Online Feedback Flow Control for Finite Wings

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

We conduct numerical flow control studies on a NACA4412 wing. Direct numerical simulation of laminar separated flows is conducted in the Nek5000 spectral element code. We identify the system dynamics in real time and continuously update the state-space model as new measurement data become available using the online dynamic mode decomposition. A linear quadratic tracking controller is designed for the separation location, such that it follows a predefined reference target. The controller gains are then recomputed online based on the new system updates, allowing for an adaptive formulation of real-time feedback control that accounts for changes of the flow due to external disturbances or due to the effects of actuation itself. For the NACA4412 airfoil at 0° angle of attack, adaptive online control of the separation location was more effective than steady blowing at moving the separation point downstream for a given actuator strength. At angle of attack of 10°, an increase of C_l/C_d by $\approx 30\%$ was achieved.

I. Nomenclature

α	=	angle of attack ($^\circ$)
c	=	wing chord
δt	=	DNS time step
Δt	=	state-space model step size
$\boldsymbol{f}$	=	vector of body forces
p	=	pressure
$\boldsymbol{u}$	=	velocity vector
U_∞	=	free-stream velocity
$\boldsymbol{u}_k$	=	control input vector at step k
K	=	controller gain
k	=	discrete time step of the state-space model
l	=	controller update step
A_k	=	state matrix of the state-space model at discrete step k
B_k	=	input matrix of the state-space model at discrete step k
C_k	=	output matrix of the state-space model at discrete step k
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
$\boldsymbol{x}_k$	=	state vector at step k
x_a, y_a, z_a	=	actuator coordinates

II. Introduction

Aerodynamic flow control aims to enhance the performance of lifting surfaces by improving lift and drag characteristics, delaying separation, and reducing the oscillatory loads due to vortex shedding. Since the goal of

*Postdoctoral fellow, Department of Aerospace Engineering and Engineering Mechanics, anton.burtsev@austin.utexas.edu
†Graduate Student, Department of Aerospace Engineering and Engineering Mechanics
‡Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Associate Fellow
§Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Associate Fellow

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 low Reynolds number. A significant number of 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 the large number of works on finite wings, the applications of online (or 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]. Despite their significant potential, such approaches do not 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. Online DMD with control was used in [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. Recently adaptive model predictive control was applied for controlling separation location over a two-dimensional airfoil flow computed using Reynolds averaged Navier Stokes equation [20].

This work is motivated by the fact that many modern (or envisioned) applications for micro-aerial vehicles include flights at stalled conditions and in unsteady (gusty) environments [21, 22] where active flow control may be necessary for safe operations. Hence, the aim of this work is to develop an online closed-loop control approach for the unsteady separated flow. Our overall concept of adaptive online control is similar to that in [20]. However, in our case, it is applied to an unsteady flow computed using DNS and online DMD is used for system identification. The rest of the paper is organized as follows. System identification, controller design and the overall control concept are explained in Sec. III. Results in section Sec. IV consider the effect of controller weights, show a comparison of the online approach with steady blowing and the effect on aerodynamic performance at 0° angle of attack. Finally, the online control is also applied at 10° angle of attack.

III. Methodology

A. Numerical Setup

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

$$\partial_t \boldsymbol{u} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + Re^{-1} \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

where $\boldsymbol{u} = (u, v, w)^{\mathrm{T}}$ is the velocity vector, p is pressure, $\boldsymbol{f}$ is the vector of external body forces acting on the flow and T denotes the transpose. 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 DNS using the spectral element code Nek5000 [23]. The two-dimensional computational domain is decomposed into quadrilateral spectral elements. Within each element, the solution variables are represented in terms of tensor-product Lagrange polynomials, the order of which is adjusted for convergence and is typically $5 \leq p \leq 9$.

We conduct a DNS of a NACA4412 wing segment at $Re_c = 5 \times 10^4$ and $\alpha = 0°$. The computational domain spans $\pm 10c$ in x and y and $0.2c$ in z and is discretized by a C-type mesh shown in Fig. 1. The spectral element mesh shown in Fig. 1(a,b) has 75,000 elements and a polynomial of order of 6 is used. Uniform inlet boundary conditions are applied at the west, south and north far field faces, with outflow at the east face. Front and back faces are assigned a periodic boundary condition.

The boundary layer is tripped by applying random volumetric body forcing normal to the wing along a spanwise line, shown in red in Fig. 1(c), at $10\%c$. The forcing is applied inside a region defined by a Gaussian distribution and is modulated by a random harmonic function in the spanwise direction and over time. The details of the tripping method used can be found in [24]. The trip forcing region extends to twice the local boundary layer thickness.

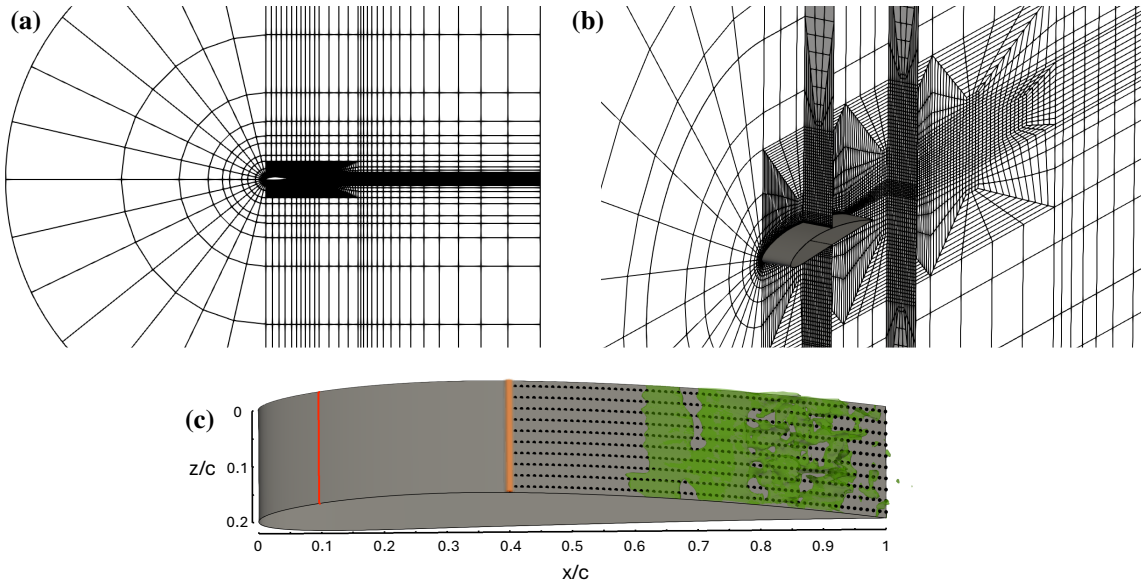

Fig. 1 Mesh used for NACA4412 wing showing full domain (a) and the near wing region (b) and the wing surface (c) showing locations of state measurement points (black), actuator forcing distribution (orange), boundary layer trip (red) and separation bubble (green). Only the spectral elements are shown in (a,b).

