

Koopman-based model predictive control with morphing surface: Regulating the flutter response of a foil with an active flap

Tso-Kang Wang  and Kourosh Shoele ^{*}*Department of Mechanical Engineering, Joint College of Engineering Florida State University–Florida A&M University, Tallahassee, Florida 32310, USA*

(Received 11 May 2023; accepted 5 December 2023; published 24 January 2024)

Active flow control can achieve substantial performance gains and meet the challenges of next-generation air vehicles and energy-harvesting devices. The use of active flow control techniques with the moving flap or morphing surfaces has been shown to be a viable path to regulating the flow-induced vibration of the foil. However, due to the complex nature of flow over morphing surfaces, all physical phenomena are intertwined, which prevents a clear understanding of the underlying flow physics and, therefore, a successful design of a controlling action to optimally modify them. In this research an active flow control framework with the model predictive control theory is proposed to modulate the flow-induced flutter of a foil using the morphing flap surface. The geometrically weighted dynamic-relevant modes are used to build surrogate models to achieve rapid model-based active control of complex systems. It is shown that the flap is capable of both facilitating and eliminating fluid-induced vibrations by regulating the lift forces exerted on the foil. Furthermore, the control framework provides full knowledge of how the structure modifies the flow and has the potential to identify the ambient environmental change simultaneously.

DOI: [10.1103/PhysRevFluids.9.014702](https://doi.org/10.1103/PhysRevFluids.9.014702)

I. INTRODUCTION

Fluid-structure coupled phenomena such as flutter or limit cycle oscillations have an impact on various engineering systems, such as the stability of the aircraft or the performance of wind turbines and hydrofoil energy extractors. Hence, regulating such phenomena has been a central topic in the fluid community. However, due to the complex nonlinear nature of the problem, the traditional control theories accustomed to linearized systems often fail to provide a simple control law. To capture the aeroelastic responses induced by the shedding of the leading-edge vortex and other unsteady vortex separations along the body, different nonlinear models have been proposed to capture the essential nonlinear responses, such as the model developed by Goman and Khrabrov [1], the ONERA model [2], and many other models [3]. Some of these models also incorporate the effect of morphing flaps, such as the ones proposed by Librescu *et al.* [4] or Block and Stragnac [5]. Following these models, control strategies have been designed to mitigate or regulate the foil response. For example, Wang *et al.* [6] used a multi-input system with active control surfaces at both leading and trailing edges and designed a full-state feedforward-feedback controller with a high-gain observer; Zhang *et al.* [7] adopted an adaptive control scheme to account for unsteady flow and eliminate vibration with a multi-DoF trailing edge flap; Lee and Singh [8] designed a robust sliding-mode control that considers only the variables within finite horizon to suppress the unwanted oscillation; and Pohl *et al.* [9] mitigated the gust impact with a feedforward-feedback control strategy designed with a simplified ONERA model. Experiments such as that carried out by

^{*}kshoele@eng.famu.fsu.edu

Herrmann *et al.* [10] or Platanitis *et al.* [11] proved that these control strategies designed with experimentally tuned models work well under the intended working conditions.

Many of the models and control strategies developed so far have been based on simplified aerodynamic loading models stemming from Theodorsen's theory [12], where the airfoil is often seen as a flat plate, and the vortex shedding and local separation effects are captured through nonlinear force coefficients. Theodorsen's theory is an algebraic model based on the potential flow theory, where a low Reynolds number and small deflection of the foil or flap are assumed. These simplified models allow quick iteration and provide a test bed for different control strategies. Their simple structure is also ideal for incorporating other effects related to unsteady disturbances such as flow fluctuations or impending gusts. However, the assumption of small deflection about the equilibrium point can be rendered invalid in many practical highly nonlinear systems. Also, these models rely on experimentally tuned coefficients and often fail to respond properly to environmental condition changes [13]. Moreover, the lack of complete knowledge of the actual flow pattern induced by the interaction between the foil, flap, and ambient flow makes it difficult to fully understand how the actuation regulates the flow system. Some of these issues can be improved by increasing the degrees of freedom of the model or switching between different models, but along with it, the complexity of the model also rises and overfitting or longer evaluation time becomes a problem. Overall, the models designed based on Theodorsen's theory and its extensions have been proven to be effective when limited to specific settings but are not able to provide detailed physical insight into how the control implemented impacts the fluid around the structure.

Recently there has been a surge of modern control methods for nonlinear high-dimensional dynamical systems utilizing modal analysis techniques. The key is building and updating reduced-order models rapidly with modal information, which can then be used to design control strategies that adapt to environmental change. The data-driven nature of the modal analysis techniques allow the model construction to be flexible with different data set and provide ample design possibility due to its relatively simple mathematical representation compared to the aforementioned nonlinear simplified models based on Theodorsen's theory. One major branch of this is using the modal analysis techniques to construct state space models via Galerkin projection of the Navier-Stokes equations onto the orthogonal vector space identified by proper orthogonal decomposition (POD) [14–16]. POD decomposes the data into orthogonal subspaces spanning the data space, and Galerkin projection can determine the dominant dynamics in each direction by orthogonal projection of the governing equations. In the case of the Navier-Stokes equation, this procedure transformed it into a set of ODEs that can be solved to acquire the coefficients for the state space model. An extension to this method is identifying linear input-output systems with balanced POD that considers the input with balanced truncation [17,18]. These methods have shown great results in modeling and controlling complex flows.

Another branch of methods that has become popular in recent years is based on the Koopman operator theory [19,20]. This is the route we took in this research since the Koopman operator is fully data-driven (while Galerkin projection requires knowledge about the governing equation), and the model is easily scalable. Furthermore, the Koopman theory captures the system's dynamics, which is straightforward in controller design, and the operator is linear, making it easy to manipulate the model. These Koopman theory-based reduced-order models are good candidates for the model predictive control (MPC) framework [21–24], which is a receding horizon-type control scheme that is widely adopted in various disciplines. The usage of MPC with Koopman theory was introduced into the fluid community very recently and has shown promising outcomes controlling chaotic systems such as the wall-driven cavity flow [25], fluidic pinball [26], or simplified aeroelastic models [27]. Here the MPC scheme along with DMD with control (DMDc) reduced-order model is used to implement feedback closed-loop control of the flow-induced fluttering of a foil with an active flap as a controller. In the authors' previous publications, the geometrical information of the deforming solid is introduced into the fluidic motion in the form of spatial stretching [28], and this methodology will be combined with DMDc to incorporate the effect of the morphing surface into identifying the reduced order model. This will provide additional knowledge on how the flap affects

the flow around it when actively actuated compared to the simplified models based on Theodorsen's theory.

In the following sections, we will introduce the background of the Koopman operator, how we can approximate it with finite-dimensional DMD algorithms, and how we can perform system identification with it to build high-accuracy state-space model for our foil-and-flap system in Sec. II. The procedure to implement the model into a MPC framework to control complex dynamical systems to regulate the heaving motion of the foil-flap system and the results will be discussed in Sec. III. Finally, we will give some remarks and future directions for this research in Sec. IV.

II. DATA-DRIVEN MODELING OF FOIL-AND-FLAP SYSTEM

In this section we will introduce the principle of the Koopman operator, how we can approximate it in a data-driven fashion with DMD, and then introduce the control input into the DMD to form state-space models. In the second section, we will discuss how we used this procedure to produce state space model of the foil-and-flap system and then evaluate the accuracy of the model.

A. Koopman operator theory and data-driven system identification

We will review the basics of the Koopman operator for discrete-time dynamical systems to align with the time-stepping simulations and DMD. Please note that many reviews, e.g., [20,29], can be referred to for a more thorough discussion on the Koopman operator, including the continuous description.

Let us consider a dynamical system $x^+ = \mathcal{T}_t(x)$ defined on a state space M , where the discrete function \mathcal{T}_t is characterized with the time t . We can call any function $g : M \rightarrow \mathbb{R}$ an observable of the system. The Koopman operator \mathcal{K}_t is a linear transformation acting on the vector space of the observables given by

$$(\mathcal{K}_t g)(x) = g(\mathcal{T}_t(x)), \quad (1)$$

wherein the Koopman operator is essentially an infinite-dimension operator that updates the observable g based on the evolution of the trajectories in the state space. In other words the Koopman operator captures the *dynamics* of the system. To cast this into a discrete-time description, consider the function to be a t -fold composition of the single time step operator \mathcal{T} , $\mathcal{T}_t(x) = \mathcal{T}(\mathcal{T}(\mathcal{T} \dots \mathcal{T}(x)))$, and likewise the discrete Koopman operator \mathcal{K} is defined as

$$g_{k+1} = \mathcal{K}(g_k). \quad (2)$$

The most intriguing property of the Koopman operator that makes it popular is that it is a *linear* operator as long as the observable space \mathcal{G} consists of the observables g is linear:

$$\begin{aligned} \mathcal{K}[\alpha g_1(x) + \beta g_2(x)] &= \alpha g_1(\mathcal{T}(x)) + \beta g_2(\mathcal{T}(x)) \\ &= \alpha \mathcal{K}(g_1(x)) + \beta \mathcal{K}(g_2(x)), \end{aligned} \quad (3)$$

and more importantly, this property holds whether the original system \mathcal{T} is linear or not, which means that the Koopman operator can approximate a nonlinear system with a linear state space.

The Koopman eigenfunctions $\psi(x)$ corresponding to the eigenvalues λ are defined as

$$\psi(x_{k+1}) = \mathcal{K} \psi(x_k) = \lambda \psi(x_k), \quad (4)$$

and with this relation, one can rewrite the observables formed as linear combinations of the Koopman eigenfunctions under the Koopman operator by

$$g(x) = \sum_k v_k \psi_k \Rightarrow \mathcal{K}_t g(x) = \sum_k v_k \lambda_k^t \psi_k. \quad (5)$$

In practice, we often need to handle multiple measurements at once. For example, computational fluid dynamic (CFD) simulations provide refined spatial information about the flow velocity and

pressure; experimentally, particle image velocimetry calculates the flow velocity by correlating the movement of particles in the flow. To expand Koopman decomposition into multiple measurements of a system, we can arrange them into a vector \mathbf{g} :

$$\mathbf{g} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_p(x) \end{bmatrix}, \quad (6)$$

and each of these observables can be expanded with the eigenfunctions in the form of Eq. (5):

$$g_i(x) = \sum_{j=1}^{\infty} v_{ij} \psi_j(x). \quad (7)$$

The vector of measurements can then be expanded in matrix form as

$$\mathbf{g} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_p(x) \end{bmatrix} = \sum_{j=1}^{\infty} \psi_j(x) \mathbf{v}_j, \quad (8)$$

where \mathbf{v}_j consists of the coefficients v_{ij} and is called the *Koopman mode*. The evolution of the multiple measurements can then be decomposed as

$$\mathbf{g}(x_t) = \mathcal{K}_t \mathbf{g}(x_0) = \sum_{j=1}^{\infty} \mathcal{K}_t \psi_j(x_0) \mathbf{v}_j = \sum_{j=1}^{\infty} \lambda_j^t \psi_j(x_0) \mathbf{v}_j. \quad (9)$$

This process is called the *Koopman mode decomposition* [30]. Often this process is approximated with the dynamic mode decomposition (DMD). We will now provide a short introduction to DMD.

