag{6.15}$$

where the relationship between $\mathbf{\Phi}$ and $\mathbf{W}$ is similar to the description of DMD in Chapter 1.

6.2 ▪ The DMDc algorithm

The following section outlines the algorithm and provides code to compute each step.

1. **Collect and construct the snapshot matrices:**
   Collect the system measurement and control snapshots and form the matrices $\mathbf{X}$, $\mathbf{X}'$, and $\mathbf{\Upsilon}$. Stack the data matrices $\mathbf{X}$ and $\mathbf{\Upsilon}$ to construct the matrix $\mathbf{\Omega}$:

```
X    = StateData(:,1:end-1);
Xp   = StateData(:,2:end);
Ups = InputData(:,1:end-1);

Omega = [X;Ups];
```

2. **Compute the SVD of the input space $\mathbf{\Omega}$.**
   Compute the SVD of $\mathbf{\Omega}$ to obtain the decomposition $\mathbf{\Omega} \approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$ with truncation value $\tilde{r}$:

```
[U,Sig,V] = svd(Omega,'econ');

thresh = 1e-10;
rtil = length(find(diag(Sig)>thresh));

Util    = U(:,1:rtil);
Sigtil  = Sig(1:rtil,1:rtil);
Vtil    = V(:,1:rtil);
```

3. **Compute the SVD of the output space $\mathbf{X}'$.**
   Compute the SVD of $\mathbf{X}'$ to obtain the decomposition $\mathbf{X}' \approx \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^*$ with truncation value r:

```
[U,Sig,V] = svd(Xp,'econ');

thresh = 1e-10;
r = length(find(diag(Sig)>thresh));

Uhat    = U(:,1:r);
Sighat  = Sig(1:r,1:r);
Vbar    = V(:,1:r);
```

4. **Compute the approximation of the operators $\mathbf{G} = [\mathbf{A}\ \ \mathbf{B}]$.**
Compute

$$\tilde{\mathbf{A}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_1^* \hat{\mathbf{U}}, \tag{6.16a}$$

$$\tilde{\mathbf{B}} = \hat{\mathbf{U}}^* \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_2^* : \tag{6.16b}$$

```
n = size(X,1);
q = size(Ups,1);
U_1 = Util(1:n,:);
U_2 = Util(n+q:n+q,:);

approxA = Uhat'*(Xp)*Vtil*inv(Sigtil)*U_1'*Uhat;
approxB = Uhat'*(Xp)*Vtil*inv(Sigtil)*U_2';
```

5. **Perform the eigenvalue decomposition of $\tilde{\mathbf{A}}$.**
Perform the eigenvalue decomposition given by

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda} : \tag{6.17}$$

```
[W,D] = eig(approxA);
```

6. **Compute the dynamic modes of the operator $\mathbf{A}$.**

$$\mathbf{\Phi} = \mathbf{X}' \tilde{\mathbf{V}} \tilde{\mathbf{\Sigma}}^{-1} \tilde{\mathbf{U}}_1^* \hat{\mathbf{U}} \mathbf{W} : \tag{6.18}$$

```
Phi = Xp * Vtil * inv(Sigtil) * U_1'*Uhat * W;
```

6.3 ▪ Examples

We illustrate DMDc with two examples. The first example is an unstable system that has a proportional controller stabilizing the system. The second example is a high-dimensional stable dynamical system with exogenous forcing.

6.3.1 ▪ Unstable linear systems

DMDc can be utilized to discover the underlying dynamics of an unstable system that has a stabilizing controller. Without the inclusion of measurements of the inputs, DMD would produce a set of stable eigenvalues and dynamic modes. DMDc, though, can recover the unstable dynamics. Consider the unstable linear dynamical system

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k, \tag{6.19}$$

where $u_k = K[x_1]_k$ and $K = -1$. The proportional controller shifts the unstable eigenvalue, $\lambda = 1.5$, to a stable value of $\lambda = 0.5$ well within the unit circle. Recall that eigenvalues outside the unit circle in the complex plane are unstable for discrete time systems. We can produce artificial data from this model using an initial condition of $\mathbf{x}_0 = [4 \;\; 7]^T$. The first five temporal snapshots can be collected in the DMDc data matrices:

$$\mathbf{X} = \begin{bmatrix} 4 & 2 & 1 & 0.5 \\ 7 & 0.7 & 0.07 & 0.007 \end{bmatrix}, \tag{6.20a}$$

