

A data-driven Koopman model predictive control framework for nonlinear flows

Hassan Arbabi, Milan Korda and Igor Mezić *

June 8, 2018

Abstract

The Koopman operator theory is an increasingly popular formalism of dynamical systems theory which enables analysis and prediction of the nonlinear dynamics from measurement data. Building on the recent development of the Koopman model predictive control framework [1], we propose a methodology for closed-loop feedback control of nonlinear flows in a fully data-driven and model-free manner. In the first step, we compute a Koopman-linear representation of the control system using a variation of the extended dynamic mode decomposition algorithm and then we apply model predictive control to the constructed linear model. Our methodology handles both full-state and sparse measurement; in the latter case, it incorporates the delay-embedding of the available data into the identification and control processes. We illustrate the application of this methodology on the periodic Burgers' equation and the boundary control of a cavity flow governed by the two-dimensional incompressible Navier-Stokes equations. In both examples the proposed methodology is successful in accomplishing the control tasks with sub-millisecond computation time required for evaluation of the control input in closed-loop, thereby allowing for a real-time deployment.

Keywords: Flow control, Koopman operator theory, Feedback control, Dynamic mode decomposition, Model predictive control

1 Introduction

Flow control is one of the central topics in fluid mechanics with an enormous impact on other fields of engineering and applied science. Its wide range of applications includes, just to name a few, reduction of aerodynamic drag on vehicles and aircrafts, mixing enhancement in combustion and chemical processes, suppression of instabilities to avoid structural fatigue, lift increase for wind turbines, and design of biomedical devices. To emphasize the impact of

*The authors are with the department of Mechanical Engineering, University of California, Santa Barbara, CA, 93106, USA. {[harbabi](mailto:harbabi@engineering.ucsb.edu), [milan.korda](mailto:milan.korda@engineering.ucsb.edu), [mezic](mailto:mezic@engineering.ucsb.edu)}@engineering.ucsb.edu

flow control, it is worth noting that discovery of efficient flow control techniques for reduction of drag on ships and cars can result in mitigation of yearly CO₂ emission by millions of tons and annual savings of billions of dollars in the global shipping industry [2, 3].

Despite all the interest and continuous effort, the flow control still poses a daunting challenge to our theoretical understanding and computational resources. The main source of difficulty is the combination of high-dimensionality and nonlinearity of fluid phenomena which results in computational or experimental models which are too complex and costly to control using well-developed strategies of modern control theory. The recent advances in numerical computation has led to partial success with active control of flows using models based on Navier-Stokes equations [2, 4–6]; however, these methods suffer from two major shortcomings: first, nonlinear models obtained from Navier-Stokes are still high-dimensional and computationally costly, thereby not allowing for fast implementation of nonlinear and computationally complex control techniques such as nonlinear model predictive control, and second, linear models used with LQR/LQG or adjoint-based controllers often rely on local linearization around equilibria or a trajectory of the flow which makes them valid only locally, and may result in suboptimal or even unstable control performance.

An alternative approach that has gained traction in the last two decades is identification of relatively low-dimensional flow models from data provided by numerical simulations or experiments. Some of the data-driven methods combine the measurement data with underlying physical model to identify models of the system. The major examples include construction of (usually autonomous) state space models via Galerkin projection of the Navier-Stokes equations onto the modes obtained by proper orthogonal decomposition (POD) of data [7–9], or identification of linear input-output systems using balanced POD [10, 11]. There are also a few applications of system identification methods to construct linear input-output models purely from data, including the eigensystem realization algorithm [12, 13], as well as subspace identification and autoregressive models [14, 15]. Utilization of the above techniques in a variety of problems has shown great promise for low-dimensional modeling and control of complex flows from data.

In this paper, we present a general and fully data-driven framework for control of nonlinear flows based on the Koopman operator theory [16, 17]. This theory is an operator-theoretic formalism of classical dynamical systems theory with two key features: first, it allows a scalable reconstruction of the underlying dynamical system from measurement data, and second, the models obtained are linear (but possibly high-dimensional) due to the fact that the Koopman operator is a linear operator whether the dynamical system is linear or not. The linearity of the Koopman models is especially advantageous since it makes them amenable to the plethora of mature control strategies developed for linear systems. This framework for design of controller, is called *Koopman-MPC* and follows the work in [1]. In the first step of our approach, we build a finite-dimensional approximation of the controlled Koopman operator from the data, using a variation of the extended dynamic mode decomposition algorithm (EDMD) [18], with a particular choice of observables assuring linearity of the resulting approximation. The ideal data would include measurements on a number of system trajectories with various input sequences. In the second step, we apply the model predictive control (MPC) to these linear models to obtain the desired objectives. The distinguishing feature is that this framework leads to a *linear* MPC, solving a convex quadratic programming problem, and thereby enables a rapid solution of the underlying optimization problem, which is

necessary for real-time deployment. This methodology can be also applied to problems with sparse measurements, i.e., problems with a limited number of instantaneous measurements (e.g. point measurements of the velocity field at several different locations). In that case, delay-embedding of the available measurements (and nonlinear functions thereof) is used to construct the Koopman-linear model; the MPC is then applied to the linear system whose state variable is the delay-embedded vector of measurements.

The outline of this paper is as follows: A brief review of related work is given in section 1.1. Section 2 gives a review of the Koopman operator theory for dynamical systems with input. In section 3, we describe the EDMD algorithm for construction of the Koopman-linear model from measurement data. In section 3.1, we discuss using delay-embedding to construct and control Koopman-linear models from sparse measurements. An overview of the MPC framework is given in section 4. In section 5, we present two numerical examples: the Burgers’ system on a periodic domain and the 2D lid-driven cavity flow. We formulate the control problem for these cases using various objectives and demonstrate our approach for both full-state and sparse measurements. We summarize the results and discuss the outlook in section 6.

1.1 Review of related work

The Koopman operator formalism of dynamical systems is rooted in the seminal works of Koopman and Von Neumann in the early 1930s [16, 19]. This formalism appeared mostly in the context of ergodic theory for much of the last century, until in mid 2000’s when the works in [20, 21] pointed out its potential for rigorous analysis of dynamical systems from data. The notion of *Koopman mode decomposition (KMD)* which is based on the expansion of observable fields in terms of Koopman operator eigenvalues and eigenfunctions was also introduced in [21]. KMD was first applied to a complex flow in [22] where its connection with the DMD numerical algorithm [23] was pointed out. The work in [22] showed the promise of this viewpoint in extracting the physically relevant flow structures and time-scales from data. Following the success of this work, KMD and its numerical implementation through DMD, has become a popular decomposition for dynamic analysis of nonlinear flows [24–28].

The application of the Koopman operator to data-driven control of high dimensional systems is much less developed. The earliest works on generalizing the Koopman operator approach to control systems was presented in [29, 30] accompanied with a numerical variation of DMD algorithm [31]. To the best of our knowledge, however, the only application for feedback control of fluid flows are the works in [32, 33]. The work in [32] considered the problem of flow control using a finite set of input values. For each value of the input, a Koopman-linear model was constructed from the data and the control problem was formulated as a switched optimal control and implemented in a receding horizon fashion. This methodology was successfully used for tracking reference output signals in Burgers equation and incompressible flow past a cylinder. The work in [33] proposed to remove the restriction of the input to a finite set, by interpolating between the Koopman-linear systems for each input value which led to an improvement of the control performance. In [1] (which this work is based on), a more general extension of the Koopman operator theory to systems with input was presented, and used to construct linear predictors especially suitable for model predictive control, demonstrating the effectiveness of the approach (among other examples)

on the control of the Korteweg-de Vries PDE. The results in [1] showed superiority of the controlled Koopman-linear predictors constructed from data to models obtained by local linearization and Carleman’s representation both for prediction and for feedback control. Let us also mention the earlier work in [34] that utilized KMD to construct the normal forms, for dynamics of the flow past an oscillating cylinder; the input forcing appeared as a bilinear term in the normal forms for this flow. See [35] for application to pulse-based control of monotone systems as well as [36] for system identification and [37] for state estimation.

The numerical engine behind the system identification part of the framework presented in this paper is the Extended Dynamic Mode Decomposition (EDMD) algorithm proposed in [18]. Although the original DMD algorithm was invented independent of the Koopman operator theory [23], the connection between the two was known from early on [22], and DMD-type algorithms have become the popular methods for computation of the Koopman operator spectral properties. Nevertheless, the convergence of DMD algorithms for approximation of Koopman operator is just recently established in [38, 39]. For application of Koopman-MPC to systems with sparse measurements, the EDMD is modified to include the delay embeddings of instantaneous measurements. Delay embedding is a classic technique in system identification literature (see, e.g., [40] for a comprehensive reference) and control as well as in linear and nonlinear time-series analysis (e.g., [41]). In the field of dynamical systems, the classical reference is the work of Takens [42] on geometric reconstruction of nonlinear attractors. The work in [43] suggested the combination of this technique with the DMD algorithm for identification of nonlinear systems and its role in approximation of the Koopman operator was studied in [38, 44]; the use for control, in the Koopman operator context, was described in [1].