Dynamic mode decomposition (DMD), first proposed by Schmid [31], decomposes data into modes and corresponding characteristic frequencies. Consider a data series $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, which can be partitioned into two time-consecutive sets $\mathbf{X}_{1:m-1} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{m-1}\}$ and $\mathbf{X}_{2:m} = \{\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_m\}$ (which don't have to be strictly one frame apart), DMD aims to find the best-fit revolution, or *dynamics*, between the two sets, i.e., find the best-fit matrix \mathbf{A} that approximates

$$\mathbf{X}_{2:m} = \mathbf{A} \mathbf{X}_{1:m-1}. \quad (10)$$

The most common best-fit definition is the minimum Frobenius norm

$$C = \sum_{i=1}^{m-1} \|\mathbf{x}_{2,i} - \mathbf{A} \mathbf{x}_{1,i}\|^2, \quad (11)$$

where $\mathbf{x}_{1,i}$ and $\mathbf{x}_{2,i}$ are the i th columns of the matrices $\mathbf{X}_{1:m-1}$ and $\mathbf{X}_{2:m}$, respectively. The solution to this problem is

$$\mathbf{A} = \mathbf{X}_{2:m} \mathbf{X}_{1:m-1}^+, \quad (12)$$

where $\mathbf{X}_{1:m-1}^+ = (\mathbf{X}_{1:m-1}^* \mathbf{X}_{1:m-1})^{-1} \mathbf{X}_{1:m-1}^*$ is the Moore-Penrose left inverse of the matrix $\mathbf{X}_{1:m-1}$. The eigenvectors of \mathbf{A} are defined as the DMD modes ψ_j , and the frequencies and growth and decay rates for these modes are the imaginary and real parts of the corresponding eigenvalues μ_j , respectively.

A system can be represented with the low-rank projected solution reconstructed at time t with the decomposition

$$\mathbf{x}_{\text{DMD}}(t) = \sum_{k=1}^K b_k(0) \psi_k e^{\mu_k t}, \quad (13)$$

where K is the reduced approximation rank and $b_k(0)$ is the initial amplitude of the k th mode. This equation can be represented in matrix form:

$$\mathbf{x}_{\text{DMD}}(t) = \Psi \text{diag}(e^{\mu_j t}) \mathbf{b}, \quad (14)$$

where Ψ is the matrix consists of DMD modes ψ , $\text{diag}(e^{\mu_j t})$ has $e^{\mu_j t}$ as diagonal entries, and \mathbf{b} is a vector calculated by

$$\mathbf{b} = \Psi^+ x_1, \quad (15)$$

where x_1 is the initial snapshot and \mathbf{b} is a vector formed from the initial amplitude $b_k(0)$.

We can now see the similarity between the Koopman mode decomposition and the DMD comparing Eqs. (9) and (13): both operators approximate the dynamics of a nonlinear system with the combination of linear modes. In fact, DMD is the finite-dimensional approximation of the Koopman operator! The DMD eigenvalues, DMD modes, and DMD mode amplitudes are the finite-dimensional approximations of the Koopman eigenvalues λ_j , Koopman modes v_j , and Koopman eigenfunctions $\psi_j(x_0)$, respectively. Hence, it is common to adopt DMD for the representation of system's dynamics due to its simplicity of operation and linear nature. Readers are recommended to refer to Mezić [20], Bagheri [32], and Tu *et al.* [33] for more details on the connections between DMD and Koopman mode decomposition. We will adopt DMD, especially the variant of DMD with control proposed by Proctor *et al.* [34], as our tool for modeling the system dynamics in the following sections.

B. Dynamic mode decomposition with control

For the purpose of developing a control strategy for nonlinear systems, it is beneficial to approximate the system with linear systems utilizing the Koopman operators (or the approximated versions). Furthermore, advanced methods are developed to differentiate the effects of the autonomous system response and control input, such as the DMD with control (DMDc) [34], extended DMD with control [35], interpolated Koopman generators [26], among others. From the research on the selection of basis functions conducted in literature [29,36], we decided that DMDc with delayed embedded coordinates is a suitable candidate for deriving the linear model for our foil-and-flap system along with the embedded geometrical weighting for its accuracy and data efficiency shown in modeling various dynamic systems.

DMDc, proposed by Proctor *et al.* [34], aims to extend DMD to actuated systems. On top of the dynamical system described in Sec. II A, where for the measured state \mathbf{x} the evolution is captured with $\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k$, additionally, we now have a control input \mathbf{u} and the dynamical model of the system described with

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k. \quad (16)$$

If we organize the system states, or measurements, and the control input sequences into matrices

$$\mathcal{X}' = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \\ | & | & \cdots & | \end{bmatrix}, \quad \mathcal{X} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & \cdots & | \end{bmatrix} \quad (17)$$

and

$$\mathcal{Y} = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{m-1} \\ | & | & \cdots & | \end{bmatrix}, \quad (18)$$

then the problem of finding the best-fit \mathbf{A} and \mathbf{B} matrices for the system

$$\mathcal{X}' \approx \mathbf{A}\mathcal{X} + \mathbf{B}\mathcal{Y} \quad (19)$$

can be recast into solving the below least-squares optimization problem:

$$\min_{[\mathbf{A} \ \mathbf{B}]} \left\| \mathcal{X}' - [\mathbf{A} \ \mathbf{B}] \begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \end{bmatrix} \right\|_2^2. \quad (20)$$

This problem conveniently has a solution given by

$$[\mathbf{A} \ \mathbf{B}] = \mathcal{X}' \begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \end{bmatrix}^+, \quad (21)$$

where the subscript $+$ again represents the Moore-Penrose left inverse. The DMDC provides a pure data-driven procedure to perform system identification, even when the data are acquired with undersampled measurement or from highly nonlinear systems [37]. We can increase the identification accuracy even more by adopting the delay embedding technique, which is a classic technique to increase the DoF of the identified systems and hence increase the accuracy [38–40]. For example, if a set of observables (e.g., measurements at different spatial locations) x^1, x^2, \dots, x^p at time t_k can be represented as $\mathbf{g}(t_k) = [x^1(t_k), x^2(t_k), \dots, x^p(t_k)]$, then the time-delayed vector of time step n_d can be constructed as $\mathbf{z}_k = [\mathbf{g}(t_{k-n_d}), \mathbf{g}(t_{k-n_d+1}), \dots, \mathbf{g}(t_k)]^T$, and the data matrix is now

$$\mathcal{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{z}_2 & \mathbf{z}_3 & \cdots & \mathbf{z}_m \\ | & | & & | \end{bmatrix}, \quad \mathcal{X} = \begin{bmatrix} | & | & & | \\ \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_{m-1} \\ | & | & & | \end{bmatrix}, \quad (22)$$

and the system identification of the dynamical system is now finding the best-fit (\mathbf{A}, \mathbf{B}) pairs that approximate

$$\mathcal{X}' \approx \mathbf{A} \mathcal{X} + \mathbf{B} \mathcal{Y}. \quad (23)$$

The above relation has the same solution as Eq. (21). This operation allows better tracking in long-term system dynamics and increases the model DoF for low-data scenarios such as limited available sensors. In the following section, we will show how we utilize this procedure to build a dynamic model for the dynamic foil-and-flap system to capture the response to different flap actuation.

C. DMDC model for foil-and-flap system

To demonstrate how the DMDC model works, we will be looking at a heaving airfoil with an active flap. A previous publication by the authors [41] has shown extensive research on how the frequencies and amplitudes of the sinusoidally oscillating flap affect the foil heaving motion. The foil-and-flap system is depicted in Fig. 1(a), with a set of translational spring-and-damper representing the aeroelastic characteristic of the foil. In the current study, the angle of attack (AoA) of the foil is fixed at 10° , and the foil is allowed to heave freely. The Reynolds number based on the chord length is set to be $\text{Re} = \frac{U_\infty L}{\nu} = 1000$, which is sufficiently large to allow the complex vortex shedding that leads to fluttering phenomena at this AoA setting. The numerical fluid-structure interaction solver, which is explained in detail and validated in the cited paper, solves the flow equations in the conformally transformed domain which provides a natural geometric weighting that represents the solid deformation [28]. This procedure offers a means to apply DMDC to flow around a deforming object. The results shown in later sections encode the geometrical information into the identified dynamic systems naturally by performing system identification in the transformed domain. Note that although this detail fades into the background as we will focus on the control aspect more in this paper, without this transformation, DMDC and its variants do not work with deforming volumetric structures as they are data-driven and don't have spatial recognition. Readers could refer to the previous publications for more details on this geometrically weighted modal decomposition (GW-MD) technique, how the authors used it to look into the foil-and-flap system, and the detailed simulation setup [28,41].

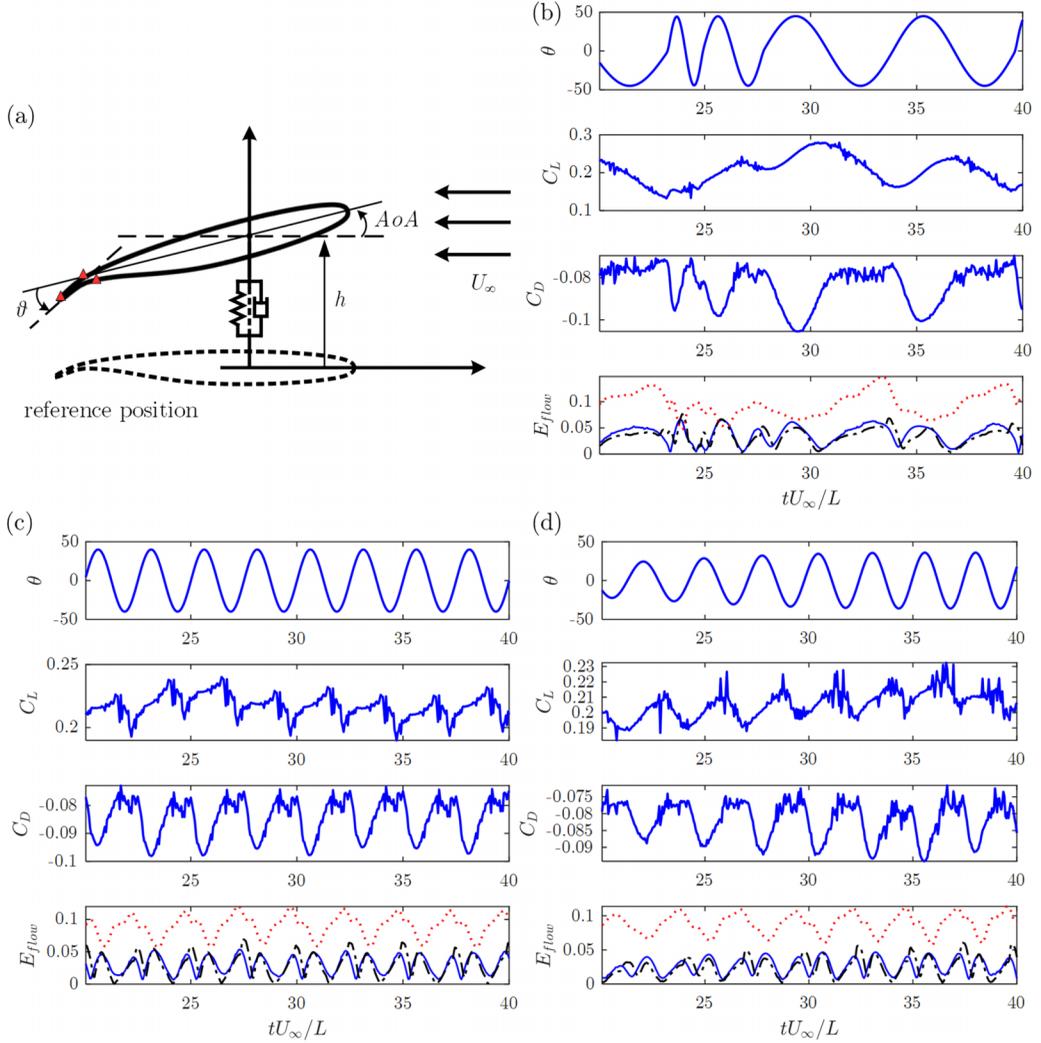


FIG. 1. (a) Numerical model and the sampling points for the flow kinetic energy. Input flap angle trajectories and corresponding lift coefficients, drag coefficients, and flow kinetic energy at specified sampling points for (b) random frequency, (c) single-frequency sinusoidal, and (d) transitional frequency and amplitude oscillating case. For flow kinetic energy, blue solid line, top; black broken line: trailing edge; red dotted line: bottom sampling point.