$$\mathbf{X}' = \begin{bmatrix} 2 & 1 & 0.5 & 0.25 \\ 0.7 & 0.07 & 0.007 & 0.0007 \end{bmatrix}, \tag{6.20b}$$

$$\Upsilon = \begin{bmatrix} -4 & -2 & -1 & -0.5 \end{bmatrix}. \tag{6.20c}$$

Utilizing these data matrices and a *known* input matrix $\mathbf{B}$, the description of DMDc in § 6.1.2 can be used to discover the underlying dynamics. The SVD of $\mathbf{X}$ gives the factorization

$$\mathbf{U} = \begin{bmatrix} -0.5239 & -0.8462 \\ -0.8462 & 0.5329 \end{bmatrix}, \tag{6.21a}$$

$$\Sigma = \begin{bmatrix} 8.2495 & 0 \\ 0 & 1.6402 \end{bmatrix}, \tag{6.21b}$$

$$\mathbf{V} = \begin{bmatrix} -0.9764 & 0.2105 \\ -0.2010 & -0.8044 \\ -0.0718 & -0.4932 \\ -0.0330 & -0.2557 \end{bmatrix}. \tag{6.21c}$$

The data matrices, the SVD approximation, and the matrix $\mathbf{B}$ allow us to compute an approximation to $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.1 \end{bmatrix}, \tag{6.22}$$

where we recover the unstable linear dynamics from data of the state and control snapshots. DMD would incorrectly recover an operator with stable eigenvalues, whereas DMDc can correctly disambiguate the underlying dynamics from exogenous inputs.

In order to also recover the input matrix $\mathbf{B}$, the problem has to be slightly modified. With perfect feedback control, DMDc will not be able to accurately reconstruct $\mathbf{B}$, since two of the rows of $\boldsymbol{\Omega}$ are equivalent up to a scaling factor. Adding a small random disturbance to the controller will break the equivalence. The form of the controller can be transformed to $u_k = K[x_1]_k + \delta_k$, where each δ is drawn from a Gaussian

distribution with zero mean. This allows both **A** and **B** to be reconstructed when the feedback is from a proportional controller.

***ALGORITHM* 6.1. DMDc for unstable systems with known B matrix.**

```
%Data collection

A = [1.5 0;0 0.1];
x0 = [4;7];
K = [-1];
m = 20;
DataX = x0;
DataU = [0];

B = [1;0];

for j = 1 : m

  DataX(:,j+1) = A * (DataX(:,j)) + B.*(K*DataX(:,j));
  DataU(:,j) = K .* DataX(1,j);
end

%Data matrices
X   = DataX(:,1:end-1);
Xp  = DataX(:,2:end);
Ups = DataU;

%SVD
[U,Sig,V] = svd(X,'econ');

thresh = 1e-10;
r = length(find(diag(Sig)>thresh));

U    = U(:,1:r);
Sig  = Sig(1:r,1:r);
V    = V(:,1:r);

%DMD

A_DMD  = Xp*V*inv(Sig)*U'

%DMDc - B is known

A_DMDc = (Xp - B*Ups)*V*inv(Sig)*U'
```

6.3.2 ▪ Linear dynamics in the Fourier domain with inputs

In this example, DMDc is applied to a dynamical system defined on a large-scale spatial grid. The dynamics have a low-rank representation in the two-dimensional spatial Fourier domain, where only five modes are chosen to be nonzero. A set of oscillating frequencies are associated with the modes to generate distinct spatiotemporal patterns in space and time. An external input can be applied to a spatial grid location at a spe-

