J. Fluid Mech. (2025), *vol.* 1016, A41, doi:10.1017/jfm.2025.10423

JFM PAPERS

Resolvent-based estimation and control of a laminar airfoil wake

Junoh Jung[1], **Rutvij Bhagwat**[1,2] **and Aaron Towne**[1]

[1]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA

[2]Department of Mechanical Engineering, Florida State University, Tallahassee, FL 32310, USA

Corresponding author: Junoh Jung, junohj@umich.edu

(Received 3 January 2025; revised 4 June 2025; accepted 23 June 2025)

We develop an optimal resolvent-based estimator and controller to predict and attenuate unsteady vortex-shedding fluctuations in the laminar wake of a NACA 0012 airfoil at an angle of attack of 6.5°, chord-based Reynolds number of 5000 and Mach number of 0.3. The resolvent-based estimation and control framework offers several advantages over standard methods. Under equivalent assumptions, the resolvent-based estimator and controller reproduce the Kalman filter and LQG controller, respectively, but at substantially lower computational cost using either an operator-based or data-driven implementation. Unlike these methods, the resolvent-based approach can naturally accommodate forcing terms (nonlinear terms from Navier–Stokes) with coloured-in-time statistics, significantly improving estimation accuracy and control efficacy. Causality is optimally enforced using a Wiener–Hopf formalism. We integrate these tools into a high-performance-computing-ready compressible flow solver and demonstrate their effectiveness for estimating and controlling velocity fluctuations in the wake of the airfoil immersed in clean and noisy free streams, the latter of which prevents the flow from falling into a periodic limit cycle. Using four shear–stress sensors on the surface of the airfoil, the resolvent-based estimator predicts a series of downstream targets with approximately 3 % and 30 % error for the clean and noisy free stream conditions, respectively. For the latter case, using four actuators on the airfoil surface, the resolvent-based controller reduces the turbulent kinetic energy in the wake by 98 %.

Key words: flow control, vortex shedding, wakes

1. Introduction

The laminar flow over an airfoil is a canonical problem in fluid mechanics due to its role in aerodynamics and status as a prototypical problem for studying wakes. Unsteady

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

perturbations in the wake are of particular interest for several reasons. First, these perturbations are intimately tied to the separation bubble that forms over the suction side of the airfoil, which in turn increases drag (Alam *et al.* 2010; Chang *et al.* 2022). Second, unsteady fluctuations can degrade aerodynamic performance in many aircraft flight control scenarios, such as manoeuvres at high angles of attack, landing, takeoff or encountering atmospheric turbulence. Third, wake perturbations significantly contribute to aerodynamic noise, which is a concern for wind turbines (Wagner, Bareiß & Guidati 1996; Agrawal *et al.* 2015) and rotorcrafts, including drones. In all these scenarios, accurate estimation and effective closed-loop control of wake perturbations are critical to improve engineering performance, i.e. to reduce drag, enhance flight control and mitigate aerodynamic noise.

A dominant feature of the laminar flow over an airfoil is vortex shedding in the wake. Vortices form on each side of the airfoil and shed periodically, creating downstream flow patterns such as the Kármán vortex street. Factors including the shape of the object, the angle of attack, and the Reynolds number influence this vortex shedding. In NACA0012 airfoils at moderate angles of attack (6°–10°), an anticlockwise vortex is generated at the trailing edge, and a clockwise vortex at the front suction side separates and is entrained downstream (Chang *et al.* 2022) – accordingly, both the front section of the suction side and the trailing edge impact vortex shedding. Many studies have been conducted to suppress vortex shedding in the wakes behind cylinders (Jin, Illingworth & Sandberg 2020; Déda *et al.* 2023; Lin & Tsai 2024) and airfoils (Colonius & Williams 2011; Broglia *et al.* 2018).

An accurate estimator is essential for successful closed-loop control (Stengel 1994; Brunton & Noack 2015). Standard estimation and control methods such as the Kalman filter (Kalman 1960) and the linear-quadratic-Gaussian (LQG) controller have been applied to fluid mechanics problems over the past two decades, e.g. Kalman filter (Rafiee, Wu & Bayen 2009; Colburn, Cessna & Bewley 2011; An *et al.* 2021) and LQG control (Bagheri, Brandt & Henningson 2009; Fabbiane *et al.* 2014, 2015, 2017; Sasaki *et al.* 2018*a*). However, these standard methods have two significant limitations when applied to flow-control problems. First, solving the Riccati equations required to obtain the Kalman and LQG gains scales poorly with problem size and becomes computationally expensive, or in some cases prohibitive, for the large systems typically obtained when discretising the Navier–Stokes equations. This issue can be partially mitigated by reducing the size of the system *a priori* via some model-reduction method (Pasquale *et al.* 2017; Gomez *et al.* 2019), but this potentially degrades the performance of the controller. Second, standard methods model the nonlinear terms of the Navier–Stokes equations as a white noise forcing on the linear dynamics. These terms are not white (Towne, Brès & Lele 2017; Zare, Jovanović & Georgiou 2017; Morra *et al.* 2021), and treating them as such deteriorates estimation and control performance (Martini *et al.* 2020; Amaral *et al.* 2021).

In this paper, we use a recently developed class of estimation and control methods based on resolvent analysis (Martini *et al.* 2020, 2022; Towne, Lozano-Durán & Yang 2020). Resolvent analysis, or input–output analysis, is a powerful methodology based on a linear mapping between forcing (input) and response (output) modes, and the gain between them. These modes and gains are obtained from a singular value decomposition of the resolvent operator obtained from the linearised Navier–Stokes equations. Early studies viewed the input as an external forcing on the Navier–Stokes equations linearised about some laminar fixed point (Jovanović & Bamieh 2005; Sharma *et al.* 2006; Sipp *et al.* 2010). McKeon & Sharma (2010) extended resolvent analysis to turbulent flows by interpreting the nonlinear terms from the Navier–Stokes equations as a forcing on the linear dynamics. Towne, Schmidt & Colonius (2018) showed that resolvent modes and spectral proper orthogonal decomposition (SPOD) modes are identical when the forcing

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

is white noise, directly linking resolvent modes to coherent structures observed in flow data. However, resolvent analysis poses computational challenges for high-dimensional problems. To address these difficulties, Ribeiro, Yeh & Taira (2020) applied randomised singular value decompoistion (SVD). Martini *et al.* (2020) used a time-stepping approach to obtain the action of the resolvent operator on a specified forcing, eliminating the need to compute the inverse of large matrices. Farghadan *et al.* (2023, 2024) further extended the range of problems amenable to resolvent analysis by combining randomised SVD with an approach to minimise the cost of the time-stepping method and successfully applied the method to several three-dimensional problems.

Resolvent analysis has been extensively used to study the flow over airfoils at different Reynolds numbers and angles of attack. Thomareis & Papadakis (2018) investigated the physics of separated and attached flows around a NACA 0012 airfoil at $Re_{L_c} = 50\,000$ and angle of attack 5°. Symon, Sipp & McKeon (2019) investigated two angles of attack, 0° and 10°, for a NACA 0018 airfoil and showed that these two cases behave as an oscillator and amplifier, respectively (Huerre & Monkewitz 1990). Kojima *et al.* (2020) identified the origin of the two-dimensional transonic buffet over a NACA 0012 airfoil at $Re_{L_c} = 2000$, $Ma_\infty = 0.85$ and $\alpha = 3^\circ$. Marquet *et al.* (2022) studied the flow over a NACA 0012 at $Re_{L_c} = 5000$ for angles of attack between $\alpha = 6.5^\circ$ and 9° using an incompressible Navier–Stokes linear operator with the mean flow obtained from a numerical simulation, which was validated against experimental data.

Motivated by the ability of resolvent modes to efficiently represent coherent structures, Towne *et al.* (2020) introduced a resolvent-based method to estimate space–time flow statistics from limited sensor measurements. Martini *et al.* (2020) extended this method to reconstruct the time series of the flow state using limited measurements. Amaral *et al.* (2021) used this method to estimate the velocity field in a channel flow using pressure and shear-stress measurements at the channel wall. The aforementioned resolvent-based estimators are non-causal, i.e. they use both past and future measurements to determine the current flow state. Martini *et al.* (2022) derived a causal resolvent-based estimator and controller by enforcing causality using a Wiener–Hopf formalism (Noble 1958), making them applicable for closed-loop flow control (Jung *et al.* 2020).

The causal resolvent-based estimator and controller reproduce the Kalman filter and LQG controller, respectively, under equivalent assumptions, namely white-noise forcing. However, the resolvent-based estimator and controller have two crucial advantages over these standard methods. First, they can be efficiently computed even for large systems using a time-stepping method similar to that described earlier for computing resolvent modes or via a data-driven approach. This makes the method applicable to the large problems typical of fluid mechanics without the need for detrimental *a priori* model reduction. Second, the resolvent-based methods can accommodate coloured-in-time forcing to statistically account for the nonlinear terms from the Navier–Stokes equations, improving estimation accuracy and control efficacy. These benefits are discussed in detail by Martini *et al.* (2022).

In this paper, we aim to estimate and control unsteady fluctuations in the wake of a two-dimensional NACA 0012 airfoil at $Ma_\infty = 0.3$, $Re_{L_c} = 5000$ and $\alpha = 6.5^\circ$ using the resolvent-based methods. The overall roadmap of the paper is depicted schematically in figure 1. We begin in § 2 by computing the flow using direct numerical simulation (DNS) and validate the simulation against results in the literature. In § 3, we analyse the global eigenmodes and resolvent modes of the flow to validate the linearised compressible Navier–Stokes operator and highlight the key flow physics. We then introduce the resolvent-based estimation and control methods in § 4, discuss how the kernels are computed in § 5 and detail our implementation of these tools into a computational

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

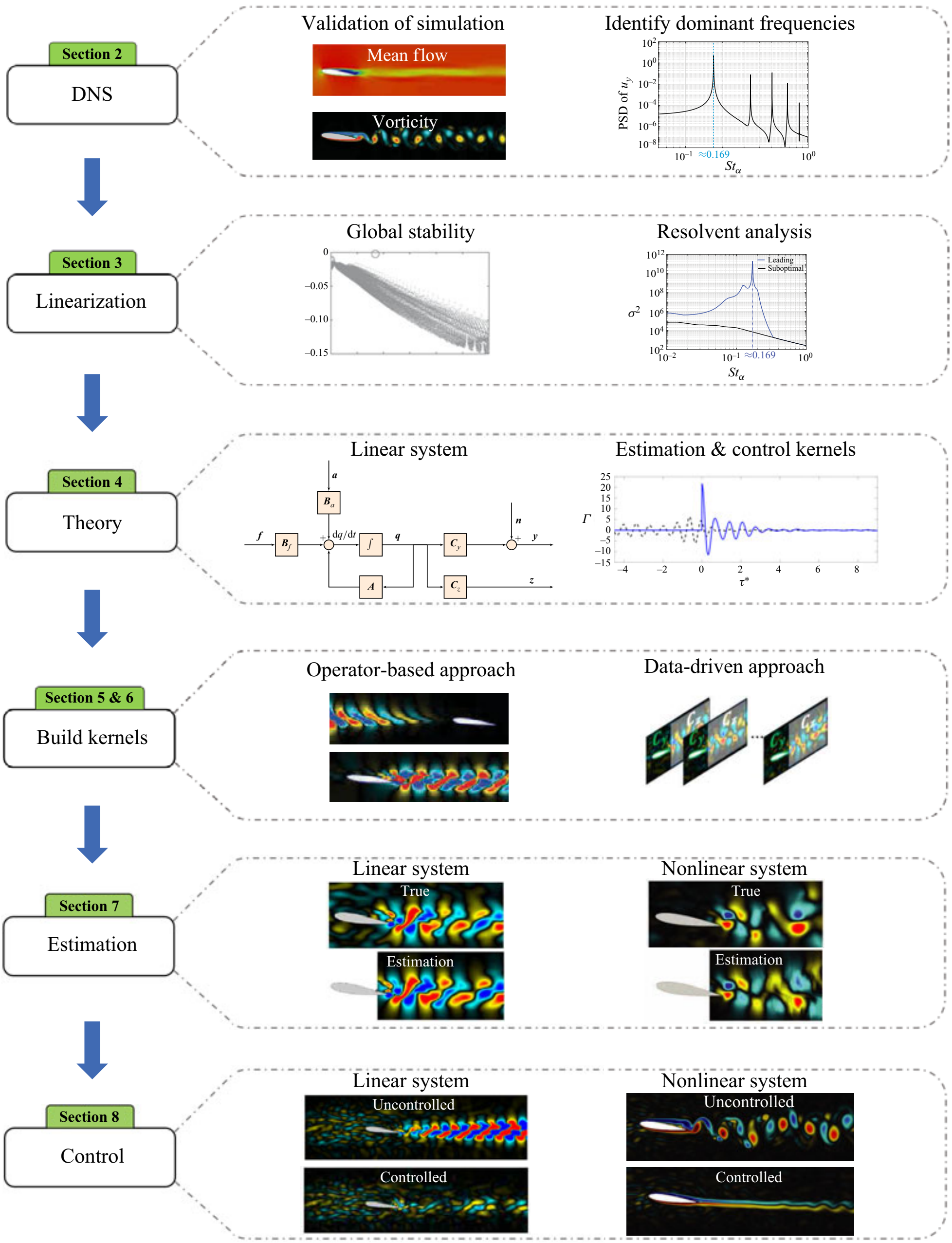


Figure 1. Roadmap for resolvent-based estimation and control of the laminar flow over an airfoil.