In the previous research, it is observed that with different actuation amplitude and frequency, at moderate angles of attack, the foil can heave with either periodic, quasiperiodic, or chaotic motion patterns. Also through GW-MD, we proved that the flap induces flow structures that interact with the wake forming from the pitching leading edge, which is the primary mechanism the flap affects the fluid-induced vibration. We will now show how the active flow control can be employed by actively commanding the flap motion to regulate the heaving motion, or more directly regulating the lift force acting on the foil through a data-driven modeling technique. The first step to that goal is to build a reduced-order model of the system with DMDC, so we can design a control strategy later on the identified linear system.

The working principle of using the Koopman-based operators like DMDC to conduct system identification is finding the system trajectory evolution by observing the data collected from

TABLE I. Tracking error of the DMDc model constructed with different parameters.

Input type	Trajectory type and count	Delayed dimension	Random	Single frequency	Transitional
θ	Random, 10	10	11.48%	15.91%	17.6%
θ	Random, 10	20	10.37%	10.87%	17.35%
θ	Random, 10	40	8.41%	7.76%	16.17%
θ	Random, 10	80	8.27%	5.23%	11.02%
θ	Random, 5	80	7.92%	5.92%	10.6%
θ	Mixed, 10	80	10.48%	4.62%	12.12%
θ & $\dot{\theta}$	Random, 10	80	7.68%	5.65%	9.84%
θ & $\dot{\theta}$	Mixed, 15	80	8.92%	4.14%	9.83%

actuating the system with different inputs. A diverse input set can be seen as moving the starting point of a system trajectory to a different spot in the response phase space. Assuming there is a single attracting limit cycle or sink, eventually every trajectory would converge to them and not much data will be needed to reconstruct the system; but when there are multiple attractors, distributed starting points are required to observe more of the system dynamics. Hence, the first step to building the data-driven state-space model for our foil-and-flap system is to actuate the flap in different settings.

Three types of flap angle input are prescribed to the active flap: random input trajectories, each consisting of 25 cycles of sinusoidal flapping motion with the same amplitude but randomized frequencies from $St_f = [0.05, 1.5]$; the periodic trajectories with fixed oscillating amplitude and frequency chosen from $St_f = [0.1, 1.5]$; and the transitional trajectories having different combinations of smoothly changing amplitude and frequencies. The lift, drag, heaving motion, and flow field information are recorded for each input trajectory. Figure 1 demonstrates three different types of input flap angle trajectory and the corresponding lift coefficient, drag coefficient, and flow kinetic energy recorded at specified points on the foil shown in Fig. 1(a). The flow kinetic energy is collected at the foil surface, as this is comparable to the actual application scenarios where the sensors embedded in the foil are generally limited to detecting the flow states adjacent to the foil. While incorporating more flow states might enhance predictive performance, expanding the states could result in potential overfitting and increased model complexity. To balance applicability and model efficiency, we opted to constrain the states to the current selection. It is important to emphasize that the flow kinetic energy, in this context, is derived from the geometrically weighted space, encompassing both structural deformation and flow velocity. This formulation results in a geometrically weighted (GW) DMDc. For the sake of conciseness, we will omit the GW prefix in this paper.

The calculated responses for different flap trajectories are collected to build the state-space model with DMDc. Four parameters could potentially affect the model accuracy: the trajectory type, the number of trajectories used for identification, the number of delay-embedded time steps, and the type of input included. Table I reports the tracking error of using models constructed by different parameter combinations to track the system evolution with the same set of initial conditions. For each trajectory type, three different initial conditions and input sequences (flap angle θ or both θ and $\dot{\theta}$) not in the training data are used to evaluate the tracking error, which is defined as

$$TE = \text{rms} \left[\frac{C_L^{\text{ref}} - C_L^{\text{model}}}{C_L^{\text{ref}}} \right], \quad (24)$$

where C_L^{ref} is the lift coefficient recorded from the response of the full-scale CFD simulation of the foil-and-flap plant, and C_L^{model} is recorded from the DMDc model with the same input sequence and initial state.

From Table I we can make some key observations, demonstrated graphically in Fig. 2. First, increasing the number of delayed time steps vastly improves the tracking performance, where

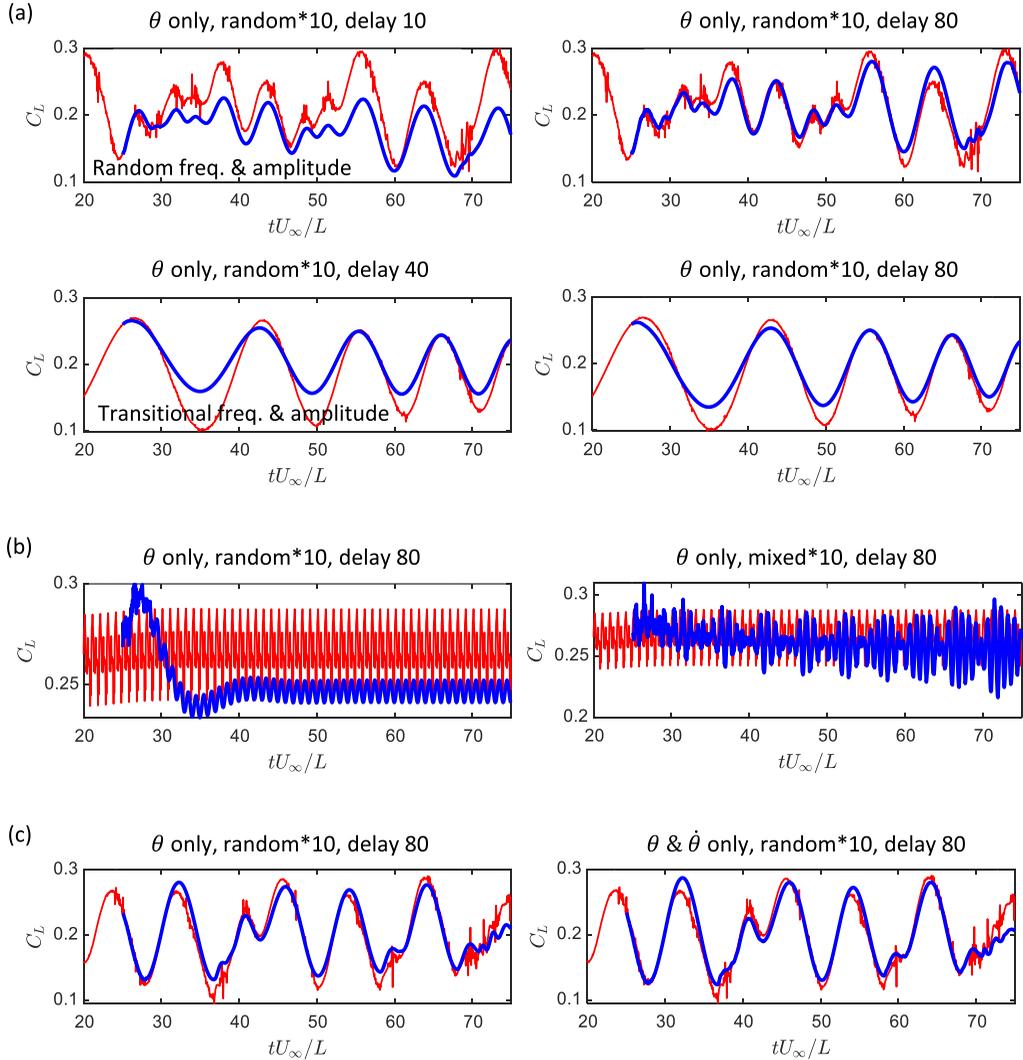


FIG. 2. Model performance evaluation for (a) delay-embedded time step, (b) trajectory type, and (c) input type. Red thin line: ground truth; blue thick line: model reconstruction.

in some cases up to 300% improvement can be obtained. Including more data allows long-term memory for the model, which helps capture slower dynamics better. Second, reducing the number of trajectories included in the system identification from 10 to 5 minimally changes the accuracy, which aligns with the observation in [37] that with delay embedding, the Koopman operator-based models work relatively well at very low-data conditions. Another observation is that including high-frequency periodic trajectories in the model, construction improves the model performance for periodic cases. From the previous study [41], we learned that at certain flap oscillation frequencies, quasiperiodic motion is present, and this phenomenon is not obvious in the randomly actuated cases. Although the tracking error is not that much different, only the mixed trajectory cases can capture the quasiperiodic nature of those higher frequency periodic cases. Finally, including an extra DoF for the control by including the $\dot{\theta}$ in the input sequences improves the tracking of the extrema, but only marginally better since our two control DoFs are related.

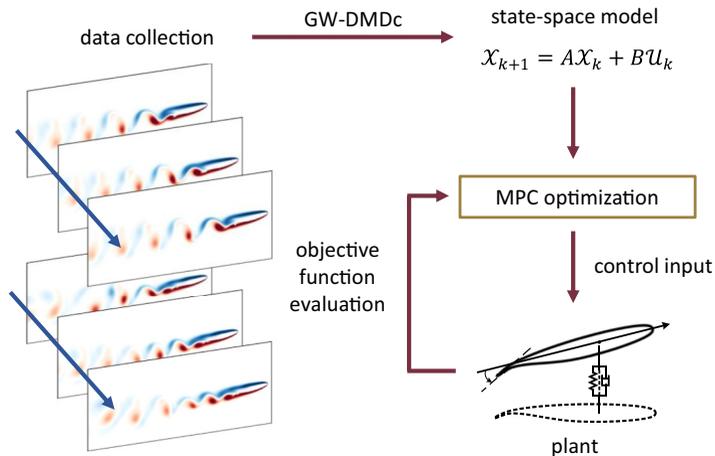


FIG. 3. Schematic representation of the MPC control framework.

However, in practice, there are some limitations for implementing what was found to be “good” for the model. Increasing the delayed time step also increases the dimension of the data matrix and hence causes difficulty in the optimization of the controller design. The more delayed time step also means data over a longer period have to be stored, which in realistic cases could be demanding for the hardware. Increasing the numbers of included trajectories also requires more delayed embedded time steps and/or finer temporal resolution to capture finer details of different scales. Adding an extra DoF of θ improves the model accuracy; however, in practice, this poses a strong constraint for finding the optimal control sequence, requiring $\sum_{\Delta t} \theta dt = \theta$ over each time step, which could cause the optimization to fail. Considering all these trade-offs, in this research we determined the suitable model uses 80 delayed time steps, 10 mixed trajectories including randomly actuated and some high-frequency periodic cases, and the input will be solely θ . From Table I we can see that this combination offers reasonable tracking performance over all scenarios tested.

Now that we have the linear model capable of capturing the essential dynamics of a nonlinear system, in the next section we will discuss how to implement this model into the MPC scheme to achieve rapid control of the flow-induced fluttering of our foil-and-flap system.