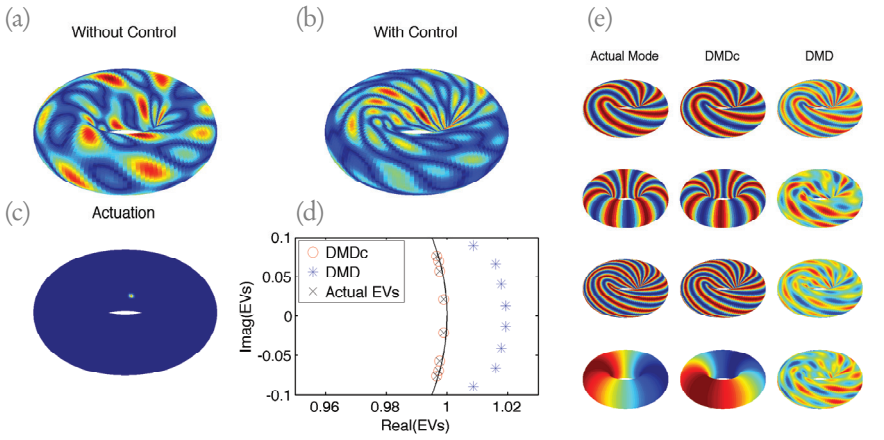

Figure 6.5. (a) *illustrates one realization of the example in* § 6.3.2 *without actuation over time.* (b) *illustrates the same dynamical system, but with actuation.* (c) *illustrates the actuation applied in the spatial domain.* (d) *shows a comparison between the actual eigenvalues and the eigenvalues found from DMD and DMDc.* (e) *shows four dynamic modes of DMD and DMDc compared to the actual underlying spatial modes* [222].

cific time. The motivation for this example comes from epidemiology and infectious disease spread, where the number of measurements can include a substantial number of physical locations; the dynamics often appear to be low dimensional; and interventions, like vaccination campaigns, can be directed at specific locations.

We construct a sparse dynamical system in the two-dimensional spatial Fourier domain. Only five modes are allowed to be nonzero. For each nonzero mode, a linear dynamical system is constructed by choosing an eigenvalue that has both an oscillating component and a small stable damping rate. The boundary conditions are periodic, restricting the dynamics to the torus. Actuation can be applied to single pixels in space and time. To investigate the impact of the actuation on the five modes, the inputs in space and time need to be inverse Fourier transformed. The spatial grid is 128×128.

The underlying dynamics of this large-scale dynamical system can be reconstructed from state and control snapshots using DMDc. Figure 6.5 shows the dynamic behavior of the five oscillating modes on the torus with and without control in the top left of the illustration. The actuation is localized to a single pixel on the torus and is shown in the lower left plot. A comparison of the output of DMD and DMDc is shown for both the eigenvalues and the dynamic modes. The dynamic modes are illustrated on the torus. DMDc is able to reconstruct the underlying dynamic modes with high fidelity in this large-scale system.

6.4 ▪ Connections to system identification methods

DMDc is a modal decomposition method that discovers coherent spatiotemporal modes from measurements of complex systems with inputs. DMD and DMDc have been previously connected to system identification methods from the control community [290, 222]. A number of important general differences exist, though, between DMDc and other system identification methods. DMD and DMDc are modal decomposition methods discovering coherent spatiotemporal patterns from high-dimensional data [247]. DMD has also been connected to Koopman spectral analysis, providing

theoretical justification for analyzing data from nonlinear systems [162, 235, 195]. These differences can be partially attributed to the originating field of study. DMD and DMDc were developed in the context of scientific fields, such as fluid dynamics, where large-scale governing equations of the nonlinear system are difficult to analyze [250, 247, 222]. System identification methods were developed in the engineering context of producing linear input-output models for the design of controllers [293, 154]. Despite these differences, a number of algorithmic similarities exist.

DMD and DMDc are intimately connected to two system identification methods: ERA and OKID, discussed in Chapter 7. These methods were originally developed for applications in the aerospace community, where the number of measurements collected was smaller than the rank of the system [151, 150]. ERA was also previously shown to be connected to a modal decomposition method called balanced proper orthogonal decomposition (BPOD), where both produce equivalent balanced input-output models [201, 233, 184]. ERA, though, can produce these balanced models directly from data without the need for computationally expensive adjoint simulations. Under impulse response experiments, DMDc and ERA are algorithmically connected [290, 222]. OKID is a generalization of ERA when convenient impulse response data is unavailable [152, 220, 218, 150]. OKID is also a part of a larger set of system identification methods called subspace identification methods [226]. There are a number of connections between DMDc and other popular subspace identification methods, such as N4SID, MOESP, and CVA [292, 293, 154, 226]. These methods involve regression, model reduction, and parameter estimation, similar to DMDc, as described in [226]. With these strong connections in the field of system identification, we believe DMDc is well poised to be impactful in this field for high-dimensional, complex systems.

6.5 ▪ Connections to Koopman operator theory

DMD has been rigorously connected to the analysis of nonlinear dynamical systems through Koopman spectral analysis, as discussed in Chapter 3. In this section, we briefly describe how Koopman operator theory can be generalized to handle complex systems with external inputs. The generalized Koopman operator called Koopman with inputs and control (KIC) can be connected to DMDc for linear observable measurements [223].