fluids dynamics (CFD) code in § 6. The resolvent-based estimator is applied to both the linearised and nonlinear problems for clean and noisy free stream conditions, the latter of which is designed to disrupt the periodic vortex shedding and induce chaotic fluctuations in the wake, in § 7. Following this, we demonstrate the effectiveness of resolvent-based control for the same systems and flow conditions in § 8. Finally, § 9 summarises the paper, highlights its novel contributions and discusses future research directions.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

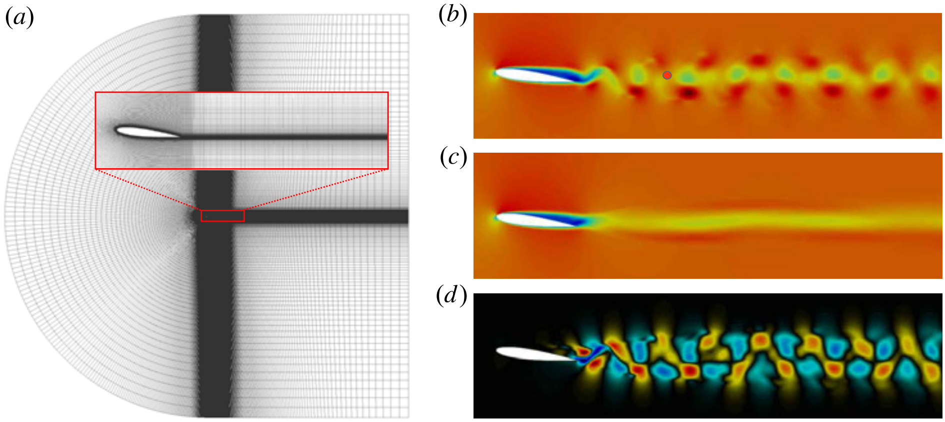


Figure 2. Direct numerical simulation: (*a*) the full computational C-shaped grid with a close-up view for the wall and wake regions; (*b*) a snapshot of the instantaneous streamwise velocity u_x with the red dot indicating the probe location at $(x, y)/L_c = (2.1, -0.11)$ for the power spectral density in figure 3; (*c*) the mean streamwise velocity $\bar{u}_x$; and (*d*) a snapshot of the instantaneous streamwise velocity fluctuation u'_x.

2. Problem set-up and simulation

We aim to use a resolvent-based approach to estimate and mitigate chaotic velocity fluctuations in a laminar airfoil wake. Following Marquet *et al.* (2022), we consider the flow around a NACA0012 airfoil at a low chord-based Reynolds number of $Re_{L_c} = 5000$, Mach number of $Ma_\infty = 0.3$ and an angle of attack of $\alpha = 6.5^\circ$. We consider two different freestream conditions: (i) a clean free stream with no ambient fluctuations and (ii) a noisy free stream with substantial ambient fluctuations generated by random forcing upstream of the airfoil. The flow falls into a periodic limit cycle due to vortex shedding for the clean free stream; the noisy free stream kicks the flow out of this limit cycle, leading to chaotic fluctuations in the wake – a far more challenging problem for estimation and control.

We simulate the flow via a direct numerical simulation (DNS) using the compressible flow solver CharLES (Brès *et al.* 2017). A C-shape mesh is created using Pointwise, as shown in figure 2(*a*), where the computational grid near the airfoil is also shown in the red box. The leading edge of the airfoil is located at the origin, $x/L_c = y/L_c = 0$. The size of the domain in the streamwise and normal direction is $x/L_c \in [-49, 50]$ and $y/L_c \in [-50, 50]$, respectively. To create a two-dimensional simulation, the spanwise direction is one cell thick ($z/L_c \in [0, 0.1]$) with symmetry boundary conditions. The grid consists of approximately 148 000 cells, with a finer grid resolution applied in the wake region, as studied in Appendix B. Characteristic far-field boundary conditions are applied along the outer boundary of the domain, and a sponge layer is used in the region $x/L_c \in [30, 50]$ to prevent reflections and ensure an effective outflow boundary condition as the wake exits the domain. Time integration is performed using a third-order total-variation-diminishing Runge–Kutta scheme (Gottlieb & Shu 1998). A constant time step is maintained by setting the Courant–Friedrichs–Lewy (CFL) number to approximately 1. After passing initial transients, data are collected for the duration $tU_\infty/L_c \in [0, 350]$. This extended time window ensures convergence of the mean and the second-order space–time statistics of vortex shedding in the wake. To verify statistical convergence, we analysed an extended time window $tU_\infty/L_c \in [0, 1000]$ and observed no change.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

	Present study	Marquet *et al.* (2022)	Error
$\bar{C}_D$	0.0862	0.088	2.05 %
$\bar{C}_L$	0.2941	0.289	1.76 %

Table 1. Comparison of the time-averaged drag and lift coefficients at $\alpha = 6.5^\circ$ with the results from incompressible periodic solution for a NACA 0012 airfoil at $Re_{L_c} = 5000$ and $Ma_\infty = 0.3$.

Figure 2(*b*) shows an instantaneous snapshot of the streamwise velocity field around the airfoil. A separation bubble on the suction side of the airfoil and vortex shedding in the wake are clearly observed. The mean streamwise velocity is shown in figure 2(*c*). The small spatial oscillation observed in the mean wake is likely a consequence of the top-bottom asymmetry of the instantaneous vortex shedding observed in figure 2(*b*). This mean flow (along with the mean of the other state variables) is used to define the linearisation used to construct the estimation and control kernels. Figure 2(*d*) shows an instantaneous snapshot of the streamwise velocity fluctuation, i.e. the mean-subtracted velocity. We aim to estimate and control (suppress) such fluctuations.

We validate the DNS via comparisons of the aerodynamic forces and vortex-shedding frequency against the results of Marquet *et al.* (2022), who considered incompressible flow over the same airfoil at the same Reynolds number and angle of attack. The time-averaged drag and lift coefficients,

$$C_D = \frac{F_D}{\frac{1}{2}\rho_\infty U_\infty^2 A}, \quad C_L = \frac{F_L}{\frac{1}{2}\rho_\infty U_\infty^2 A}, \tag{2.1}$$

are reported in table 1. Our results match those of Marquet *et al.* (2022) within a 2 % error. The power spectral density (PSD) of the transverse velocity at $x/L_c = 2.1$, $y/L_c = -0.11$ is shown in figure 3. The vortex-shedding frequency $St_\alpha \equiv \omega_r (L_c \sin\alpha)/(2\pi Ma_\infty)$ is approximately 0.169 in the present study, close to the value of 0.18 found by Marquet *et al.* (2022). This slight difference in the aerodynamic forces and vortex-shedding frequency could result from minor differences between the incompressible and compressible flows or differences in the grid refinement (the present grid is more finely resolved).

3. Global stability and resolvent analyses

3.1. *Linearisation and global stability analysis*

We start with the compressible Navier–Stokes equations written as

$$\frac{\partial \boldsymbol{q}}{\partial t} = \mathcal{F}(\boldsymbol{q}), \tag{3.1}$$

where $\boldsymbol{q}$ is a state vector of flow variables $[\rho, \rho u_x, \rho u_y, \rho u_z, \rho E]^{\mathrm{T}}$ and $\mathcal{F}$ is the nonlinear Navier–Stokes operator. We linearise about the mean flow rather than a steady fixed point due to the superior ability of the linear operator obtained from the mean flow to model the vortex-shedding frequency and corresponding flow structures observed in data (Barkley 2006; Colonius & Towne 2025). The equations are linearised using a Reynolds decomposition, giving

$$\frac{\partial \boldsymbol{q}'}{\partial t} - \boldsymbol{A}\boldsymbol{q}' = \boldsymbol{B}\boldsymbol{f}(\bar{\boldsymbol{q}}, \boldsymbol{q}'), \tag{3.2}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

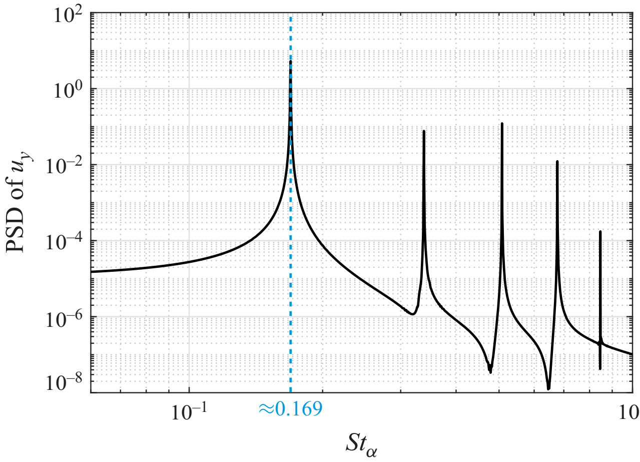


Figure 3. PSD of the wall-normal velocity at $(x, y)/L_c = (2.1, -0.11)$.

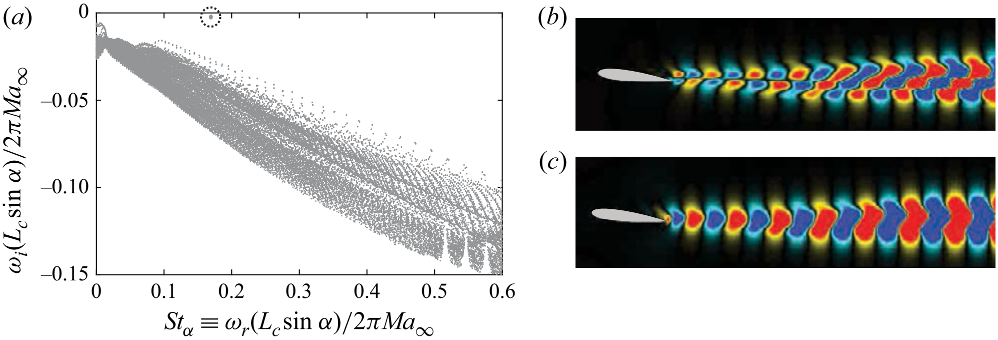


Figure 4. Eigenspectrum: (*a*) eigenspectrum, the dotted circle shows the dominant wake eigenmode at the vortex-shedding frequency $St_\alpha \approx 0.17$; (*b*) the corresponding streamwise velocity eigenmode; and (*c*) cross-streamwise velocity eigenmode.

where $\bar{\boldsymbol{q}}$ and $\boldsymbol{q}'$ represent the mean and perturbation state vectors, respectively, $\boldsymbol{A} = \partial\mathcal{F}(\bar{\boldsymbol{q}})/\partial\boldsymbol{q}$ is the linearised Navier–Stokes operator, $\boldsymbol{f}$ comprises the remaining nonlinear terms and any exogenous forcing, and $\boldsymbol{B}$ is an input matrix that can be used to restrict the form of $\boldsymbol{f}$. For convenience, we omit $(\cdot)'$ for perturbation from this point on. Our approach for constructing $\boldsymbol{A}$ is detailed in § 6.

Resolvent-based estimation and control are nominally applicable and robust only for globally stable systems (Schmid & Sipp 2016; Martini *et al.* 2020, 2022). Thus, we first conduct a global stability analysis to ensure that our linear operator is stable. Figure 4(*a*) shows the spectrum of the linearised Navier–Stokes operator $\boldsymbol{A}$, represented in terms of the complex frequency $\omega = i\lambda$, where λ is the eigenvalue of $\boldsymbol{A}$. The imaginary part ω_i is negative for all eigenvalues, indicating that the flow around the airfoil is globally stable. The least-damped eigenvalue appears at the frequency $St_\alpha \approx 0.169$, which matches the vortex-shedding frequency observed in the PSD in figure 3. The corresponding eigenmode, shown in figures 4(*b*) and 4(*c*), exhibits the expected characteristics of vortex shedding (Noack *et al.* 2003).

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

3.2. *Resolvent analysis*

Resolvent analysis is central to our estimation and control methods. Therefore, we compute the leading resolvent modes for the airfoil flow as a preliminary step to validate our implementation and to gain insights into the flow physics and the appropriate sensor and actuator placements. After transforming the linear system (3.2) into the frequency domain and solving for the state, we obtain

$$\hat{\boldsymbol{q}} = \boldsymbol{R}\boldsymbol{B}\hat{\boldsymbol{f}}, \tag{3.3}$$

where the resolvent operator is defined as $\boldsymbol{R} = (-i\omega \boldsymbol{I} - \boldsymbol{A})^{-1}$. The notation $\hat{(\cdot)}$ indicates a quantity in the frequency domain throughout this paper.

Resolvent analysis seeks input and output modes that maximise the resolvent gain

$$\sigma^2 = \frac{\|\hat{\boldsymbol{y}}\|^2}{\|\hat{\boldsymbol{f}}\|^2}, \tag{3.4}$$

where $\hat{\boldsymbol{y}} = \boldsymbol{C}\hat{\boldsymbol{q}}$ is an output of interest extracted from the state by the output matrix $\boldsymbol{C}$. The norm is induced by the inner product $\langle \boldsymbol{q}_1, \boldsymbol{q}_2 \rangle = \boldsymbol{q}_1^* \boldsymbol{W} \boldsymbol{q}_2$, where $\boldsymbol{W}$ is a weight matrix used to set a desired norm and $(\cdot)^*$ indicates the conjugate transpose. We use the Chu compressible energy norm (Chu 1965), and set $\boldsymbol{B}$ and $\boldsymbol{C}$ to the identity matrix for the preliminary resolvent analysis in this section (non-identity values will be used later for estimation and control).