2 Koopman operator theory

In this section, we first review the basics of the Koopman operator formalism for autonomous dynamical systems and then discuss its extension to systems with input and output. We will focus on *discrete-time* dynamical systems to be consistent with the discrete-time nature of the measurement data, but most of the analysis easily carries over to the continuous-time systems. We refer the reader to [17, 45] for a more detailed discussion of the Koopman operator basics.

Consider the dynamical system

$$x^+ = T(x), \quad x \in M \tag{1}$$

defined on a state space M . We call any function $g : M \rightarrow \mathbb{R}$ an *observable* of the system, and we note that the set of all observables forms a (typically infinite-dimensional) vector space. The Koopman operator, denoted by \mathcal{K} , is a linear transformation on this vector space given by

$$\mathcal{K}g = g \circ T, \tag{2}$$

where \circ denotes the function composition, i.e., $(\mathcal{K}g)(x) = g(T(x))$. Informally speaking, the Koopman operator updates the observable g based on the evolution of the trajectories in the

state space. The key property of the Koopman operator that we exploit in this work is its linearity, that is, for any two observables g and h , and scalar values α and β , we have

$$\mathcal{K}(\alpha g + \beta h) = \alpha \mathcal{K}g + \beta \mathcal{K}h, \quad (3)$$

which follows from the definition in eq. (2). We call the observable ϕ a Koopman eigenfunction associated with Koopman eigenvalue $\lambda \in \mathbb{C}$ if it satisfies

$$\mathcal{K}\phi = \lambda\phi. \quad (4)$$

The spectral properties of the Koopman operator can be used to characterize the state space dynamics; for example, the Koopman eigenvalues determine the stability of the system and the level sets of certain Koopman eigenfunctions carve out the invariant manifolds and isochrons [46–48]. Moreover, for smooth dynamical systems with simple nonlinear dynamics, e.g., systems that possess hyperbolic fixed points, limit cycles and tori, the evolution of observables can be described as a linear expansion in Koopman eigenfunctions [49]. In these systems, the spectrum of the Koopman operator consists of only point spectrum (i.e. eigenvalues) which fully describes the evolution of observables;

$$\mathcal{K}^n g = \sum_{j=0}^{\infty} v_j \phi_j \lambda_j^n. \quad (5)$$

where v_j is called the Koopman mode associated with Koopman eigenvalue-eigenfunction pair (λ_j, ϕ_j) and it is given by the projection of the observable g onto ϕ_j . See [22] for more detail on Koopman modes, and [49] on the expansion in (5).

The extension of the Koopman operator theory to a controlled system denoted by

$$x^+ = T(x, u), \quad x \in M, \quad u \in \mathcal{U}, \quad (6)$$

requires one to work on the *extended state space*, which is the Cartesian product of the state space M and the space of all input sequences $\ell(\mathcal{U}) = \{(u_0)_{i=0}^{\infty} \mid u_i \in \mathcal{U}\}$. We denote the extended state space by $S = M \times \ell(\mathcal{U})$. Now, given an observable $g : S \rightarrow \mathbb{R}$ we can define the non-autonomous Koopman operator,

$$(\mathcal{K}g)(x, (u_i)_{i=0}^{\infty}) = g(T(x, u_0), (u_i)_{i=1}^{\infty}). \quad (7)$$

See [1] for more details on this extension. We emphasize that the linear representation of the nonlinear system by the Koopman operator is globally valid and generalizes the local linearization around equilibria [49].

3 Construction of Koopman-linear system

In this section, we review the construction of the Koopman-linear system as proposed by [1] using the EDMD algorithm [18]. We are looking to approximate the dynamics of the nonlinear flow via a linear time-invariant system such as

$$\begin{aligned} z^+ &= Az + Bu & z \in \mathbb{R}^n, \quad u \in \mathbb{R}^k, \\ \hat{x} &= Cz. \end{aligned} \quad (8)$$

Consider the set of states and inputs of the nonlinear system in the form of

$$X = [x_1, \dots, x_K], \quad X^+ = [x_1^+, x_2^+, \dots, x_K^+], \quad U = [u_1, \dots, u_K]. \quad (9)$$

where $x_j^+ = T(x_j, u_j)$. Let

$$\mathbf{g}(x) = [g_1(x) \quad \dots \quad g_m(x)]^\top \quad (10)$$

be a given vector of possibly nonlinear observables. These functions may represent user-specified nonlinear functions of the state as well as physical measurements (i.e., outputs) taken on the dynamical system (or nonlinear functions of such outputs). We are going to assume that we only have access to values of the observables, and therefore, explicit knowledge of the state variable in (9) is not required. By collecting data on the dynamical system, we can form the lifted snapshot data matrices

$$X_{\text{lift}} = [\mathbf{g}(x_1), \dots, \mathbf{g}(x_K)], \quad X_{\text{lift}}^+ = [\mathbf{g}(x_1^+), \dots, \mathbf{g}(x_K^+)], \quad U = [u_1, \dots, u_K]. \quad (11)$$

These data matrices are the lifted coordinates of the system in the space of observables. Note that, as in [1], we have not lifted U coordinates to preserve the linear dependence of the predictor on the original input. The matrices A , B and C are then given by the solution to the linear least-squares problems

$$\min_{A,B} \|X_{\text{lift}}^+ - AX_{\text{lift}} - BU\|_F, \quad \min_C \|X - CX_{\text{lift}}\|_F \quad (12)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The analytical solution to these two problems can be compactly written as

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} = \begin{bmatrix} X_{\text{lift}}^+ \\ X \end{bmatrix} \begin{bmatrix} X_{\text{lift}} \\ U \end{bmatrix}^\dagger. \quad (13)$$

When snapshot matrix X_{lift} is fat (i.e. number of columns exceeds number of rows), it is more efficient to compute the matrices by solving the normal equations

$$V = \mathcal{M}G, \quad (14)$$

with the unknown matrix variable \mathcal{M} and given matrices

$$V = \begin{bmatrix} X_{\text{lift}}^+ \\ X \end{bmatrix} \begin{bmatrix} X_{\text{lift}} \\ U \end{bmatrix}^\top, \quad G = \begin{bmatrix} X_{\text{lift}} \\ U \end{bmatrix} \begin{bmatrix} X_{\text{lift}} \\ U \end{bmatrix}^\top.$$

The solution \mathcal{M} to (14) provides the matrices A , B , C through

$$\mathcal{M} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}.$$

The matrices A and B describe the linear dynamics of the Koopman-linear state $z = \mathbf{g}(x)$. The prediction of the original state x is obtained simply by $\hat{x} = Cz$. See [39] for a convergence analysis of EDMD for approximation of Koopman operator.

3.1 Sparse measurements and delay embedding

When the number of observables measured on a dynamical system is insufficient for construction of an accurate model, we can use the delay embedding of the observables. Delay embedding (i.e., embedding several consecutive output measurements into a single data point) is a classical technique ubiquitous in system identification literature (e.g., [40]) but also in the theory of dynamical systems for geometric reconstruction of nonlinear attractors [42]. It has also been utilized in the context of Koopman framework in [20, 38, 50]. The key feature of delay embedding here is that it provides samplings of extra observables to realize the Koopman operator. To be more precise, if we have a sequence of measurements on a single observable h at the n_d time instants $t_i, t_{i+1}, \dots, t_{i+n_d-1}$, we can think of them as sampling of the n_d observables $[h, \mathcal{K}h, \dots, \mathcal{K}^{n_d-1}h]$ at the single time instant t_i . Here, we describe how we can incorporate delay-embedding into identification and control of Koopman-linear models. The only requirement for identification is that we should have access to at least $n_d + 1$ *sequential* time samples on the trajectories where n_d is the chosen number of delays.