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 [25, 26]. For every actuator the force at any point in the domain at step k is given by

$$\boldsymbol{f} = \boldsymbol{e}_\theta G_{\text{act}}(x, y, z) G_{\text{const}} u_k \tag{2}$$

where $\boldsymbol{e}_\theta = [\cos\theta,\ \sin\theta,\ 0]^{\text{T}}$ is a unit vector specifying the direction of the forcing which is pitched by θ with respect to the free stream, $\boldsymbol{u}_k$ is the control input for the actuator at step k, 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:

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

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 $a_x = a_y = a_z = 0.005$, which adequately scale the forcing region such that its height is on the order of boundary layer thickness were chosen and G_{const} was set to 10. The forcing region was placed at $x/c = 0.4$ as shown by the orange contours of G_{act} in Fig. 1(c). The region extend all the way across the span and approximates a blowing slot. For reference, the separation bubble of the uncontrolled turbulent flow is shown in green.

B. Dynamic Mode Decomposition with Control

Dynamic mode decomposition with control (DMDc) [27] is an extension of DMD. 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

$$X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{m-1}], \quad X' = [\boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_m], \quad U = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{m-1}], \tag{4}$$

where X is the $n \times (q-1)$ 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

$$\Omega = \begin{bmatrix} X \\ U \end{bmatrix}. \tag{5}$$

Next, the SVD of the input matrix is taken

$$\Omega = U_\Omega \Sigma_\Omega V_\Omega^{\mathrm{T}}. \tag{6}$$

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

$$\begin{aligned} A &= X' V_\Omega^{\mathrm{T}} \Sigma^{-1} U_1^{\mathrm{T}}, \\ B &= X' V_\Omega^{\mathrm{T}} \Sigma^{-1} U_2^{\mathrm{T}}, \end{aligned} \tag{7}$$

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

$$AW = W\Lambda, \tag{8}$$

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:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k, \tag{9}$$

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

C. Online Dynamic Mode Decomposition with Control

The classical DMD [18] 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) [27] can only be done once enough state vector measurements are collected as training data. Such *a posteriori* analysis is not suitable for situations when sufficient number of state vector measurements at different time instances are 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) [28] 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 number of state vector components n) but high temporal resolution of measurements. The extension of oDMD for control based on the approach of [19] is summarized below.

Recall that the classical DMD describes a linear system $\boldsymbol{x}_{k+1} = A_{\mathrm{DMD}}\boldsymbol{x}_k$, which can be solved as

$$A_{\mathrm{DMD}} = X' X^{+}, \tag{10}$$

where the superscript + denotes the pseudo-inverse. Using the definition of the pseudo-inverse $X^{+} = X^{\mathrm{T}} \left[X X^{\mathrm{T}} \right]^{-1}$ and assuming that A can change with time, we can write

$$\tilde{A}_{k-1} = M_{k-1} P_{k-1}, \tag{11}$$

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

$$M_k = \begin{bmatrix} X' & \boldsymbol{x}_k \end{bmatrix} \begin{bmatrix} X & \boldsymbol{x}_{k-1} \end{bmatrix}^{\mathrm{T}} = M_{k-1} + \boldsymbol{x}_k \boldsymbol{x}_{k-1}^{\mathrm{T}}, \tag{12}$$

$$P_k = \left(\begin{bmatrix} X & \boldsymbol{x}_{k-1} \end{bmatrix} \begin{bmatrix} X & \boldsymbol{x}_{k-1} \end{bmatrix}^{\mathrm{T}} \right)^{-1} = \left[P_{k-1}^{-1} + \boldsymbol{x}_{k-1} \boldsymbol{x}_{k-1}^{\mathrm{T}} \right]^{-1}. \tag{13}$$

DMD can be expanded to include control inputs by simply taking $\tilde{A} = [A \;\; B]$ and substituting $\boldsymbol{x}_k$ for $[\boldsymbol{x}_k \;\; \boldsymbol{u}_k]^{\mathrm{T}}$ as demonstrated in [27]. After some algebraic manipulations, the details of which can be found in [28], the following set of update equations (Eq. (14-16)), which include the effect of control, can be written for step k as follows:

$$P_k = \frac{1}{\kappa}\left(P_{k-1} + \gamma_k P_{k-1}\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}^{\mathrm{T}} P_{k-1}\right), \tag{14}$$

where $\boldsymbol{x}_k$ is the new state vector value, $\boldsymbol{x}_{k-1}$ is the previous state measurement and $\boldsymbol{u}_{k-1}$ is the previous control input. Note that that P_k in Eq. 14 is $(n+m)\times(n+m)$ to account for the fact that the state measurement vectors are size n and the control vector is of size m, with n and m being the number of states and actuators, respectively. Here, $\boldsymbol{u}_{k-1}$ is the control input applied at $k-1$ to get from $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}_k$. The scalar γ_k in Eq. 14 is defined as

$$\gamma_k = 1 + \left(1 + \begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}^{\mathrm{T}} P_{k-1}\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}\right). \tag{15}$$

Equation 14 has a weighting factor $\kappa \in (0, 1]$, which affects the contribution of older snapshots to the current system estimate. Setting a low value ($\kappa \to 0$) 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

$$\begin{bmatrix}A_k & B_k\end{bmatrix} = \begin{bmatrix}A_{k-1} & B_{k-1}\end{bmatrix} + \gamma_k\left(\boldsymbol{x}_k - \begin{bmatrix}A_{k-1} & B_{k-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{x}_{k-1}\\ \boldsymbol{u}_{k-1}\end{bmatrix}^{\mathrm{T}} P_{k-1}, \tag{16}$$

which considers the error between the current measured state $\boldsymbol{x}_k$ 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 pair $[A_0 \;\; B_0]$ 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

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + B_k\boldsymbol{u}_k, \tag{17}$$

that is amenable for controller design.