After defining the weighted resolvent operator

$$\tilde{\boldsymbol{R}} = \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{C}\boldsymbol{R}\boldsymbol{B}\boldsymbol{W}^{-\frac{1}{2}}, \tag{3.5}$$

the resolvent gains and modes are obtained from the SVD

$$\tilde{\boldsymbol{R}} = \tilde{\boldsymbol{U}} \Sigma \tilde{\boldsymbol{V}}^*. \tag{3.6}$$

The resolvent gains are contained within the diagonal matrix $\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \ldots, \sigma_n]$. The forcing and response modes that maximise (3.4) are recovered as $\boldsymbol{V} = \boldsymbol{W}^{-1/2}\tilde{\boldsymbol{V}}$ and $\boldsymbol{U} = \boldsymbol{W}^{-1/2}\tilde{\boldsymbol{U}}$, respectively (Towne *et al.* 2018). We compute the resolvent modes by transforming the SVD into an equivalent eigenvalue problem, which is solved using an Arnoldi iteration in which the action of $\tilde{\boldsymbol{R}}$ is obtained by computing its LU decomposition (Jeun, Nichols & Jovanović 2016; Schmidt *et al.* 2018).

In figure 5, the peak of the leading resolvent gain is observed at the vortex-shedding frequency $St_\alpha \approx 0.169$. The streamwise and transverse velocity components of the optimal forcing and response modes are shown in figure 5(*b*–*e*). The optimal forcing mode is primarily located upstream and above the airfoil, while the optimal response mode is clearly observed downstream in the wake, as expected for a convective flow. The optimal response mode is similar to the dominant eigenmode.

4. Resolvent-based estimation and control framework

In this section, we provide a brief overview of the resolvent-based estimation and control framework developed by Towne *et al.* (2020) and Martini *et al.* (2020, 2022).

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

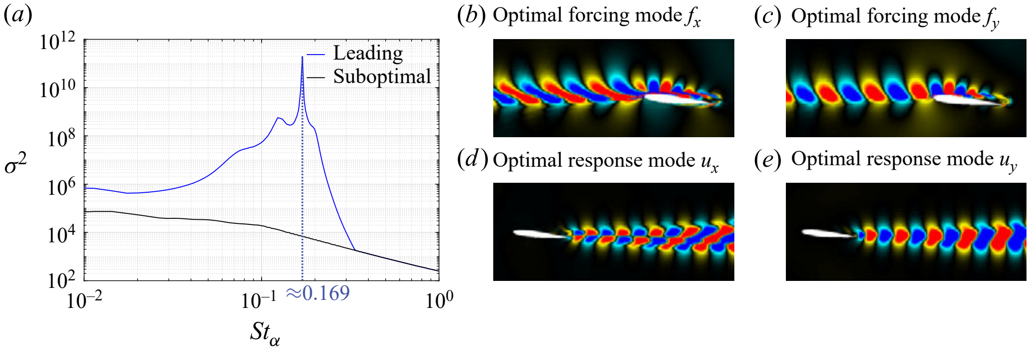


Figure 5. Resolvent gains, optimal forcing and response modes: (*a*) leading and second optimal gains; (*b*) optimal forcing mode of u_x; (*c*) optimal forcing mode of u_y; (*d*) optimal response mode of u_x; and (*e*) optimal response mode of u_y.

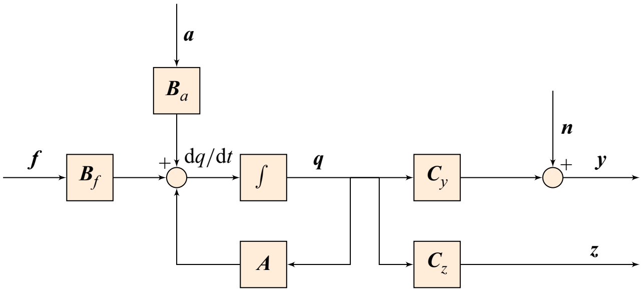


Figure 6. Block diagram representation of the linear system.

4.1. *System set-up*

As a generalisation of (3.2), we consider the linear time-invariant system

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}_f\boldsymbol{f}(t) + \boldsymbol{B}_a\boldsymbol{a}(t), \tag{4.1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}_y\boldsymbol{q}(t) + \boldsymbol{n}(t), \tag{4.1b}$$

$$\boldsymbol{z}(t) = \boldsymbol{C}_z\boldsymbol{q}(t). \tag{4.1c}$$

The system matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is the linearised compressible Navier–Stokes operator, $\boldsymbol{q} \in \mathbb{C}^n$ is the full state of flow and n is the total size of the state. The forcing $\boldsymbol{f} \in \mathbb{C}^{n_f}$ represents the nonlinear terms from the Navier–Stokes equations and any exogenous forcing. The forcing matrix $\boldsymbol{B}_f \in \mathbb{C}^{n\times n_f}$ restricts the form of the forcing $\boldsymbol{f}$, e.g. localises it to a certain spatial region. The actuation signal $\boldsymbol{a} \in \mathbb{C}^{n_a}$, which will be the output of the controller, is mapped onto the system by the actuation matrix $\boldsymbol{B}_a \in \mathbb{C}^{n\times n_a}$, where n_a is the total number of actuators. The sensor measurement $\boldsymbol{y} \in \mathbb{C}^{n_y}$ is extracted from the state by the measurement matrix $\boldsymbol{C}_y \in \mathbb{C}^{n_y\times n}$, which defines the sensor locations and types, and n_y is the total number of sensors. The sensor measurements are corrupted by the sensor noise $\boldsymbol{n} \in \mathbb{C}^{n_y}$. The target $\boldsymbol{z} \in \mathbb{C}^{n_z}$, i.e. the quantity that we wish to estimate and control, is extracted from the state by the target matrix $\boldsymbol{C}_z \in \mathbb{C}^{n_z\times n}$, where n_z is the total number of targets. Analogous to other optimal estimation and control methods, the forcing and noise are described by correlations that are assumed to be known.

Following Martini *et al.* (2022), we split the system (4.1) into two parts: the so-called forcing system that is driven by the forcing $\boldsymbol{f}(t)$ and corrupted by the sensor noise $\boldsymbol{n}(t)$,

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

$$\frac{\mathrm{d}\boldsymbol{q}_f}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_f(t) + \boldsymbol{B}_f\boldsymbol{f}(t), \tag{4.2a}$$

$$\boldsymbol{y}_f(t) = \boldsymbol{C}_y\boldsymbol{q}_f(t) + \boldsymbol{n}(t), \tag{4.2b}$$

$$\boldsymbol{z}_f(t) = \boldsymbol{C}_z\boldsymbol{q}_f(t), \tag{4.2c}$$

and the actuation system that is driven only by the actuation signal $\boldsymbol{a}(t)$,

$$\frac{\mathrm{d}\boldsymbol{q}_a}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_a(t) + \boldsymbol{B}_a\boldsymbol{a}(t), \tag{4.3a}$$

$$\boldsymbol{y}_a(t) = \boldsymbol{C}_y\boldsymbol{q}_a(t), \tag{4.3b}$$

$$\boldsymbol{z}_a(t) = \boldsymbol{C}_z\boldsymbol{q}_a(t). \tag{4.3c}$$

The state, sensor measurements and targets for the full system are recovered as

$$\boldsymbol{q} = \boldsymbol{q}_f + \boldsymbol{q}_a, \qquad \boldsymbol{y} = \boldsymbol{y}_f + \boldsymbol{y}_a \quad \text{and} \quad \boldsymbol{z} = \boldsymbol{z}_f + \boldsymbol{z}_a. \tag{4.4}$$

By applying the Fourier transform to (4.2) and (4.3), and using the resolvent operator defined in (3.3), we obtain the frequency-domain representations of sensor measurements and targets of the forcing and actuation systems,

$$\hat{\boldsymbol{y}}_f = \boldsymbol{R}_{yf}\hat{\boldsymbol{f}} + \hat{\boldsymbol{n}}, \tag{4.5a}$$

$$\hat{\boldsymbol{z}}_f = \boldsymbol{R}_{zf}\hat{\boldsymbol{f}}, \tag{4.5b}$$

$$\hat{\boldsymbol{y}}_a = \boldsymbol{R}_{ya}\hat{\boldsymbol{a}}, \tag{4.5c}$$

$$\hat{\boldsymbol{z}}_a = \boldsymbol{R}_{za}\hat{\boldsymbol{a}}. \tag{4.5d}$$

Here, $\boldsymbol{R}_{yf} = \boldsymbol{C}_y\boldsymbol{R}\boldsymbol{B}_f$, $\boldsymbol{R}_{zf} = \boldsymbol{C}_z\boldsymbol{R}\boldsymbol{B}_f$, $\boldsymbol{R}_{ya} = \boldsymbol{C}_y\boldsymbol{R}\boldsymbol{B}_a$ and $\boldsymbol{R}_{za} = \boldsymbol{C}_z\boldsymbol{R}\boldsymbol{B}_a$ are modified resolvent operators (sometimes called input–output operators) that will appear in the resolvent-based estimation and control kernels.

4.2. *Resolvent-based estimation*

The resolvent-based estimates of the target $\tilde{\boldsymbol{z}}$ are obtained via a convolution between the sensor measurements and a kernel. Here, we define three distinct resolvent-based estimation kernels: non-causal, truncated non-causal and causal kernels.

First, we define a non-causal estimator

$$\tilde{\boldsymbol{z}}_{nc}(t) = \int_{-\infty}^{\infty} \boldsymbol{T}_{nc}(t-\tau)\ \boldsymbol{y}(\tau)\mathrm{d}\tau, \tag{4.6}$$

where $\boldsymbol{T}_{nc} \in \mathbb{C}^{n_z \times n_y}$ is a non-causal estimation kernel. The optimal estimation kernel is obtained by minimising a cost function defined as the time-integrated expected value of the estimation error,

$$J_{nc} = \int_{-\infty}^{\infty} \mathbb{E}\big\{\boldsymbol{e}(t)^\dagger\boldsymbol{e}(t)\big\}\mathrm{d}t, \tag{4.7}$$

where the estimation error $\boldsymbol{e}(t) = \tilde{\boldsymbol{z}}(t) - \boldsymbol{z}(t)$ is the difference between the estimated and true target, $(\cdot)^\dagger$ denotes the adjoint operator using a suitable inner product, and $\mathbb{E}\{\cdot\}$ is the expectation operator. The cost function J_{nc} is minimised by setting its derivative with respect to $\boldsymbol{T}_{nc}$ to zero, yielding the optimal non-causal estimation kernel (Martini *et al.* 2020)

$$\hat{\boldsymbol{T}}_{nc}(\omega) = \boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}})^{-1}, \tag{4.8}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

where $\hat{\boldsymbol{F}} = \mathbb{E}\{\hat{\boldsymbol{f}}\hat{\boldsymbol{f}}^\dagger\}$ and $\hat{\boldsymbol{N}} = \mathbb{E}\{\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^\dagger\}$ are the cross-spectral densities (CSDs) of the forcing and sensor noise, respectively. The time-domain estimation kernel $\boldsymbol{T}_{nc}$ is obtained by inverse Fourier transforming the frequency domain kernel $\hat{\boldsymbol{T}}_{nc}$. In general, this kernel will be non-causal, i.e. $\boldsymbol{T}_{nc}(\tau)$ will not be strictly zero for $\tau < 0$, therefore requiring future sensor data, $\boldsymbol{y}(\tau > 0)$, which is unavailable for real-time applications, to evaluate (4.6).

Second, we define a truncated non-causal estimation kernel as a simple baseline approach to address the aforementioned lack of causality. This is done by simply truncating the integral in (4.6),

$$\tilde{z}_{tnc}(t) = \int_{-\infty}^{0} \boldsymbol{T}_{tnc}(t-\tau)\boldsymbol{y}(\tau)\mathrm{d}\tau, \tag{4.9}$$

which is equivalent to truncating the non-causal part of the kernel,

$$\boldsymbol{T}_{tnc}(\tau) = \begin{cases} \boldsymbol{T}_{nc}(\tau), & \tau \geqslant 0, \\ 0, & \tau < 0. \end{cases} \tag{4.10}$$

The downside of this approach is that the *ex post facto* truncation of the optimal non-causal kernel ruins its optimality.

Third, we define an optimal causal resolvent-based estimation kernel by enforcing causality using the Wiener–Hopf formalism (Martinelli 2009; Martini *et al.* 2022). The causal estimator,

$$\tilde{z}_{c}(t) = \int_{-\infty}^{0} \boldsymbol{T}_{c}(t-\tau)\boldsymbol{y}(\tau)\mathrm{d}\tau, \tag{4.11}$$

is defined in terms of the causal estimation kernel $\boldsymbol{T}_c \in \mathbb{C}^{n_z \times n_y}$. To find the optimal kernel under the constraint of causality, we modify the cost function (4.7) to read

$$J_c = \int_{-\infty}^{\infty} \mathbb{E}\left\{e(t)^\dagger e(t)\right\} + \left(\boldsymbol{\Lambda}_-(t)\boldsymbol{T}_c(t) + \boldsymbol{\Lambda}_-^\dagger(t)\boldsymbol{T}_c^\dagger(t)\right)\mathrm{d}t, \tag{4.12}$$

where $\boldsymbol{\Lambda}$ is a Lagrange multiplier that is used to force the causal kernel to be zero for the non-causal part ($\tau < 0$). The ($+$) and ($-$) subscripts indicate that the non-causal ($\tau < 0$) and causal ($\tau > 0$) parts, respectively, of a matrix or function are set to zero using a Wiener–Hopf factorisation. An introduction to the Wiener–Hopf method is described in Appendix B. Similar to the derivation of (4.8), we minimise the causal cost function (4.12) by setting its derivative with respect to $\boldsymbol{T}_c$ to zero. In doing so, we encounter the Wiener–Hopf problem (B4) with $\hat{\boldsymbol{G}} = \boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger$ and $\hat{\boldsymbol{H}} = \boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}$. Using the solution of this Wiener–Hopf problem given by (B8), the causal estimation kernel (Martini *et al.* 2022) is

$$\hat{\boldsymbol{T}}_c(\omega) = \left[\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger\left(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}\right)_-^{-1}\right]_+ \left(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}\right)_+^{-1}. \tag{4.13}$$

4.3. *Resolvent-based control*

Analogous to the estimation problem, the actuation signal $\boldsymbol{a}(t)$ that will be used to control the flow is obtained via a convolution between the sensor measurements and a control kernel (also referred to as control law in some studies). Again, we consider non-causal, truncated non-causal and causal variations of the resolvent-based control kernels (Martini *et al.* 2022).