Let \mathbf{h} be the vector of instantaneously measured observables on the dynamical system (e.g., point measurements of the velocity field), and n_d be the delay embedding dimension. Consider the state and input matrices described in (9), but now assume that they contain a string of sequential samples with length $n_d + 1$, i.e., for some j , we have

$$x_{i+1} = T(x_i, u_i), \quad i = j, \dots, j + n_d - 1. \quad (15)$$

We can delay embed the measurements on this string to construct a pair of lifted coordinates in the space of observables,

$$\zeta_j = \begin{bmatrix} \mathbf{h}(x_i) \\ \vdots \\ \mathbf{h}(x_{i+n_d-1}) \\ u_i \\ \vdots \\ u_{i+n_d-1} \end{bmatrix}, \quad \zeta_j^+ = \begin{bmatrix} \mathbf{h}(x_{i+1}) \\ \vdots \\ \mathbf{h}(x_{i+n_d}) \\ u_{i+1} \\ \vdots \\ u_{i+n_d} \end{bmatrix}, \quad i = j, \dots, j + n_d - 1. \quad (16)$$

It is easy to check that $\zeta_j^+ = \mathcal{K}\zeta_j$. By delay-embedding the observations on all sequential strings of data, we can form the new matrices

$$\tilde{X} = [\zeta_1, \zeta_2, \dots, \zeta_L], \quad \tilde{Y} = [\zeta_1^+, \zeta_2^+, \dots, \zeta_L^+]. \quad (17)$$

We can once again lift the data using a vector of nonlinear user-specified functions \mathbf{g} to form the new lifted matrices,

$$\begin{aligned} X_{\text{lift}} &= [\mathbf{g}(\zeta_1), \mathbf{g}(\zeta_2), \dots, \mathbf{g}(\zeta_L)], \\ Y_{\text{lift}} &= [\mathbf{g}(\zeta_1^+), \mathbf{g}(\zeta_2^+), \dots, \mathbf{g}(\zeta_L^+)]. \end{aligned} \quad (18)$$

Having X_{lift} , Y_{lift} and input matrix U defined, we solve the the least-squares problems (12) to find the linear system matrices. It is very important for the lifting function \mathbf{g} to have a meaningful dependence on u_i, \dots, u_{i+n_d} . This allows EDMD to approximate the dynamics

of the *extended* state space and discern the effect of previous inputs in the evolution of the state.

The linear predictor in this case would be

$$\begin{aligned} z^+ &= Az + Bu \\ \hat{\zeta} &= Cz, \end{aligned} \tag{19}$$

Here $\hat{\zeta}$ denotes the prediction of the “embedded” state ζ (note that when $\hat{\zeta}$ is employed for controller design, typically only the part of $\hat{\zeta}$ corresponding to the most recent output prediction is used).

4 Model predictive control

The methodology presented in the last section allows us to construct a model of the flow in the form of a linear dynamical system (8). In this work, we will apply MPC to this linear model to control the original nonlinear flow, but other techniques from modern control theory could be applied as well; see the survey [51] or the book [52] for an overview of MPC. In the context of MPC, we formulate the control objective as minimization of a cost function over a finite-time horizon. The general strategy is to use the model in eq. (8) to predict the system evolution over the horizon, and use these predictions to compute the optimal input sequence minimizing the given cost function along this horizon. Then we *apply only the first element* of the computed input sequence to the real system, thereby producing a new value of the output, and repeat the whole process. This technique is sometimes called the *receding horizon control*. In the following, we describe the notation and some mathematical aspects of this technique. The distinguishing feature when using the lifted linear predictor (8) is that the resulting MPC problem is a convex quadratic program (QP) despite the original dynamics being nonlinear. In addition, the complexity of solving the quadratic problem can be shown to be independent of the size of the lift if the so-called *dense form* is used [1], thereby allowing for a rapid solution using highly efficient QP solvers tailored for linear MPC applications (in our case qpOASES [53]).

Let N be the length of the prediction horizon, and $\{u_i\}_{i=0}^{N-1}$ and $\{y_i\}_{i=1}^N$ denote the sequence of input and output values over that horizon. A very common choice of cost functions is the convex quadratic form,

$$\begin{aligned} J(\{u_i\}_{i=0}^{N-1}, \{y_i\}_{i=1}^N) &= y_N^\top Q_N y_N + q^\top y_N \\ &+ \sum_{i=1}^{N-1} y_i^\top Q_i y_i + u_i^\top R_i u_i + q_i^\top y_i + r_i^\top u_i \\ &+ u_0^\top R_0 u_0 + r_0^\top u_0, \end{aligned} \tag{20}$$

where $Q_{i=0,\dots,N}$ and $R_{i=0,\dots,N-1}$ are real symmetric positive-definite matrices. The above cost function can be used to formulate many of the common control objectives including the tracking of a reference signal. For example, assume that we want to control the flow such that its output measurements follow an arbitrary time-dependent output sequence denoted

by $\{y^{\text{ref}}\}_i$. We can formulate this objective as minimization of the distance between $\{y\}_i$ and $\{y^{\text{ref}}\}_i$, and the corresponding cost function over the finite horizon would be

$$\begin{aligned} J_1(\{u_i\}_{i=0}^{N-1}, \{y_i\}_{i=1}^N) &= \sum_{i=1}^N (y_i - y_i^{\text{ref}})^\top Q (y_i - y_i^{\text{ref}}), \\ &= \sum_{i=1}^N y_i^\top Q y_i - 2(y_i^{\text{ref}})^\top Q y_i + (y_i^{\text{ref}})^\top Q y_i^{\text{ref}}. \end{aligned} \quad (21)$$

where Q is the weight matrix that determines the relative importance of measurements in y . Note that the last term in the above equation is not dependent on the input or output, and therefore it does not affect the optimal solution. By dropping this term, and letting $q_i = -Q^\top y_i^{\text{ref}}$, we obtain

$$J_1(\{u_i\}_{i=0}^{N-1}, \{y_i\}_{i=1}^N) = \sum_{i=1}^N y_i^\top Q y_i + q_i^\top y_i, \quad (22)$$

which is a special form of eq. (20). In the numerical examples presented in this paper, we will use this type of cost function.

The MPC controller solves the following optimization problem at *each time step* of the closed loop operation

$$\begin{aligned} (\{u_i^*\}_{i=0}^{N-1}, \{y_i^*\}_{i=1}^N) &= \arg \min J(\{u_i\}_{i=0}^{N-1}, \{y_i\}_{i=1}^N) \\ \text{s.t.} \quad & z_{i+1} = A z_i + B u_i, \quad i = 0, \dots, N \\ & y_i = C z_i \\ & E_i^y y_i + E_i^u u_i \leq b_i, \quad i = 0, \dots, N-1, \\ & E_N y_N \leq b_N \\ & z_0 = \mathbf{g}(\zeta_c), \end{aligned} \quad (23)$$

where ζ_c is the delay-embedded vector of measurements

$$\zeta_c = [\mathbf{h}(x_{k-n_d+1}), \dots, \mathbf{h}(x_k), u_{k-n_d}, \dots, u_{k-1}]^\top \quad (24)$$

The matrices $E_{i=0, \dots, N-1}^x$, $E_{i=0, \dots, N-1}^u$ and E_N define polyhedral state and input constraints. This is a standard form of a convex quadratic programming problem which can be efficiently solved using many available QP solvers - in our case qpOASES [53]. The computational complexity can be further reduced by expressing the lifted state variables z in terms of the control inputs u , thereby eliminating the dependence on the possible very large dimension of z ; see [1] for details.

Once the optimal input sequence $\{u_i^*\}_{i=0}^{N-1}$ is computed, we apply its first element u_0^* to the system to obtain a new output measurement which updates the current state ζ_c ; the whole process is then repeated in a receding horizon fashion. Algorithm 1 summarizes the closed-loop control operation, and the entire algorithm for implementation of the Koopman-MPC is illustrated in Figure 1.

Algorithm 1 Koopman MPC – closed-loop operation

Initialization: $\mathbf{h}(x_{-n_d}), \dots, \mathbf{h}(x_{-1}), u_{-n_d}, \dots, u_{-1}$

- 1: **for** $k = 0, 1, \dots$ **do**
 - 2: Measure $\mathbf{h}(x_k)$.
 - 3: Set $\zeta_c = [\mathbf{h}(x_{k-n_d+1}), \dots, \mathbf{h}(x_k), u_{k-n_d}, \dots, u_{k-1}]^\top$.
 - 4: Set $z_0 := \mathbf{g}(\zeta_c)$
 - 5: Solve (23) to get an optimal solution $(u_i^*)_{i=1}^N$
 - 6: Apply u_1^* to the nonlinear system
-

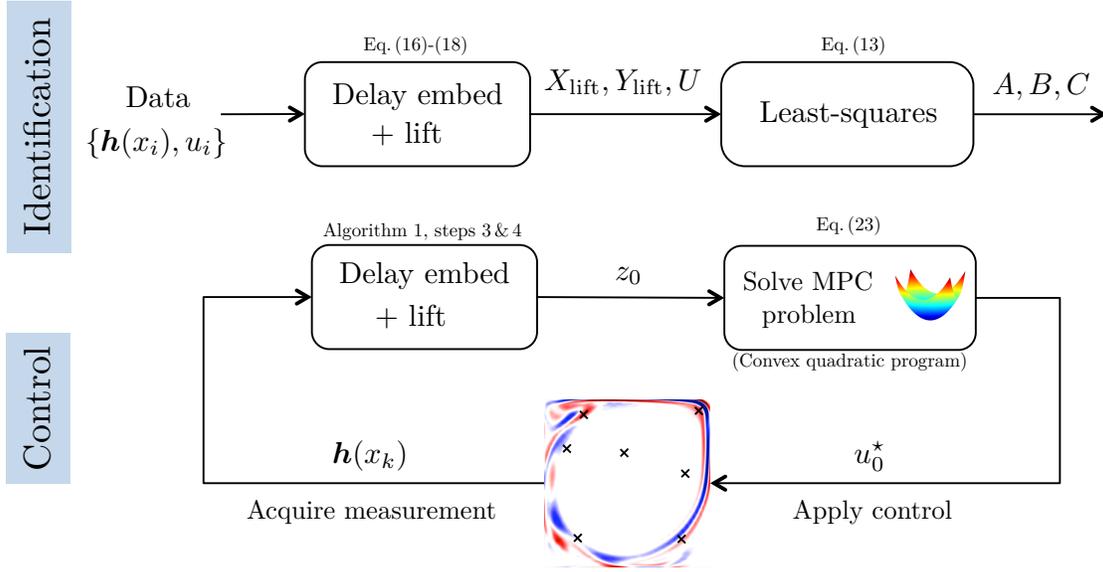


Figure 1: Schematic representation of Koopman-MPC framework for identification and closed-loop control of nonlinear flows.