III. MODEL PREDICTIVE CONTROL FRAMEWORK FOR THE FOIL-AND-FLAP SYSTEM

With the DMDc model that can recover the foil-and-flap system dynamics in hand, a control strategy can be found to regulate the flow-induced flutter. The control scheme we decided to adopt here is the *Koopman-MPC* innovated by Korda and Mezić [35]. MPC, overall, is not a specific control law but a control framework, and here we will provide a simple introduction. For more information, readers can refer to reviews and books, e.g., [42,43]. The standard procedure of MPC consists of three steps: first, a model is built for the plant, which we have done through the DMDc process. Then at each time step of controlling, an optimization problem is solved to find the optimal input that drives the *model* from the initial state acquired from the *plant* to the desired state. Finally, the optimal control input is applied back to the actual plant, and the modified state is fed into the optimizer again in the next loop. These three steps are shown graphically in Fig. 3. Note that for MPC framework, the control input is actually produced in an open-loop fashion by solving the optimization problem in the model space, and no feedback from the plant is given until the next instance when the controller is called. This is also where using the Koopman-based model in the MPC framework shines: the Koopman-based models are *linear*, and hence, the optimization problem can be cast into a *convex* quadratic problem which can be solved rapidly by a plethora of available solvers, e.g., qpOASES [44], OSQP [45], and the FORTRAN version of the quadprog [46]

adopted in this work. Next, we will explain how we can cast our control problem into a solvable form, and we will show the results of applying active control to the foil-and-flap plant later in this section.

A. Define control problem and quadratic programming formulation for MPC

The goal of this project is to control the flow-induced flutter phenomena with the active flap as the actuator. In other words, we want to regulate the lift force of the system. The control problem can be recast into a trajectory-tracking problem, where we want the lift coefficient of the foil to follow the designated reference trajectory. If this trajectory is a constant, then the flutter is eliminated; the trajectory can also be oscillatory for the purpose of enhanced energy extraction. For each control time step t_j , we have the dynamic model

$$\begin{aligned} x_{j+1} &= \mathbf{A}x_j + \mathbf{B}u_j, \\ y_j &= \mathbf{C}x_j, \end{aligned} \quad (25)$$

where $x_j \in \mathbb{R}^{n \times 1}$ are the state vector at current control time step and x_{j+1} at the next time step. In our case each x_k consists of the delay-embedded lift coefficient, drag coefficient, and flow kinetic energy at selected points on the foil [as shown in Fig. 1(a)]. The positions of these data acquisition points are chosen at the spot where the flow shows more fluctuation. The input vector $u_j \in \mathbb{R}^{m \times 1}$ contains the specified flap angle. $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ are the state-transition matrix and loading matrix, respectively, acquired from the DMDc system identification process. $y_j \in \mathbb{R}^{p \times 1}$ is the observation vector that contains the desired measurements of the system, which is the lift coefficient at the current time step in our case here. $\mathbf{C} \in \mathbb{R}^{p \times n}$ is the user-specified measurement-sensitivity matrix, and here it will be 1 at where the lift at current time step is and 0 at all the other locations. The control problem of lift trajectory tracking can be expressed in terms of optimizing the input to minimize the tracking error over the next N time steps, written in the form of the objective function:

$$\min_{u_0, \dots, u_{N-1}} \frac{1}{2} \sum_{k=0}^N [(y_k - y_{k,\text{ref}})^T \mathbf{Q}_k (y_k - y_{k,\text{ref}})] \quad (26)$$

subject to Eq. (25). $y_{k,\text{ref}} \in \mathbb{R}^{p \times 1}$ is the reference trajectories, and $\mathbf{Q}_k \in \mathbb{R}^{p \times p}$ defines the weighting at each time step. Common choices for the weighting \mathbf{Q}_k are uniform or a descending function to improve the short-term control accuracy. However, for a standard QP problem, the control target and the control input should be in the same input space:

$$\min_x \frac{1}{2} x^T H x + f^T x, \quad \text{s.t.} \quad \begin{cases} \mathbf{A}x \leq b \\ y_j = \mathbf{C}x_j \\ lb \leq x \leq ub \end{cases}. \quad (27)$$

To write Eq. (26) in the form of Eq. (27), we will have to utilize the evolutionary property of Eq. (25),

$$y_{k+1} = \mathbf{C}x_{k+1} = \mathbf{C}(\mathbf{A}x_k + \mathbf{B}u_k) = \mathbf{C}\mathbf{A}x_k + \mathbf{C}\mathbf{B}u_k, \quad (28)$$

and for the following time step:

$$\begin{aligned} y_{k+2} &= \mathbf{C}x_{k+2} = \mathbf{C}(\mathbf{A}x_{k+1} + \mathbf{B}u_{k+1}) \\ &= \mathbf{C}[\mathbf{A}(\mathbf{A}x_k + \mathbf{B}u_k) + \mathbf{B}u_{k+1}] \\ &= \mathbf{C}\mathbf{A}^2x_k + \mathbf{C}\mathbf{A}\mathbf{B}u_k + \mathbf{C}\mathbf{B}u_{k+1}. \end{aligned} \quad (29)$$

We can see that there is a pattern in this. Derive this for N step, collect all the states into a new state vector and all input into an input vector,

$$\tilde{y}_k \equiv [y_k^T, y_{k+1}^T, \dots, y_{k+N}^T]^T \quad \text{and} \quad \tilde{u}_k \equiv [u_k^T, u_{k+1}^T, \dots, u_{k+N-1}^T]^T, \quad (30)$$

then

$$\tilde{y}_k = \tilde{\mathbf{A}}x_k + \tilde{\mathbf{B}}\tilde{u}_k, \quad (31)$$

where

$$\tilde{\mathbf{A}} \equiv \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^N \end{bmatrix}, \quad \tilde{\mathbf{B}} \equiv \begin{bmatrix} 0 & 0 & 0 \\ \mathbf{CB} & 0 & 0 \\ \mathbf{CAB} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \mathbf{CA}^{N-2}\mathbf{B} & & \mathbf{B} \end{bmatrix}. \quad (32)$$

The dimensions of these vectors and matrices are $\tilde{y}_k \in \mathbb{R}^{p(N+1) \times 1}$, $\tilde{u}_k \in \mathbb{R}^{mN \times 1}$, $\tilde{\mathbf{A}} \in \mathbb{R}^{p(N+1) \times n}$, and $\tilde{\mathbf{B}} \in \mathbb{R}^{p(N+1) \times mN}$. Note that in Eq. (31), the x_k is the *known* initial state measured from the plant. If Eq. (31) is substituted to the objective function Eq. (26), we arrive at

$$\begin{aligned} J(x_k, \tilde{y}_{k,\text{ref}}, \tilde{u}_k) &= \sum_{k=1}^{N+1} [(y_k - \tilde{y}_{k,\text{ref}})^T \mathbf{Q}_k (y_k - \tilde{y}_{k,\text{ref}})] \\ &= (\tilde{\mathbf{A}}x_k + \tilde{\mathbf{B}}\tilde{u}_k - \tilde{y}_{k,\text{ref}})^T \tilde{\mathbf{Q}} (\tilde{\mathbf{A}}x_k + \tilde{\mathbf{B}}\tilde{u}_k - \tilde{y}_{k,\text{ref}}) \\ &= (x_k^T \tilde{\mathbf{A}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{A}} x_k + \tilde{y}_{k,\text{ref}}^T \tilde{\mathbf{Q}} \tilde{y}_{k,\text{ref}} - 2x_k^T \tilde{\mathbf{A}}^T \tilde{\mathbf{Q}} \tilde{y}_{k,\text{ref}}) \\ &\quad + \tilde{u}_k^T (\tilde{\mathbf{B}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}}) \tilde{u}_k + 2(x_k^T \tilde{\mathbf{A}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}} - \tilde{y}_{k,\text{ref}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}}) \tilde{u}_k. \end{aligned} \quad (33)$$

The weighting matrix $\tilde{\mathbf{Q}} \in \mathbb{R}^{p(N+1) \times p(N+1)}$ has all the \mathbf{Q}_j , $j \in [k, k+N]$ on the diagonal axis. Notice that the first three terms of the last expansion of Eq. (33) are constants with the same initial state x_k and reference trajectory $\tilde{y}_{k,\text{ref}}$ since the modified state-transition matrix $\tilde{\mathbf{A}}$ and loading matrix $\tilde{\mathbf{B}}$ are derived outside of the optimization process. This allows dropping these terms in the optimization problem and forming the equivalent QP problem of Eq. (26) as

$$\min_{\tilde{u}_k} \{ \tilde{u}_k^T (\tilde{\mathbf{B}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}}) \tilde{u}_k + 2(x_k^T \tilde{\mathbf{A}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}} - \tilde{y}_{k,\text{ref}}^T \tilde{\mathbf{Q}} \tilde{\mathbf{B}}) \tilde{u}_k \}, \quad \text{s.t.} \begin{cases} \tilde{u}_{\min} \leq \tilde{u}_k \leq \tilde{u}_{\max} \\ x_k, \tilde{y}_{k,\text{ref}} \text{ given} \end{cases}. \quad (34)$$

Comparing this to the standard QP formulation, Eq. (27), it is clear that this has the exact same form, and the existing tools can be readily deployed to solve this problem.

Two constraints are enforced through the inequality matrix \tilde{u}_{\max} and \tilde{u}_{\min} . The first constraints are the range of the flap motion, limited to $-50^\circ \leq u_k \leq 50^\circ$. This range is determined *a priori* and here is set based on the stability of the computational solver. When the flap deflects over 50° , a sharp angle is generated at the connection point between the flap and the main foil body, rendering the conformal mapping to be invalid. Another constraint is named the continuity constraint, $|u_{k+1} - u_k| \leq u_{\text{range}}$. We call this the continuity constraint since this condition ensures the flap does not perform any sudden acceleration, which could make the FSI solver unstable due to the interpolation required to generate a new computational grid at each time. If we include the angular flap velocity $\dot{\theta}$ in the input vector, we will have to enforce another constraint to connect the flap angle and velocity, e.g., the central difference scheme $\dot{\theta}_k = (\theta_{k+1} - \theta_{k-1})/dt$. In practice, this is proven to be too strong of a constraint, and the optimizer used here could not always find the optimal solution. Hence, we only use the flap angle as the input.

Once the QP problem, Eq. (34), is solved, we get an input sequence \tilde{u}_k for time step t_k to t_{k+N-1} that returns the minimal objective function over the next N time step. For closed-loop MPC control, we will apply only the first element of this input sequence to the plant, and then at the next control time step, new states are collected to be the initial condition of the new optimization process. This is called the *receding horizon* MPC control, which provides the feedback mechanism for the MPC framework.

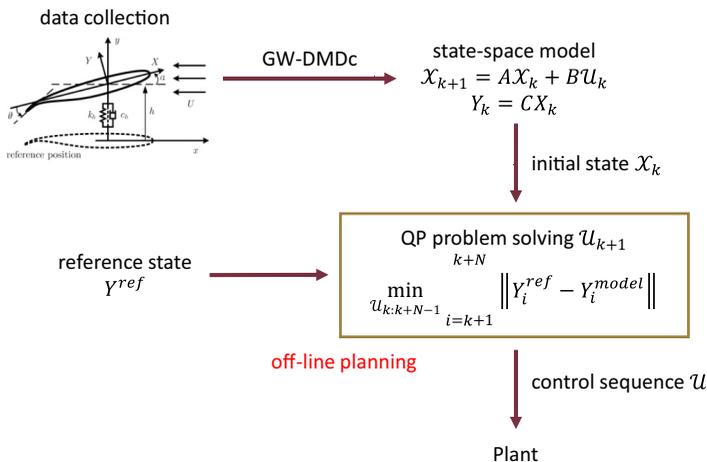


FIG. 4. Block diagram for the open-loop model-based control.

