

Data-Driven Flow Control for Finite Wings at Low Reynolds Numbers

Anton Burtsev*, Akshit Jariwala†, Efstathios Bakolas‡ and David Goldstein§
University of Texas at Austin, Austin, TX, 78712, USA

Numerical flow control studies are conducted on a NACA 0015 airfoil and finite wing of aspect ratio 4 at a chord based Reynolds number of 200 and high angles of attack (α). Direct Numerical Simulation (DNS) of laminar separated flows is conducted in the Nek5000 spectral element code. A time-varying state space model of the fluid flow that is amenable to control design is identified by using a data-driven modal decomposition. An optimal stabilizing controller that can (asymptotically) stabilize the flow is obtained by solving a linear quadratic optimal control problem and is introduced in the form of feedback control in the DNS as a body force actuation. The performance of the online control based on the time-varying model is compared to that of the time-invariant one. The ability of this control scheme to adapt to changes of the system is tested by changing the angle of attack α . The online control approach based on the time-varying model works well for small changes in α but is unable to fully stabilize the flow when the increase in α is above 4° . The time-varying model offers more accurate predictions, but is more sensitive to modifications of the flow. Further work is required to fully leverage its advantages for control design.

I. Nomenclature

$\alpha, \Delta\alpha$	=	angle of attack and change in angle of attack ($^\circ$)
c	=	wing chord
t	=	non-dimensional time
$\delta t, \Delta t$	=	DNS time step and linear model discrete step
$\mathbf{f}$	=	vector of body forces
p	=	pressure
$\mathbf{u}$	=	velocity vector
U_∞	=	free-stream velocity
$\mathbf{u}_k$	=	control input vector at step k
K	=	controller gain
k	=	discrete time state-space model step
l	=	control step
κ	=	oDMDC weighting factor
A_k, B_k	=	state and input matrices of the state-space model at step k
N	=	time horizon
n	=	number of states
m	=	number of control inputs
q	=	number of temporal snapshots
Re	=	Reynolds number
ν	=	kinematic viscosity
λ	=	eigenvalue of the state matrix
$\mathbf{x}_k$	=	state vector at step k
X	=	snapshot matrix

*Postdoctoral fellow, Department of Aerospace Engineering and Engineering Mechanics, AIAA Member, anton.burtsev@austin.utexas.edu

†Graduate Student, Department of Aerospace Engineering and Engineering Mechanics

‡Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Senior Member

§Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Associate Fellow

II. Introduction

AERODYNAMIC flow control aims to enhance the performance of lifting surfaces often by improving lift and drag characteristics, delaying separation, and reducing the oscillatory loads due to vortex shedding. Since the goal of any flow control is to alter the flow dynamics with as little actuation as possible, the approach of many studies is to leverage the inherent hydrodynamic instabilities. Using the physical knowledge of the flow, small control inputs can be applied in regions where the instabilities are most receptive to actuation and can greatly affect the flow.

Flow control aimed at mitigating separation over lifting bodies has predominantly relied on exploiting the receptivity of the separated flow to periodic actuation within a narrow frequency range. A lot of effort has been devoted to studies of various actuation parameters with the effects on the flow evaluated *a posteriori* [1–3].

In the past decade, there has been an abundance of studies analyzing the wake dynamics of finite-wing flows at very low Reynolds number. Many studies focused on analyzing the vortex dynamics during unsteady maneuvers such as translation and rotation [4, 5], surging and plunging [6, 7], pitching [8–10], and flapping [11, 12]. The flows at these conditions are massively separated and dominated by unsteady vortex shedding. Large scale vortex structures present in these unsteady flows, such as leading edge vortices [13, 14], can augment unsteady vortical lift. The use of these structures for flow control [15] have also been studied.

Despite there being a significant body of work on finite wings, the applications of adaptive active feedback control for these flows have been rather scarce. Neural networks have been used for adaptive system identification and control for lift augmentation over an airfoil based on limited surface measurements [16] and for wake stabilization and drag reduction [17]. While certainly successful, such approaches do not fully account for the flow physics. On the other hand, data-driven modal decompositions, such as the widely used dynamic mode decomposition (DMD) [18], are known to capture the physics of the limit cycle oscillations of a separated flow. In particular, DMD can identify various instabilities as discrete frequency modes. Online DMD with control was used by [19] to iteratively update a dynamic model and recompute the feedback gains of a linear quadratic regulator (LQR) in experimental control of separation over a plate. While still data-driven, such an approach relies on a physics based reduced order model of the flow.

This work is motivated by the fact that modern goals for micro-aerial vehicles include flights at stalled conditions and in unsteady (gusty) environments [20, 21] where active flow control may be necessary for safe operations. So far, adaptive active feedback control has not been applied for a finite wing at such conditions. Hence, the aim of this work is to develop an adaptive closed-loop control approach for unsteady separated flow. We conduct numerical flow control simulations of separated flows over airfoils and finite-aspect-ratio wings at low Reynolds numbers and large angles of attack. The low Reynolds number conditions serve as a simpler training ground before attempting higher Re regimes. To check the ability of the control scheme to cope with variations in conditions, we subject the controlled flow to a quick increase of the angle of attack. The rest of the paper is organized as follows: the data-driven algorithms used for system identification and the control approach are explained in the following section. The results section starts with a comparison of online DMD and classical DMD for prediction of the evolution of the system subjected to a change of the angle of attack. The online control approach is then applied to two-dimensional airfoil flow and eventually to a three-dimensional finite wing flow.

III. Methodology

A. Equations of Motion

The flow is governed by nondimensional, incompressible Navier-Stokes and continuity equations:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + Re^{-1} \nabla^2 \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \quad (1)$$

where $\mathbf{u} = (u, v, w)^T$ is the velocity vector, p is pressure and $\mathbf{f}$ are the external body forces acting on the flow. The Reynolds number defined as $Re \equiv U_\infty c / \nu$, where U_∞ is the free-stream velocity, c is the wing chord, and ν the kinematic viscosity. These governing equations are solved by direct numerical simulation (DNS) using the spectral element code Nek5000 [22]. The computational domain extends $15c$ upstream and $20c$ downstream in x , $\pm 15c$ in y and $15c$ past the wing tip in z for the three-dimensional case. The domain is discretized by a C-type mesh with 2438 (46,000 in the three-dimensional case) quadrilateral elements. Within each element, the solution variables are represented in terms tensor-product Lagrange polynomials of order $P = 7$. A DNS time step $\delta t = 10^{-3}$ is used.

B. Actuators

For the numerical simulations reported here, the control actuators were modeled as a spatially localized body forces acting on the flow. This is similar to the actuation approach previously used by [23, 24]. For every actuator the force at any point in the domain at step k is given by

$$\mathbf{f} = \mathbf{e}_\theta G_{\text{act}}(x, y, z) G_{\text{const}} \mathbf{u}_k \quad (2)$$

where $\mathbf{e}_\theta = [\cos \theta, \sin \theta, 0]^T$ is a unit vector specifying the direction of the forcing which is pitched by θ with respect to the free stream, $\mathbf{u}_k$ is the control input for the actuator, and G_{const} is a constant used to scale the input. The location where the actuator affects the flow is determined by $G_{\text{act}}(x, y, z)$, which is a spatial distribution of the following form:

$$G_{\text{act}}(x, y, z) = \exp \left(-0.5 \left(\frac{(x - x_a)^2}{a_x^2} + \frac{(y - y_a)^2}{a_y^2} + \frac{(z - z_a)^2}{a_z^2} \right) \right), \quad (3)$$

where x_a, y_a, z_a are the coordinates of the actuator and a_x, a_y, a_z are parameters determining the shape of the distribution in respective directions. The value of G_{act} is set to zero at locations where $G_{\text{act}} < 0.01$. The values of $a_x = a_y = a_z = 0.02$ and $G_{\text{const}} = 10$ were chosen, which adequately scale the forcing region.

The control setup is shown in Fig. 1. The green contours of G_{act} show the region where the body forcing is applied to the flow. This is located just upstream of the separation point of the uncontrolled flow. For the the finite wing case, this region extends to $z/c = 3.7$ in the spanwise direction and approximates a slot. The blowing ratio of the body force actuator for the chosen (a_x, a_y, a_z) and G_{const} values was characterized by measuring the streamwise velocity U_{act} induced by the forcing at the location of the actuator (x_a, y_a, z_a) , i.e. the center of the spatial distribution of Eq. 3. This test was conducted in the two-dimensional domain without the airfoil (but otherwise identical conditions) and with the forcing direction aligned with the free stream. The ratio of the steady state velocity surplus due to actuation to the free stream velocity was computed. The dependence of this blowing ratio on the magnitude of the control signal u_k supplied to the actuator is plotted in Fig. 1c).