D. Controller Design

When non-zero steady state needs to be reached, or when the final steady state requires a non-zero steady-state control input, the linear quadratic tracker (LQT) controller can be used [29]. The LQT performance index is defined as

$$J_i = \frac{1}{2}\left(\boldsymbol{y}_N - \bar{\boldsymbol{y}}_N\right)^{\mathrm{T}}\Pi\left(\boldsymbol{y}_N - \bar{\boldsymbol{y}}_N\right) + \frac{1}{2}\sum_{k=i}^{i+N-1}\left(\left(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k\right)^{\mathrm{T}} Q\left(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_k\right) + \boldsymbol{u}_k^{\mathrm{T}} R\boldsymbol{u}_k\right), \tag{18}$$

where $\bar{\boldsymbol{y}}_k = [\bar{\boldsymbol{y}}_i, ..., \bar{\boldsymbol{y}}_{i+N-1}]$ is the reference signal that the system's output trajectory must track and N (positive integer) is the (finite) time horizon. 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 output vector component. Larger Q_{ii} values mean that it is more important to minimize the error in the component y_i of the output vector. Similarly, the value of R for a specific actuator penalizes actuator usage. Lower value of R_{jj} meaning that control polity is allowed to use control input vectors, where the component u_j can be larger than the other input components. The additional symmetric, positive definite matrix Π weights the steady-state error between the output vector $\boldsymbol{y}_N$ and the reference (target) output vector $\bar{\boldsymbol{y}}_N$. The LQT optimal control input contains a feedback and a feed-forward term

$$\boldsymbol{u}_k = -K_k\boldsymbol{x}_k + F_k. \tag{19}$$

The feedback gain K_k is defined as

$$K_k = \left(\hat{B}^{\mathrm{T}} S_{k+1}\hat{B} + R\right)^{-1}\hat{B}^{\mathrm{T}} S_{k+1}\hat{A}. \tag{20}$$

Here the matrix S_{k+1} is the solution of the Riccati difference equation, which is given by

$$S_k = \hat{A}^{\mathrm{T}} S_{k+1}\left(\hat{A} - \hat{B}K_k\right) + C^{\mathrm{T}} QC, \tag{21}$$

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 F is given by,

$$F_k = \left(\left(\hat{B}^{\mathrm{T}} S_{k+1} \hat{B} + R\right)^{-1} \hat{B}^{\mathrm{T}}\right) \boldsymbol{v}_{k+1} \tag{22}$$

where

$$\boldsymbol{v}_k = \left(\hat{A} - \hat{B} K_k\right)^{\mathrm{T}} \boldsymbol{v}_{k+1} + C^{\mathrm{T}} Q \bar{\boldsymbol{y}}_k. \tag{23}$$

Eqs. (20) and (23) are iterated backwards in time starting at step N with the starting conditions of $S_N = C^{\mathrm{T}} \Pi C$ and $\boldsymbol{v}_N = C^{\mathrm{T}} \Pi \bar{\boldsymbol{y}}_N$, 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. Overall Control Concept

We wish to control the separation location on the wing, such that it tracks a predefined reference target. The state vector x consists of streamwise velocity fluctuations measured at n points slightly above the wing surface. For the purposes of flow control, the separation point is taken as the location where the value of the streamwise velocity changes sign to negative (the start of the reversed flow region of the separation bubble). The number of states, n, is chosen to give reasonable resolution of the separation location. The streamwise velocity for the state vector is measured at 600 points just above the wing suction surface. These are uniformly spaced with 60 points in the streamwise direction starting from $x/c = 0.4$ to 1 (the trailing edge) and with 10 points across the span as shown in Fig. 1(c). The streamwise velocity is spanwise averaged along the 10 points to generate single state vector. The state dimension, n, is therefore 60. At each linear model step k, a new value of the state vector is interpolated from the DNS solution. The online DMDc is then applied to this current state vector (x_{k-1}) and the one stored from the previous step (x_k) to update the time-varying A_k and B_k.

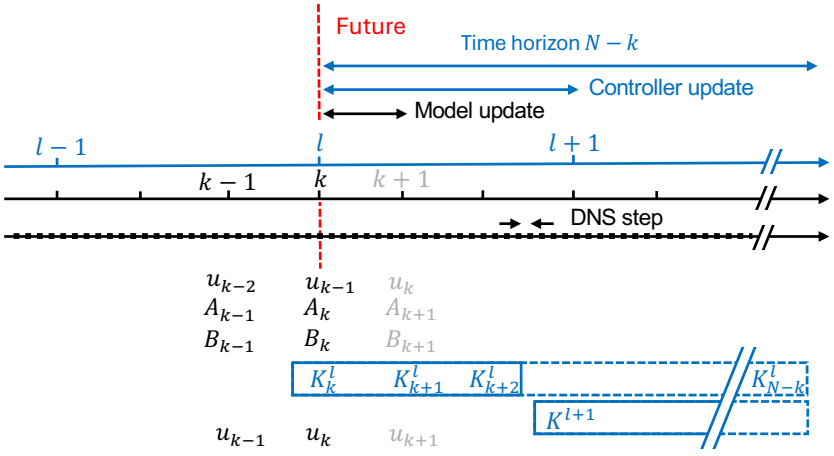

Fig. 2 Relationship between the various timescales of the online control problem.

We introduce the output y_k that is equal to the index of the state vector, where separation occurs. This simplifies the generation of the target signal for the tracking controller, as instead of specifying a vector of desired streamwise velocity distribution, only a single value of the desired separation location is required. An output matrix C_k, which maps the state vector ($\boldsymbol{x}_k$) to the output vector (y_k) is needed to complete the state-space model. Since, in our case, there is no analytical relationship between the separation location and the values of the state, it is found by solving the linear system

$$\boldsymbol{y}_k = C_k \boldsymbol{x}_k. \tag{24}$$

At every step k, y_k is computed as the first index of $\boldsymbol{x}_k$, where the value of the streamwise velocity changes sign to negative and Eq. 24 is solved for C_k. Because the output takes integer values, this is only done when the error between the current output y_k (computed from current $\boldsymbol{x}_k$) and $\tilde{y}_k = C_{k-1} \boldsymbol{x}_k$ (output obtained using the old output matrix) is greater than 0.5.