5 Numerical Examples ²

5.1 Burgers equation

As the first example, we consider the Burgers equation with periodic boundary condition,

$$\frac{\partial v}{\partial t} + v \frac{\partial v}{\partial z} = \nu \frac{\partial^2 v}{\partial z^2} + f(z, t), \quad z \in [0, 1], \quad t \in [0, \infty). \quad (25)$$

$$v(0, t) = v(1, t) \quad (26)$$

where ν is the kinematic viscosity. Note that we have used z to denote the spatial coordinates in the flow examples, hoping that it will not be confused with the Koopman-linear state in

²The MATLAB implementation of the examples is available at https://github.com/arbabiha/KoopmanMPC_for_flowcontrol.

(8). Similar to [32], we assume the forcing $f(z, t)$ is given by

$$f(z, t) = u_1(t)f_1(z) + u_2(t)f_2(z), \quad (27)$$

$$= u_1(t)e^{-\left(15(z-0.25)\right)^2} + u_2(t)e^{-\left(15(z-0.75)\right)^2} \quad (28)$$

with the control input $u = (u_1, u_2) \in \mathbb{R}^2$.

The control objective is to follow the reference state

$$\begin{aligned} v_{\text{ref}}(z, 0 \leq t < 2) &= \frac{1}{2}, \\ v_{\text{ref}}(z, 1 \leq t < 4) &= 1, \\ v_{\text{ref}}(z, 4 \leq t < 6) &= \frac{1}{2}, \end{aligned} \quad (29)$$

starting from the initial condition,

$$v(z, 0) = ae^{-\left(5(z-0.5)\right)^2} + (1-a)\sin(4\pi z). \quad (30)$$

with $a \in [0, 1]$ chosen randomly, and with the input signals constrained as $|u_1|, |u_2| < 0.1$.

To construct the Koopman-linear system, we have used 50 two-second long trajectories with $\nu = 0.01$. Each trajectory starts from a random initial condition as in (30). The input control at each time instant is randomly drawn from the uniform distribution on $(u_1, u_2) \in [-0.1, 0.1]^2$. The Burgers equation upwind finite-difference scheme for advection and central difference for diffusion term, with 4th-order Runge-Kutta time stepping performed on 150 spatial grid points with time steps of 0.01 second.

In case of full-state measurements, we use the vector of values of v at the computational grid points, the kinetic energy of v , and the constant observable ($\psi(v) = 1$). The cost function to be minimized is the kinetic energy (L^2 -norm) of the state tracking error,

$$e(t) = \int_0^1 |v(z, t) - v_{\text{ref}}(z, t)|^2 dz. \quad (31)$$

For the sparse measurement scenario, we assume that we only have access to the vector of sparse measurements $v^s = \mathbf{h}(v)$ which consists of values of v at 10 random grid points. We form the Koopman-linear state vector by including delay embedding of v^s with embedding dimension $n_d = 5$, the kinetic energy of instantaneous measurements $\|v^s\|^2$, and the constant observable. That is

$$\zeta_c = [v^s(t_{i-4}), \dots, v^s(t_i), u(t_{i-4}), \dots, u(t_{i-1})]^\top, \quad (32)$$

$$\mathbf{g}(\zeta_c) = [\zeta_c^\top, \|v^s(t_i)\|^2, 1]^\top \in \mathbb{R}^{60}. \quad (33)$$

We define the tracking error as

$$e(t) = \frac{1}{m} \|v^s(t) - v_{\text{ref}}^s(t)\|^2, \quad (34)$$

and the predicted objective function used within the MPC is then

$$J = \int_0^T e(t) dt,$$

where the prediction horizon is set to $T = 0.1$. After spatio-temporal discretization, this objective readily translates to the form (20) with $N = 10$.

The results of the controlled simulation for both scenarios are depicted in fig. 2. In both cases, the state successfully tracks the reference signal; however, the controller built via sparse measurements is slightly delayed compared to the case of full-state measurement which can be attributed to the construction of the Koopman-linear state using delay-embedding. We note that the tracking error during the transient phases is caused by input saturation as documented by the plot of the control input signal.

Robustness with respect to the parameter ν . One question that arises in the context of low-dimensional modeling is whether the models constructed at some parameter value would be robust enough for prediction at other values. In order to test the robustness of Koopman-linear model in case of Burgers, we use the model constructed above using sparse measurements to control the Burgers system at various values of $\nu \in [10^{-4}, 0.1]$. The results in fig. 3 indicate that Koopman-linear model constructed at the parameter regime $\nu = 0.01$ is remarkably effective over a wide parameter range, and the control performance is very robust. As expected, however, the input signal and tracking error in the diffusion-dominated regime (large ν) is less fluctuating, as the diffusion helps the controller to stabilize the state around the spatially-uniform reference state in (29).

5.2 2D lid-driven cavity flow

In the second example, we consider an incompressible viscous flow in a square cavity which is driven by motion of the top lid. The dynamics of the cavity flow is governed by the Navier-Stokes equation, which, in terms of the stream function variable reads

$$\frac{\partial}{\partial t} \nabla^2 \psi + \frac{\partial \psi}{\partial z_2} \frac{\partial}{\partial z_1} \nabla^2 \psi - \frac{\partial \psi}{\partial z_1} \frac{\partial}{\partial z_2} \nabla^2 \psi = \frac{1}{Re} \nabla^4 \psi, \quad (z_1, z_2) \in [-1, 1]^2, \quad t \in [0, \infty), \quad (35)$$

$$\psi \Big|_{z_1=\pm 1} = \psi \Big|_{z_2=-1} = 0 \quad \text{and} \quad \frac{\partial \psi}{\partial z_2} \Big|_{z_2=1} = f(z_1, t), \quad (36)$$

where $\psi(z_1, z_2, t)$ is the stream function, Re is the Reynolds number, and $f_1(z_1, t)$ is the velocity of the top lid which acts as the forcing on the system. We assume that we can control the top lid velocity,

$$f(z_1, t) = (1 + u(t))(1 - z_1^2)^2, \quad (37)$$

with the control input $u \in \mathbb{R}$.

The autonomous cavity flow with $Re \leq 10000$ converges to a steady velocity profile (i.e. fixed point in the state space) which consists of a large central vortex with downstream corner eddies. At around $Re = 10500$, a Hopf bifurcation makes the fixed point unstable