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

The non-causal controller takes the form

$$\boldsymbol{a}_{nc}(t) = \int_{-\infty}^{\infty} \boldsymbol{\Gamma}_{nc}(t-\tau)\boldsymbol{y}_f(\tau)\mathrm{d}\tau, \tag{4.14}$$

where $\boldsymbol{\Gamma}_{nc} \in \mathbb{C}^{n_a \times n_y}$ is the non-causal control kernel. The optimal kernel is obtained by minimising the cost function

$$J_{nc} = \int_{-\infty}^{\infty} \mathbb{E}\big\{\boldsymbol{z}(t)^\dagger \boldsymbol{z}(t) + \boldsymbol{a}(t)^\dagger \boldsymbol{P}\boldsymbol{a}(t)\big\}\mathrm{d}t, \tag{4.15}$$

where $\boldsymbol{P}$ is a weight matrix that penalises the actuation effort. Minimising this cost function yields the optimal non-causal control kernel

$$\hat{\boldsymbol{\Gamma}}_{nc}(\omega) = \big(\boldsymbol{R}_{za}^\dagger \boldsymbol{R}_{za} + \hat{\boldsymbol{P}}\big)^{-1}\big(-\boldsymbol{R}_{za}^\dagger\big)\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger\big(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}\big)^{-1}. \tag{4.16}$$

The truncated non-causal control kernel is defined as

$$\boldsymbol{\Gamma}_{tnc}(\tau) = \begin{cases} \boldsymbol{\Gamma}_{nc}(\tau), & \tau \geqslant 0, \\ 0, & \tau < 0, \end{cases} \tag{4.17}$$

from which the actuation signal is computed as

$$\boldsymbol{a}_{tnc}(t) = \int_{-\infty}^{0} \boldsymbol{\Gamma}_{tnc}(t-\tau)\boldsymbol{y}_f(\tau)\mathrm{d}\tau. \tag{4.18}$$

As before, truncating the optimal non-causal kernel yields a non-optimal causal kernel.

Finally, the optimal causal controller is

$$\boldsymbol{a}_{c}(t) = \int_{-\infty}^{0} \boldsymbol{\Gamma}_{c}(t-\tau)\boldsymbol{y}_f(\tau)\mathrm{d}\tau, \tag{4.19}$$

and the optimal causal control kernel is obtained by using Lagrange multipliers to enforce causality, expressed by the cost function

$$J_c = \int_{-\infty}^{\infty} \mathbb{E}\big\{\boldsymbol{z}(t)^\dagger \boldsymbol{z}(t) + \boldsymbol{a}(t)^\dagger \boldsymbol{P}\boldsymbol{a}(t) + (\boldsymbol{\Lambda}_-(t)\boldsymbol{\Gamma}_c(t) + \boldsymbol{\Lambda}_-^\dagger(t)\boldsymbol{\Gamma}_c^\dagger(t))\big\}\mathrm{d}t. \tag{4.20}$$

Minimising (4.20) leads to the Wiener–Hopf problem (B5) with $\hat{\boldsymbol{K}} = \boldsymbol{R}_{za}^\dagger \boldsymbol{R}_{za} + \hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{L}} = -\boldsymbol{R}_{za}^\dagger$. Using the solution of this Wiener–Hopf problem given in (B9), the optimal causal resolvent-based control kernel is

$$\begin{aligned}\hat{\boldsymbol{\Gamma}}_{c}(\omega) = \big(\boldsymbol{R}_{za}^\dagger \boldsymbol{R}_{za} + \hat{\boldsymbol{P}}\big)_+^{-1}\big[\big(\boldsymbol{R}_{za}^\dagger \boldsymbol{R}_{za} + \hat{\boldsymbol{P}}\big)_-^{-1}\big(-\boldsymbol{R}_{za}^\dagger\big)\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger \\ \big(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}\big)_-^{-1}\big]_+\big(\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger + \hat{\boldsymbol{N}}\big)_+^{-1}.\end{aligned} \tag{4.21}$$

In (4.14), (4.18) and (4.19), the actuation signal is obtained as a convolution between the control kernel and the sensor measurement $\boldsymbol{y}_f$ for the forcing system (4.2), i.e. the control kernels were derived using a measurement excluding the response of the system (4.3) to actuation. In practice, only the complete measurement $\boldsymbol{y} = \boldsymbol{y}_f + \boldsymbol{y}_a$ is available. Considering the full (combined) linear system (4.1), the final closed-loop controller takes the form

$$\boldsymbol{a}(t) = \int_{-\infty}^{0} \boldsymbol{\Gamma}_{cl}(t-\tau)\boldsymbol{y}(\tau)\mathrm{d}\tau, \tag{4.22}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

where the final closed-loop kernel is

$$\hat{\boldsymbol{\Gamma}}_{cl} = \left(\boldsymbol{I} + \hat{\boldsymbol{\Gamma}} \boldsymbol{R}_{ya}\right)^{-1} \hat{\boldsymbol{\Gamma}}, \tag{4.23}$$

with $\hat{\boldsymbol{\Gamma}}$ replaced by $\hat{\boldsymbol{\Gamma}}_{tnc}$ or $\hat{\boldsymbol{\Gamma}}_c$ for truncated non-causal and optimal causal resolvent-based control, respectively.

5. Computing estimation and control kernels

In this section, we present two approaches to compute resolvent-based estimation and control kernels (Martini *et al.* 2022). First, we describe an operator-based approach that allows for efficient implementation of linear simulations without the need for inverting the operator or performing prior model reduction, making it particularly efficient for large-scale problems. Second, we explain a data-driven approach that does not require the construction of the linearised Navier–Stokes operator. Instead, this method uses training data from simulations or experiments to build cross-spectral densities, which are then used to compute the components of the estimation and control kernels.

5.1. *Operator-based approach*

The resolvent operator $\boldsymbol{R}$ is defined in terms of an inverse, the cost of which scales poorly with the problem dimension n and becomes computationally expensive for large systems. To circumvent this, we employ a time-stepping approach that avoids the inverse operation and instead constructs the necessary modified resolvent operators that appear in the estimation and control kernels by solving linear equations in the time domain (Martini *et al.* 2020, 2022; Farghadan, Martini & Towne 2023). For both cases, the cost of this approach scales linearly with the problem dimension, avoiding the need to reduce the system via *a priori* model reduction and the associated loss of accuracy. Individual modified resolvent operators such as $\boldsymbol{R}_{za}$ and $\boldsymbol{R}_{ya}$ can be obtained using a single-stage run of the direct linear equations. Products of modified resolvent operators (and the forcing CSD) such as $\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^{\dagger}$ and $\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^{\dagger}$ can be obtained with greater efficiently using two-stage runs of both the adjoint and direct linear equations.

5.1.1. *Single-stage run*

The operators $\boldsymbol{R}_{za}$ and $\boldsymbol{R}_{ya}$ appearing in (4.21) and (4.23), respectively, can be constructed by computing a series of impulse responses of the actuation system (4.3),

$$\frac{\mathrm{d}\boldsymbol{q}_{a,k}}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_{a,k}(t) + \boldsymbol{B}_{a,k}\delta(t), \tag{5.1a}$$

$$\boldsymbol{y}_{a,k}(t) = \boldsymbol{C}_y\boldsymbol{q}_{a,k}(t), \tag{5.1b}$$

$$\boldsymbol{z}_{a,k}(t) = \boldsymbol{C}_z\boldsymbol{q}_{a,k}(t). \tag{5.1c}$$

Here, $\boldsymbol{y}_{a,k} \in \mathbb{C}^{n_y}$ and $\boldsymbol{z}_{a,k} \in \mathbb{C}^{n_z}$ are the sensor and target measurement of the direct system forced by an impulse $\delta(t)$ located at the $k = 1, 2, 3, \ldots$ actuator, as encoded by the $k = 1, 2, 3, \ldots$ column of the actuation matrix, $\boldsymbol{B}_{a,k}$. By collecting these data for each actuator $k = 1, \ldots, n_a$ and taking a Fourier transform, we obtain

$$\hat{\boldsymbol{Y}}_a = \begin{bmatrix}\hat{\boldsymbol{y}}_{a,1} & \hat{\boldsymbol{y}}_{a,2} & \ldots & \hat{\boldsymbol{y}}_{a,n_a}\end{bmatrix} = \boldsymbol{R}_{ya}, \tag{5.2a}$$

$$\hat{\boldsymbol{Z}}_a = \begin{bmatrix}\hat{\boldsymbol{z}}_{a,1} & \hat{\boldsymbol{z}}_{a,2} & \ldots & \hat{\boldsymbol{z}}_{a,n_a}\end{bmatrix} = \boldsymbol{R}_{za}, \tag{5.2b}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

with $\hat{\boldsymbol{Y}}_a \in \mathbb{C}^{n_y \times n_a}$ and $\hat{\boldsymbol{Z}}_a \in \mathbb{C}^{n_z \times n_a}$. That is, the Fourier transform of each measurement $\boldsymbol{y}_{a,k}$ and $\boldsymbol{z}_{a,k}$ yields a column of the modified resolvent operators $\boldsymbol{R}_{ya}$ and $\boldsymbol{R}_{za}$, respectively.

5.1.2. *Two-stage run*

While single-stage runs could be used to construct all of the modified resolvent operators in the estimation and control kernels, certain products thereof can be constructed more efficiently using pairs of adjoint and direct runs. The procedure begins with solving the adjoint system

$$-\frac{\mathrm{d}\boldsymbol{q}_{f,i}}{\mathrm{d}t}(t) = \boldsymbol{A}^\dagger \boldsymbol{q}_{f,i}(t) + \boldsymbol{C}_{y,i}^\dagger \delta(t), \tag{5.3a}$$

$$\boldsymbol{s}_i(t) = \boldsymbol{B}_f^\dagger \boldsymbol{q}_{f,i}(t), \tag{5.3b}$$

where $\boldsymbol{A}^\dagger$ is the adjoint linearised Navier–Stokes operator and the subscript i indicates the sensor defined by the ith row of the sensor measurement matrix $\boldsymbol{C}_y$. The output $\boldsymbol{s}_i$ of the adjoint run is used as a forcing in a corresponding direct run of the forcing system (4.2),

$$\frac{\mathrm{d}\boldsymbol{q}_{f,i}}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_{f,i}(t) + \boldsymbol{B}_f \boldsymbol{s}_i(t), \tag{5.4a}$$

$$\boldsymbol{y}_{f,i}(t) = \boldsymbol{C}_y \boldsymbol{q}_{f,i}(t) + \boldsymbol{n}_i(t), \tag{5.4b}$$

$$\boldsymbol{z}_{f,i}(t) = \boldsymbol{C}_z \boldsymbol{q}_{f,i}(t). \tag{5.4c}$$

As in the single-stage run, by collecting each of the final sensor and target measurements, and taking a Fourier transform, we obtain

$$\hat{\boldsymbol{Y}}_f = \begin{bmatrix} \hat{\boldsymbol{y}}_1 & \hat{\boldsymbol{y}}_2 & \cdots & \hat{\boldsymbol{y}}_{n_y} \end{bmatrix} = \boldsymbol{R}_{yf}\boldsymbol{R}_{yf}^\dagger, \tag{5.5a}$$

$$\hat{\boldsymbol{Z}}_f = \begin{bmatrix} \hat{\boldsymbol{z}}_1 & \hat{\boldsymbol{z}}_2 & \cdots & \hat{\boldsymbol{z}}_{n_y} \end{bmatrix} = \boldsymbol{R}_{zf}\boldsymbol{R}_{yf}^\dagger, \tag{5.5b}$$

with $\hat{\boldsymbol{Y}}_f \in \mathbb{C}^{n_y \times n_y}$ and $\hat{\boldsymbol{Z}}_f \in \mathbb{C}^{n_z \times n_y}$. To account for the nonlinearity of the flow, the coloured forcing statistics, $\hat{\boldsymbol{F}}$, can be incorporated into (5.5) during adjoint and direct simulations, resulting in $\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger$ and $\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}\boldsymbol{R}_{yf}^\dagger$ (Jung 2024).