Consider the nonlinear dynamical system with inputs

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \tag{6.23}$$

where $\mathbf{x} \in \mathcal{M}$ and $\mathbf{u} \in \mathcal{N}$; both $\mathcal{M}$ and $\mathcal{N}$ are smooth manifolds. We define a set of scalar-valued functions $g : \mathcal{M} \times \mathcal{N} \to \mathbb{R}$, called observables. Each observable is an element of an infinite-dimensional Hilbert space $\mathscr{H}$. The KIC $\mathcal{K} : \mathscr{H} \to \mathscr{H}$ acts on the Hilbert space of observables given by

$$\mathcal{K}\, g(\mathbf{x}, \mathbf{u}) \triangleq g(\mathbf{f}(\mathbf{x}, \mathbf{u}), *), \tag{6.24}$$

where $*$ indicates a choice of definition, based on the type of complex system under analysis. The following are the two main choices:

1. $* = \mathbf{u}$: The inputs are evolving dynamically. The inputs could be generated by state-dependent controllers or by externally evolving systems found in multiscale modeling.

2. $* = 0$: The inputs are not evolving dynamically. The inputs affect the state but could be random exogenous disturbances or impulse-response experiments.

The linear characteristics of KIC allow us to perform an eigendecomposition of $\mathcal{K}$ given in the standard form:

$$\mathcal{K}\varphi_j(\mathbf{x},\mathbf{u}) = \lambda_j \varphi_j(\mathbf{x},\mathbf{u}), \quad j = 1,2,\ldots. \tag{6.25}$$

The operator is now spanned by eigenfunctions that are defined on the inputs and state. Using the infinite expansion shown in (6.25), the observable functions g_j can be rewritten in terms of the right eigenfunctions φ_j,

$$\mathbf{g}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} g_1(\mathbf{x},\mathbf{u}) \\ g_2(\mathbf{x},\mathbf{u}) \\ \vdots \\ g_p(\mathbf{x},\mathbf{u}) \end{bmatrix} = \sum_{j=1}^{\infty} \varphi_j(\mathbf{x},\mathbf{u})\mathbf{v}_j, \tag{6.26}$$

where p is the number of measurements. The new Koopman operator can be applied to this representation of the measurement:

$$\mathcal{K}\mathbf{g}(\mathbf{x},\mathbf{u}) = \mathbf{g}(\mathbf{f}(\mathbf{x},\mathbf{u}),\mathbf{u}) = \sum_{j=1}^{\infty} \lambda_j \varphi_j(\mathbf{x},\mathbf{u})\mathbf{v}_j. \tag{6.27}$$

Note that the expansion is in terms of Koopman eigenfunctions φ with vector-valued coefficients $\mathbf{v}_j$ called Koopman modes. The terminology of Koopman operator theory now allows for measurement functions that accept inputs.

6.5.1 ▪ KIC for linear systems

KIC is directly connected to DMDc for linear dynamical systems. Consider the linear system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k. \tag{6.28}$$

We choose a set of linear measurements on the state and inputs, e.g. $g_1(\mathbf{x},\mathbf{u}) = x_1$ and $g_2(\mathbf{x},\mathbf{u}) = u_1$. The linear dynamical system can be rewritten in terms of a new set of measurements $\mathbf{z}_k$

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \tag{6.29a}$$

$$\mathbf{z}_{k+1} = \mathbf{G}\mathbf{z}_k. \tag{6.29b}$$

The eigenvalues and eigenvectors of $\mathbf{G}$ are fundamentally connected to the eigenfunctions and Koopman modes of $\mathcal{K}$. The linear dynamical system in (6.29) with inputs is not the canonical version of an input-output system. The future state $\mathbf{z}_{k+1}$ rarely includes the future input $\mathbf{u}_{k+1}$. If there are dynamics on the inputs, the Koopman operator will discover them. If not, the definition $* = 0$ should be employed. This allows for a reduction of (6.29) to

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{u}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \tag{6.30}$$

which can be further reduced to

$$\begin{bmatrix} \mathbf{x}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}, \tag{6.31a}$$

$$\mathbf{x}_{k+1} = \mathbf{G}\ \mathbf{z}_k. \tag{6.31b}$$

The formulation in (6.31) is equivalent to DMDc. The KIC formulation generalizes DMDc in a number of important ways. Systems where inputs are dynamically evolving can be considered. A larger set of observables, including nonlinear libraries, can now also be considered since $\mathscr{K}$ is defined on an entire set of observables in $\mathscr{H}$.