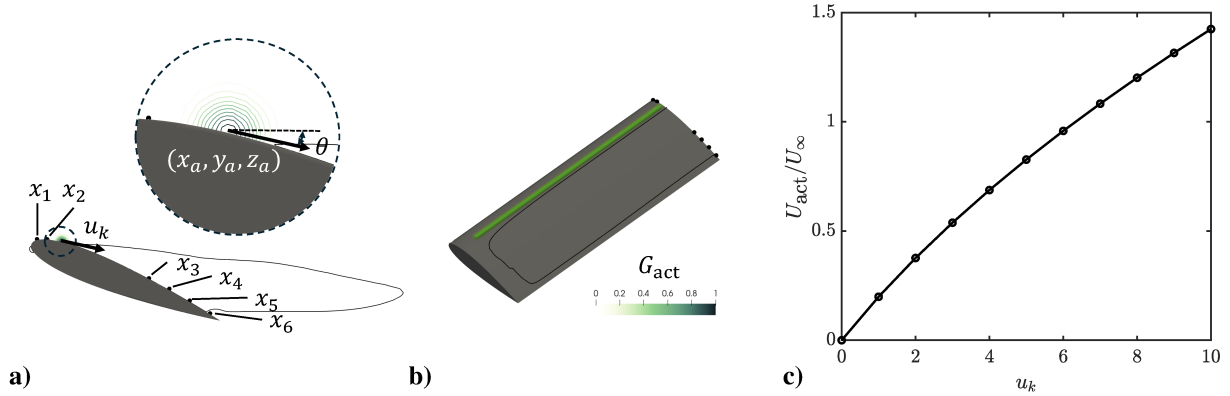


Fig. 1 Actuator setup for the airfoil a) and finite wing b) setup showing the location of the state measurements and the control force. Green contours indicate the spatial distribution of the actuator body force. Black lines indicate the reversed flow region of the uncontrolled flow. The blowing ratio U_{act}/U_∞ corresponding to different values of the input signal u_k for a two-dimensional case c).

C. System Identification

1. Dynamic Mode Decomposition

The classical DMD describes a linear system of the form $\mathbf{x}_{k+1} = A_{\text{DMD}} \mathbf{x}_k$, which can be written in matrix form as

$$\mathbf{X}' = A_{\text{DMD}} \mathbf{X}, \quad (4)$$

where $\mathbf{X}$ is the matrix of snapshots $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{l-1}]$ and $\mathbf{X}'$ is $\mathbf{X}$ shifted by one snapshot $\mathbf{X}' = [\mathbf{x}_2, \dots, \mathbf{x}_l]$. The least-squares problem in Eq. 4 can be solved as

$$A_{\text{DMD}} = \mathbf{X}' \mathbf{X}^+, \quad (5)$$

where the superscript $+$ denotes the pseudo-inverse defined as $X^+ = X^T [XX^T]^{-1}$. Typically, a reduced order problem is formulated by taking the singular value decomposition (SVD) of the snapshot matrix $X = U\Sigma V^T$. This results in

$$A_{\text{DMD}} = X'V\Sigma^{-1}U^T. \quad (6)$$

The SVD can also be optionally truncated to r , with $r < \min(m, q - 1)$. The DMD matrix can then be projected onto the space spanned by the first r columns of X

$$\tilde{A} = \tilde{U}^T A_{\text{DMD}} \tilde{U} = \tilde{U}^T X' \tilde{V} \tilde{\Sigma}^{-1}, \quad (7)$$

where $X = \tilde{U} \tilde{\Sigma} \tilde{V}^T$ is the truncated SVD. The eigenvalue problem (EVP)

$$\tilde{A} \tilde{W} = \lambda_i \tilde{W} \quad i = 1, 2, \dots, r, \quad (8)$$

gives the DMD eigenvalues, λ . The matrix $\tilde{W}$ contains the eigenvectors of the reduced order problem. The corresponding DMD eigenvectors can be obtained by

$$W = \tilde{U} \tilde{W}. \quad (9)$$

The (continuous-time) oscillation frequency f and the growth rate g of a DMDc mode can be determined from the (discrete-time) eigenvalue λ of matrix A using the following equations:

$$f = \angle \lambda / (2\pi \Delta t), \quad (10)$$

$$g = \log |\lambda| / \Delta t, \quad (11)$$

where Δt is the time step of the linear model.

2. Dynamic Mode Decomposition with Control

Dynamic mode decomposition with control (DMDc) [25] is an extension of the widely used dynamic mode decomposition (DMD) method [18]. The main feature of DMDc is its ability to distinguish the underlying system dynamics from the effects of actuation, which produces accurate input-output models for an actuated system. The training snapshots are arranged in three matrices

$$X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{m-1}], \quad X' = [\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_m], \quad U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{m-1}], \quad (12)$$

where X is the $n \times m - 1$ snapshot matrix, X' is the snapshot matrix shifted by one snapshots and U is the $m \times q - 1$ control matrix containing the corresponding actuator signals. Here n is the size of the control grid, q is the total number of snapshots and m is the number of control actuators. The DMDc input matrix is constructed by appending the training data control matrix to the snapshot matrix

$$\Omega = \begin{bmatrix} X \\ U \end{bmatrix}. \quad (13)$$

Next, the SVD of the input matrix is taken

$$\Omega = U_\Omega \Sigma_\Omega V_\Omega^T. \quad (14)$$

The state and input matrices of the reduced order system are then computed as follows:

$$A = X' V_\Omega^T \Sigma_\Omega^{-1} U_1^T, \quad (15)$$

$$B = X' V_\Omega^T \Sigma_\Omega^{-1} U_2^T, \quad (16)$$

where U_1 includes the 1 to n rows of U_Ω and U_2 includes the $n + 1$ to q rows of U_Ω . Assuming that A has a set of linearly independent eigenvectors, the dynamic modes are then obtained by solving the eigenvalue problem

$$AW = W\Lambda, \quad (17)$$

where Λ is a diagonal matrix of complex DMDc eigenvalues λ and W is a matrix containing the corresponding eigenvectors in its columns. The A and B matrices obtained with DMDc can be used to construct a discrete-time linear (control) system:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k, \quad (18)$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are the state vectors of the state-space model at the discrete stage k and $k + 1$, respectively. The discrete stage k corresponds to time t_k and discrete stage $k + 1$ corresponds to time t_{k+1} with $t_{k+1} = t_k + \Delta t$.

3. Online Dynamic Mode Decomposition with Control

The classical DMD [26] formulation requires access to all prior snapshots to form the data matrix and compute the singular value decomposition. Therefore, system identification using DMD with control (DMDc) [25] can only be done once enough state vector measurements are collected as training data. Such *a posteriori* analysis is not suitable for situations when a sufficient number of state vector measurements at different time instances is not available or if the data matrix is too large to store in memory. If the flow in question significantly changes its dynamics, for example due to external disturbances, these will not be accounted for unless they were present in the original training data. Once a control strategy is employed, the effect of the control input itself can significantly vary the flow dynamics, potentially rendering the original model obsolete. Online dynamic mode decomposition (oDMD) [27] provides a way to continuously update the DMD estimate, representing the flow dynamics as a time-varying linear system. The added benefit is that the full data matrix does not have to be formed or stored. This makes oDMDc particularly attractive for problems with low spatial resolution (low dimension of the state space n) but high temporal resolution of measurements. The extension of oDMD for control based on the approach of [28] is summarized below.

Recall that the classical DMD describes a linear system of the form $\mathbf{x}_{k+1} = A_{\text{DMD}}\mathbf{x}_k$, which can be written in matrix form as Eq. 4. The resulting least-squares problem in Eq. 5 can be written as

$$A_{\text{DMD}} = MP, \quad (19)$$

where $M = X'X^T$ and $P = [XX^T]^{-1}$. At step k , a new state measurement, $\mathbf{x}_k$, and the previous state measurement, $\mathbf{x}_{k-1}$, are available and the matrices M and P can be updated using

$$M_k = [X' \ \mathbf{x}_k] [X \ \mathbf{x}_{k-1}]^T = M_{k-1} + \mathbf{x}_k \mathbf{x}_{k-1}^T, \quad (20)$$

$$P_k = \left([X \ \mathbf{x}_{k-1}] [X \ \mathbf{x}_{k-1}]^T \right)^{-1} = \left[P_{k-1}^{-1} + \mathbf{x}_{k-1} \mathbf{x}_{k-1}^T \right]^{-1}. \quad (21)$$

DMD can be expanded to include control inputs by simply taking $\tilde{A} = [A \ B]$ and substituting $\mathbf{x}_k$ for $[\mathbf{x}_k \ \mathbf{u}_k]^T$ as demonstrated in [25]. After some algebraic manipulations, the details of which can be found in [27], the following set of update equations (Eq. (22-24)), which include the effect of control, can be written for step k as follows:

$$P_k = \frac{1}{\kappa} \left(P_{k-1} + \gamma_k P_{k-1} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix}^T P_{k-1} \right), \quad (22)$$

where $\mathbf{x}_k$ is the new state vector value, $\mathbf{x}_{k-1}$ is the previous state measurement and $\mathbf{u}_{k-1}$ is the previous control input. Note that that P_k in Eq. 22 is $(n+m) \times (n+m)$ to account for the fact that the state measurement vectors are of size n and the control vector is of size m , with n and m being the number of states and actuators, respectively. Here, $\mathbf{u}_{k-1}$ is the control input applied at $k-1$ to get from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$. The scalar γ_k in Eq. 22 is defined as

$$\gamma_k = 1 + \left(1 + \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix}^T P_{k-1} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix} \right). \quad (23)$$