Finally, (5.2) and (5.5) are used to write the estimation kernels in (4.8) and (4.13), and control kernels in (4.16) and (4.21) in terms of $\hat{\boldsymbol{Y}}_a$, $\hat{\boldsymbol{Z}}_a$, $\hat{\boldsymbol{Y}}_f$ and $\hat{\boldsymbol{Z}}_f$. The final operator-based estimation and control kernels are

$$\hat{\boldsymbol{T}}_{nc,O} = \hat{\boldsymbol{Z}}_f(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})^{-1}, \tag{5.6a}$$

$$\hat{\boldsymbol{T}}_{c,O} = [\hat{\boldsymbol{Z}}_f(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})_-^{-1}]_+(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})_+^{-1}, \tag{5.6b}$$

and

$$\hat{\boldsymbol{\Gamma}}_{nc,O} = (\hat{\boldsymbol{Z}}_a^\dagger \hat{\boldsymbol{Z}}_a + \hat{\boldsymbol{P}})^{-1}(-\hat{\boldsymbol{Z}}_a^\dagger)\hat{\boldsymbol{Z}}_f(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})^{-1}, \tag{5.7a}$$

$$\hat{\boldsymbol{\Gamma}}_{c,O} = (\hat{\boldsymbol{Z}}_a^\dagger \hat{\boldsymbol{Z}}_a + \hat{\boldsymbol{P}})_+^{-1}[(\hat{\boldsymbol{Z}}_a^\dagger \hat{\boldsymbol{Z}}_a + \hat{\boldsymbol{P}})_-^{-1}(-\hat{\boldsymbol{Z}}_a^\dagger)\hat{\boldsymbol{Z}}_f(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})_-^{-1}]_+(\hat{\boldsymbol{Y}}_f + \hat{\boldsymbol{N}})_+^{-1}. \tag{5.7b}$$

5.2. *Data-driven approach*

When the linearised Navier–Stokes operator is unavailable, a data-driven approach (Martini *et al.* 2022) can be employed to build the required modified resolvent operators

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

using CSDs computed from data (Towne *et al.* 2018, 2020). This approach extends the applicability of resolvent-based estimation and control to experimental settings and circumvents the need for adjoint solvers. Additionally, when the CSD is computed from the dataset of a nonlinear system, the resulting kernels automatically include the forcing CSD $\hat{\boldsymbol{F}}$, which statistically accounts for the nonlinearity of the flow, thereby improving the estimation and control performance for the nonlinear system.

The nonlinear terms in the Navier–Stokes equations act as a forcing of the resolvent operator (McKeon & Sharma 2010) and their influence is crucial in complex dynamic systems (Amaral *et al.* 2021). To explicitly address the nonlinear terms, we split the forcing vector $\boldsymbol{f}$ into two components: the external forcing $\boldsymbol{f}_{ext}$ and the nonlinear terms $\boldsymbol{f}_{nl}$. This distinction helps us better understand their impact when building the CSDs from the data. The overall forcing term in (4.1) can be split as

$$\boldsymbol{B}_f \boldsymbol{f} = \begin{bmatrix} \boldsymbol{B}_{ext} & \boldsymbol{B}_{nl} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_{ext} \\ \boldsymbol{f}_{nl} \end{bmatrix}, \tag{5.8}$$

where $\boldsymbol{B}_{ext} \in \mathbb{C}^{n\times n_{ext}}$ and $\boldsymbol{B}_{nl} \in \mathbb{C}^{n\times n}$. Typically, the region subject to external forcing is smaller than the overall domain, such that $n_{ext} < n$. In a linear system, the nonlinear terms $\boldsymbol{f}_{nl}$ are not included, allowing us to analyse the linear dynamics of the flow, and to build linear estimators and controllers. However, our ultimate goal is to manipulate the unsteady fluctuations inherent in the actual flow, which necessitates considering nonlinearity. For the nonlinear system, we collect data from the systems without and with external forcing to better capture the behaviour of the systems influenced by external forcing. The forcing system (4.2) without and with the external forcing can be expressed as

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}_{nl}\boldsymbol{f}_{nl}(t), \tag{5.9}$$

$$\frac{\mathrm{d}\boldsymbol{q}_e}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_e(t) + \boldsymbol{B}_{ext}\boldsymbol{f}_{ext} + \boldsymbol{B}_{nl}\boldsymbol{f}_{nl,e}(t). \tag{5.10}$$

The subscript e indicates the flow quantity that contains the development of nonlinearity, which was impacted by the external forcing. The $\boldsymbol{f}_{nl,e}$ term is evolved by the external forcing in time and space, so in the nonlinear system, the nonlinear effect can not be neglected. Equation (5.9) is the DNS or LES system without any source term. We assume external and nonlinear forcings are uncorrelated. Then, we obtain

$$\begin{bmatrix} \hat{\boldsymbol{y}}_{f,nl} \\ \hat{\boldsymbol{z}}_{f,nl} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{yf,nl} \\ \boldsymbol{R}_{zf,nl} \end{bmatrix} \hat{\boldsymbol{f}}_{nl} + \begin{bmatrix} \hat{\boldsymbol{n}} \\ 0 \end{bmatrix}, \tag{5.11}$$

$$\begin{bmatrix} \hat{\boldsymbol{y}}_{f,ext,nl} \\ \hat{\boldsymbol{z}}_{f,ext,nl} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{yf,ext} & \boldsymbol{R}_{yf,nl} \\ \boldsymbol{R}_{zf,ext} & \boldsymbol{R}_{zf,nl} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{f}}_{ext} \\ \hat{\boldsymbol{f}}_{nl,e} \end{bmatrix} + \begin{bmatrix} \hat{\boldsymbol{n}} \\ 0 \end{bmatrix}. \tag{5.12}$$

Computing the cross-spectral density of $[\hat{\boldsymbol{y}} \quad \hat{\boldsymbol{z}}]^T$ from (5.11) and (5.12) gives

$$\begin{bmatrix} \boldsymbol{S}_{yy} \\ \boldsymbol{S}_{zy} \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{S}_{yy,f,nl} \\ \boldsymbol{S}_{zy,f,nl} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{yf,nl}\hat{\boldsymbol{F}}_{nl}\boldsymbol{R}^{\dagger}_{yf,nl} + \hat{\boldsymbol{N}} \\ \boldsymbol{R}_{zf,nl}\hat{\boldsymbol{F}}_{nl}\boldsymbol{R}^{\dagger}_{yf,nl} \end{bmatrix}, \tag{5.13}$$

$$\begin{bmatrix} \boldsymbol{S}_{yy,e} \\ \boldsymbol{S}_{zy,e} \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{S}_{yy,f,ext,nl} \\ \boldsymbol{S}_{zy,f,ext,nl} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{yf,nl}\hat{\boldsymbol{F}}_{nl,r}\boldsymbol{R}^{\dagger}_{yf,nl} + \boldsymbol{R}_{yf,ext}\hat{\boldsymbol{F}}_{ext}\boldsymbol{R}^{\dagger}_{yf,ext} + \hat{\boldsymbol{N}} \\ \boldsymbol{R}_{zf,nl}\hat{\boldsymbol{F}}_{nl,r}\boldsymbol{R}^{\dagger}_{yf,nl} + \boldsymbol{R}_{zf,ext}\hat{\boldsymbol{F}}_{ext}\boldsymbol{R}^{\dagger}_{yf,ext} \end{bmatrix}, \tag{5.14}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

with $\boldsymbol{S}_{yy} = \mathbb{E}\{\hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^{\dagger}\}$ and $\boldsymbol{S}_{zy} = \mathbb{E}\{\hat{\boldsymbol{z}}\hat{\boldsymbol{y}}^{\dagger}\}$. Since the right-hand sides of (5.13) and (5.14) contain the terms needed to build the estimation and control kernels (4.8), (4.13), (4.16) and (4.21), this shows that the CSDs on the left-hand side can be used in their place. Note that the CSDs inherently contain statistical information about the nonlinearity of the flow within the forcing CSD matrix.

The data-driven non-causal and causal estimation kernels in (4.8) and (4.13) are computed using the CSDs from (5.13), yielding

$$\hat{\boldsymbol{T}}_{nc,D} = \boldsymbol{S}_{zy}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})^{-1}, \tag{5.15a}$$

$$\hat{\boldsymbol{T}}_{c,D} = [\boldsymbol{S}_{zy}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})_{-}^{-1}]_{+}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})_{+}^{-1}. \tag{5.15b}$$

The control kernels require $\boldsymbol{R}_{ya}$ and $\boldsymbol{R}_{za}$, which do not include the forcing CSD matrix and can be obtained by imposing an impulse forcing at the actuator location in the nonlinear system,

$$\frac{\mathrm{d}\boldsymbol{q}_{a,k}}{\mathrm{d}t}(t) = \boldsymbol{A}\boldsymbol{q}_{a,k}(t) + \boldsymbol{B}_{a,k}\delta(t) + \boldsymbol{B}_{nl}\boldsymbol{f}_{nl,k}(t). \tag{5.16}$$

The Fourier-transformed sensor and target reading from the nonlinear system forced by impulses at the actuator locations are subtracted by the same quantities, $\hat{\boldsymbol{y}}_{f,nl}$ and $\hat{\boldsymbol{z}}_{f,nl}$, from the nonlinear system without the actuators. Then, we obtain

$$\begin{bmatrix} \hat{\boldsymbol{Y}}_s \\ \hat{\boldsymbol{Z}}_s \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{y}}_a - \hat{\boldsymbol{y}}_{f,nl} \\ \hat{\boldsymbol{z}}_a - \hat{\boldsymbol{z}}_{f,nl} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{R}_{ya} \\ \boldsymbol{R}_{za} \end{bmatrix}, \tag{5.17}$$

where the subscript s denotes the Fourier-transformed readings computed through subtraction using the data-driven approach. The resulting CSDs are

$$\begin{bmatrix} \boldsymbol{S}_{yy,s} \\ \boldsymbol{S}_{zz,s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{ya}\boldsymbol{R}_{ya}^{\dagger} \\ \boldsymbol{R}_{za}\boldsymbol{R}_{za}^{\dagger} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{Y}_a\boldsymbol{Y}_a^{\dagger} \\ \boldsymbol{Z}_a\boldsymbol{Z}_a^{\dagger} \end{bmatrix}. \tag{5.18}$$

Using this result, we can finally obtain the data-driven control kernels,

$$\hat{\boldsymbol{\varGamma}}_{nc,D} = (\boldsymbol{S}_{zz,s}^{\dagger} + \hat{\boldsymbol{P}})^{-1}(-\hat{\boldsymbol{Z}}_s^{\dagger})\boldsymbol{S}_{zy}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})^{-1}, \tag{5.19a}$$

$$\hat{\boldsymbol{\varGamma}}_{c,D} = (\boldsymbol{S}_{zz,s}^{\dagger} + \hat{\boldsymbol{P}})_{+}^{-1}[(\boldsymbol{S}_{zz,s}^{\dagger} + \hat{\boldsymbol{P}})_{-}^{-1}(-\hat{\boldsymbol{Z}}_s^{\dagger})\boldsymbol{S}_{zy}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})_{-}^{-1}]_{+}(\boldsymbol{S}_{yy} + \hat{\boldsymbol{N}})_{+}^{-1}. \tag{5.19b}$$

To apply these kernels to the system that is excited by the external forcing, $\boldsymbol{S}_{yy}$ and $\boldsymbol{S}_{zy}$ can be replaced with $\boldsymbol{S}_{yy,r}$ and $\boldsymbol{S}_{zy,r}$.

6. Implementation of a resolvent-based estimation and control tool

In this section, we describe our implementation of the resolvent-based estimation and control tools described in §§ 4 and 5 within CharLES, an unstructured compressible flow solver for large-scale problems within a high-performance computing environment. The conceptional flow chart for the software is depicted in figure 7. These tools can be used for any flows simulated within the CharLES solver (Brès *et al.* 2017) and can be linked with any external packages written in C/C++. To parallelise linear algebra computation and the streaming Fourier transform, we integrated the solver with PETSc (Balay *et al.* 2019) and FFTW (Frigo & Johnson 2005).

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

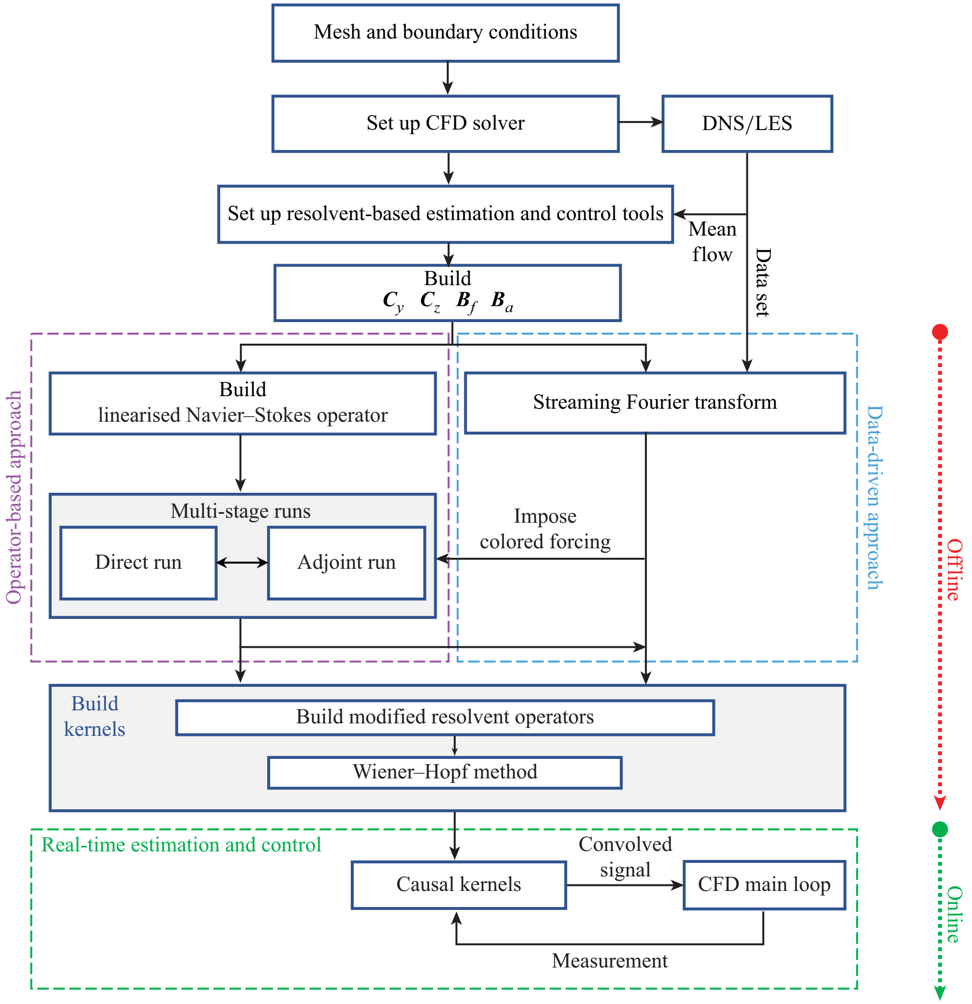

Mesh and boundary conditions
Set up CFD solver
DNS/LES
Set up resolvent-based estimation and control tools
Mean flow
Data set
Build
$\boldsymbol{C}_y$ $\boldsymbol{C}_z$ $\boldsymbol{B}_f$ $\boldsymbol{B}_a$
Operator-based approach
Build
linearised Navier–Stokes operator
Streaming Fourier transform
Data-driven approach
Offline
Multi-stage runs
Direct run
Adjoint run
Impose
colored forcing
Build
kernels
Build modified resolvent operators
Wiener–Hopf method
Real-time estimation and control
Convolved
signal
Causal kernels
CFD main loop
Measurement
Online

Figure 7. Flow chart for the new implementation of resolvent-based estimation and control tools within the compressible flow solver CharLES.