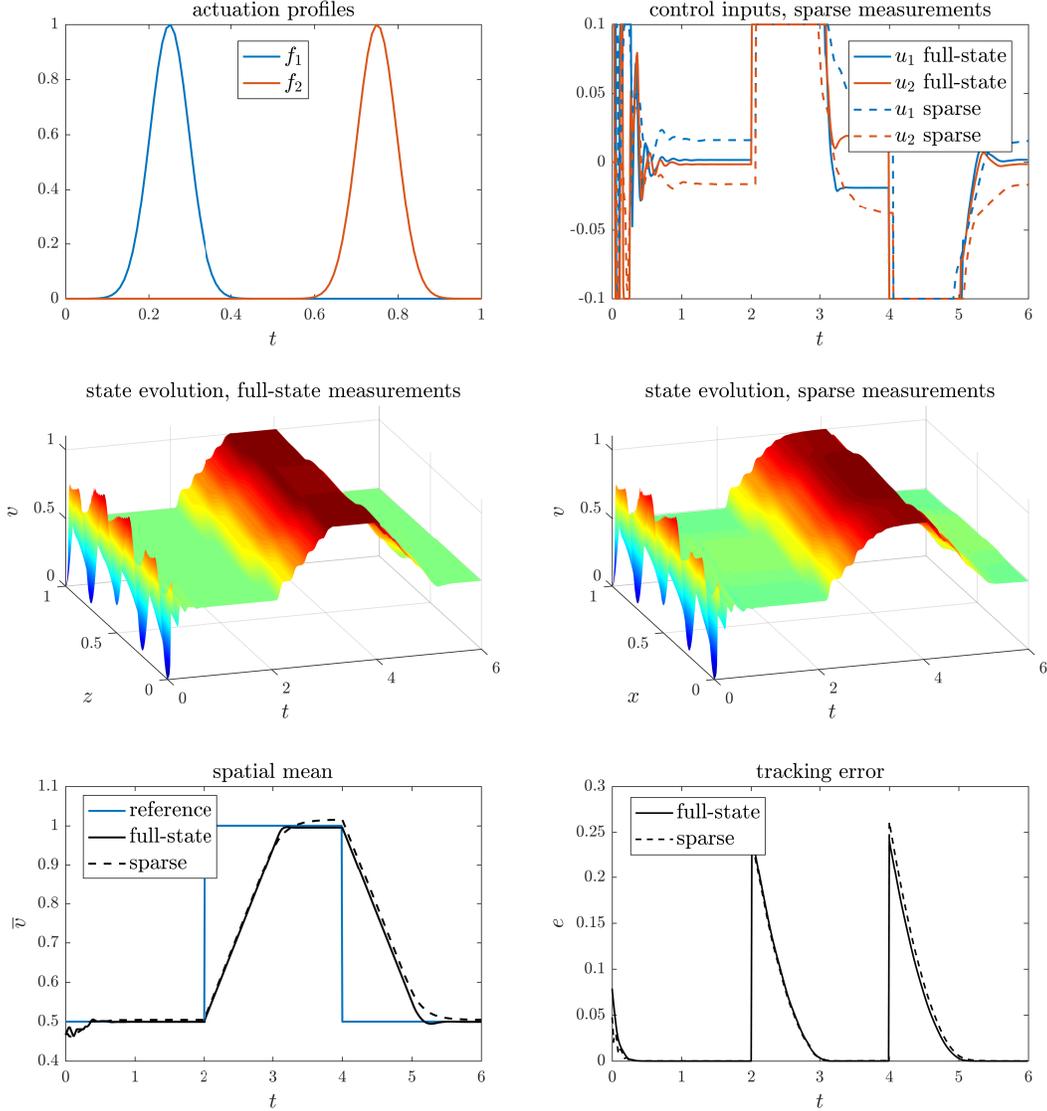


Figure 2: **Koopman-MPC for control of Burgers system.** Input signals, tracking error and state evolution in for closed-loop simulation using Koopman-linear models constructed by full-state measurements (150 observables) and sparse measurements (10 observables).

and the solutions up to $Re = 15000$ converge to a limit cycle. In this regime, the boundary of the central vortex oscillates due to the periodic shedding of vortices from the downstream corners. At higher Reynolds, the flow dynamics grows more complicated and ultimately becomes chaotic at high Reynolds. More details on the dynamics and the numerical scheme used to solve (35) can be found in [54].

We consider two control problems for the lid-driven cavity flow:

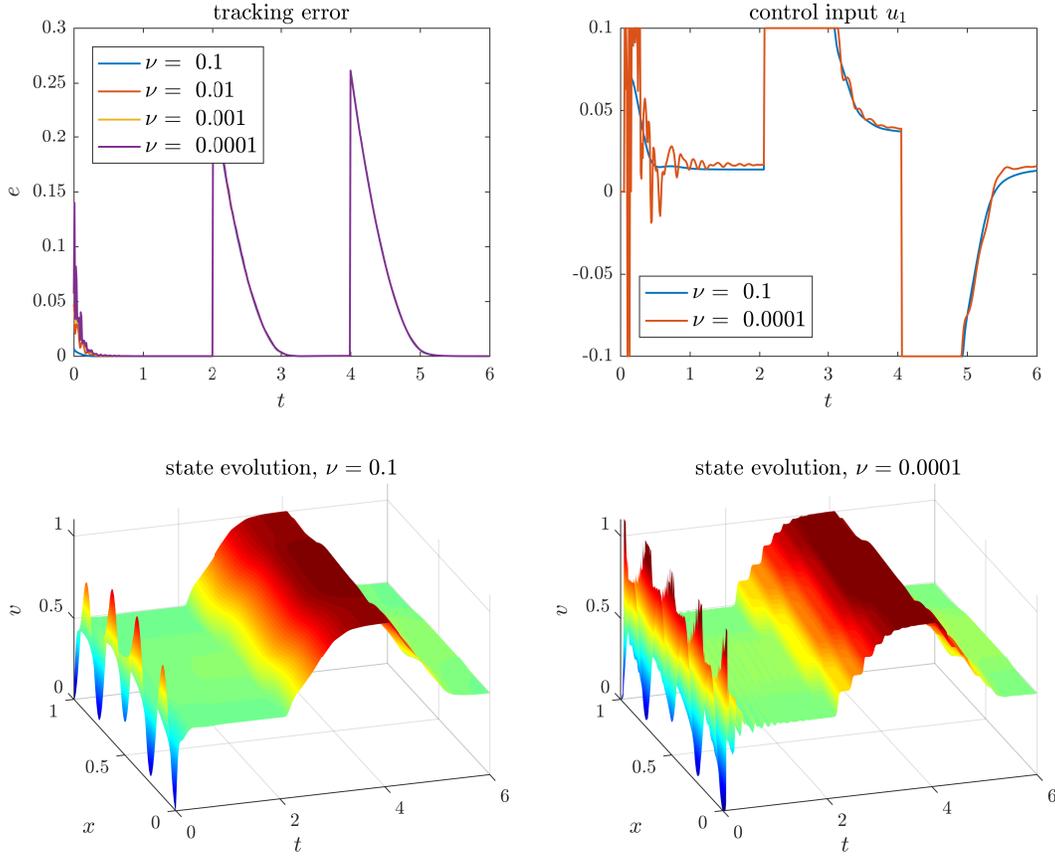


Figure 3: **Robustness of Koopman-MPC with respect to parameter ν .** The tracking error and state evolution for controlling the flow at various values of ν using a controller constructed at $\nu = 0.01$.

- Problem 1) We aim to stabilize the limit cycling flow at $Re = 13000$ around the fixed point solution at $Re = 10000$. This problem has the trivial solution $u_0 = -3/13$, since the effective Re is proportional to the top velocity and $u = u_0$ would set back the flow to the fixed point at $Re = 10000$. To avoid the trivial solution, we use the input constraints $-2/13 < u < 2/13$.
- Problem 2) Using the same input constraints, we try to stabilize the limit cycling flow at $Re = 13000$ around the unstable fixed point solution at the same Re . This problem is specially challenging since the linearization around this fixed point has eigenvalues with positive real part that are not controllable. This implies that the nonlinear system is not stabilizable and there is no linear or nonlinear (regular) feedback control that could achieve the full stabilization [55].

The construction of the Koopman-linear system is similar to the Burgers system; we have used 300 two-second long trajectories of the system with control inputs that are randomly drawn from $[-3/13, 3/13]$. The initial condition for each trajectory is a random convex combination of the stable fixed point at $Re = 10000$, a point on the limit cycle and the

unstable fixed point at $Re = 13000$. The unstable fixed point is computed via the method proposed in [56]. For the full-state observation, we use the values of the stream function on the 50×50 computational grid, the kinetic energy and the constant observable. For the case of sparse measurement, we use the values of stream functions at $k = 2, 5, 50, 100$ random points inside the flow domain, the l^2 vector norm of observed stream function values, and the constant observable. According to section 3.1, the dimension of the state space for the Koopman-linear system built from sparse measurements will be $n = 16, 31, 256, 506$ respectively, which is considerably smaller than the Koopman-linear system with full-state observation ($n = 2502$).

Let ψ_{ref} denote the stream function at the target fixed point. In the case of the full-state measurements, we define the tracking error to be the kinetic energy of the flow distance from the reference state, i.e.,

$$e(t) = e_k(t) := \int_{\Omega} |\mathbf{v}(t) - \mathbf{v}_{\text{ref}}|^2 dz_1 dz_2, \quad (38)$$

where $\mathbf{v} = (\partial\psi/\partial z_2, -\partial\psi/\partial z_1)$ is the velocity field, and Ω is the flow domain.

In the case of sparse measurements, let ψ_s be the vector of stream function measurements. Then the tracking error will be the l^2 -norm of distance from the reference measurements, that is,

$$e(t) = \|\psi_s(t) - \psi_{s,\text{ref}}\|^2. \quad (39)$$

The objective function of the MPC is then given by

$$J = \int_0^T e(t) dt,$$

where the prediction horizon is set to $T = 0.2$. After spatio-temporal discretization, this objective function readily translates to to the form (20) with $N = 20$.