Interval		t	Δ_t	δ_t
Model update	Δt_{oDMDc}	0.01	1	10
Control update	Δt_{LQT}	0.5-1	50-100	500-1,000
Time horizon	t_N	100	10,000	100,000

Table 1 Timescales for NACA4412 separation control.

A tracking controller is then designed. For the purposes of controller design, the matrices A, B and C are treated as if they were constant for the whole time horizon over which the controller is designed. This simplification has to be made because future A, B and C matrices are not known. In the following, we use $\hat{A} = A_k$, $\hat{B} = B_k$ and $\hat{C}_k$ to denote these constant state, input and output matrices obtained from the current estimate of the online DMDc. We then apply time-invariant versions of the controllers on this time-invariant system

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \hat{A}\boldsymbol{x}_k + \hat{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \hat{C}_k\boldsymbol{x}_k. \end{aligned} \tag{25}$$

The resulting gains will not be strictly optimal in the optimal control sense since the actual system in Eq. (17) 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.

At each controller update step l, a sequence of feedback and feed-forward gains K^l and F^l is computed for the LQT time horizon N. Here, K^l refers to an m by n by N array of gains computed at controller update step l. The controller gains do not have to be updated at each linear models step, as long as the rate of its updates is sufficient to account for any expected variations of the system and the update interval is less than or equal to the LQT time horizon. In between the controller updates, the latest sequence of gains is used to obtain the control signal, which is introduced as a linear feedback to the DNS as

$$\boldsymbol{u}_k = -K_k^l\boldsymbol{x}_k + F_k^l, \tag{26}$$

where K_k^l and F_k^l mean the LQT gains for the k-th step. The relationship between the DNS step (δt), the linear model step (Δt), controller update interval (Δt_{LQT}) and the time horizon (N) is summarized in Fig. 2. The typical values of these time intervals are show in table 1.

IV. Results

A. System Identification

The performance of the the classical DMDc in predicting the future evolution of the states is compared to the oDMDc in Fig. 3. The color contours show the variation of the state vector (streamwise flow velocity near the surface of the wing) with time. Blue indicates reversed flow and the contour of zero streamwise velocity (corresponding to separation or reattachment) is shown with a dashed line. For both cases, training data were collected for $14 \leq t \leq 30$ every 10 DNS time steps. A training control input was applied at the beginning of the training window. For DMDc, Eq. 7 is used to obtain the state-space system, while for oDMDc, the update equations Eq. 14-16, initialized with a random A and B pair were used. The prediction of the states after $t = 30$ assuming zero control input using these state-space models is plotted in middle and bottom subplots in Fig. 3. For oDMDc, the last pair of A_k,B_k corresponding to $t = 30$ was used. Both approaches fail to capture the exact dynamics of the turbulent separation, but oDMDc prediction is more accurate in the mean location of the separation and reattachment lines. The DMDc prediction blows up due to uns unstable mode identified by DMDc. It should be noted that the DMDc model is highly dependent on the training data used. Various training control inputs and different lengths of the overall training data may result in better performance than that shown in Fig. 3. However, for same conditions, the oDMDc model preforms better. Another important advantage of the online system identification, which is not shown here, is its ability to accommodate changes of the conditions flow conditions, for example, changes of angle of attack due to gusts or maneuvers.

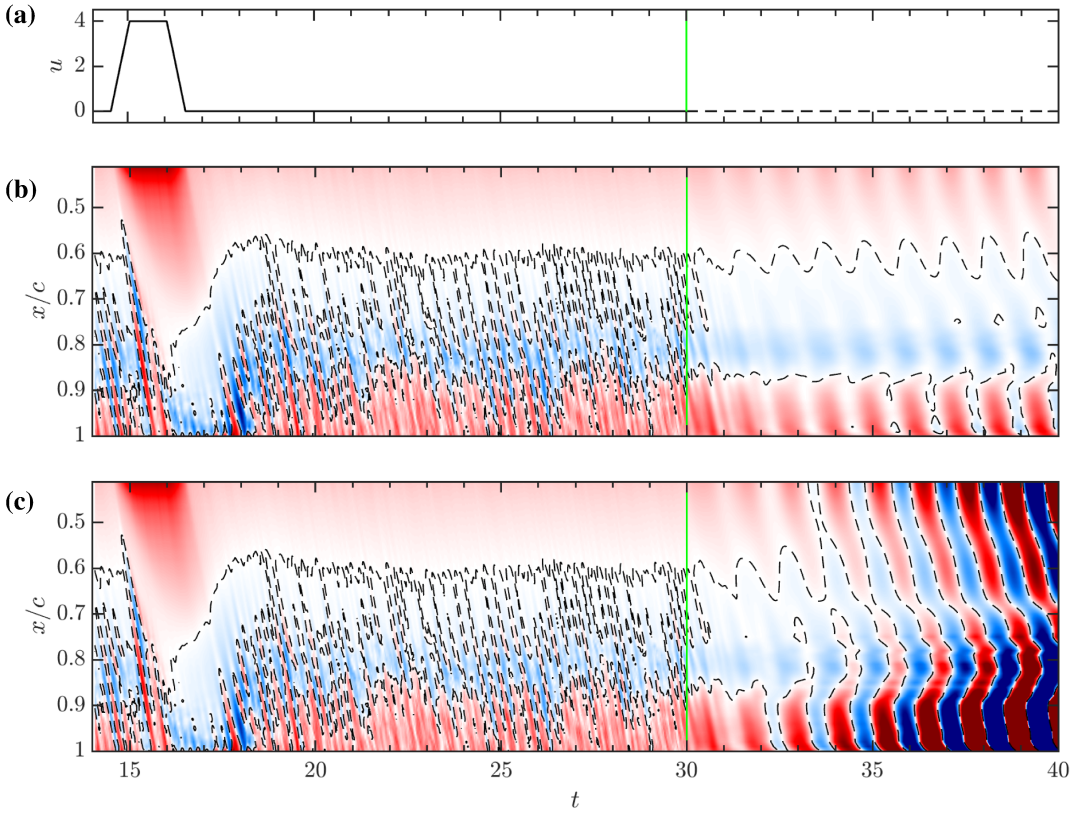

Fig. 3 Comparison of state predictions. Training data is to the left of the green line, prediction to the right. Control input (a), oDMDc (b) and DMDc (c).