6.1. *Linearisation*

The operator-based approach in § 5 requires access to the linear operator $\boldsymbol{A}$ or the actions of both the linear and adjoint operators. We use a matrix-forming approach in which the linear operator is directly computed and stored within the nonlinear solver. This approach accounts for the numerical schemes and boundary conditions employed in the nonlinear simulations, and ensures that the linear operator is readily available at run-time. Extracting the linear operator from large-scale, compressible CFD solvers is not a trivial task and several approaches have been suggested in the literature (Nielsen & Kleb 2006; Fosas, Sipp & Schmid 2012; Cook *et al.* 2018; Bhagwat 2020). The approach we adopt most closely mirrors that of Nielsen & Kleb (2006) and Cook *et al.* (2018).

The CharLES solver uses a control-volume-based finite volume method. A straightforward way to extract the linear operator numerically is to use a finite-difference approximation of the Jacobian, computing the linear operator one column at a time. For example, the jth column of the linear operator can be extracted using a

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

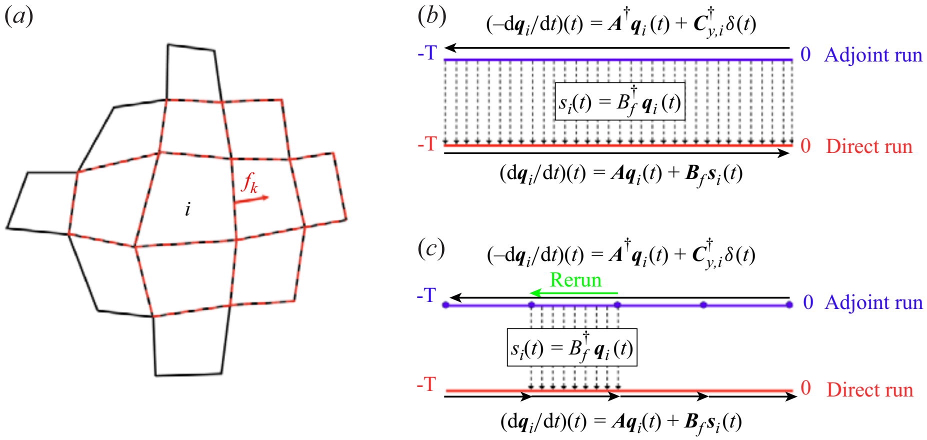

(a)
i
f_k
(b)
$(-\mathrm{d}\boldsymbol{q}_i/\mathrm{d}t)(t) = \boldsymbol{A}^\dagger \boldsymbol{q}_i(t) + \boldsymbol{C}^\dagger_{y,i}\delta(t)$
-T
0
Adjoint run
$s_i(t) = B^\dagger_f \boldsymbol{q}_i(t)$
-T
0
Direct run
$(\mathrm{d}\boldsymbol{q}_i/\mathrm{d}t)(t) = \boldsymbol{A}\boldsymbol{q}_i(t) + \boldsymbol{B}_f \boldsymbol{s}_i(t)$
(c)
$(-\mathrm{d}\boldsymbol{q}_i/\mathrm{d}t)(t) = \boldsymbol{A}^\dagger \boldsymbol{q}_i(t) + \boldsymbol{C}^\dagger_{y,i}\delta(t)$
Rerun
-T
0
Adjoint run
$s_i(t) = B^\dagger_f \boldsymbol{q}_i(t)$
-T
0
Direct run
$(\mathrm{d}\boldsymbol{q}_i/\mathrm{d}t)(t) = \boldsymbol{A}\boldsymbol{q}_i(t) + \boldsymbol{B}_f \boldsymbol{s}_i(t)$

Figure 8. Linearisation within the CharLES solver: (*a*) computational stencil for a control volume (CV_i) labelled i. The computational stencil needed to compute the flux at a specific face k of CV_i is shown with red dashed lines. The CV-based stencil is the union of all face-based stencils for a given CV. A schematic of the two-stage adjoint-direct run: (*b*) without checkpointing and (*c*) with checkpointing. Checkpoints are marked with solid blue circles.

second-order approximation as

$$\boldsymbol{A}(:, j) = \frac{\mathcal{F}\left(\bar{\boldsymbol{q}} + \epsilon \boldsymbol{e}_j\right) - \mathcal{F}\left(\bar{\boldsymbol{q}} - \epsilon \boldsymbol{e}_j\right)}{2\epsilon}, \tag{6.1}$$

where j refers to the jth degree of freedom and $\mathcal{F}$ represents the right-hand side of the CFD code (the discretised nonlinear compressible Navier–Stokes operator). However, this approach is computationally expensive, as the number of the global right-hand-side evaluations ($\mathcal{F}$) required to form the operator equals the problem dimension n.

Instead, we use an approach adopted by Nielsen & Kleb (2006) and more recently applied by Cook *et al.* (2018) that relies on perturbing multiple degrees of freedom (DOFs) simultaneously. The key insight is that perturbing an element of the state vector $\boldsymbol{q}$ affects only a small number of nearby control volumes, known as its computational stencil. This stencil is determined by the numerics used in the CharLES solver (Brès *et al.* 2017). Thus, it is feasible to perturb multiple elements of the state vector at once without the perturbations interfering with each other. This approach allows us to compute multiple columns of $\boldsymbol{A}$ simultaneously, significantly reducing the computational cost. This is accomplished by replacing unit vector $\boldsymbol{e}_j$ in (6.1) with $\tilde{\boldsymbol{e}}_k$, which represents the multiple degrees of freedom perturbed simultaneously. Using this approach, the number of right-hand-side evaluations $\mathcal{F}(\boldsymbol{q})$ scales with the extent of the computational stencil, not with the size of the problem. In the spatial discretisation in CharLES, the flux at a face depends on the adjoining control volumes and their immediate face neighbours, as illustrated in figure 8(*a*).

To optimise the process of building the $\tilde{\boldsymbol{e}}_k$ vectors, we sort the computational grid into lists of non-overlapping degrees of freedom on a single processor and then broadcast this information to all other processors. The linearised compressible Navier–Stokes operator $\boldsymbol{A}$ is extracted and saved at the initial step, and the routine is performed only once before the time-stepping or nonlinear runs (DNS or LES). Typically, the number of right-hand-side evaluations required for standard hexahedral/quadrilateral grids is approximately 100 and 300 for two-dimensional (2-D) and three-dimensional (3-D) problems, respectively. The

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

small parameter ϵ in (6.1) is empirically chosen to minimise the error in the numerical derivatives. We have found $\epsilon = \epsilon_0 ||\boldsymbol{q}||$ with $\epsilon_0 = 10^{-6}$ to be an effective and robust choice. The $\|\boldsymbol{q}\|$ is computed separately for each quantity (density, velocities and energy).

6.2. *Efficient implementation of the estimation and control tools*

To effectively store and use the linear operator for large-scale numerical linear algebra computations such as matrix-vector products, we parallelise our implementation using the open-source linear algebra package PETSc (Balay *et al.* 2019). PETSc leverages the underlying domain decomposition used by the CFD solver to partition the computational grid. Once the matrices such as $\boldsymbol{A}$, $\boldsymbol{C}_y$, $\boldsymbol{C}_z$, $\boldsymbol{B}_a$ and $\boldsymbol{B}_f$ are constructed, PETSc is used to advance the linear dynamics for the single- or two-stage runs described in § 5.1. To advance these equations in time, we use the TVD-RK3 scheme (Gottlieb & Shu 1998), the same scheme used by the nonlinear solver.

In the two-stage run described in § 5.1.2, the time-series data of $\boldsymbol{s}_i(t)$ from (5.3*b*) must be stored to serve as the forcing term $\boldsymbol{B}_f \boldsymbol{s}_i(t)$ in (5.4*a*), as shown schematically in figure 8(*b*). However, storing all snapshots from the initial step of the adjoint run to the time T where the direct run begins becomes prohibitively expensive over long time horizons. We use checkpointing, as shown in figure 8(*c*), to address this issue by only storing snapshots at particular intervals during the adjoint run in (5.3*a*). After completing the first full adjoint run, the direct run is advanced in chunks. The adjoint run is then rerun between the last two checkpoints, using only the stored snapshots within that interval, before conducting the direct run through the same interval. This approach reduces memory usage, which is particularly beneficial for large-scale problems. For example, if advancing N_t time steps in the adjoint run, the storage requirement can be reduced from $O(N_t)$ to $O(W_t + N_t/W_t)$, where W_t is the length of the interval between two checkpoints and N_t/W_t roughly indicates the number of checkpoints. The minimum memory requirement is achieved when $W_t \approx \sqrt{N_t}$. Thus, checkpointing reduces the memory required from $O(N_t)$ to $O(2\sqrt{N_t})$.

Constructing the data-driven estimation and control kernels (5.15) and (5.19) requires the computation of CSDs, as described in § 5.2. Typically, CSDs are computed using Fast Fourier transforms (FFTs), which require simultaneous access to many snapshots of the state (to be precise, the number of desired frequencies n_{freq}). However, this approach quickly becomes infeasible when each snapshot is large, i.e. when the state dimension n is large. To reduce data size and memory usage, we employ streaming discrete Fourier transforms (DFTs), as proposed by Schmidt & Towne (2019) within the context of a streaming algorithm for spectral proper orthogonal decomposition (SPOD) and further used by Farghadan *et al.* (2023) within a scalable time-stepping algorithm for resolvent analysis.

The streaming algorithm requires access to only one instantaneous snapshot of the state at a time, avoiding the need to store the entire time series. This is achieved by using the definition of the discrete Fourier transform (DFT), which yields results equivalent to the FFT. Each snapshot contributes to the summation of the Fourier modes as

$$\hat{\boldsymbol{f}}^l_k = \sum_{j=1}^{n_{freq}} \boldsymbol{f}^l_j \, p_{jk}, \tag{6.2}$$

where $p_{jk} = e^{(k-1)(j-1)(-i2\pi/n_{freq})}$, with k representing the kth frequency, j the jth snapshot and l the lth block of data. The full time series data are divided into multiple blocks, each windowed with a 50 % overlap. Each snapshot is multiplied by the complex

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

scalar p_{jk} and added to the summation of the Fourier modes. Our implementation is integrated with FFTW (Frigo & Johnson 2005) and stores the DFT matrix $\mathbb{C}^{n_{freq} \times n_{freq}}$ during the initialisation step of the CFD solver.

6.3. *Extracting nonlinear terms*

Extracting the nonlinear terms ($\boldsymbol{f}_{nl}$ in (5.8)) of Navier–Stokes equations is useful to investigate the nonlinear interactions. The nonlinear terms are extracted within the application developed for the resolvent-based estimation and control tool. The principle is described here.

The nonlinear Navier–Stokes operator $\mathcal{F}$ can be expressed as

$$\mathcal{F}(\boldsymbol{q}) = \mathcal{F}(\bar{\boldsymbol{q}}) + \frac{\partial \mathcal{F}(\bar{\boldsymbol{q}})}{\partial \boldsymbol{q}} \boldsymbol{q}' + nl(\boldsymbol{q}'), \tag{6.3}$$

where $nl(\boldsymbol{q}')$ represents all remaining nonlinear terms after linearisation. The forcing vector that accounts for the nonlinear terms can be derived as follows:

$$\boldsymbol{f}_{nl}(\boldsymbol{q}') = \mathcal{F}(\bar{\boldsymbol{q}}) + nl(\boldsymbol{q}') = \underbrace{\mathcal{F}(\boldsymbol{q})}_{\text{from DNS}} - \underbrace{\boldsymbol{A}\boldsymbol{q}'}_{\text{from linear run}}. \tag{6.4}$$

We run the nonlinear simulation (DNS or LES) in time and, within the same loop, compute the term $\boldsymbol{A}\boldsymbol{q}'$ to subtract from $\mathcal{F}(\boldsymbol{q})$. The resulting nonlinear term $\boldsymbol{f}_{nl}$ is saved in the solver. Computing the CSDs of the nonlinear term requires significant memory due to the large n_{nl}. Therefore, to efficiently compute $\hat{\boldsymbol{F}}$, we use a streaming Fourier transform in (6.2), which does not require to save all the snapshot $\boldsymbol{f}_{nl}$.