B. Open-loop model-based control

Now let us design the controller for the foil-and-flap system using the DMDc model derived above. First, an open-loop model-based control is employed to serve as a baseline case where no system feedback is given, or the feedback time interval is infinite. The block diagram of the open-loop control is shown in Fig. 4. The open-loop term here means that there is no feedback from the plant and trajectory planning is done offline. At $t = t_0$, we acquire the delay-embedded initial states from the plant as \tilde{x}_0 , then start the control with the reference trajectories \tilde{y}_{ref} . After completing optimization at each control time step, we apply the first element of the optimal control sequence to the *model*, then take the states returned from the *model* as the new initial state. Following the calculation for a specified time span, the collected control sequence is fed to the plant and we let it evolve on its own. This control strategy has no feedback from the plant and is hence called the open-loop model-based control. Note that to reduce the computational resource requirement and facilitate real-time application, the control time step dt_{ctrl} is much larger than the FSI simulation time step. We have $dt_{ctrl} U_\infty/L = 0.05$ for all of our controlled cases, and the trajectory planning block is in charge of producing the input trajectory in between. For the current controller, where only the flap angle serves as the control target, the trajectory is connected piecewise linearly, and the flap angular velocity is calculated with the slope of the linear piece. This works the same for the closed-loop cases discussed later.

Since our ultimate goal is to regulate the flow-induced flutter, we choose a scenario that is realistic to the possible application for energy extraction, which is the sinusoidal lift coefficient with transitioning amplitude and frequency. The reference lift coefficient is acquired by prescribing flap motion to the plant. Figure 5 shows three trajectories of the input flap angle and corresponding lift coefficients in thin red lines. It also shows the results from the open-loop model-based control in thick blue lines, and the vertical broken line marks where the control is initiated after the flap follows the original prescribed trajectory. Three different sample trajectories will be used to demonstrate the control efficacy: trajectory A is the low-frequency trajectory with flap waving at $St_f = 0.075$ to 0.1 with slightly decreasing amplitude; trajectory B oscillates at a higher-frequency range $St_f = 0.5$ to 0.6; and the frequency of trajectory C gradually decreases from $St_f = 0.5$ to 0.4.

The results show that the open-loop control can keep the system responses within 20% range from the desired trajectories. Table II shows the average tracking error of applying the open-loop control over every 300 control time steps, i.e., $\Delta t U_\infty/L = 15$. For all three cases, we can see that at the initiation of the control engagement, a large flap deflection is requested by the controller. This stems from the discrepancy between the desired states and the model-predicted states using the

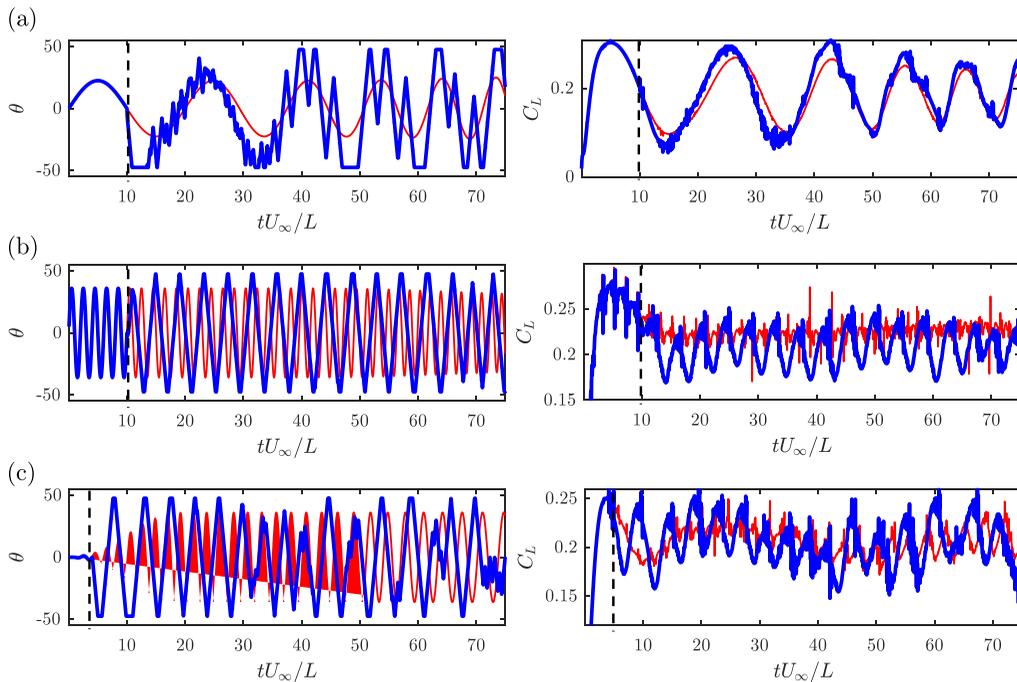


FIG. 5. Open-loop model-based control performance evaluation for (a) trajectory A: low-frequency; (b) trajectory B: high-frequency; and (c) trajectory C: high-to-low frequency test cases. Red thin line: ground truth; blue thick line: controlled cases; vertical broken line: control initiation.

plant-provided initial condition. As the open-loop controller has no feedback from the plant, this error will require some time to be mitigated, and after that the overall range of the lift force can be reasonably recovered. Given that this control strategy lacks feedback from the system response, the paramount factor influencing control effectiveness is the accuracy of the model predictions. For cases with higher oscillation frequencies, the control input deviates significantly from the reference input after a brief control action interval, sometimes exhibiting an almost 180° phase difference. Surprisingly, the overall error remains insubstantial. This is attributed to the DMDC model, which, despite its imperfect reconstruction of detailed higher frequency pulsing responses in quasiperiodic systems, effectively captures the general trend of fluidic forces and states [see Fig. 2(b)]. Given the controller's ability to follow the average, the overall tracking error is consequently confined to a lower range. However, for trajectory A, characterized by a larger amplitude, it is evident that the initial error propagates and leads to intense fluctuations in the control input. After $t U_\infty/L > 60$, the controller is able to track the lift coefficient well since the DMDC model performs better at small amplitude cases and the initial error is mostly mitigated.

TABLE II. Average tracking error with open-loop model-based control corresponding to the three cases shown in Fig. 5.

Trajectory case	$t = 15 \rightarrow 30$	$30 \rightarrow 45$	$45 \rightarrow 60$	$60 \rightarrow 75$
A	14.63%	21.26%	10.04%	10.79%
B	10.66%	12.93%	10.38%	12.48%
C	9.71%	10.0%	12.78%	13.78%

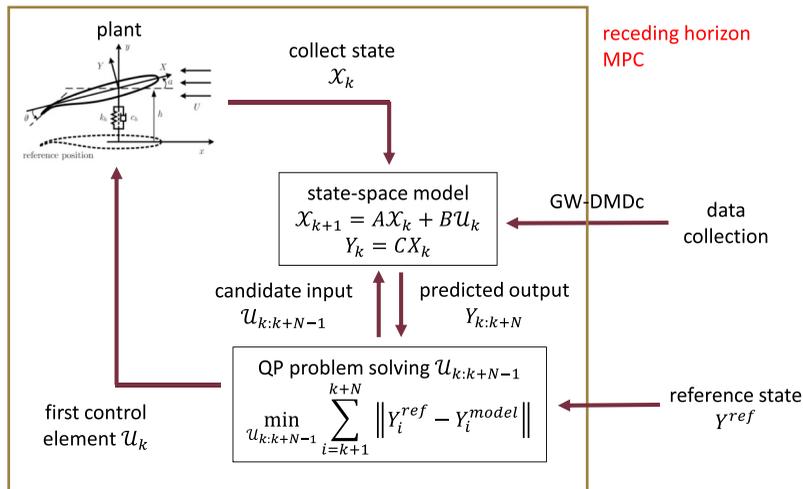


FIG. 6. Block diagram for the closed-loop MPC control.

Overall, open-loop model-based control could not achieve accurate trajectory tracking. Moreover, owing to model inaccuracies, the controller tends to demand excessive actuation. Across all three cases, an overshoot of the control signal compared to the originally prescribed motion is evident, presenting an undesirable characteristic for real-world actuators. Nonetheless, the controller maintains the response within a reasonable range. We will now explore enhancements to the controller by incorporating system response feedback.

C. Closed-loop MPC control

To further improve the effectiveness of our control scheme, we can take the systems states acquired from the *plant* and reinitiate the optimization procedure at each control time step. The block diagram of this closed-loop MPC control scheme is shown in Fig. 6. We can see that compared to the open-loop design, the plant is now included in the control cycle to feed back the state altered by the last control input.

The same three cases used in the open-loop cases are shown here to demonstrate how the closed-loop MPC performs. The reference lift coefficient and the original input sequence, along with the results of the controlled cases, are shown in Fig. 7. It is evident that closed-loop control works better than the open-loop case, especially for the low-frequency cases, where the lift coefficient nearly perfectly follows the reference trajectory shortly after the control is engaged. The large error at the initialization is quickly corrected by the optimization, and within 300 iteration cycles after initiation ($t U_\infty/L = 15$), the controller settles down on its steady input sequence. Table III indicates the tracking error in a different time span, which demonstrates that the closed-loop controller can track the lift coefficient to a great sub-5% error range for most cases. However, for high-frequency cases, again, the trend is recovered very well but the high-amplitude pulsing behavior is suboptimally captured. We can see from the input sequence that the input θ never goes to the extrema even though the peak is not matched, which reflects the observation that the DMDc model could not capture the full-rank dynamics of the system. In the authors' previous publication [41], we concluded that the quasiperiodic motion is caused by the interaction between the flow-induced mode and the flap-induced mode. In the phase-space perspective, this means these two trajectories get so close in the phase space that they intersect with each other and the system keeps switching between the two trajectories. DMDc cannot distinguish them either due to the smoothing effect when multiple trajectories are considered or because the data acquisition time step should be smaller to preserve more details for the identification procedure to recognize the

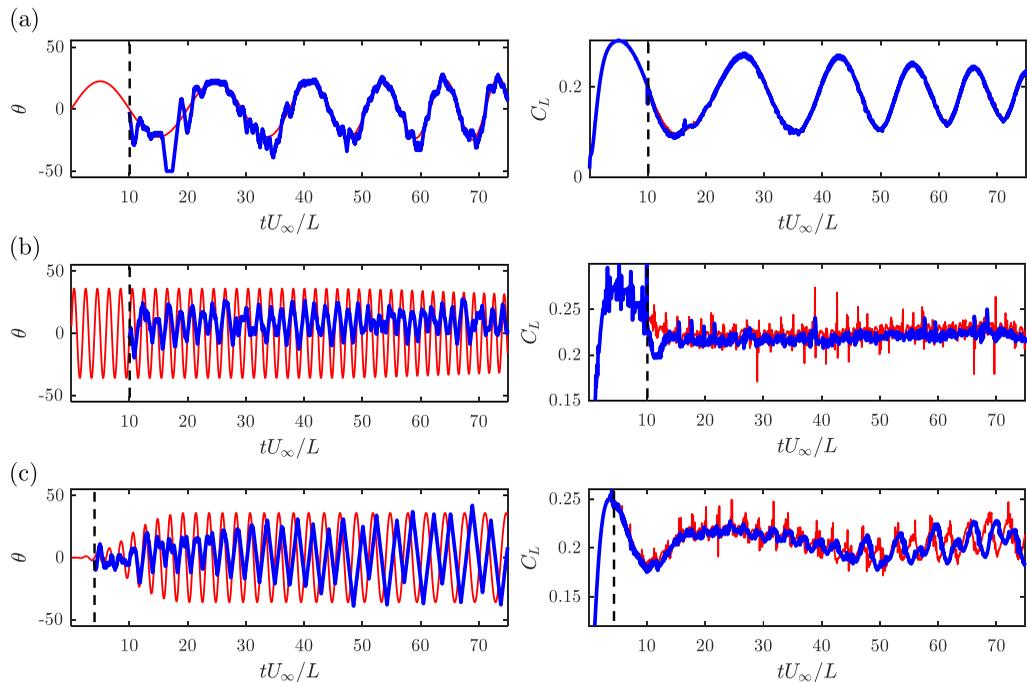


FIG. 7. Closed-loop MPC control performance evaluation for (a) low-frequency, (b) high-frequency, and (c) high-to-low frequency test cases. Red thin line: ground truth; blue thick line: controlled cases.