B. Adaptive Control at $\alpha = 0°$

Online adaptive control as described in the previous section is then used to move the (spanwise-averaged) separation location downstream. We first apply the online control scheme for the flow at zero angle of attack to determine the optimal weights for the LQT problem. The effect of varying the R weight of the LQT for the online control is considered in Fig. 4. For this test, $P = Q = 1$ and the time horizon was fixed at 100 chord time units. The mean strength of the jet control signal ($\bar{u}$) and mean tracking error ($\bar{E}_{track}$) are plotted on the left and right y-axes, respectively. The tracking error taken to be the difference between the actual and target separation location, expressed as the percentage of the reference value. The filled regions indicate the variation of the instantaneous value and are equal to the mean value ± the standard deviation. As expected, higher values of R result in a reduction of the mean strength of $\bar{u}$ and increase $\bar{E}_{track}$. However, $\bar{E}_{track}$ increases slowly for $R \lesssim 50$ and remains below 10%. The effect of varying the interval (Δ_{LQT}), at which the LQT problem is resolved is illustrated in Fig. 5. This interval is varied from 0.01 (the model step) to 3 chord time units. Smaller values of Δ_{LQT} lead to larger average tracking error with larger fluctuations. Because of this, Δ_{LQT} was fixed at 1 chord time unit.

Figure 6 compares the performance of the active online control with open loop control inputs of constant strength. The time-averaged mean separation location, $\bar{x}_s/c$, and the average tracking error, $\bar{E}_{track}$, are plotted with the mean strength of the jet control signal ($\bar{u}$). For online runs, different levels of $\bar{u}$ were achieved by changing the R weight of the LQT, while the target was constant and equal to 80% of chord. Active online control achieves separation locations further downstream for any given $\bar{u}$ compared to a constant input of the same strength. The tracking accuracy of the online approach is better up to $\bar{u} \approx 6$. This is because for the online case, higher levels of $\bar{u}$ also correspond to larger instantaneous fluctuations of u, as can be seen by increased width of the shaded region around the $\bar{u}$ curve in Fig. 4. The comparison of constant blowing with $u = 4$ and the online control run having same average strength of u, corresponding to $P = Q = 1, R = 20$, is plotted in Fig. 7.

The effect of open loop ($u = 4$) and adaptive online control of the aerodynamic performance of the NACA4412 wing

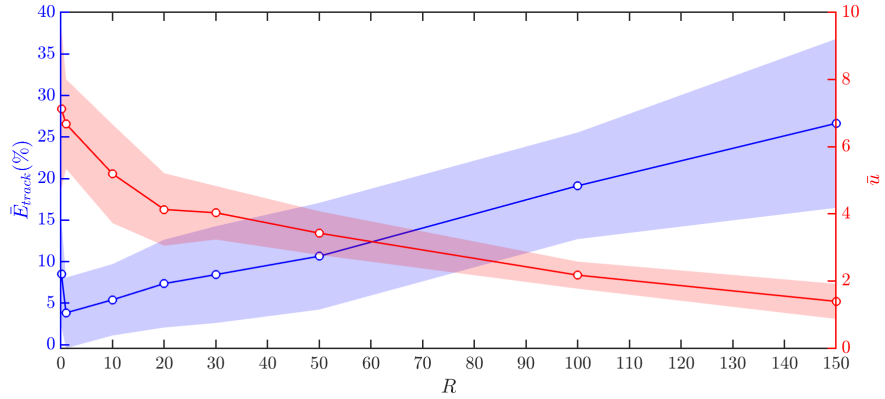

Fig. 4 Effect of the R weight of the LQT on the mean strength of the jet control signal and the average tracking error.

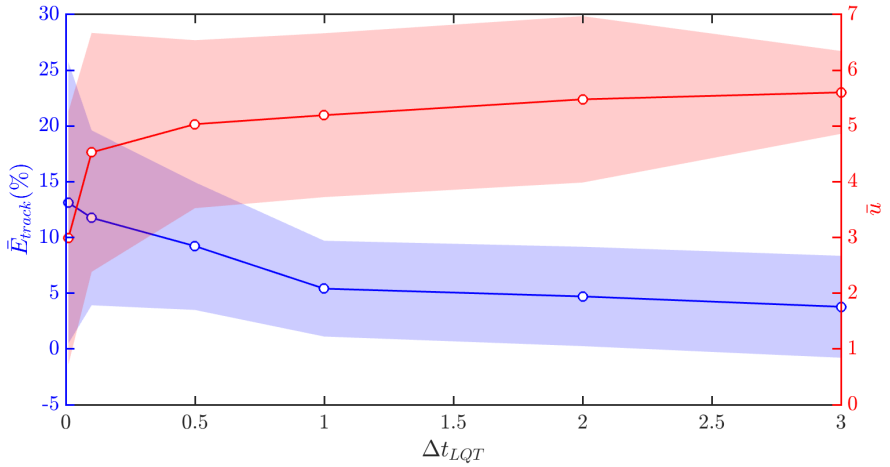

Fig. 5 Effect of the controller update interval on the mean strength of the jet control signal and the average tracking error.

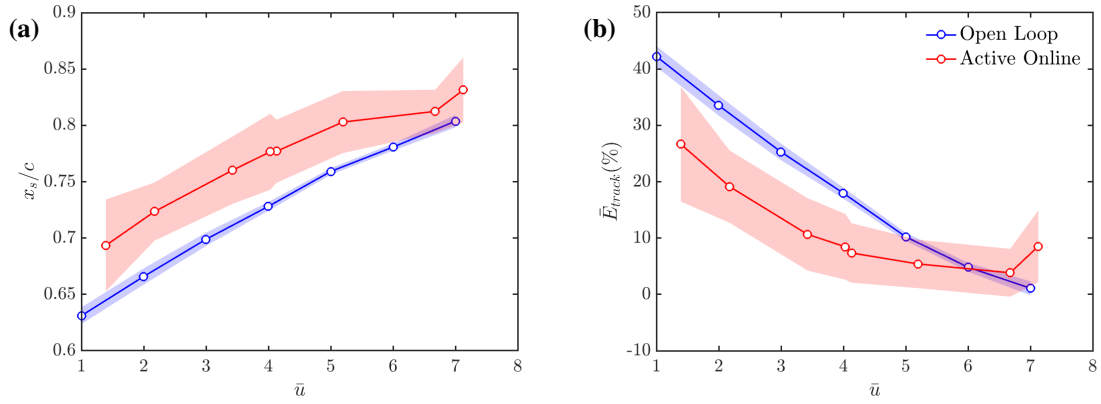

Fig. 6 Variation of mean separation location (a) and tracking error (b) with mean control input strength for open loop and adaptive online control.