Equation 22 has a weighting factor $\kappa \in (0, 1]$, which affects the contribution of older snapshots to the current system estimate. Setting a low value ($\kappa \rightarrow 0$) will aggressively attenuate old data, while a value close to unity will result in a gradual decay of old snapshots. The state-space model can be updated using

$$\begin{bmatrix} A_k & B_k \end{bmatrix} = \begin{bmatrix} A_{k-1} & B_{k-1} \end{bmatrix} + \gamma_k \left(\mathbf{x}_k - \begin{bmatrix} A_{k-1} & B_{k-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_{k-1} \\ \mathbf{u}_{k-1} \end{bmatrix}^T P_{k-1}, \quad (24)$$

which considers the error between the current measured state $\mathbf{x}_k$ and its prediction using an “old” state-space model and state at $k-1$. This algorithm can be initialized either with normal DMDc or using a random $[A_0 \ B_0]$ and setting $P_0 = \beta I$, where I is the identity matrix and β is a large positive scalar. This results in a time-varying (discrete-time) state-space model

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k, \quad (25)$$

that is amenable to controller design.

D. Controller Design

We apply two standard controllers to the discrete-time system defined in Eq. (25). For the purposes of controller design, the matrices A and B are treated as if they were constant for the whole time horizon over which the controller is designed (which means that every time A and / or B change, the control design process must be repeated based on the most recent values of these two matrices). In the following, we use $\hat{A} = A_k$ and $\hat{B} = B_k$ to denote these constant state and input matrices obtained from the current estimate of the online DMDc. We then apply time-invariant versions of the controllers on this time-invariant system $\mathbf{x}_{k+1} = \hat{A}\mathbf{x}_k + \hat{B}\mathbf{u}_k$. The resulting gains will not be strictly optimal in the optimal control sense since the actual system in Eq. (25) to which we will apply the feedback is time-varying and nonlinear. However, for the current estimate of the oDMDc when the gains are recomputed, they will be in the set of optimal feedback gains.

1. Linear Quadratic Regulator

We currently employ the classical linear quadratic regulator (LQR) controller [29], which in its infinite horizon formulation has a performance index defined by

$$J = \sum_{k=0}^{\infty} \left(\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k \right), \quad (26)$$

to be minimized subject to time-invariant dynamics $\mathbf{x}_{k+1} = \hat{A}\mathbf{x}_k + \hat{B}\mathbf{u}_k$. Here, Q and R are positive definite diagonal weight matrices of sizes $n \times n$ and $m \times m$, respectively. The value of Q determines how much weight should be placed on that state component (how far the state is from the origin). Larger Q_{ii} values mean that it is more important to minimize the error in the component x_i of the state vector. Similarly, the value of R for a specific actuator penalizes actuator usage, with lower value of R_{jj} meaning that one is allowed to use control input vectors whose component u_j (or more precisely, the absolute value of the latter) can be larger than the other input components. The LQR controller gain $\hat{K}$ is defined as

$$\hat{K} = \left(R + \hat{B}^T S \hat{B} \right)^{-1} \hat{B}^T S \hat{A}, \quad (27)$$

where S is the solution of the discrete algebraic Riccati equation (DARE)

$$S = \hat{A}^T S \hat{A} - \hat{A}^T S \hat{B} \left(R + \hat{B}^T S \hat{B} \right)^{-1} \hat{B}^T S \hat{A} + Q. \quad (28)$$

The optimal control input for step k is then applied to the system as closed loop feedback

$$\mathbf{u}_k = -\hat{K}\mathbf{x}_k, \quad (29)$$

where $\mathbf{x}_k$ is the current state value.

2. Linear Quadratic Tracker

As time approaches infinity, the LQR controller aims to bring the state vector and the control input vector to zero. If it is needed to reach a non-zero state, or if the final steady state requires a non-zero steady-state control input, the linear quadratic tracker (LQT) extension of LQR can be used instead [29]. The LQT performance index is defined as

$$J_i = \frac{1}{2} (\mathbf{x}_N - \bar{\mathbf{x}}_N)^T \Pi (\mathbf{x}_N - \bar{\mathbf{x}}_N) + \frac{1}{2} \sum_{k=i}^{i+N-1} \left((\mathbf{x}_k - \bar{\mathbf{x}}_k)^T Q (\mathbf{x}_k - \bar{\mathbf{x}}_k) + \mathbf{u}_k^T R \mathbf{u}_k \right), \quad (30)$$

where $\{\bar{\mathbf{x}}_k\}$ is the reference signal that the system's state trajectory must track and N (positive integer) is the (finite) time horizon. The weight matrices Q and R have the same meaning as for the LQR problem. The additional symmetric, positive definite matrix Π weights the steady-state error between the state vector $\mathbf{x}_N$ and the reference (target) state vector $\bar{\mathbf{x}}_N$. The LQT optimal control input contains a feedback and a feed-forward term

$$\mathbf{u}_k = -K_k \mathbf{x}_k + K_k^v \mathbf{v}_{k+1}. \quad (31)$$

The feedback gain K_k is defined as

$$K_k = \left(\hat{B}^T S_{k+1} \hat{B} + R \right)^{-1} \hat{B}^T S_{k+1} \hat{A}. \quad (32)$$

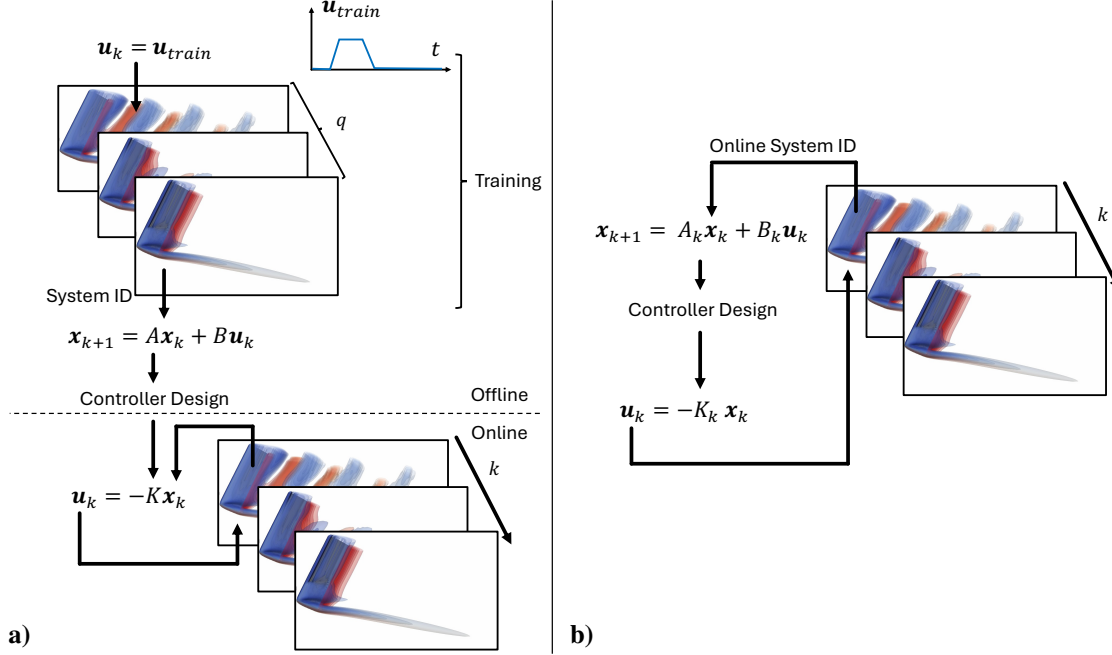


Fig. 2 Conceptual overview of the normal, DMDc based a) and the online oDMDc b) control methods.

Here the matrix S_{k+1} is the solution of the Riccati difference equation, which for the case of full state information is given by

$$S_k = \hat{A}^T S_{k+1} (\hat{A} - \hat{B} K_k) + Q, \quad (33)$$

and is solved backwards in time. Note that $\hat{A}$ and $\hat{B}$ are assumed to be constant for the time horizon N . The feed-forward gain K^v is given by,

$$K_k^v = (\hat{B}^T S_{k+1} \hat{B} + R)^{-1} \hat{B}^T \quad (34)$$

where

$$v_k = (\hat{A} - \hat{B} K_k)^T v_{k+1} + Q \bar{x}_k. \quad (35)$$

Eqs. (32) and (35) are iterated backwards in time starting at step N with the starting conditions of given by $S_N = \Pi$ and $v_N = \Pi \bar{x}_N$, respectively. Note that in contrast with the infinite horizon LQR problem, the LQT gains are, in general, time-varying even for the time-invariant system defined by $\hat{A}$ and $\hat{B}$.