7. Resolvent-based estimation results

In this section, we use the resolvent-based estimation framework to estimate velocity fluctuations in the wake of the airfoil. We expect this framework to be well suited for this task since the flow is globally stable and resolvent modes capture the vortex shedding, as demonstrated in figures 4 and 5, respectively.

Before considering the actual nonlinear flow of interest, we first evaluate the performance of the resolvent-based estimator applied to the linear system (excited by upstream disturbances) on which the estimator is based. We then turn our attention to the true, nonlinear system with clean and noisy free stream conditions. As discussed previously, the noisy free stream prevents the flow from falling into a periodic limit cycle, resulting in chaotic fluctuations in the wake.

The measurements $\boldsymbol{y}$ used by the estimator correspond to one or more shear stress sensors on the surface of the airfoil, extracted from the state $\boldsymbol{q}$ by an appropriately defined measurement matrix $\boldsymbol{C}_y$. To mimic the effect of sensors of finite size and to facilitate convergence on a finite grid, the sensors and targets have Gaussian spatial support of the form

$$\alpha e^{-(x-x_c)^2/2\sigma_x^2-(y-y_c)^2/2\sigma_y^2}, \tag{7.1}$$

where σ_x and σ_y set the width of the Gaussian function and the constant α is set so that the kernel integrates to one. Consistent with other recent applications of the resolvent-based framework (Amaral *et al.* 2021; Towne *et al.* 2024), initial tests suggested little difference between sensing shear stress and pressure, the other candidate quantity that could be measured on the airfoil surface.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

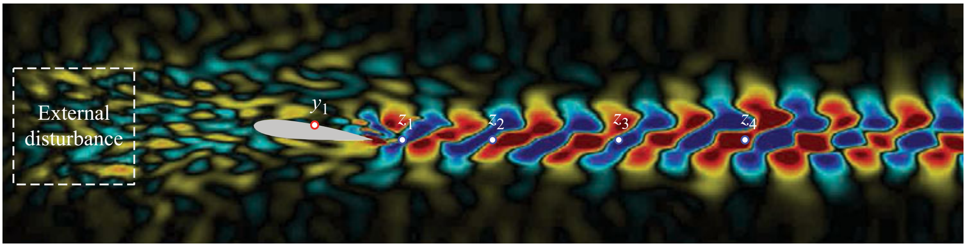


Figure 9. Estimation set-up for the linear system, showing the locations of the upstream forcing (white box), sensors (red circle) and targets (blue circle). The contours show an instantaneous snapshot of the streamwise velocity fluctuation.

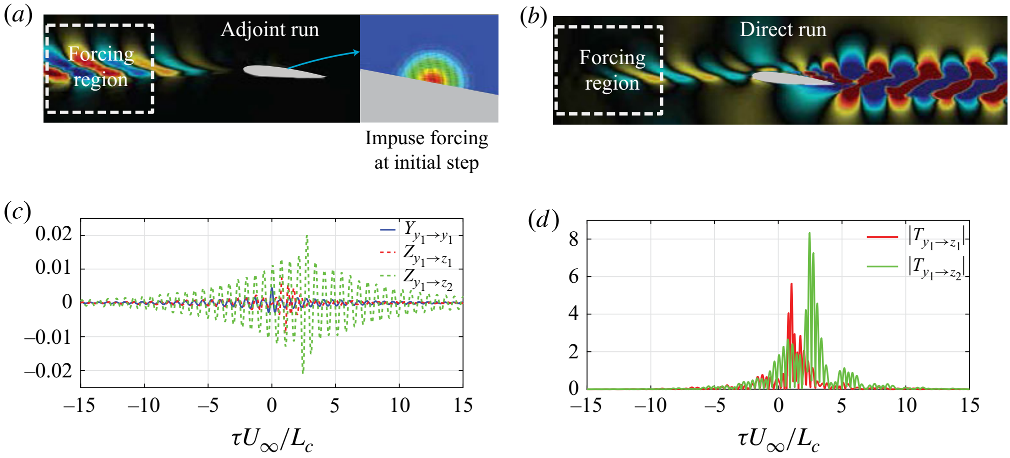


Figure 10. Operator-based estimation approach: (*a*) snapshot of the adjoint run at a specific time instant, including a zoom-in of the impulsive forcing applied at time zero; (*b*) snapshot of the direct run forced by the $\boldsymbol{B}_f$ readings from the adjoint run; (*c*) sensor and target readings y_1 $[x/L_c = 0.5]$, z_1 $[x/L_c = 1.2]$ and z_2 $[x/L_c = 2.0]$ from the direct run in panel (*b*); (*d*) non-causal estimation kernels constructed using the readings from panel (*c*).

7.1. *Linear system*

Figure 9 shows an instantaneous snapshot of the streamwise velocity fluctuation for the linear system driven by an external disturbance in the upstream region. The dominant wake mode, closely resembling the least stable eigenmode and the optimal resolvent response mode, is clearly visible.

7.1.1. *Building estimation kernels with white-noise forcing*

We begin by demonstrating how to construct the estimation kernel using the operator-based approach described in § 5.1. A sensor (y_1) is placed on the suction surface of the airfoil at $x/L_c = 0.5$, and four targets (z_1, z_2, z_3, z_4) are positioned in the wake aligned with the trailing edge at $x/L_c = [1.2, 2.0, 3.0, 4.0]$ and $y/L_c = -0.11$, as illustrated in figure 9. In figure 10, we show an example of the two-stage run used to build the operator-based kernel between sensor y_1 and targets z_1 or z_2. An impulsive forcing is applied to the adjoint system at the sensor location at time zero, as shown in figure 10(*a*). This forcing has Gaussian temporal and spatial support, with $\sigma_t = 12.5$ and $\sigma_x = \sigma_y = 0.02$, and the sensors

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

and targets have the same spatial support. The input matrix $\boldsymbol{B}_f$ is defined such that the external forcing $\boldsymbol{f}_{ext}$ is defined only in the prescribed upstream region. The forcing CSD matrix is assumed to be white noise, i.e. $\boldsymbol{F}_{ext} = \boldsymbol{I}$, implying that the forcing is uncorrelated in space and time. The sensor noise CSD is $\hat{\boldsymbol{N}} = \epsilon I$, with ϵ equal to 0.1 of the maximum value of $\hat{\boldsymbol{Y}}$. While sensor noise amplitude can affect the smoothness of the estimated data, it does not significantly impact its amplitude or phase.

The readings $\boldsymbol{s}_i$ from the adjoint run are then used to force the direct run, leading to the response shown in figure 10(*b*). The corresponding sensor and target measurements are recorded in $\boldsymbol{Y}$ and $\boldsymbol{Z}$, as defined in (5.5) and shown in figure 10(*c*). $\boldsymbol{Y}$ is symmetric about $\tau U_\infty / L_c = 0$ as it is autocorrelation. The perturbation grows as it travels downstream, so $Z_{y_1 \rightarrow z_2}$ is generally greater than $Z_{y_1 \rightarrow z_1}$. To achieve convergence in the adjoint and direct runs, we simulate the two-stage run over a time span $tU_\infty / L_c \in [-36, 36]$ with a time step twice that of the DNS.

The Fourier-transformed data, $\hat{\boldsymbol{Y}}$ and $\hat{\boldsymbol{Z}}$, are equivalent to $\boldsymbol{R}_{yf} \boldsymbol{R}_{yf}^{\dagger}$ and $\boldsymbol{R}_{zf} \boldsymbol{R}_{yf}^{\dagger}$, as shown in (5.5). Finally, the Fourier-transformed data are used to form the resolvent-based kernels using (5.6). Figure 10(*d*) illustrates the estimation kernel $\boldsymbol{T}_{nc}$ in (4.8) computed from $\boldsymbol{Y}$ and $\boldsymbol{Z}$. The peaks of $T_{y_1 \rightarrow z_1}$ and $T_{y_1 \rightarrow z_2}$ can be explained as the travel time from the sensor location (where the impulse forcing is imposed) to the target. A second dominant peak is observed for $T_{y_1 \rightarrow z_1}$ and $T_{y_1 \rightarrow z_2}$ on the left side of the main peaks, which may be the result of acoustic waves, which are faster than the hydrodynamic waves. For the estimation kernels, $\tau U_\infty / L_c > 0$ and $\tau U_\infty / L_c < 0$ represent past and future times, respectively. Both kernels shown in figure 10(*d*) have non-zero amplitude mainly for $\tau U_\infty / L_c > 0$, i.e. they are nearly causal. If a significant non-causal part is present, truncating it will degrade the performance of the estimator; optimality, under the constraint of causality, can be restored using the Wiener–Hopf decomposition. This impact is more significant for the nonlinear system, so we will discuss it in greater detail in that section.

7.1.2. *Estimation results for the linear system*

We present the causal resolvent-based estimation result only using a single sensor, comparing the true streamwise velocity fluctuation u'_x with the estimated value over time. Additionally, we estimate the fluctuations of both the streamwise u'_x and cross-streamwise u'_y velocity components in an extended region of the targets using a small number of sensors. To quantify the accuracy of the estimates, we calculate the estimation error

$$E = \frac{\Sigma_i \int (\tilde{z}_i(t) - z_i(t))^2 \, \mathrm{d}t}{\Sigma_i \int (z_i(t))^2 \, \mathrm{d}t}, \tag{7.2}$$

where $\tilde{z}_i$ and z_i represent the estimated and true values for the ith target, respectively. In computing the estimation error, we assume the system is ergodic, allowing the ensemble average to be replaced by the time average.

Since the estimator was designed for this linear system, the causal estimates are theoretically optimal. Figure 11 shows examples of the true and estimated target readings as a function of time. The result indicates that the target near the trailing edge, positioned in a more complex flow region, is estimated less accurately. In contrast, the other three targets (z_2, z_3 and z_4) show better estimation with an error of 0.07–0.08. Overall, the frequency and amplitude of the fluctuations are well estimated.

Next, we explore the impact of the sensor location on the estimation accuracy. Figure 12 shows the estimation error as a function of the target location x/L_c (aligned with the trailing edge) for six different sensor locations on the airfoil surface. For targets near the trailing edge ($x/L_c < 1.5$), the front sensors on the suction side (y_1) and pressure side

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

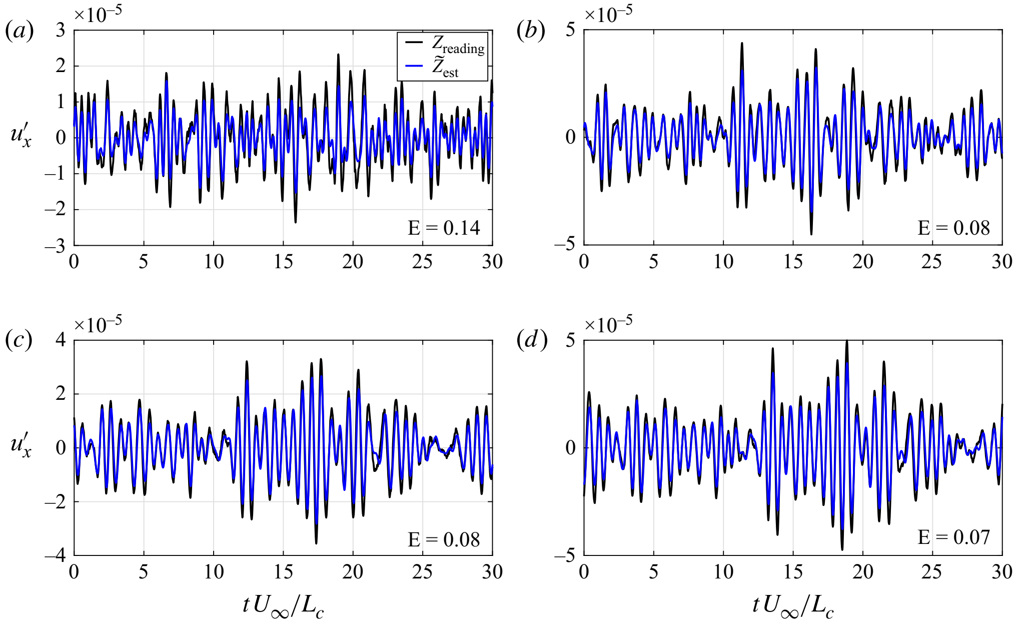


Figure 11. Causal estimation using an operator-based approach for the linear system at the targets: (*a*) z_1; (*b*) z_2; (*c*) z_3; and (*d*) z_4 at positions $[x/L_c = 1.2, 2.0, 3.0, 4.0]$, as shown in figure 9. The estimation error (7.2) is reported for each case.

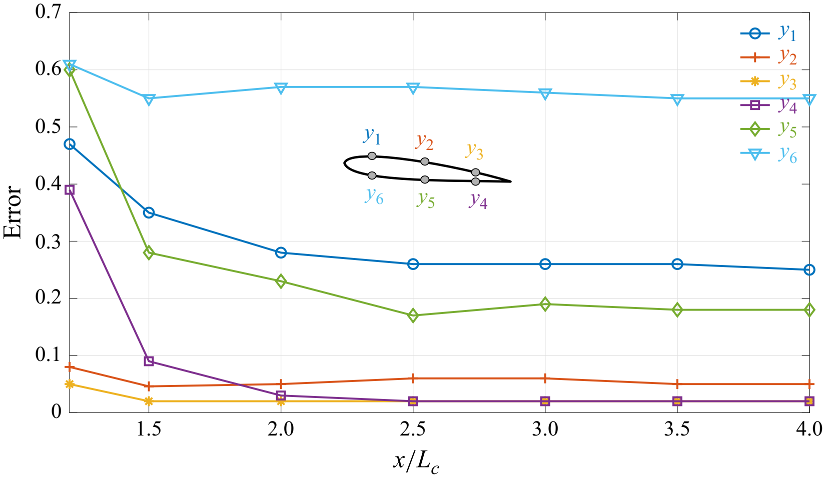


Figure 12. Estimation error for the linear system as a function of the target location x/L_c and $y/L_c = -0.11$.