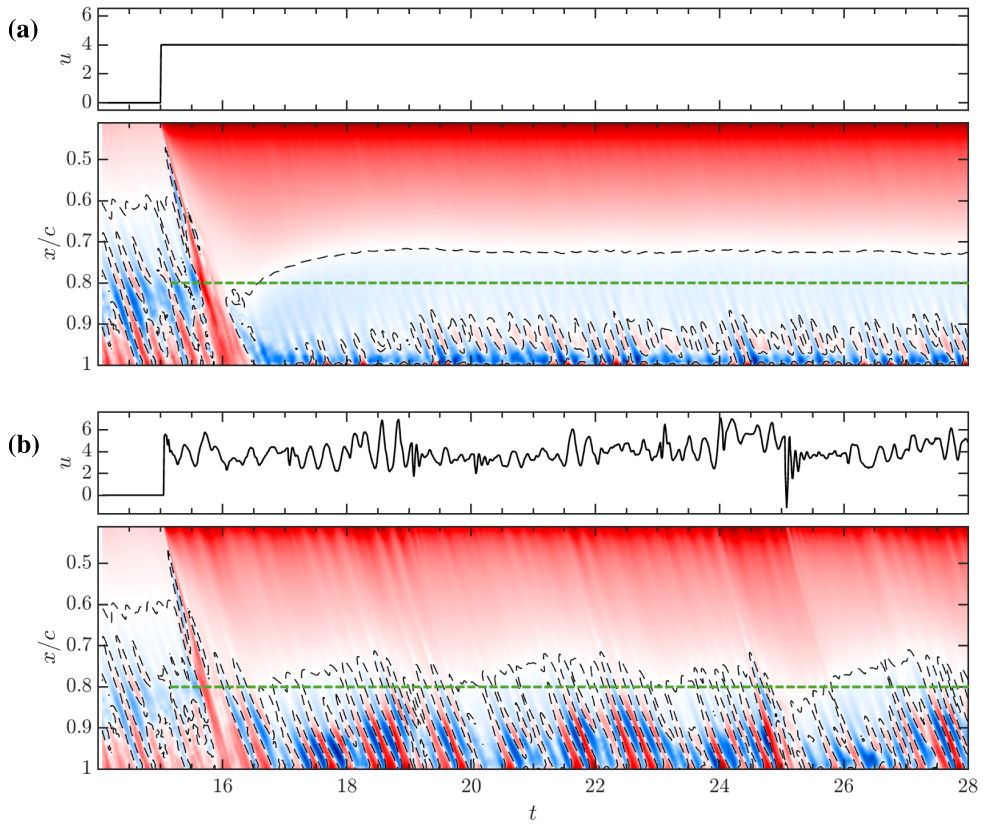

Fig. 7 Comparison of the control signal (top) and separation location (bottom) for open loop control with $\bar{u} = 4$ (a) and online oDMDc control ($P = Q = 1, R = 10$) (b). The target separation location is shown by the green line.

at zero angle of attack is plotted in Fig. 9. Solid lines represent running averages, with instantaneous fluctuations plotted with transparent lines. The average values of the uncontrolled baseline case are plotted with the black dashed lines. It should be noted that the computation of lift and drag forces is done by taking the surface integral of pressure and viscous forces on the airfoil surface and does not account for the thrust (reaction force) induced by the presence of the actuator. Open loop control decreases C_l and slightly increases C_d resulting in a reduction of the average C_l/C_d from 15 to 9. This occurs because the flow is almost re-lamiliarized. This can be seen in Fig. 8, as the separation location becomes more uniform in the spanwise direction and the starts to exhibit larger scale spanwise vortices. Hence, the C_l/C_d value moves close to the average of the untripped flow, which is about 5. However, for the online control case, the mean C_l is only marginally reduced and C_d remains virtually constant, resulting in only a slightly lower mean C_l/C_d as seen in Fig. 9. Hence, online control reduces the separation bubble, while avoiding the reduction in aerodynamic performance caused by open loop steady blowing of equivalent mean strength. However, it should be noted that the amplitude of fluctuations of both lift and drag coefficients is substantially increased for the online case. This is at least in part due to the highly unsteady nature of the control signal as can be seen in Fig. 7(b).

C. Adaptive Control at $\alpha = 10°$

Because the effect of control at $\alpha = 0°$ leads to flow re-laminarization and subsequent reduction of the C_l, we also apply the control scheme to flow at higher angle of attack at $\alpha = 10°$. The baseline plot of the states for $\alpha = 10°$ is shown in Fig. 10. The separation location varies in the region of $5.5 \leq x/c \leq 7.5$ and unlike at $\alpha = 0°$, there is no clear reattachment visible. Online control was run with $P = Q = 1, R = 10, N = 100s, \Delta_{LQT} = 1s$ and the target separation location set to $x/c = 0.9$. The control signal and resulting separation location are plotted in Fig. 11. Online control moves the separation location to about $x/c = 7$ but is unable to reach the target of $x/c = 0.9$, at least at this combination of LQT weights. However, the separation location becomes more steady the extent of the separation bubble