Figure 4 shows the kinetic energy of the state discrepancy (e_k defined in (38)) in applying the Koopman-MPC to problem 1. All the closed-loop simulations start from the same initial condition on the limit cycle. The Koopman-MPC, except for $k = 1, 2$, is successful in considerably reducing the flow distance from the desired state over finite time. The control inputs and the flow evolution for some values of k and full-state observation is shown in fig. 5. In the case of full-state observation, the input signal is mostly saturated at the lower bound which results in a lower effective Re for the flow, and hence getting closer to the fixed point at $Re = 10000$. However, the controller occasionally uses bursts to speed up the stabilization. The effect of these intermittent bursts on the control can be deduced by comparison with the control input with $k = 5$ which is saturated at the lower bound at all times.

Figure 4 also suggests that the control performance of the Koopman-linear systems generally scales with the number of measured observables, i.e., larger number of observables results in better control performance. This indicates that there is a reasonable tradeoff between the sparsity of measurements and the control performance. Moreover, the full-state observation offers less than 10 percent improvement over $k = 50$ in the terminal discrepancy, which indicates that the cavity flow dynamics is approximately low-dimensional and it can

be effectively captured using low-dimensional models from data. We have observed that the choice of measurement location in the flow domain may significantly affect the control performance for very small k , e.g. $k = 1, 2, 5$, and results reported in the figures only represent the typical behavior of controllers built on sparse measurements.

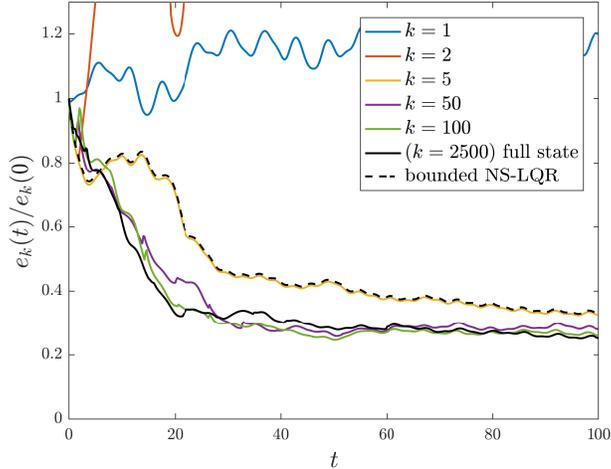


Figure 4: **Control Performance for stabilization around the steady solution at $Re = 10000$.** Normalized kinetic energy of flow distance for cavity flow controllers built by Koopman-MPC with various number of measurements (k), as well as LQR based on local linearization of Navier-Stokes.

A standard technique for flow stabilization is to use linearized Navier-Stokes with linear control strategies. Figure 4 shows the performance of such technique (labeled bounded NS-LQR) in achieving the control objective of the first problem. In this method, the Navier-Stokes equation is linearized around the reference fixed point, and an optimal state feedback gain is computed that would minimize the cost function in (38) over an infinite-time horizon for the linearized system (see Appendix for detail). At each time step, the computed optimal input is bounded by the constraints identical to the MPC setting and then applied to the nonlinear system. This method results in an input signal which is saturated at the lower bound and therefore its performance is identical to the case of Koopman-MPC with $k = 5$ measurements. This method is successful in substantially reducing the tracking error, but unlike the Koopman-MPC framework, it is not capable of exploiting the nonlinearities far from the fixed point to speed up the stabilization. Moreover, this method is model-based and its performance is likely to degrade when uncertainties in estimating fluid properties, or input modeling errors are present.

The performance of the Koopman-MPC framework for stabilization around the fixed point at $Re = 13000$ (problem 2) is shown in fig. 6. Recall that the target fixed point is not stabilizable and no feedback solutions exist that could asymptotically bring the state to the fixed point. Nevertheless, the controllers based on Koopman-linear systems are capable of substantially reducing the tracking error (e.g. down to 40% with $k = 100$). The behavior of controllers is similar to the previous problem, i.e., they tend to decrease the effective Re and use occasional bursts to accelerate the stabilization. An interesting observation is that the controllers built on delay-embedding of measurements perform better than the one

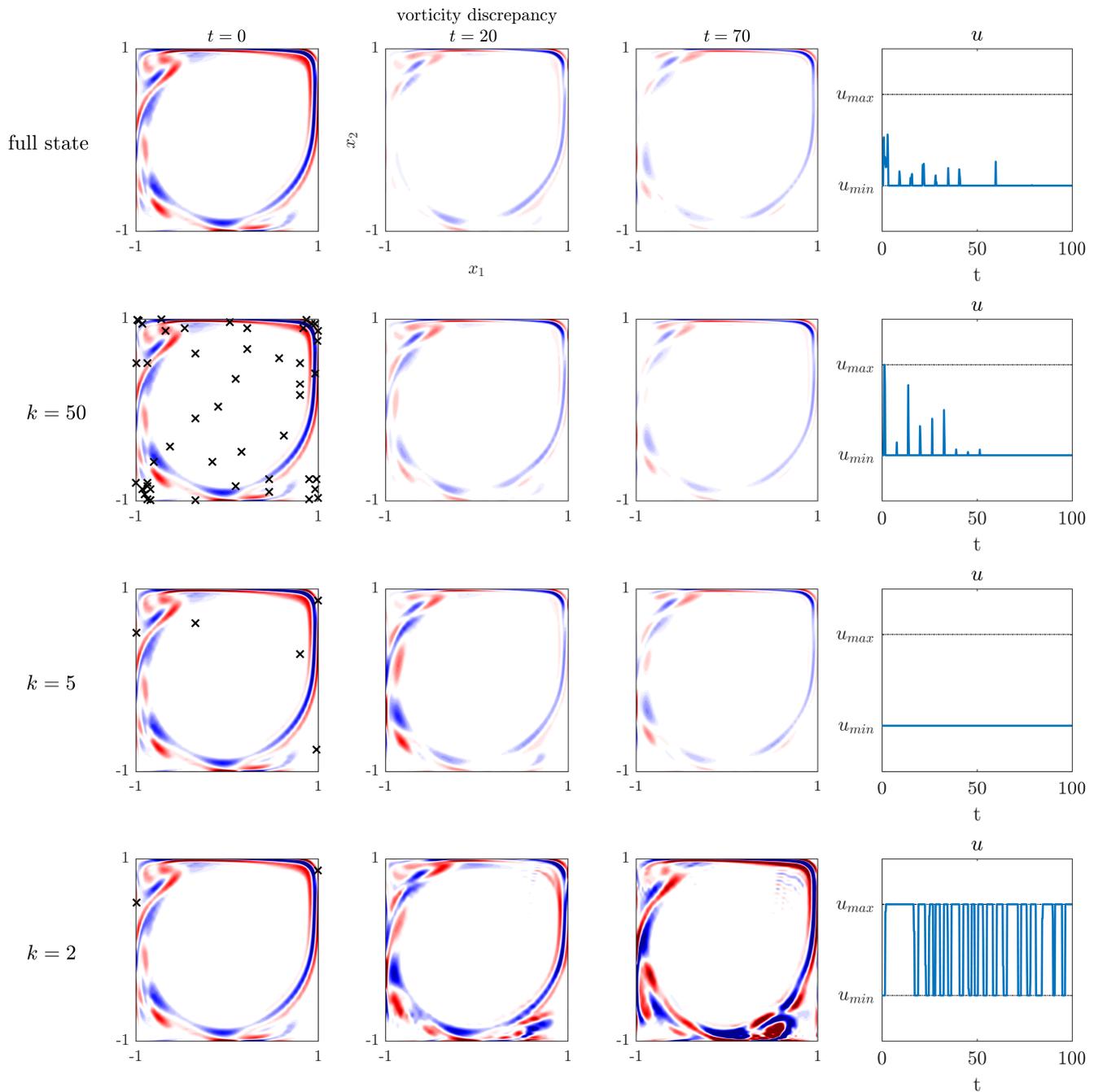


Figure 5: **Closed-loop control of cavity flow evolution with Koopman-MPC.** Discrepancy in the vorticity of controlled state, and input signal, for full-state and sparse measurements. The measurement locations are marked via crosses in the leftmost column. The performance of bounded NS-LQR is identical to the Koopman-MPC with $k = 5$.

with full-state measurements. This shows the effectiveness of using nonlinear observables such as delay-embedded measurements to predict the nonlinear evolution. Note that for this problem, the linearized system around the fixed point is not stabilizable and there is no clear start point for designing feedback control based on linearization techniques commonly used in flow control.