E. Summary of the Flow Control Approach

The overall idea of the two control schemes used is shown in Fig. 2. The first step of classical, DMDc-based control is the system identification process. An arbitrary control input, which we will refer to as the training input, is applied to the actuator and the evolution of the flow is recorded for q steps. The frequency of the training input is chosen such as to excite the most dominant modes of the system that one wishes to infer. We have tested several such inputs including a one period of a sin wave at the shedding frequency and step inputs of finite duration. A trapezoidal step as shown in Fig. 2a) was chosen. Experiments with adding Gaussian noise of various amplitudes to the step training input did not show any significant variations in the performance of the system identification process. DMDc is then used to obtain a time-invariant state-space model (Eq. 18). The controller is then designed and the Riccati equation (Eq. 33) is solved once to obtain a time independent gain. The feedback control is obtained by multiplying this gain by the current state vector. In principle, the system identification and controller design can be conducted offline in this case.

In the online control approach shown in Fig. 2b), the oDMDc algorithm is used to update the time-varying state-space model. In principle, there is no need to apply a training control input and the control can be initiated as soon as the oDMDc produces the first system estimate. However, we have still applied the same training input as was used for the

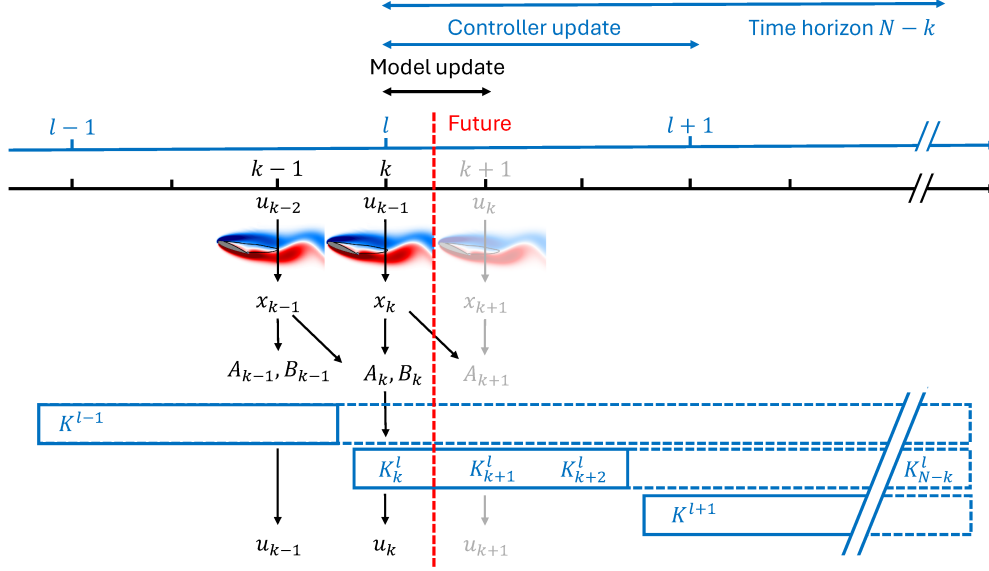


Fig. 3 Time scales of the state-space model and controller updates.

offline approach and initiated the control after q steps to ensure that both methods start with a similar initial system estimate. A controller is then designed. The weighting matrices of the LQT can be modified to place more emphasis on stabilizing the unstable modes, if necessary. Once the controller gains are computed, the control signal is introduced as a linear feedback to the DNS. The corresponding update timescales are shown in Fig. 3. The state-space model is updated at steps k (which is equal to 20 DNS time steps) as schematically illustrated in Fig. 3. At step k the control input u_{k-1} , computed at the previous step $k-1$, is applied to the actuator. The DNS code then solves the flow for that DNS step giving x_k , which along with the previous state vector (x_{k-1}) and the previous state-space model (A_{k-1}, B_{k-1}) is used to update A_k, B_k . The controller gain does not have to be updated so frequently as long as the rate of its updates is sufficient to account for any expected variations of the system. So due to the higher computational expense, the LQT controller is updated every 50 model steps (1000 DNS time steps). Figure 3 shows that if the current step k corresponds to the controller update step l , a new set of time-varying gains K^l is computed by solving the Riccati difference equation for the receding time horizon of $N-k$. The LQT gains are time-varying, so K_k^l denotes the gain for step k computed at the update step l . In between the controller updates, the elements of this array (K_k^l, K_{k+1}^l , etc.) are used to obtain the input u_k . Note that only a portion of K^l is used (indicated by solid blue box) before the controller is updated and the next set of gains K^{l+1} is obtained.

In both cases sparse measurements of pressure on the suction side of the wing are used as states. These are interpolated from the DNS solution at each step k . Both approaches use a classical LQT controller [29]. To obtain the reference steady-state target ($\bar{x}$) for the LQT, several open loop control simulations with increasing magnitude of (constant with time) u_k were conducted. The steady-state values of the state vector for a case which achieved stabilization were used as $\bar{x}$ for the closed loop control runs.

IV. Results

The LQT controller is applied to a two-dimensional airfoil and a three-dimensional finite wing flow. The performance of the online scheme is compared to that based on the time-invariant model. The robustness of the control to changes of the flow is tested by introducing a change in the airfoil angle of attack. Figure 4 shows the process. The starting angle of attack α_0 is constant and equal to 22° . At $t = 200$ the free-stream velocity vector is changed by $\Delta\alpha$, which causes the effective angle of attack of the airfoil to increase to α_1 resulting in an increase of the mean and amplitude of the C_L/C_D fluctuations.

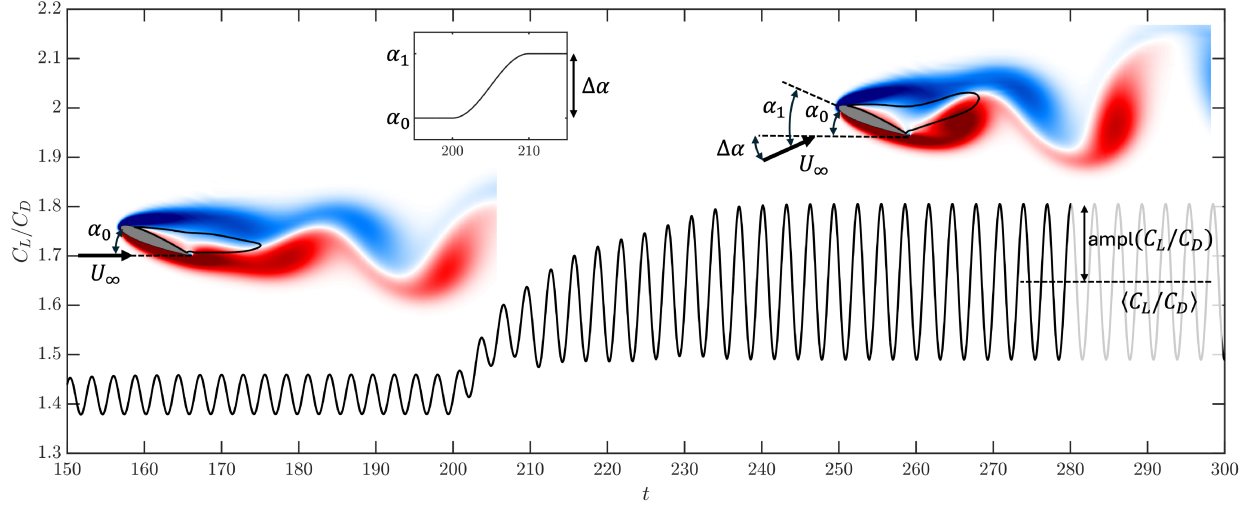


Fig. 4 Change of angle of attack used to test the robustness of the control scheme. Uncontrolled baseline for $\Delta\alpha = 2^\circ$ is shown.

A. Online DMDc for State Prediction

The performance of the oDMDc model in predicting the state evolution with time is tested on a baseline (uncontrolled) case of $\Delta\alpha = 8^\circ$. Figure 5 shows the time evolution of six states (pressure measurements along the suction surface of the airfoil illustrated in Fig 1). The black line is the DNS result. A training control input was applied at $t = 130$ to initialize the oDMDc. The airfoil is pitched up by 8° between $t = 200$ and 210 . The prediction based on a traditional time-invariant DMDc model, trained on data in the interval $120 \leq t \leq 190$, is plotted with blue lines. Online DMDc model was initialized with a random matrix at $t = 120$ and then was continuously updated based on new measurements from the baseline DNS. DMDc prediction cannot account for the changes in the system dynamics caused by the pitch up of the wing at $t = 200$. oDMDc, on the other hand, accurately captured both the mean and the amplitude of the fluctuations with only a minor phase discrepancy with the DNS result.