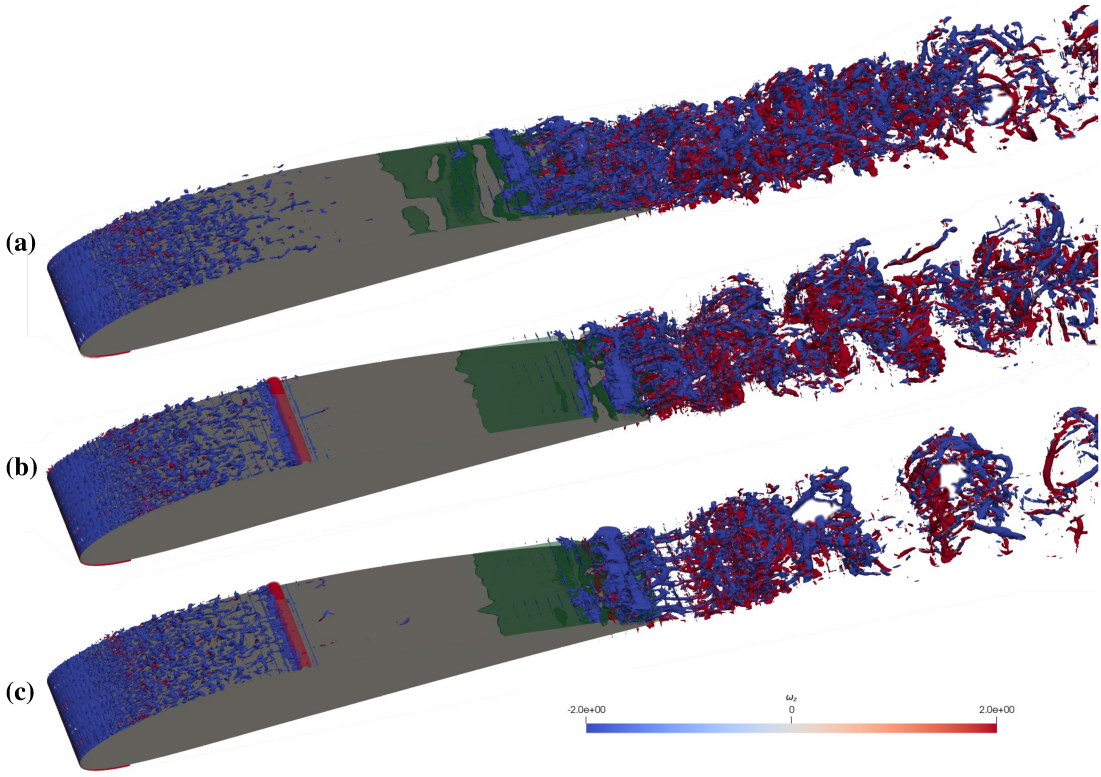

Fig. 8 Comparison of the instantaneous flow over the NACA4412 at $\alpha = 0°$ showing the uncontrolled baseline at $t = 14$ (a), steady blowing control at $t = 28$ (b), and adaptive online control at $t = 28$ (d). Contours of λ_2 colored by spanwise vorticity. The separation bubble is shown in green and the jet force field is in orange.

is reduced with intermittent reattachment observed at $x/c \approx 8.5$. The comparison of aerodynamic performance between the baseline and controlled cases is plotted in Fig. 12. After the initial transients pass, C_l increases by $\approx 20\%$ and C_d decreases by $\approx 16\%$, which results in an increase of C_l/C_d of $\approx 30\%$.

V. Conclusions

An adaptive online control strategy for separation location of a turbulent flow over a NACA4412 wing was proposed. The online system identification using oDMDc was used to identify a time-varying state-space model. A discrete time, LQT controller was used to repeatedly generate optimal control gains as the state-space model updates. Adaptive online control was more effective than open loop steady blowing at moving the separation point downstream for a given actuator strength. The online approach also avoided the reduction in mean aerodynamic performance caused by steady blowing of similar strength at 0° angle of attack, albeit at a cost of increased amplitude of fluctuations. When applied at 10° angle of attack, the online control achieved an increase of C_l/C_d of $\approx 30\%$. Another important advantage of the online control strategy, which was not shown here, is its ability to accommodate time-varying reference signals and the potential to adapt to changes of flow conditions. In addition, the fact that the control problem is solved repeatedly allows changing LQT weights and time horizon with time. The present control approach can be generalized to use an arbitrary number of actuators which may be necessary for finite and swept wing flows. Furthermore, having discrete spanwise actuators opens the potential to design smarter control algorithms that take advantage of the structures of the oncoming turbulent boundary layer [26]. Therefore, these results serve as a first step towards applying adaptive online control to more complex, fully three-dimensional wing flows at higher Reynolds numbers.

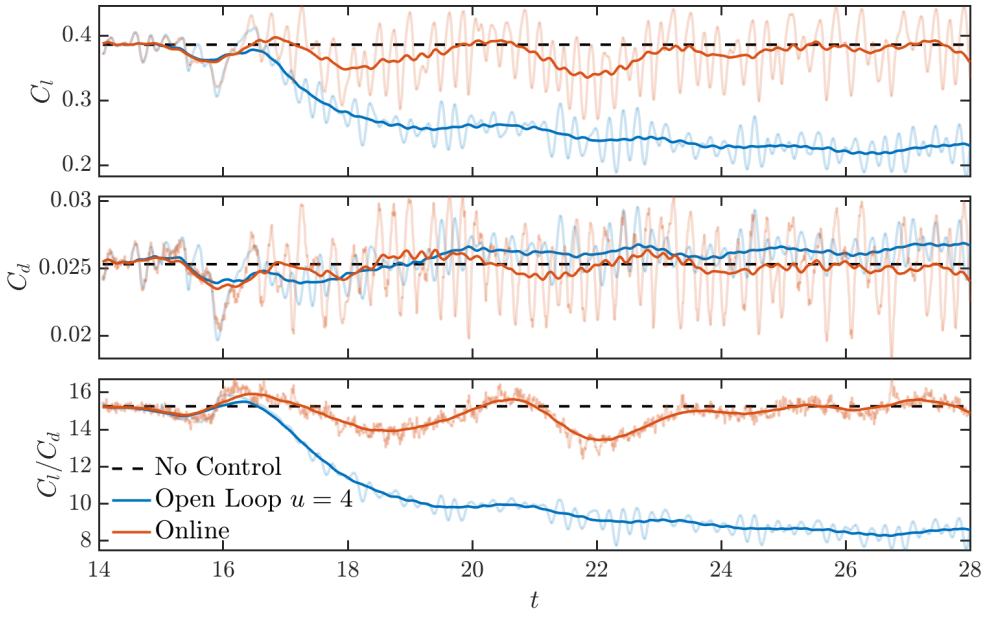

Fig. 9 Effect of control on the aerodynamic coefficients at $\alpha = 0°$. Solid lines represent running averages, with instantaneous fluctuations plotted with transparent lines. Online control run is using $P = Q = 1, R = 10, N = 100s, \Delta_{LQT} = 0.5s$.