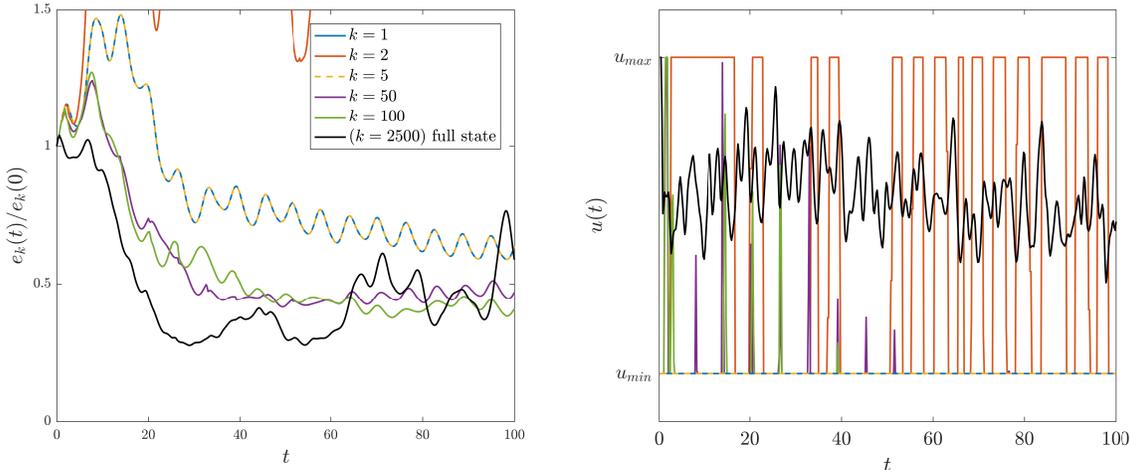


Figure 6: **Control Performance for stabilization around the steady solution at $Re = 13000$:** Normalized kinetic energy of state discrepancy and the input signals for cavity flow controllers built by Koopman-MPC with various number of measurements (k). The steady solution is not stabilizable and there is no optimal feedback solution for LQR.

Computation time Table 1 summarizes the average computational time³ required to evaluate the control input at each time step of the closed-loop operation. We report separately the computation time t_{embed} required to build the state of the Koopman linear system by embedding the available measurements and the time t_{MPC} required to solve the optimization problem (23) in the *dense form*⁴ using the active set qpOASES solver [53]. As evident from the table, combination of the Koopman linear representation and convex quadratic programming of the MPC framework leads to computation of the control input in a fraction of a millisecond. Note that in both examples the bulk of the computation time is spent on embedding the sparse measurements to build the Koopman linear state; this step requires data manipulation carried out purely in MATLAB and could be sped up by a tailored implementation (e.g., in C). Of course, in a real-world implementation of this framework on nonlinear flows, other factors including the time to record and process the physical measurements should also be considered.

³The computations were carried out in MATLAB running on a 3.40 GHz Intel Xeon CPU and 64 GB RAM.

⁴The conversion of the optimization problem (23) to the dense form consist in solving for the state variables (z_1, \dots, z_N) in terms of the control inputs (u_0, \dots, u_{N-1}) and the initial state z_0 using the linear recursion $z^+ = Az + Bu$; the result of this straightforward linear algebra exercise can be found in the appendix of [1].

Table 1: Computation time for Koopman MPC of cavity flow and Burgers equations and NS-LQR control for cavity flow. The symbol “—” signifies a negligible embedding time in the case of full state measurement.

	# of measurements k	embedding dimension n	t_{embed} [sec]	t_{MPC} [sec]
Burgers	10	62	$4.71 \cdot 10^{-5}$	$2.28 \cdot 10^{-7}$
	150	152	—	$2.95 \cdot 10^{-7}$
Cavity	1	11	$1.51 \cdot 10^{-4}$	$8.96 \cdot 10^{-6}$
	2	16	$1.54 \cdot 10^{-4}$	$5.05 \cdot 10^{-5}$
	5	31	$1.52 \cdot 10^{-4}$	$4.19 \cdot 10^{-5}$
	50	256	$1.55 \cdot 10^{-5}$	$3.07 \cdot 10^{-5}$
	100	506	$1.66 \cdot 10^{-4}$	$7.59 \cdot 10^{-5}$
	2500	2502	—	$4.44 \cdot 10^{-6}$
NS-LQR	2500	2501	—	$t_{\text{LQR}} = 6.55 \cdot 10^{-5}$

6 Conclusion and outlook

In this work, we discussed the application of the Koopman-linear MPC framework, first proposed in [1], for data-driven control of nonlinear flows. The key idea is to approximate the Koopman operator from data to obtain finite-dimensional linear systems that approximate the nonlinear global evolution of the system and use these systems as the predictor in the model predictive control framework. The combination of Koopman-linear representation of the dynamics and MPC leads to a convex quadratic programming problem that is solved at each time step; this is accomplished using highly efficient and tailored solvers for linear MPC. Moreover, the proposed framework is based solely on data and therefore robust to uncertainties and errors in available models of the nonlinear system. In the problems considered in this work, the Koopman MPC framework showed superior performance compared to feedback strategies based on local linearization and with sub-millisecond computation time.

An important direction for the future work would be to optimize the data collection process to obtain more accurate and efficient Koopman linear models. This requires addressing two problems: first, finding efficient methods for sampling the extended state space of the nonlinear system; in this work we used random initial condition and random input sequences in the domain of interests to generate data for the EDMD algorithm. The second problem is identifying observables that provide the best finite-dimensional approximation of the Koopman operator in the space of observables. Using machine learning techniques (e.g., [57, 58]) combined with sampling approaches (e.g., [59]) within the Koopman-MPC framework could automatize the choice of observables as well as improve control performance.

Acknowledgements

The authors would like to thank Dr. Sebastian Peitz for a constructive exchange of ideas on the subject. This research was supported in part by the ARO-MURI grant W911NF-17-1-0306, with program managers Dr. Matthew Munson and Dr. Samuel Stanton. The

research of M. Korda was partly supported by the Swiss National Science Foundation under grant P2ELP2165166.

Appendix: Model-based optimal control of cavity flow

In this section, we describe the design of LQR controller for lid-driven cavity flow based on linearization around the steady solutions. Consider the Navier-Stokes equation in (35) written as

$$\frac{\partial}{\partial t} E\psi = f(\psi, u) \quad (40)$$

and let ψ_0 be the fixed-point solution corresponding to the input u_0 , i.e., $f(\psi_0, u_0) = 0$. The linearized Navier-Stokes equations around this equilibrium is given by

$$\frac{\partial}{\partial t} E\tilde{\psi} = A\tilde{\psi} \quad (41)$$

with

$$E := \nabla^2(\cdot),$$

$$A := \frac{1}{Re} \nabla^4(\cdot) - \frac{\partial}{\partial z_2}(\cdot) \frac{\partial}{\partial z_1} \nabla^2 \psi_0 + \frac{\partial}{\partial z_1}(\cdot) \frac{\partial}{\partial z_2} \nabla^2 \psi_0 - \frac{\partial \psi_0}{\partial z_2} \frac{\partial}{\partial z_1} \nabla^2(\cdot) + \frac{\partial \psi_0}{\partial z_1} \frac{\partial}{\partial z_2} \nabla^2(\cdot).$$

and $\tilde{\psi}$ is stream function in the linearized equations. Similar to (37), the control input to the system is the amplitude of the top velocity lid, which results in the following boundary conditions,

$$\tilde{\psi} \Big|_{\partial\Omega} = 0, \quad \frac{\partial \tilde{\psi}}{\partial n} \Big|_{z_1=\pm 1 \text{ or } z_2=-1} = 0, \quad \text{and} \quad \frac{\partial \tilde{\psi}}{\partial z_2} \Big|_{z_2=+1} = u(t)(1 - z_1^2)^2, \quad (42)$$

with \tilde{u} as the deviation from the base input u_0 .

In order to transform the boundary control problem into the standard linear time-invariant (LTI) format, we introduce the extension function

$$H(z_1, z_2) = \frac{1}{4}(1 - z_1^2)^2(1 + z_2)^2(z_2 - 1), \quad (43)$$

and use the change of variables

$$\eta(z_1, z_2) = \tilde{\psi}(z_1, z_2) - H(z_1, z_2)\tilde{u}(t), \quad (44)$$

The linear system in the new variable reads

$$\frac{\partial}{\partial t} E\eta = A\eta + AH\tilde{u} - EH\frac{d\tilde{u}}{dt}, \quad (45)$$

with homogeneous boundary condition,

$$\eta \Big|_{\partial\Omega} = \frac{\partial \eta}{\partial n} \Big|_{\partial\Omega} = 0. \quad (46)$$

The above is in fact an LTI *descriptor* system which can be written as

$$\frac{\partial}{\partial t} \begin{bmatrix} E & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta \\ \tilde{u} \end{bmatrix} = \begin{bmatrix} A & AH \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \tilde{u} \end{bmatrix} + \begin{bmatrix} -EH \\ 1 \end{bmatrix} \frac{d\tilde{u}}{dt}.$$

or in a more compact form,

$$\mathbf{E}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\dot{\tilde{u}}, \quad (47)$$

where $\mathbf{x} = [\eta, \tilde{u}]^\top$ is the embedded state. We are interested in finding the optimal input $d\tilde{u}/dt$ (and \tilde{u}) for the above system that minimizes the cost function,

$$J(\mathbf{x}, \dot{u}) = \int_0^\infty \left[\int_\Omega |\mathbf{v}|^2 dA + \alpha_1 \tilde{u}^2 + \alpha_2 (\dot{\tilde{u}})^2 \right] dt \quad (48)$$

where $\mathbf{v} = (\partial\tilde{\psi}/\partial z_2, -\partial\tilde{\psi}/\partial z_1)$ is the velocity field of the linearized system.