The good agreement of the oDMDc prediction with the uncontrolled baseline was found to be heavily dependent on the value of the forgetting factor κ in Eq. 22. Figure 6 shows the effect of κ on the percentage errors in the prediction of the mean and amplitude of the state fluctuations (averaged across n states). The errors in the prediction are minimized when κ approached 0.9. However, the time fluctuations of the A_k matrix increase dramatically as shown by the plot of the matrix 2-norm of A_k in Fig. 5a). These large fluctuations of A_k are undesirable for controller design as they will yield different gains. Because of this, the following results were obtained with κ set to 1, for which changes in the norm of A_k are not erratic. This meant that old snapshot data was set to decay gradually. The error in the resulting model is not an issue since the prediction of the states is not actually required for the control strategy. Instead, the control problem is reevaluated repeatedly as described in section III.C.3. Furthermore, the nature of feedback control is somewhat forgiving for inaccuracies of the controller gain (inherent robustness properties of the feedback controller).

B. Two-Dimensional Airfoil Flow

The effect of varying LQT weight is considered for a standard time-invariant DMDc model. Figure 7 shows the variation of C_L/C_D and the corresponding control input produced by the LQT controller applied to the time-invariant DMDc model. For all cases shown, all weight matrices were set to influence all modes equally (constant multiplied by identity matrix), $\Pi = Q$, and $R = 0.1$. The DMDc model was trained on data in the interval $120 \leq t \leq 160$ and the controller was turned on at $t = 160$. The flow was then stabilized in 10-20 time units. The effect of increasing Q was faster flow stabilization and higher final time C_L/C_D but at a cost of higher final time actuator input.

A sample time history from an online control run is shown in Fig. 8. The control input and LQT gains are plotted in Fig. 8a). The oDMDc algorithm is initialized with random values for A . A training control input is applied to the actuator starting at $t = 130$. After 10-15 (chord) time units oDMDc has converged to a state-space model since the norms of A_k and B_k stop fluctuating as one can observe in Fig. 8b). The LQT controller is turned on at $t = 160$ and

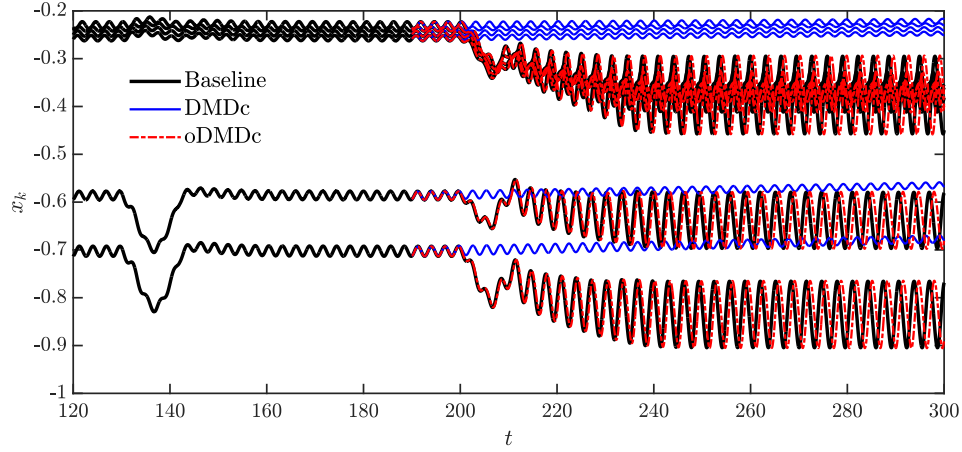


Fig. 5 Comparison of state prediction using different models without control ($\alpha_0 = 22^\circ$, $\Delta\alpha = 8^\circ$, $Re = 200$, $\kappa = 0.9$).

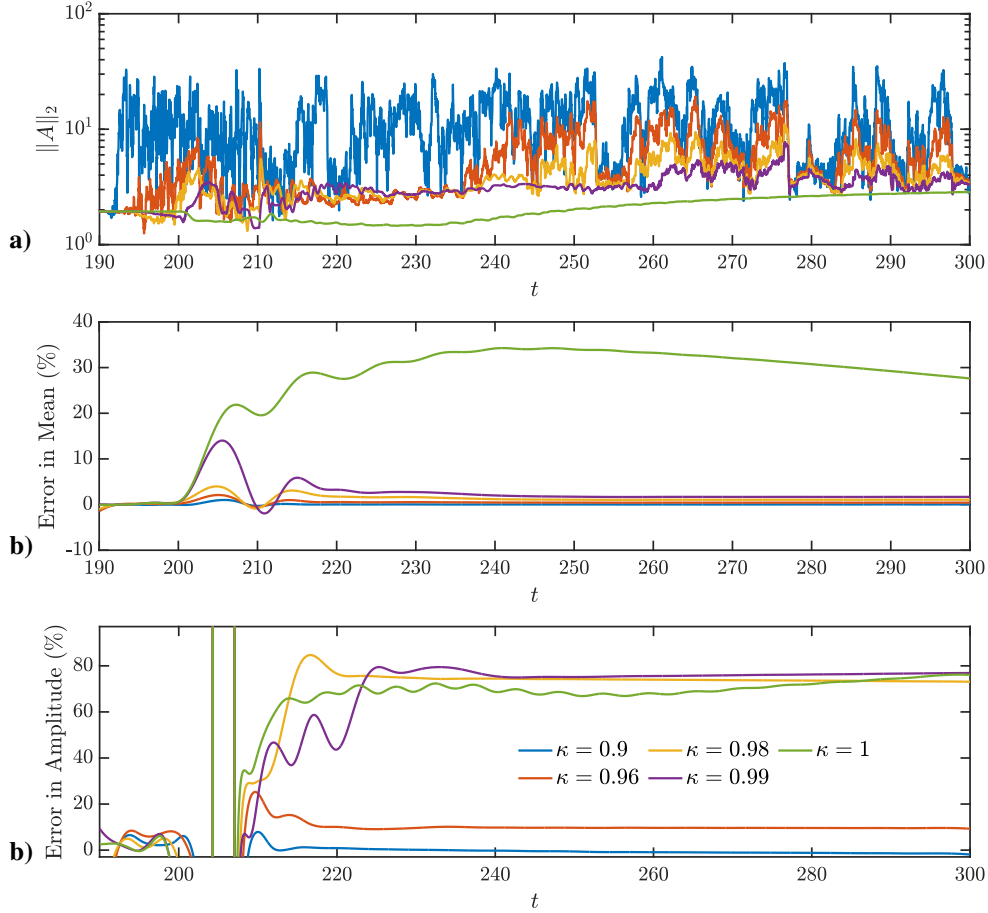


Fig. 6 Effect of the forgetting factor on oDMDc state prediction. Matrix 2-norm of the A matrix a), average error in the predicted mean value of the states b), average error in the predicted amplitude of the states c). ($\alpha_0 = 22^\circ$, $\Delta\alpha = 8^\circ$, $Re = 200$).

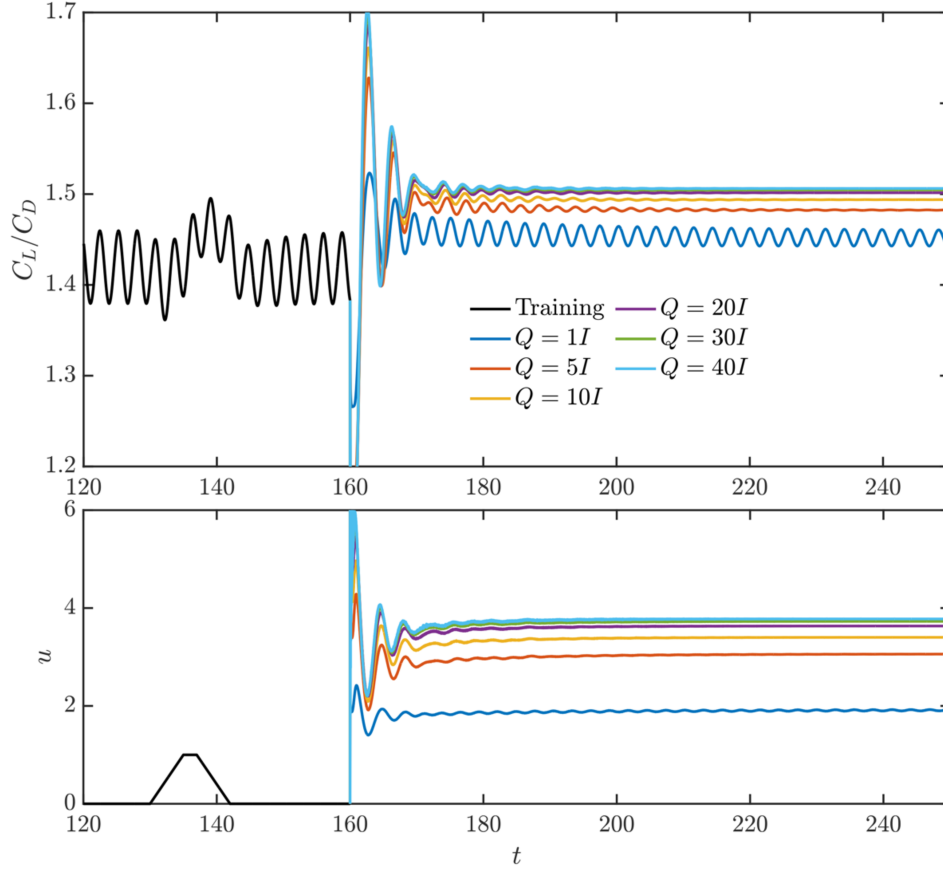


Fig. 7 Effect of the LQT weights for the time-invariant DMDc model ($\alpha_0 = 22^\circ$, $Re = 200$). For all cases Π was equal to Q and $R = 0.1$.