difference between the trajectories. From the control perspective, at higher-frequency conditions, this foil-and-flap system is not controllable only by the active flap actuation, or the system state space cannot be reached entirely by moving the flap. From the various tests conducted for the open-loop and closed-loop control, we conclude that the active control is especially valid within the flap oscillating frequency range of $St_f < 0.5$ as seen in Figs. 6 and 7. This range matches that found in the cited previous research, where the flap-induced modes are the dominant flow mechanism only when separated farther enough from the flow-induced mode. Still, for the high-frequency cases, the tracking error is minimal since the long-term average behaviors are still captured well by the DMDC model and the flap can impact the slower long-range trends.

A crucial parameter distinguishing the open-loop and closed-loop cases is the feedback interval, as the open-loop case can be regarded as the extreme scenario with an infinite feedback interval. To further assess the influence of the feedback interval on control effectiveness, we examine four cases with varying settings of Δt , $U_\infty/L = 0.1, 0.2, 0.4, 0.8$, all employing the same low-frequency trajectory as in the preceding instances [refer to Figs. 5(a) and 7(a)]. Figure 8 and Table IV show, respectively, the flap angle with corresponding lift coefficient and average tracking error

TABLE III. Average tracking error with closed-loop MPC control corresponding to the three cases shown in Fig. 7.

Trajectory case	$t = 15 \rightarrow 30$	$30 \rightarrow 45$	$45 \rightarrow 60$	$60 \rightarrow 75$
A	2.97%	2.47%	2.0%	3.17%
B	4.27%	4.23%	3.78%	4.17%
C	3.27%	4.27%	6.82%	8.87%

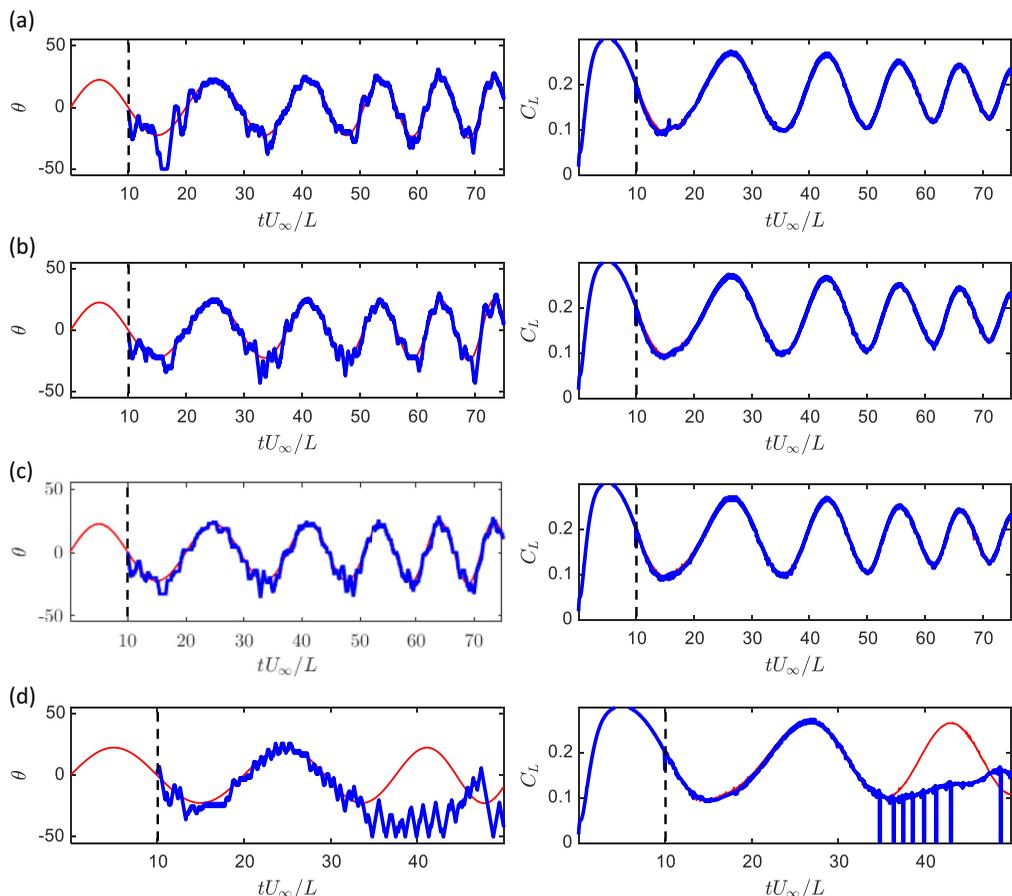


FIG. 8. Closed-loop MPC control performance with feedback interval $\Delta t U_\infty/L =$ (a) 0.1, (b) 0.2, (c) 0.4, and (d) 0.8. Red thin line: ground truth; blue thick line: controlled cases.

for the four cases along with the original $\Delta t, U_\infty/L = 0.05$ case. We can see that when the feedback interval increases beyond a certain threshold, the controller performance suddenly drops to an undesirable extent – over 500%, which is worse than the open-loop approach. Upon further inspection, the reason behind this is that when there is no system state feedback, the system slowly sways away from the target trajectory as the planned piecewise trajectory does not optimally track. This does not present an issue if the target trajectory exhibits a trend with a low frequency, as the

TABLE IV. Average tracking error with closed-loop MPC control corresponding to the four cases shown in Fig. 8.

$\Delta t U_\infty/L$	$t = 15 \rightarrow 30$	$30 \rightarrow 45$	$45 \rightarrow 60$	$60 \rightarrow 75$
0.05	2.97%	2.47%	2.0%	3.17%
0.1	2.95%	2.46%	2.0%	2.27%
0.2	2.50%	2.43%	1.91%	2.49%
0.4	3.42%	2.82%	2.74%	3.24%
0.8	3.75%	500.03%	–	–

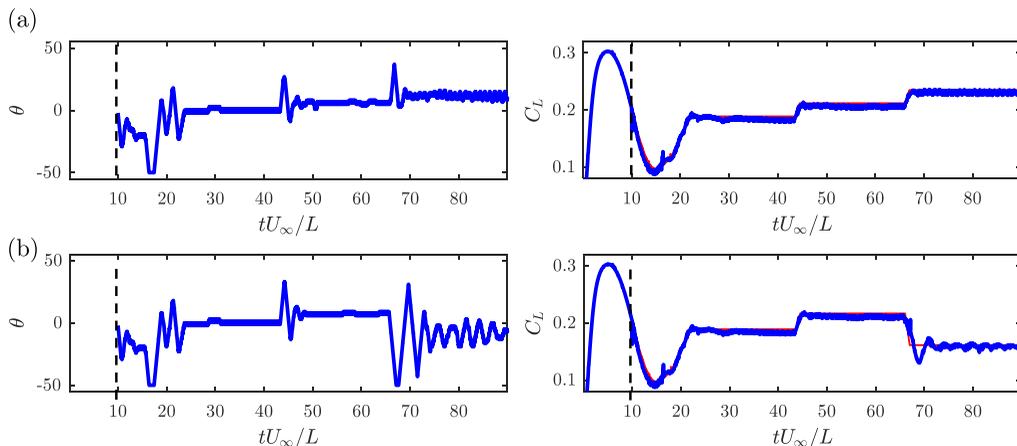


FIG. 9. Closed-loop MPC control performance evaluation for two different step-function trajectories. Red thin line: ground truth; blue thick line: controlled cases.

open-loop approach is proven to have good tracking performance, given the model is well constructed. When a new feedback signal arrives, the controller is able to regain control. Yet, as the tracking frequency increases, the state feedback introduces a notable discrepancy between the current state and the desired state, as observed in the open-loop scenario. In such instances, the model predictive control (MPC) controller fails to find a feasible solution, resulting in the propagation of errors and eventual destabilization of the system. Conversely, when the feedback interval is sufficiently small, the tracking performance seems to exhibit reduced dependence on the interval.

The previous tests aim to demonstrate how the flap can help facilitate a large amplitude of flow-induced vibration; how about the opposite, trying to use the flap to stop the vibration? We will now tackle this problem. Figure 9 shows the desired lift coefficient and the closed-loop MPC-controlled results. In this set of tests, two lift coefficient trajectories are tested, both of which start with the same oscillating lift and settle down at the average lift of waving the flap at a large range (-30° to 30°), which is about $C_L = 0.189$. Then the target lift coefficient shifts to the other constant value after a period, then to another. The flap angle trajectories generated by the closed-loop MPC controller to track these lift trajectories are shown, where we can see the flap has to be constantly adjusted to cope with the shedding leading edge vortices so that the lift can stay at a constant value. The oscillation especially becomes intense when the flap goes to negative values (pointing upward), which creates stronger nonlinear interaction with the vortex wake created at the leading edge. Nonetheless, the constant lift target is well achieved, with the average tracking error between all piecewise segments being 3.19%. This shows that the presented control mechanism can successfully eliminate the flow-induced vibration.

The closed-loop control achieves the goal that we set at the beginning of this work: regulation of the flow-induced flutter of a foil with the active flap as the actuator. With the Koopman-based MPC, this task is satisfied with great accuracy for low-frequency oscillation and constant lift scenarios (no vibration), and reasonable response for high-frequency oscillation. Let us further consider the practical application situation, where the speed of the controller is crucial. Differently from the simulation where we can wait until the optimization is finished before evolving the flow, in practice the flow just keeps going. If the controller is not fast enough, when the optimized input is acquired, the flow is already in a different state. The performance of utilizing QP solvers to perform MPC depends on many factors, including the model type, number of inputs to optimize, the number of states considered, tolerance required, etc., and it is also an iterative process so there is no single performance criteria that can describe its efficiency. Readers are referred to papers with benchmark

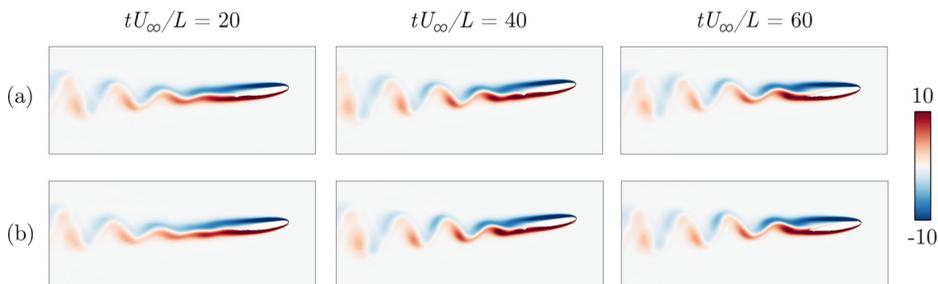


FIG. 10. Vorticity field of the low-frequency flap trajectory case for the (a) prescribed flap motion and (b) closed-loop MPC-controlled cases.

tests, such as Kouzoupis *et al.* [47], that compare the performance of different QP solvers to learn more on this aspect. A crucial benefit of using a Koopman-based model is that solving the convex optimization problem can be rapid to enable real-time controlling of nonlinear systems. For our case, each call of the quadprog in MATLAB takes about 0.006 sec to perform with the delayed dimension 80 and control horizon $N = 10$ using a desktop PC equipped with the Intel Xeon CPU E5-1603 v4 running at 2.8 GHz.