We have used the Chebyshev collocation scheme [60] to spatially discretize the linear system in (47) and the cost function in (48). We have chosen $\alpha_1 = \alpha_2 = 10^{-6}$ to minimally penalize the input and avoid infinitely large solutions. If the linear system is stabilizable (for example in the case of steady solution at $Re = 10000$), solving the continuous-time algebraic Riccati equation (ARE) (see e.g. [61] for descriptor formulation of ARE), would give the optimal feedback gain $k = [k_\eta \ k_u]$. The LQR optimal input \tilde{u} is then computed by time stepping the following ordinary differential equation

$$\dot{\tilde{u}} = -k_u u - k_\eta \eta = -k_u \tilde{u} - k_\eta (\tilde{\psi} - H\tilde{u}),$$

and the input $u = u_0 + \tilde{u}$ is applied to the nonlinear system. If the fixed point is not linearly stabilizable, such as the steady solution at $Re = 13000$, then ARE does not have a solution and there is no stabilizing input.

References

- [1] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.
- [2] John Kim. Physics and control of wall turbulence for drag reduction. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 369(1940):1396–1411, 2011.
- [3] Steven L Brunton and Bernd R Noack. Closed-loop turbulence control: progress and challenges. *Applied Mechanics Reviews*, 67(5):050801, 2015.
- [4] Thomas R Bewley. Flow control: new challenges for a new renaissance. *Progress in Aerospace sciences*, 37(1):21–58, 2001.
- [5] Thomas R Bewley, Parviz Moin, and Roger Temam. DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *Journal of Fluid Mechanics*, 447:179–225, 2001.

- [6] John Kim and Thomas R Bewley. A linear systems approach to flow control. *Annu. Rev. Fluid Mech.*, 39:383–417, 2007.
- [7] Philip Holmes, John L Lumley, Gal Berkooz, and Clarence Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.
- [8] Bernd R Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [9] Maciej J Balajewicz, Earl H Dowell, and Bernd R Noack. Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier–stokes equation. *Journal of Fluid Mechanics*, 729:285–308, 2013.
- [10] Karen Willcox and Jaime Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA journal*, 40(11):2323–2330, 2002.
- [11] CW Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.
- [12] Randolph H Cabell, Michael A Kegerise, David E Cox, and Gary P Gibbs. Experimental feedback control of flow-induced cavity tones. *AIAA journal*, 44(8):1807–1816, 2006.
- [13] Steven L Brunton, Clarence W Rowley, and David R Williams. Reduced-order unsteady aerodynamic models at low reynolds numbers. *Journal of Fluid Mechanics*, 724:203–233, 2013.
- [14] Shao-Ching Huang and John Kim. Control and system identification of a separated flow. *Physics of Fluids*, 20(10):101509, 2008.
- [15] Aurelien Hervé, Denis Sipp, Peter J Schmid, and Manuel Samuelides. A physics-based approach to flow control using system identification. *Journal of Fluid Mechanics*, 702:26–58, 2012.
- [16] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
- [17] Igor Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.
- [18] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.
- [19] B. O. Koopman and J. von Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255, 1932.
- [20] Igor Mezić and Andrzej Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1):101–133, 2004.

- [21] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- [22] C.W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D.S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641(1):115–127, 2009.
- [23] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [24] Peter J Schmid, L Li, MP Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.
- [25] Jia-Chen Hua, Gemunu H Gunaratne, Douglas G Talley, James R Gord, and Suresh Roy. Dynamic-mode decomposition based analysis of shear coaxial jets with and without transverse acoustic driving. *Journal of Fluid Mechanics*, 790:5–32, 2016.
- [26] Shervin Bagheri. Koopman-mode decomposition of the cylinder wake. *J. Fluid Mech.*, 726:596–623, 2013.
- [27] Taraneh Sayadi, Peter J Schmid, Joseph W Nichols, and Parviz Moin. Reduced-order representation of near-wall structures in the late transitional boundary layer. *Journal of Fluid Mechanics*, 748:278–301, 2014.
- [28] Pramod K Subbareddy, Matthew D Bartkowicz, and Graham V Candler. Direct numerical simulation of high-speed transition due to an isolated roughness element. *Journal of Fluid Mechanics*, 748:848–878, 2014.
- [29] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Generalizing Koopman theory to allow for inputs and control. *arXiv preprint arXiv:1602.07647*, 2016.
- [30] Matthew O. Williams, Maziar S. Hemati, Scott T. M. Dawson, Ioannis G. Kevrekidis, and Clarence W. Rowley. Extending data-driven Koopman analysis to actuated systems. In *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2016.
- [31] Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.
- [32] Sebastian Peitz and Stefan Klus. Koopman operator-based model reduction for switched-system control of pdes. *arXiv preprint arXiv:1710.06759*, 2017.
- [33] Sebastian Peitz. Controlling nonlinear pdes using low-dimensional bilinear approximations obtained from data. *arXiv preprint arXiv:1801.06419*, 2018.
- [34] Bryan Glaz, Igor Mezić, Maria Fonoberova, and Sophie Loire. Quasi-periodic intermittency in oscillating cylinder flow. *Journal of Fluid Mechanics*, 828:680–707, 2017.
- [35] Aivar Sootla, Alexandre Mauroy, and Damien Ernst. An optimal control formulation of pulse-based control using koopman operator. *arXiv preprint arXiv:1707.08462*, 2017.

- [36] Alexandre Mauroy and Jorge Goncalves. Koopman-based lifting techniques for nonlinear systems identification. *arXiv preprint arXiv:1709.02003*, 2017.
- [37] Amit Surana and Andrzej Banaszuk. Linear observer synthesis for nonlinear systems using Koopman operator framework. In *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2016.
- [38] Hassan Arbabi and Igor Mezić. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.
- [39] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, pages 1–24, 2017.
- [40] Lennart Ljung. System identification. In *Signal Analysis and Prediction*, pages 163–173. Springer, 1998.
- [41] Dag Tjøstheim and Bjørn H Auestad. Nonparametric identification of nonlinear time series: projections. *Journal of the American Statistical Association*, 89(428):1398–1409, 1994.
- [42] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical systems and turbulence, Warwick 1980*, pages 366–381. Springer, 1981.
- [43] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1:391–421, 2014.
- [44] Milan Korda, Mihai Putinar, and Igor Mezić. Data-driven spectral analysis of the Koopman operator. *arXiv preprint arXiv:1710.06532*, 2017.
- [45] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism a). *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.
- [46] Alexandre Mauroy and Igor Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016.
- [47] Igor Mezić. On applications of the spectral theory of the Koopman operator in dynamical systems and control theory. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 7034–7041. IEEE, 2015.
- [48] Alexandre Mauroy and Igor Mezić. On the use of fourier averages to compute the global isochrons of (quasi) periodic dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(3):033112, 2012.
- [49] Igor Mezić. Koopman operator spectrum and data analysis. *arXiv preprint arXiv:1702.07597*, 2017.

- [50] Dimitrios Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Applied and Computational Harmonic Analysis*, 2017.
- [51] David Q Mayne, James B Rawlings, Christopher V Rao, and Pierre OM Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [52] Lars Grüne and Jürgen Pannek. Nonlinear model predictive control. In *Nonlinear Model Predictive Control*. Springer, 2011.
- [53] Hans Joachim Ferreau, Christian Kirches, Andreas Potschka, Hans Georg Bock, and Moritz Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.
- [54] Hassan Arbabi and Igor Mezić. Study of dynamics in post-transient flows using Koopman mode decomposition. *Phys. Rev. Fluids*, 2:124402, 2017.
- [55] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media, 2013.
- [56] Bastien E Jordi, Colin J Cotter, and Spencer J Sherwin. Encapsulated formulation of the selective frequency damping method. *Physics of Fluids*, 26(3):034101, 2014.
- [57] Enoch Yeung, Soumya Kundu, and Nathan Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. *arXiv preprint arXiv:1708.06850*, 2017.
- [58] Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.
- [59] Ryan Mohr and Igor Mezić. Construction of eigenfunctions for scalar-type operators via laplace averages with connections to the Koopman operator. *arXiv preprint arXiv:1403.6559*, 2014.
- [60] Lloyd N Trefethen. *Spectral methods in MATLAB*, volume 10. Siam, 2000.
- [61] William F Arnold and Alan J Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proceedings of the IEEE*, 72(12):1746–1754, 1984.