is recomputed every 1 time unit. After some initial transient fluctuations, caused mainly by the modification of the time-varying model due to the effects of control, the online control scheme settles on a gain value which remains constant. When the system is perturbed by the change of the wing incidence angle at $t = 200$, the controller gain is automatically adjusted. The states and the reference targets are plotted in Fig. 8c). The flow is then stabilized in 10-20 time units after the control is turned on.

Figure 9 shows the performance of the LQT controller. The subplots show percentage change of the mean and amplitude of C_L/C_D fluctuations with respect to the uncontrolled baseline. The mean and amplitude of the control input u_k is also shown. The steady state values of both quantities are shown at $290 \leq t \leq 300$. The results in 9a) were obtained with a time-invariant model, where the control gains were evaluated once at $t = 160$ and introduced as a feedback to the flow. The results in 9b) show the time-varying online model, where the controller gains were recomputed every 1 time unit. The values of the actuator weight and the LQT time horizon were held constant at $R = 0.1$ and $t_N = 100$, respectively. The desired target solution of the LQT ($\bar{x}$) is similarly constant for both cases. For both the time-invariant and online cases, the higher values of the Π and Q LQT weights result in better performance of the controller as indicated by a larger increase in $\langle C_L/C_D \rangle$ due to higher control effort. The time-invariant controller fully suppresses wake oscillations for the unperturbed ($\Delta\alpha = 0$) case. The amplitude of control input for flow stabilization is approximately 4, which approximately corresponds to a blowing ratio of 0.68 (see Fig. 1c)). As $\Delta\alpha$ increases, the controller performance gets progressively worse and the control signal becomes more oscillatory as indicated by the growing amplitude of u_k .

Contrary to our expectation, the online control using the time-varying model performed worse. While the mean C_L/C_D values were similar to the time-invariant model, the amplitude of fluctuations was substantially reduced only for the unperturbed case. Moreover, for higher values of $\Delta\alpha$ the fluctuation amplitude of C_L/C_D exceeded that of the

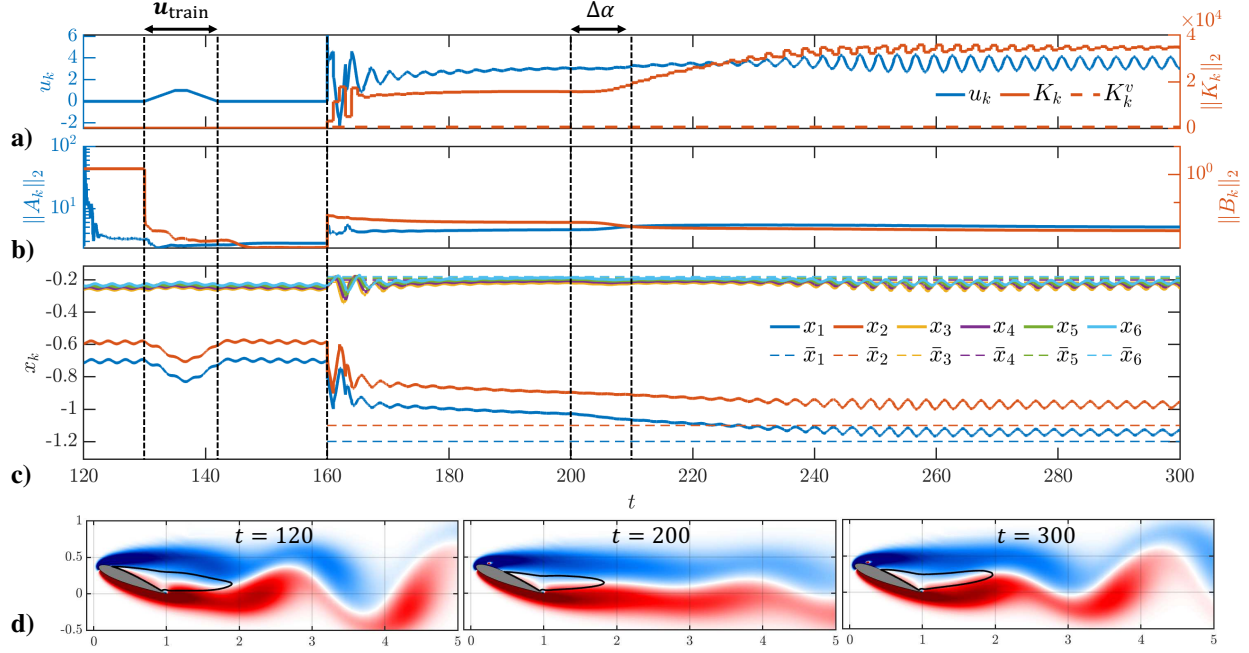


Fig. 8 Time history of the online control run for $\alpha_0 = 22^\circ, \Delta\alpha = 2^\circ, \Pi = Q = 10, R = 0.1, Re = 200$. **a)** Control input and second norms of the LQT gains, **b)** second norm of A_k and B_k , **c)** state vector entries (x) and corresponding targets ($\bar{x}$), **d)** flow field at various times (contours of spanwise vorticity, black line denotes the reversed flow region).

uncontrolled baseline. This was caused by high fluctuations of u_k . It should be noted that the final steady state target provided to the LQT controller was same in both cases. This simplification has not fully utilized the advantage of the online approach, which allows one to vary the target with time as the tracking problem is reevaluated. It is expected that if the reference solution was adjusted to account for the pitch up disturbance, the online control would have likely performed better.

It should be noted that the nonzero mean values of the actuator input contributes to increase of the mean lift in several ways. The reduction of the separation region, Coandă effect due to blowing accelerating the flow on the suction side, and direct contribution of the body force vector components to lift and drag forces all can contribute to increased $\langle C_L/C_D \rangle$. Differentiating the effect of these contributions is not trivial. We are not directly interested in improving the aerodynamic performance as such but rather in reducing the unsteadiness in the flow.

C. Finite Wing Flow

We extend the same control approaches to a finite wing of aspect ratio 4 at the same (chord based) Reynolds number of 200. The same pitch up disturbance as described in Fig. 4 is applied to the three-dimensional wing.

The performance of the online control approach is shown in Fig. 10. Two cases are compared, one with training and one without. In the case with training, a training control signal was first applied to the actuator and the controller was turned on only after the changes in the A_k and B_k of the time-varying model have subsided (similar to Fig. 8). In the case without training, the controller was turned on as soon as the first estimate of A_k and B_k has been produced. In this case the oDMDC algorithm was identifying the system as it was being controlled. The long term performance of these two approaches is nearly identical as shown in Fig. 10. Such ability of the online approach to stabilize the flow immediately, without having to wait for system identification using a training signal is an advantage.