D. Flow field observations

As elaborated in Sec. I, another benefit of using the DMDc-based MPC structure is that the control is designed on the fly and applied to the nonlinear full-scale plant. This allows us to examine how the flow responds to the flap actuation, and this could provide real-time insight on the actuator impact and inform about the optimal sensor placement. Combining the MPC framework with methods that can incorporate new observations online [48,49], one can create more accurate models without conducting system identification again. This property could be crucial to practically implement this control method to real-world systems, as the ambient environment can provide more accurate working conditions to improve the model prediction.

Let us take a look at the flow field of the low-frequency and high-frequency cases shown in the previous sections. The vorticity field snapshots of the ground-truth simulations with prescribed flap motion compared with the closed-loop MPC controlled cases are shown in Figs. 10 and 11. We can now see why the MPC controller cannot fully capture the high-frequency response regimes. The smoothing effect of the DMDc model treats the desired lift trajectory as a slowly varying sinusoidal pattern and categorizes the behavior into a simple limit cycle oscillation. However, the spiking peaks come from the nonlinear interaction between the leading and trailing edge vortex shedding, which creates a much more chaotic wake in the ground truth case. The small MPC-generated flap

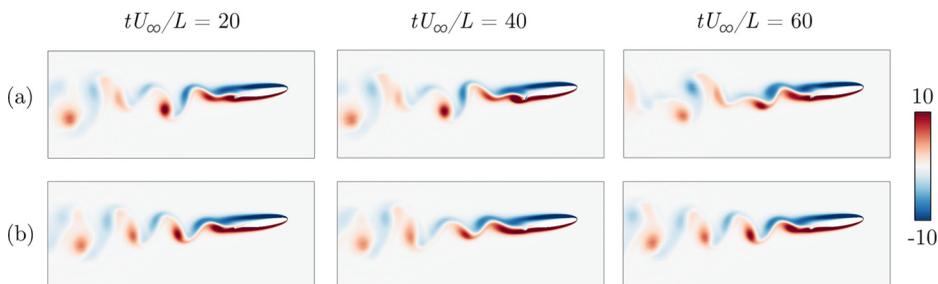


FIG. 11. Vorticity field of the high-frequency flap trajectory case for the (a) prescribed flap motion and (b) closed-loop MPC-controlled cases.

amplitude could not excite the intense trailing edge separation that kicks the system out of the stable limit cycle. Including an online state estimator in the system could improve the identification of this kind of behavior. On the other hand, in the low-frequency cases, the two scenarios exhibit an almost identical wake patterns after the control sequence settles down after the initial error is mitigated.

E. Unsteady ambient flow condition identification

Although the optimization process of the MPC framework could be sufficiently fast for real-world scenarios, in practice robustness of the solver against environmental condition disturbances is another crucial factor for the controller to work. For example, for a flying airfoil, the flow conditions constantly change due to humidity, temperature, turbulence, and many other factors to consider. The model used here is built with just one set of environmental conditions (uniform ambient flow), and it showed good potential at tracking different kinds of trajectories produced from diverse flap motion. However, how it will respond to other forms of disturbances remains to be explored. For the control purpose, MPC is very robust at hiding the model's inaccuracy [29], but a model that inherently considers unsteady conditions could provide more robust control of the system and a better understanding of the underlying physics. This remains an open question to be scrutinized thoroughly with realistic environmental conditions, but here we will provide a preliminary discussion with simplified conditions.

A number of reduced order models developed for flow-induced fluttering systems also take unsteadiness into account [50], and some of them can be easily fitted into the current MPC structure. For example, Deem *et al.* [51] designed an adaptive separation control of a laminar boundary layer based on online DMD [48]. By periodically updating the DMD model after modifying the flow, the control gain based on the new flow state can be adjusted accordingly. Another example is the DMD with exogenous input proposed by Kou and Zhang [52] that models the disturbance as an external input and separates the effect of the natural unperturbed response and the external input. We will adopt this idea and treat the flow fluctuations as an external input characterized by certain parameters and test our MPC control framework on a foil subject to unsteady flow.

We will use a simple model to represent the unsteady condition caused by the incoming gust. This canonical disturbance can be described with a sinusoidal oscillating streamwise velocity and is a classic case described in Theodorsen's original work [12], which then has been adopted by various research [53,54]. The streamwise ambient flow velocity is represented as

$$U_\infty(t) = U_{\text{mean}} + U_A(t) \sin[\omega_\infty(t)t + \phi_\infty(t)], \quad (35)$$

where $U_A(t)$ is the oscillation amplitude of the free stream velocity, and $\omega_\infty(t)$ and $\phi_\infty(t)$ are the frequency and the phase of the ambient flow oscillation, respectively. Recall that in Sec. III, we defined the control input u as the flap angle. To incorporate the effect of the oscillating flow, we can treat the ambient flow velocity as a separate external input and write the control input at time t_k as

$$u_k = [\theta_k, U_{\infty,k}]^T, \quad (36)$$

and the rest of the optimization procedure follows Eq. (21) and following. Figure 12 shows an example of a foil subject to oscillating flap actuation and streamwise oscillating gust. Both the frequency of the flap motion and that of the ambient flow transition from 0.1 to 0.05 in this specific case.

The GW-DMDc model with both flap angle and ambient flow velocity as input is again constructed with 10 different trajectories consisting of lift and drag coefficients and flow kinetic energy sampled on the foil surface. These trajectories all have smoothly transitioned input magnitudes. The following cases are not part of the training data. Performing the same open-loop model-based control described in Sec. III B on the case shown in Fig. 12, we are able to recover the trajectory of both the flap motion and ambient flow oscillation to a reasonable accuracy, as shown in Fig. 13. A similar issue that with open-loop model-based control, the lift coefficient cannot be perfectly recovered due to the model imperfection still being present, and the amplitude of the flap angle

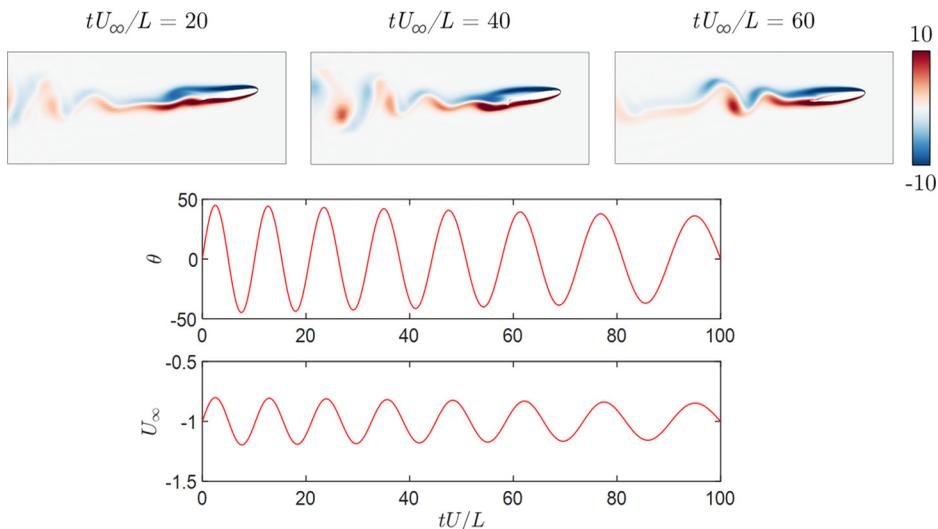


FIG. 12. Vorticity field and the corresponding magnitude of the flap angle θ and ambient flow velocity U_∞ .

is incorrect. Still, even with the added disturbance in the ambient flow velocity, the model exhibits good simultaneous predictability for the ambient environmental conditions and the active actuation required to follow the reference lift coefficient trajectory.

With this simple example, we see the potential in both control and identification of the MPC framework. Moreover, with the geometrically weighted DMDC, the identification process can be extended to identify spatial structure changes. The nonlinear impact of the actuators on the flow can finally be scrutinized. This could enable a new territory of morphing control development, with multiple actuators responsible for mitigating different kinds of disturbances. Furthermore, the modal information can be used to reconstruct the whole flow field with limited available sensors.

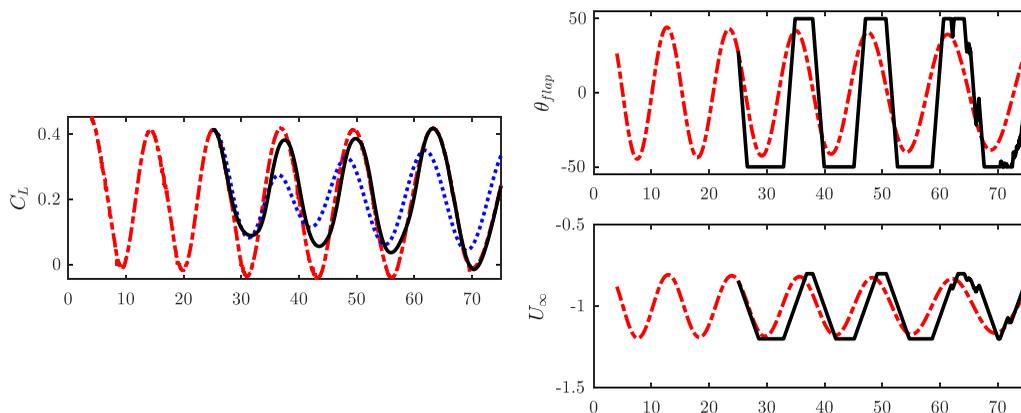


FIG. 13. The lift coefficients, flap angle, and ambient flow velocity trajectories. Red broken line: reference; blue dotted line: model response with original input; black solid line: open-loop model-based control results.

IV. CONCLUSION

In this paper we introduce how we can build a linear model with control input utilizing a geometrically weighted DMDc procedure. This model is then used to derive the objective function used in the MPC scheme to satisfy trajectory tracking requirements. Both open-loop model-based control and closed-loop MPC control are tested on our foil-and-flap system, which exhibits good tracking performance. The results show that we successfully regulate the flow-induced flutter with the active flap by directly controlling the lift forces acting on the foil.

However, the controllability of the flap actuator and the limitation of the DMDc model offer some restrictions on the control scheme, mainly in tracking high-frequency nonlinear pulsing responses. From the previous study [41], we know an active flap interacts with the flow-induced mode very differently at different amplitudes and frequencies, and at a high flapping frequency limit, the flow-induced mode is more dominant. If one wish to track these higher-frequency dynamics, adding another high-frequency actuator such as a fluidic actuator at the leading edge of the foil is a possible solution. Another possible solution to this problem is to utilize some forms of state estimators to update the model to match the observed responses. For example, the Kalman filter [55] is a recursive state estimation technique that can estimate the future state with the recorded state history. Through a Kalman filter, the underlying nonlinear interactions between the flap- and flow-induced modes can be estimated in real time and provide another layer of predictability to the GW-DMDc model. Another benefit of including a state estimator is the improvement of the controller's robustness. As discussed in Sec. III E, by adding an extra control input, the effects of coherent flow structures can possibly be isolated to provide better control sequence design. Moreover, the state estimator could help screen the turbulence fluctuation, model inaccuracy, or other unknown noncoherent disturbances and identify the dynamics controllable by the available controller.