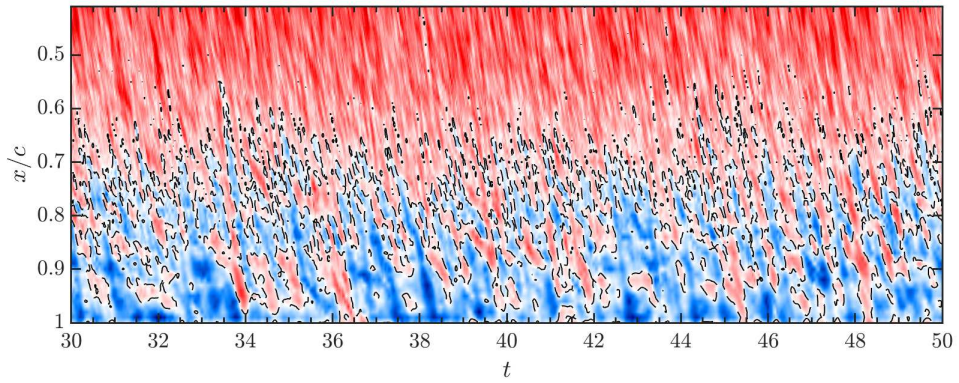

Fig. 10 Separation location at $\alpha = 10°$ without control.

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.

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.

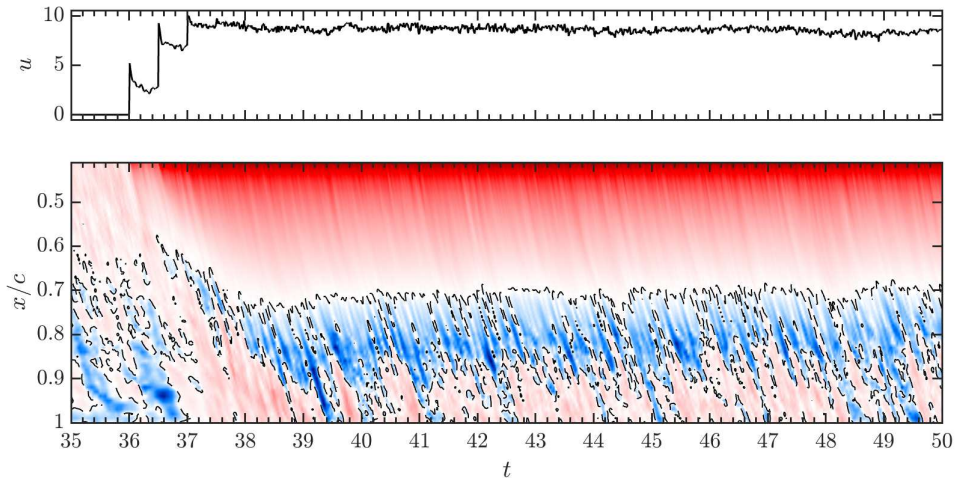

Fig. 11 Control signal and separation location at $\alpha = 10°$ with online control ($P = Q = 1, R = 10, N = 100s, \Delta_{LQT} = 1s$).

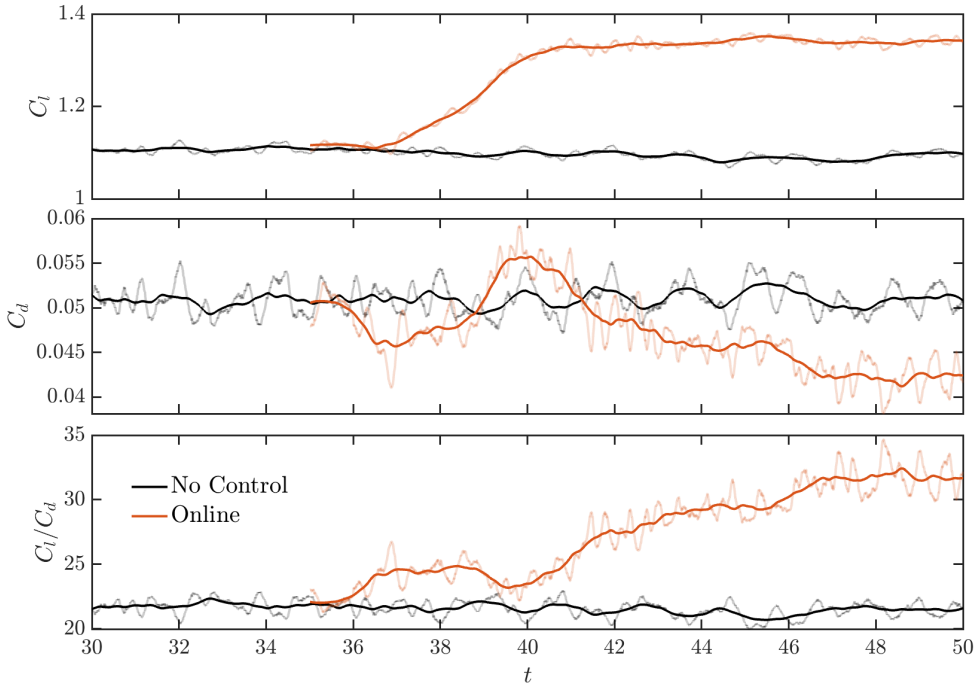

Fig. 12 Effect of control on the aerodynamic coefficients at $\alpha = 10°$. Solid lines represent running averages, with instantaneous fluctuations plotted with transparent lines. Online control run is using $P = Q = 1, R = 10, N = 100s, \Delta_{LQT} = 1s$.

[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.

[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., 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.

[20] Vander Schaaf, J. C., Lu, Q., Fidkowski, K. J., and Bernstein, D. S., "Data-Driven Model Predictive Control of Airfoil Flow Separation," *2024 American Control Conference (ACC)*, IEEE, 2024, pp. 1568–1573.

[21] Mueller, T. J., *Fixed and flapping wing aerodynamics for micro air vehicle applications*, AIAA, 2001. https://doi.org/10.2514/4.866654.

[22] Pines, D. J., and Bohorquez, F., "Challenges facing future micro-air-vehicle development," *Journal of aircraft*, Vol. 43, No. 2, 2006, pp. 290–305.

[23] Fischer, P. F., Lottes, J. W., and Kerkemeier, S. G., "nek5000 Web page," , 2008. Http://nek5000.mcs.anl.gov.

[24] Schlatter, P., and Örlü, R., "Turbulent Boundary Layers at Moderate Reynolds Numbers: Inflow Length and Tripping Effects," *Journal of Fluid Mechanics*, Vol. 710, 2012, pp. 5–34. https://doi.org/10.1017/jfm.2012.324.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Lewis, F. L., Vrabie, D., and Syrmos, V. L., *Optimal Control*, 3rd ed., John Wiley & Sons, 2012.