Fig. 11 shows the instantaneous snapshots of the baseline and controlled flow after the pitch-up disturbance. For $\Delta\alpha$ of 2° and 4° , the online control was able to fully stabilize the flow. At $\Delta\alpha = 6^\circ$, full attenuation of shedding was not achieved, however, the amplitude of C_L/C_D fluctuations is reduced by about 50%. At $\Delta\alpha = 8^\circ$, the amplitude of lift fluctuations is unchanged with respect to the baseline as seen in Fig. 10. However, an examination of the controlled

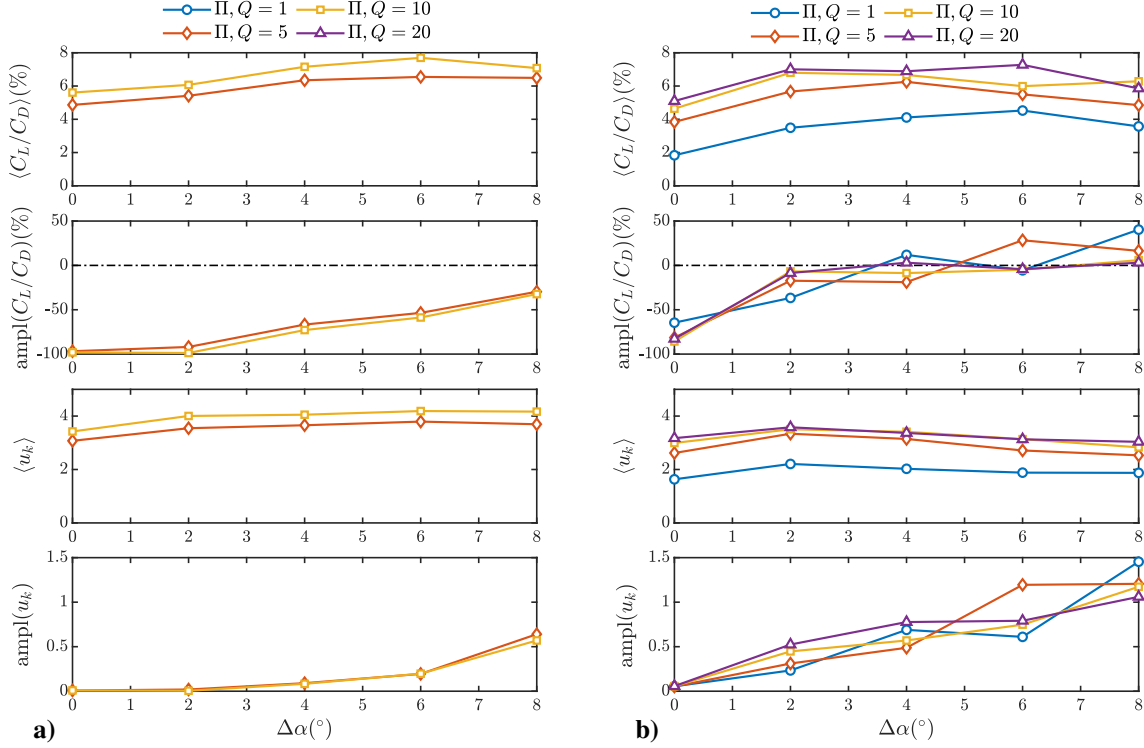


Fig. 9 Comparison of the normal a) and online b) LQT controllers for different $\Delta\alpha$ and Π, Q values for the two-dimensional case ($R = 0.1, \alpha_0 = 22^\circ, Re = 200$).

flow field shown in Fig. 4 for $\Delta\alpha = 6^\circ$ and 8° shows that there is far less three-dimensionality in the flow compared to the baseline. The wake vortices consist of predominantly spanwise vorticity and are mostly parallel to the wing. The vortices with significant streamwise and crossflow vorticity components, which connected spanwise vortices of opposite sign in the uncontrolled flow are significantly weakened as a result of control. To further illustrate this point, the frequency content of the C_L/C_D signal is plotted in Fig. 12 for the uncontrolled baseline and controlled flow. The power spectral density, obtained by a fast Fourier transform of the C_L/C_D signal after the pitch up, is plotted with non-dimensional frequency $St = \frac{f_c}{U_\infty}$. The vertical line indicates the shedding frequency of a two-dimensional airfoil at corresponding conditions. The frequency of the controlled flow at $\Delta\alpha = 6^\circ$ and 8° is closer to the corresponding two-dimensional case. Hence, the effect of the uniform actuation along the span in these cases is to reduce the spanwise variation of shedding, which results in frequency closer to the two-dimensional equivalent.

V. Conclusions

An online feedback control method based on the online DMDc and classical LQT controller was implemented for stabilizing two-dimensional airfoil and three-dimensional finite wing flows subject to a quick change of angle of attack $\Delta\alpha$. The online control approach based on the time-varying model worked well at low $\Delta\alpha$ but failed to fully stabilize the flow when the increase in α was above 4° for both two- and three-dimensional cases. For the finite wing, the reduction in three-dimensionality of the flow was noticed due to the effects of the control. The wake vortices of the controlled flow had predominantly spanwise vorticity and the shedding frequency was closer to the corresponding two-dimensional airfoil flow.

While the time-varying model offers better accuracy of predictions, it is more sensitive to perturbations both in the form of modifications of the flow (such as the change of the angle of attack considered here) as well as sudden changes of the system due to control inputs. More work is required to address these issues. A potential change that may improve the performance of the online control scheme would be to adapt the target solution when the pitch up

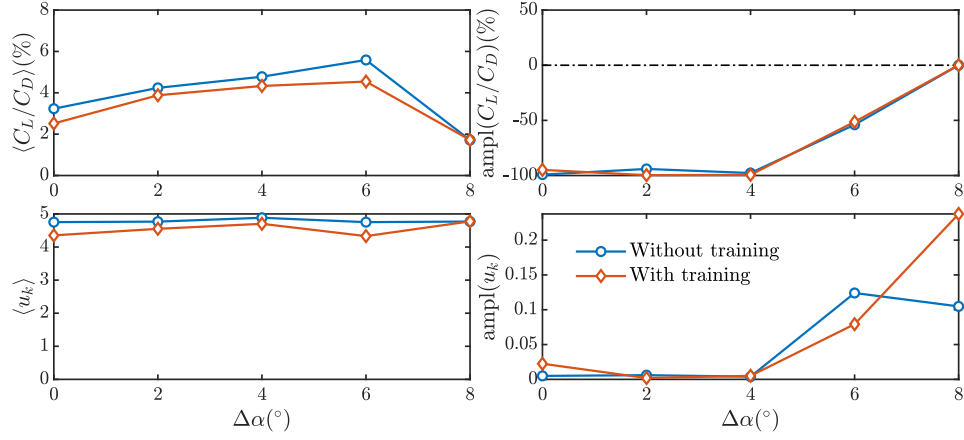


Fig. 10 Performance of the online control for the three-dimensional wing.

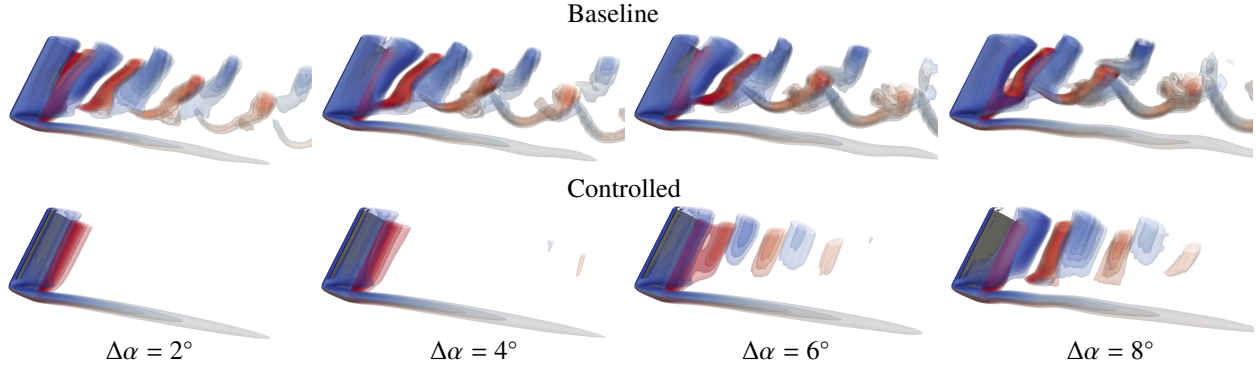


Fig. 11 Instantaneous baseline (top) and online controlled (bottom) flow over wings after pitch up disturbance ($R = 0.1$, $\Pi = Q = 10$, $\alpha_0 = 22^{\circ}$, $Re = 200$). Contours of $0.1 \leq Q \leq 1$ colored by spanwise vorticity.

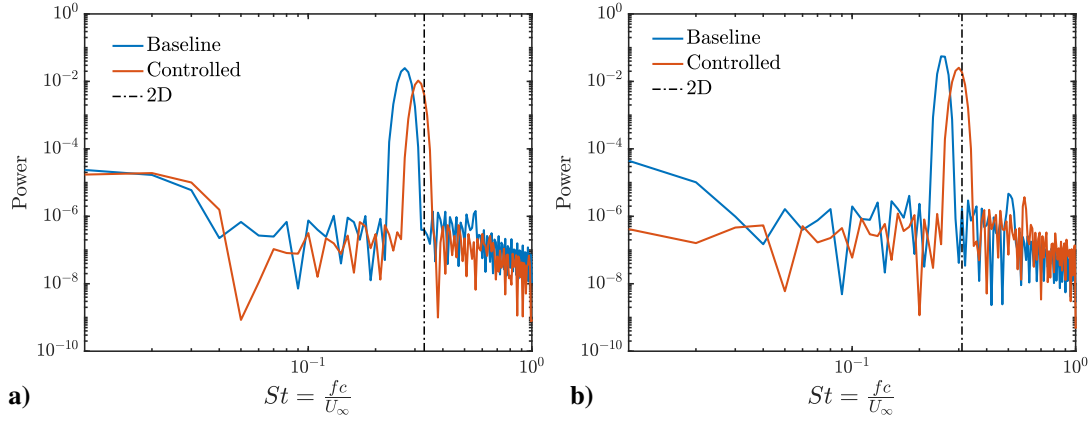


Fig. 12 Frequency spectrum of the C_L/C_D for the finite wing at $\Delta\alpha = 6^{\circ}$ a) and $\Delta\alpha = 8^{\circ}$ b) cases.

disturbance is introduced. Future work will also address multiple actuators, for example an array of discrete actuators along the span of the wing. We also intend to extend these methods to separation control for turbulent flow.