In this work we adopted the idea of linear modeling with the Koopman operator, which has the benefits of data-driven and linearity. There are still many options of models that can fit into the MPC framework, as the only requirement is the prediction the system response with a certain accuracy, and that the model can be evaluated quickly. The classical Theodorsen's model is constantly being improved and could be a great candidate for the model used in the MPC framework. For example, the work by Pohl and Hermann *et al.* [9,10] links multiple models acquired from different airfoil postures to capture the hysteresis effect of the foil-and-flap system. Other works aim to expand the modeling horizon of Theodorsen's model, such as those conducted by Wang *et al.* [6], which formulates the effects of gust, or by Platanitis *et al.* [11], which considers the interaction of multiple morphing surfaces. These efforts have been improving the elegant classical model and could be used in the MPC framework. Another branch of methods worth mentioning is the vortex model, which can capture the vortex distribution and evolution in the wake. For example, Darakananda *et al.* [56] represented the flux of vorticity into the wake by a continual release of time-variable vortex elements from both edges of the airfoil, and the strengths of the vortices are corrected by a Kalman filter based on local measurements. Mathieu *et al.* [57] further extended the model to identify the possible connection between the vortex strength and a cluster of variables to reject disturbances of the local measurement. These methods are all part of a continuous effort for better modeling of aerodynamic flows with moving and deforming structures, and the authors are excited to see how the combined effort of the community can bring a better control and design philosophy to the FSI systems.

ACKNOWLEDGMENTS

This study is partially supported by the Defense Advanced Research Projects Agency, Grant No. D19AP00035. The authors also acknowledge the Research Computing Center at Florida State University, which was made available for conducting the research reported in this paper.

- [1] M. Goman and A. Khrabrov, State-space representation of aerodynamic characteristics of an aircraft at high angles of attack, *J. Aircr.* **31**, 1109 (1994).
- [2] K. W. McAlister, O. Lambert, and D. Petot, Application of the onera model of dynamic stall, Aeromechanics Laboratory AVSCOM, Tech. Rep. 2399 (U.S. Army Research and Technology Laboratories, 1984).
- [3] H. Babinsky, R. J. Stevens, A. R. Jones, L. P. Bernal, and M. V. Ol, Low order modelling of lift forces for unsteady pitching and surging wings, in *54th AIAA Aerospace Sciences Meeting* (AIAA SciTech, San Diego, CA, 2016), p. 0290.
- [4] L. Librescu, S. Na, P. Marzocca, C. Chung, and M. K. Kwak, Active aeroelastic control of 2-D wing-flap systems operating in an incompressible flowfield and impacted by a blast pulse, *J. Sound Vib.* **283**, 685 (2005).
- [5] J. J. Block and T. W. Strganac, Applied active control for a nonlinear aeroelastic structure, *J. Guid. Control. Dyn.* **21**, 838 (1998).
- [6] Z. Wang, A. Behal, and P. Marzocca, Model-free control design for multi-input multi-output aeroelastic system subject to external disturbance, *J. Guid. Control. Dyn.* **34**, 446 (2011).
- [7] K. Zhang, P. Marzocca, and A. Behal, Adaptive aeroelastic control of nonlinear airfoil with multiple flaps under unsteady flow, *J. Vib. Control* **23**, 1593 (2017).
- [8] K. W. Lee and S. N. Singh, Robust finite-time continuous control of an unsteady aeroelastic system, *J. Guid. Control. Dyn.* **41**, 978 (2018).
- [9] J. E. Pohl, R. Radespiel, B. Herrmann, S. L. Brunton, and R. Semaan, Gust mitigation through closed-loop control. I. Trailing-edge flap response, *Phys. Rev. Fluids* **7**, 024705 (2022).
- [10] B. Herrmann, S. L. Brunton, J. E. Pohl, and R. Semaan, Gust mitigation through closed-loop control. II. Feedforward and feedback control, *Phys. Rev. Fluids* **7**, 024706 (2022).
- [11] G. Plataniotis and T. W. Strganac, Control of a nonlinear wing section using leading-and trailing-edge surfaces, *J. Guid. Control. Dyn.* **27**, 52 (2004).
- [12] T. Theodorsen, General theory of aerodynamic instability and the mechanism of flutter, NACA Technical Report No. NACA-TR-496, 1935.
- [13] M. S. Hemati, S. T. Dawson, and C. W. Rowley, Parameter-varying aerodynamics models for aggressive pitching-response prediction, *AIAA J.* **55**, 693 (2017).
- [14] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *J. Fluid Mech.* **497**, 335 (2003).
- [15] C. W. Rowley, T. Colonius, and R. M. Murray, Model reduction for compressible flows using POD and Galerkin projection, *Physica D* **189**, 115 (2004).
- [16] M. J. Balajewicz, E. H. Dowell, and B. R. Noack, Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation, *J. Fluid Mech.* **729**, 285 (2013).
- [17] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *Int. J. Bifurcation Chaos* **15**, 997 (2005).
- [18] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA J.* **40**, 2323 (2002).
- [19] B. O. Koopman, Hamiltonian systems and transformation in hilbert space, *Proc. Natl. Acad. Sci. USA* **17**, 315 (1931).
- [20] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, *Annu. Rev. Fluid Mech.* **45**, 357 (2013).
- [21] C. E. Garcia, D. M. Prett, and M. Morari, Model predictive control: Theory and practice—A survey, *Automatica* **25**, 335 (1989).
- [22] M. Morari and J. H. Lee, Model predictive control: Past, present, and future, *Comput. Chem. Eng.* **23**, 667 (1999).
- [23] J. H. Lee, Model predictive control: Review of the three decades of development, *Int. J. Control Autom. Syst.* **9**, 415 (2011).
- [24] E. F. Camacho and C. B. Alba, *Model Predictive Control* (Springer Science & Business Media, 2013).
- [25] H. Arbabi, M. Korda, and I. Mezić, A data-driven Koopman model predictive control framework for nonlinear partial differential equations, in *2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 2018* (IEEE, Piscataway, NJ, 2018), pp. 6409–6414.

- [26] S. Peitz, S. E. Otto, and C. W. Rowley, Data-driven model predictive control using interpolated Koopman generators, *SIAM J. Appl. Dyn. Syst.* **19**, 2162 (2020).
- [27] N. Fonzi, S. L. Brunton, and U. Fasel, Data-driven nonlinear aeroelastic models of morphing wings for control, *Proc. R. Soc. A* **476**, 20200079 (2020).
- [28] T.-K. Wang and K. Shoele, Geometrically weighted modal decomposition techniques, *J. Fluid Mech.* **911**, A41 (2021).
- [29] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, Modern Koopman theory for dynamical systems, *SIAM Rev.* **64**, 229 (2022).
- [30] I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dyn.* **41**, 309 (2005).
- [31] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *J. Fluid Mech.* **656**, 5 (2010).
- [32] S. Bagheri, Koopman-mode decomposition of the cylinder wake, *J. Fluid Mech.* **726**, 596 (2013).
- [33] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: Theory and applications, *J. Comput. Dyn.* **1**, 391 (2014).
- [34] J. L. Proctor, S. L. Brunton, and J. N. Kutz, Dynamic mode decomposition with control, *SIAM J. Appl. Dyn. Syst.* **15**, 142 (2016).
- [35] M. Korda and I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica* **93**, 149 (2018).
- [36] S. Peitz and S. Klus, Feedback control of nonlinear pdes using data-efficient reduced order models based on the Loopman operator, in *The Koopman Operator in Systems and Control* (Springer, 2020), pp. 257–282.
- [37] E. Kaiser, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proc. R. Soc. A* **474**, 20180335 (2018).
- [38] F. Takens, Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence, Warwick 1980* (Springer, 1981), pp. 366–381.
- [39] L. Ljung, System identification, in *Signal Analysis and Prediction* (Springer, 1998), pp. 163–173.
- [40] I. Mezić and A. Banaszuk, Comparison of systems with complex behavior, *Physica D* **197**, 101 (2004).
- [41] T.-K. Wang and K. Shoele, Mode competition in a plunging foil with an active flap: A multiscale modal analysis approach, *Phys. Rev. Fluids* **7**, 044701 (2022).
- [42] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica* **36**, 789 (2000).
- [43] L. Grüne and J. Pannek, Nonlinear model predictive control, in *Nonlinear Model Predictive Control* (Springer, 2017), pp. 45–69.
- [44] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, qpOASES: A parametric active-set algorithm for quadratic programming, *Math. Program. Comput.* **6**, 327 (2014).
- [45] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, OSQP: An operator splitting solver for quadratic programs, *Math. Program. Comput.* **12**, 637 (2020).
- [46] B. A. Turlach and A. Weingessel, quadprog: Functions to solve quadratic programming problems, CRAN-Package quadprog (2007).
- [47] D. Kouzoupis, A. Zanelli, H. Peyrl, and H. J. Ferreau, Towards proper assessment of QP algorithms for embedded model predictive control, in *2015 European Control Conference (ECC)* (IEEE, 2015), pp. 2609–2616.
- [48] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, Online dynamic mode decomposition for time-varying systems, *SIAM J. Appl. Dyn. Syst.* **18**, 1586 (2019).
- [49] T. Nonomura, H. Shibata, and R. Takaki, Dynamic mode decomposition using a Kalman filter for parameter estimation, *AIP Adv.* **8**, 105106 (2018).
- [50] J. Kou and W. Zhang, Data-driven modeling for unsteady aerodynamics and aeroelasticity, *Prog. Aerosp. Sci.* **125**, 100725 (2021).
- [51] E. A. Deem, L. N. Cattafesta, M. S. Hemati, H. Zhang, C. Rowley, and R. Mittal, Adaptive separation control of a laminar boundary layer using online dynamic mode decomposition, *J. Fluid Mech.* **903**, A21 (2020).

- [52] J. Kou and W. Zhang, Dynamic mode decomposition with exogenous input for data-driven modeling of unsteady flows, *Phys. Fluids* **31**, 057106 (2019).
- [53] R. Ma, Y. Yang, M. Li, and Q. Li, The unsteady lift of an oscillating airfoil encountering a sinusoidal streamwise gust, *J. Fluid Mech.* **908**, A22 (2021).
- [54] A. R. Jones, O. Cetiner, and M. J. Smith, Physics and modeling of large flow disturbances: Discrete gust encounters for modern air vehicles, *Annu. Rev. Fluid Mech.* **54**, 469 (2022).
- [55] R. E. Kalman, A new approach to linear filtering and prediction problems, *J. Basic Eng.* **82**, 35 (1960).
- [56] D. Darakananda, A. F. C. da Silva, T. Colonius, and J. D. Eldredge, Data-assimilated low-order vortex modeling of separated flows, *Phys. Rev. Fluids* **3**, 124701 (2018).
- [57] M. Le Provost, R. Baptista, Y. Marzouk, and J. Eldredge, [A low-rank nonlinear ensemble filter for vortex models of aerodynamic flows](#), in *AIAA Scitech 2021 Forum* (AIAA, 2021), p. 1937.