Acknowledgments

This work was funded by the National Science Foundation under award numbers 2129494 (monitored by Dr. Ron Joslin) and 2052811 (monitored by Dr. Yue Wang). The authors acknowledge the Texas Advanced Computing Center (TACC) at the University of Texas at Austin for providing high-performance computing resources that have contributed to the results of this work. The authors would also like to thank Dr. Alexandros Tsolovikos for comments and suggestions.

References

- [1] Seifert, A., Darabi, A., and Wyganski, I., “Delay of Airfoil Stall by Periodic Excitation,” *Journal of Aircraft*, Vol. 33, No. 4, 1996, pp. 691–698. <https://doi.org/10.2514/3.47003>.
- [2] Glezer, A., Amitay, M., and Honohan, A. M., “Aspects of Low- and High-Frequency Actuation for Aerodynamic Flow Control,” *AIAA Journal*, Vol. 43, No. 7, 2005, pp. 1501–1511. <https://doi.org/10.2514/1.7411>.
- [3] Raju, R., Mittal, R., and Cattafesta, L., “Dynamics of Airfoil Separation Control Using Zero-Net Mass-Flux Forcing,” *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3103–3115. <https://doi.org/10.2514/1.37147>.
- [4] Kim, D., and Gharib, M., “Experimental study of three-dimensional vortex structures in translating and rotating plates,” *Experiments in Fluids*, Vol. 49, No. 1, 2010, pp. 329–339.
- [5] Jones, A. R., Medina, A., Spooner, H., and Mulleners, K., “Characterizing a burst leading-edge vortex on a rotating flat plate wing,” *Experiments in Fluids*, Vol. 57, No. 4, 2016, p. 52.
- [6] Calderon, D. E., Cleaver, D. J., Gursul, I., and Wang, Z., “On the absence of asymmetric wakes for periodically plunging finite wings,” *Physics of Fluids*, Vol. 26, No. 7, 2014, p. 071907. <https://doi.org/10.1063/1.4891256>.
- [7] Mancini, P., Manar, F., Granlund, K. O., Ol, M. V., and Jones, A. R., “Unsteady aerodynamic characteristics of a translating rigid wing at low Reynolds number,” *Physics of Fluids*, Vol. 27, No. 12, 2015, p. 123102. <https://doi.org/10.1063/1.4936396>.
- [8] Jantzen, R. T., Taira, K., Granlund, K. O., and Ol, M. V., “Vortex dynamics around pitching plates,” *Physics of Fluids*, Vol. 26, No. 5, 2014, p. 053606. <https://doi.org/10.1063/1.4879035>.
- [9] Son, O., and Cetiner, O., “Three-Dimensionality Effects due to Change in the Aspect Ratio for the Flow around an Impulsively Pitching Flat Plate,” *Journal of Aerospace Engineering*, Vol. 30, No. 5, 2017, p. 04017053. [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0000765](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000765).
- [10] Smith, L. R., and Jones, A. R., “Vortex formation on a pitching aerofoil at high surging amplitudes,” *Journal of Fluid Mechanics*, Vol. 905, 2020, p. A22. <https://doi.org/10.1017/jfm.2020.741>.
- [11] Dong, H., Mittal, R., and Najjar, F. M., “Wake topology and hydrodynamic performance of low-aspect-ratio flapping foils,” *Journal of Fluid Mechanics*, Vol. 566, 2006, p. 309–343. <https://doi.org/10.1017/S002211200600190X>.
- [12] Medina, A., Eldredge, J. D., Kweon, J., and Choi, H., “Illustration of Wing Deformation Effects in Three-Dimensional Flapping Flight,” *AIAA Journal*, Vol. 53, No. 9, 2015, pp. 2607–2620. <https://doi.org/10.2514/1.J053706>.
- [13] Gursul, I., Wang, Z., and Vardaki, E., “Review of flow control mechanisms of leading-edge vortices,” *Progress in Aerospace Sciences*, Vol. 43, No. 7, 2007, pp. 246–270.
- [14] Eldredge, J. D., and Jones, A. R., “Leading-Edge Vortices: Mechanics and Modeling,” *Annual Review of Fluid Mechanics*, Vol. 51, No. 1, 2019, pp. 75–104. <https://doi.org/10.1146/annurev-fluid-010518-040334>.
- [15] Gursul, I., Cleaver, D. J., and Wang, Z., “Control of low Reynolds number flows by means of fluid–structure interactions,” *Progress in Aerospace Sciences*, Vol. 64, 2014, pp. 17–55.
- [16] Reese, B. M., Collins, E. G., Fernandez, E., and Alvi, F. S., “Nonlinear Adaptive Approach to Microjet-Based Flow Separation Control,” *AIAA Journal*, Vol. 54, No. 10, 2016, pp. 3002–3014. <https://doi.org/10.2514/1.J054307>.
- [17] Rabault, J., Kuchta, M., Jensen, A., Réglade, U., and Cerardi, N., “Artificial Neural Networks Trained through Deep Reinforcement Learning Discover Control Strategies for Active Flow Control,” *Journal of Fluid Mechanics*, Vol. 865, 2019, pp. 281–302. <https://doi.org/10.1017/jfm.2019.62>.
- [18] Schmid, P. J., “Dynamic Mode Decomposition of Numerical and Experimental Data,” *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28. <https://doi.org/10.1017/S0022112010001217>.

- [19] Deem, E. A., Cattafesta, L. N., Yao, H., Hemati, M., Zhang, H., and Rowley, C. W., “Experimental Implementation of Modal Approaches for Autonomous Reattachment of Separated Flows,” *2018 AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Kissimmee, Florida, 2018. <https://doi.org/10.2514/6.2018-1052>.
- [20] Mueller, T. J., *Fixed and flapping wing aerodynamics for micro air vehicle applications*, AIAA, 2001. <https://doi.org/10.2514/4.866654>.
- [21] Pines, D. J., and Bohorquez, F., “Challenges facing future micro-air-vehicle development,” *Journal of aircraft*, Vol. 43, No. 2, 2006, pp. 290–305.
- [22] Fischer, P. F., Lottes, J. W., and Kerkemeier, S. G., “nek5000 Web page,” , 2008. [Http://nek5000.mcs.anl.gov](http://nek5000.mcs.anl.gov).
- [23] Tsolovikos, A., Bakolas, E., Suryanarayanan, S., and Goldstein, D., “Estimation and Control of Fluid Flows Using Sparsity-Promoting Dynamic Mode Decomposition,” *IEEE Control Systems Letters*, Vol. 5, No. 4, 2021, pp. 1145–1150. <https://doi.org/10.1109/LCSYS.2020.3015776>.
- [24] Tsolovikos, A., Jariwala, A., Suryanarayanan, S., Bakolas, E., and Goldstein, D., “Separation Delay in Turbulent Boundary Layers via Model Predictive Control of Large-Scale Motions,” *Physics of Fluids*, Vol. 35, No. 11, 2023, p. 115118. <https://doi.org/10.1063/5.0169138>.
- [25] Proctor, J. L., Brunton, S. L., and Kutz, J. N., “Dynamic Mode Decomposition with Control,” *SIAM Journal on Applied Dynamical Systems*, Vol. 15, No. 1, 2016, pp. 142–161. <https://doi.org/10.1137/15M1013857>.
- [26] Schmid, P. J., “Dynamic Mode Decomposition of Numerical and Experimental Data,” *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28. <https://doi.org/10.1017/S0022112010001217>.
- [27] Zhang, H., Rowley, C. W., Deem, E. A., and Cattafesta, L. N., “Online Dynamic Mode Decomposition for Time-Varying Systems,” *SIAM Journal on Applied Dynamical Systems*, Vol. 18, No. 3, 2019, pp. 1586–1609. <https://doi.org/10.1137/18M1192329>.
- [28] Deem, E. A., Cattafesta, L. N., Hemati, M. S., Zhang, H., Rowley, C., and Mittal, R., “Adaptive Separation Control of a Laminar Boundary Layer Using Online Dynamic Mode Decomposition,” *Journal of Fluid Mechanics*, Vol. 903, 2020, p. A21. <https://doi.org/10.1017/jfm.2020.546>.
- [29] Lewis, F. L., Vrabie, D., and Syrmos, V. L., *Optimal Control*, 3rd ed., John Wiley & Sons, 2012.