(y_6) produce inaccurate estimates. In contrast, the rear sensors (y_3 and y_4) and the middle sensor y_2, which is the location used to demonstrate building the kernels in the previous section, result in better estimation accuracy. The suction-side sensors y_2 and y_3 accurately capture the flow dynamics near the trailing edge generated by the separation bubble over the airfoil. While y_4 faces challenges in capturing flow information from the bottom of the airfoil, it can still estimate the downstream targets ($x/L_c > 2$) as effectively as the sensors y_2 and y_3. These observations are consistent with the leading resolvent (response) mode

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

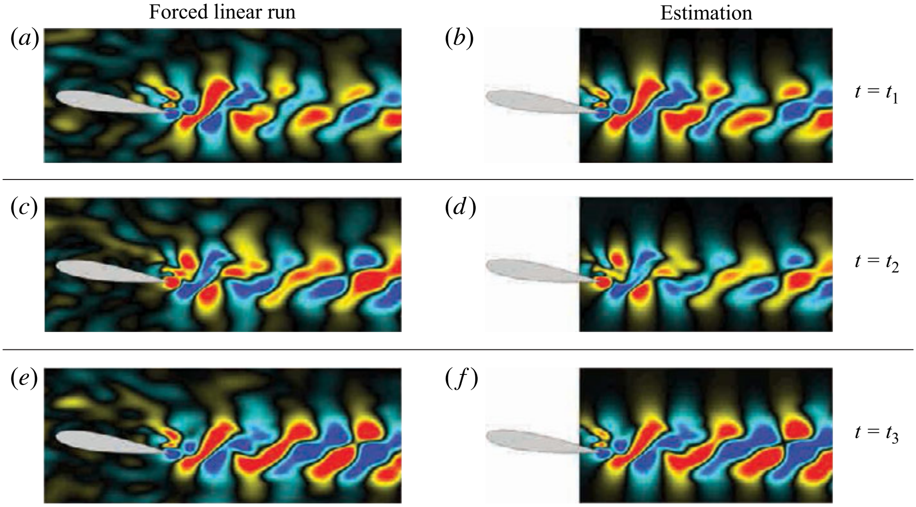


Figure 13. Estimation of streamwise velocity fluctuation for the extended target region at three different time steps for the linear system using the sensor y_3, as shown in figure 12.

shown in figure 5; qualitatively, lower errors are observed for sensors located in regions where the mode has a meaningful footprint.

Finally, we estimate the state in an extended region of the flow rather than at individual, discrete targets, allowing us to evaluate the estimation accuracy in different regions. The estimation kernel in this context takes the form

$$\begin{bmatrix} \hat{z}_1 \\ \hat{z}_2 \\ \vdots \\ \hat{z}_{n_z} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{T}}_{z_1 y_1} & \hat{\boldsymbol{T}}_{z_1 y_2} & \cdots & \hat{\boldsymbol{T}}_{z_1 n_y} \\ \hat{\boldsymbol{T}}_{z_2 y_1} & \hat{\boldsymbol{T}}_{z_2 y_2} & \cdots & \hat{\boldsymbol{T}}_{z_2 n_y} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{T}}_{z_{n_z} y_1} & \hat{\boldsymbol{T}}_{z_{n_z} y_2} & \cdots & \hat{\boldsymbol{T}}_{z_{n_z} y_{n_y}} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{y}}_1 \\ \hat{\boldsymbol{y}}_2 \\ \vdots \\ \hat{\boldsymbol{y}}_{n_y} \end{bmatrix}, \tag{7.3}$$

where n_y and n_z represent the number of sensors and targets, respectively. Based on the optimal results from figure 12, we use just one sensor located on the suction side, y_3. Figure 13 shows snapshots of the estimates in the extended target regions. The three time steps t_1, t_2, t_3 are selected to represent different phases of the vortex shedding. Figure 14 shows the estimation error as a function of position within the extended target region. As expected, the error increases as the target moves downstream or laterally away from the wake.

7.2. *Nonlinear system*

In the previous section, we confirmed that the resolvent-based kernels, derived from the resolvent operator, provide accurate estimates for the linear system. We now shift our focus to the actual (nonlinear) system, where it is crucial to statistically account for the nonlinear terms using coloured forcing. We do so by using the data-driven approach described in § 5.2, which is equivalent to the operator-based method with the appropriate forcing CSD $\hat{\boldsymbol{F}}$. As discussed earlier, we consider both clean and noisy free stream conditions for the nonlinear system, as illustrated in figures 15(*a*) and 15(*b*). The flow is simulated using

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

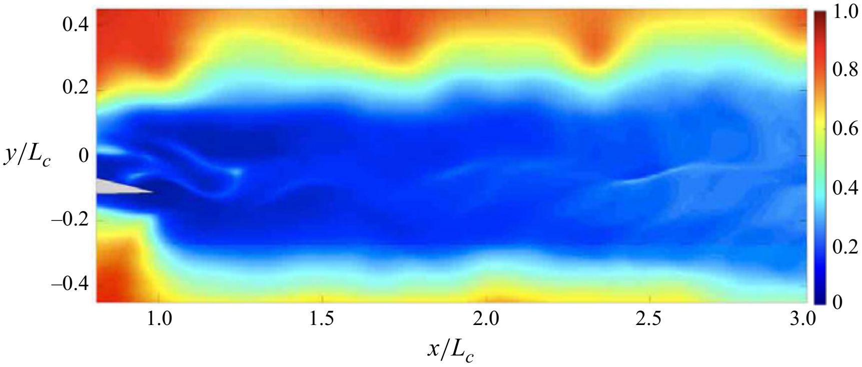


Figure 14. Estimation error in the extended target regions for the linear system using the sensor y_3.

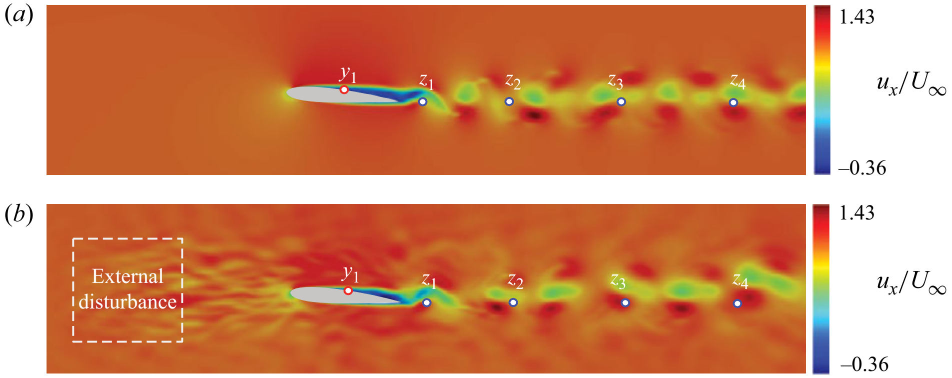


Figure 15. Instantaneous snapshot of the streamwise velocity u_x for (*a*) the clean and (*b*) the noisy DNS cases. The symbols show the sensor and target locations. The noisy free stream is generated by a random forcing within the region $x/L_c \in [-2, -1]$ and $y/L_c \in [-0.5, 0.5]$.

DNS with the same numerical set-up described in § 2. Since the estimator is defined in terms of perturbations to the mean, the mean is removed from the sensor readings before convolution with the estimation kernels.

7.2.1. *Nonlinear response to the external forcing*

The nonlinear system subject to external forcing can be expressed as

$$\frac{\partial \boldsymbol{q}}{\partial t} = \mathcal{F}(\boldsymbol{q}) + \boldsymbol{B}_{f,ext} \boldsymbol{f}_{ext}. \tag{7.4}$$

We choose the external forcing $\boldsymbol{f}_{ext}(x, t)$ to be white noise in space and time, generated using random vectors with each entry uniformly distributed over $[-1, 1] \times W$, where W controls the variance of the random vector, adjusting the noise level of the free stream. The CSD matrix for this forcing is $\hat{\boldsymbol{F}}_{ext} = W^2 \boldsymbol{I}$. In contrast, the linear system models the nonlinear terms as white noise (an identity matrix; McKeon & Sharma 2010).

To help us select an appropriate forcing amplitude W, we analyse the PSD of the state for varying noise levels, focusing on the impact on the vortex shedding. The snapshots in figure 16 demonstrate that while vortex shedding persists at all noise levels, the spatial

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

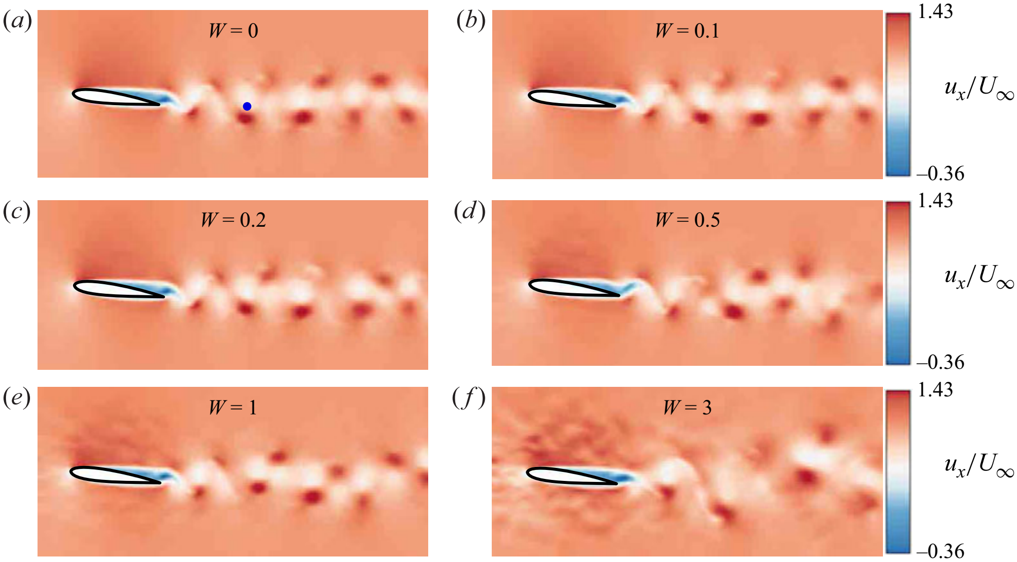


Figure 16. Instantaneous snapshots of the streamwise velocity u_x for varying free stream noise intensities, as determined by the forcing amplitude W: (*a*) $W = 0$ (clean); (*b*) $W = 0.1$; (*c*) $W = 0.2$; (*d*) $W = 0.5$; (*e*) $W = 1$; and (*f*) $W = 3$. The blue dot in panel (*a*) indicates the location for which the PSD is analysed in figure 17.

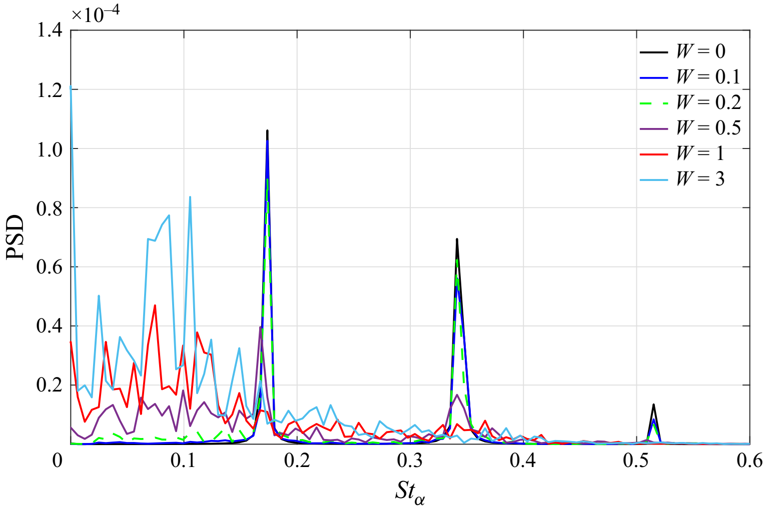


Figure 17. Power spectral density of the streamwise velocity u_x for the nonlinear system in terms of the noise level W at the point $[x/L_c, y/L_c] = [2.11, -0.11]$ in figure 16.

periodic pattern in the streamwise direction is disrupted. Figure 17 makes this point quantitative by showing the PSD of the state at the location indicated in figure 16(*a*). The results confirm that the vortex-shedding frequencies ($St_\alpha \approx 0.17 \times n$ with $n = 1, 2, 3$) are suppressed for $W \geqslant 1$, indicating a strong nonlinear modification to the linear dynamics. Further insights are provided in figure 18, which compares the mean streamwise velocity

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

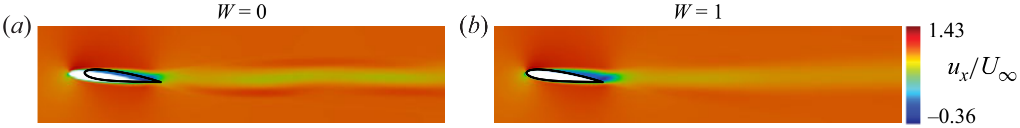


Figure 18. Mean streamwise velocity $\bar{u}_x$ fields for (*a*) the clean ($W = 0$) and (*b*) the noisy free stream ($W = 1$) cases.

$\bar{u}_x$ for the clean ($W = 0$) and noisy ($W = 1$) free stream cases. The noticeable modification to the mean flow further highlights the nonlinearity induced by the noisy free stream; in particular, the increased randomness of the vortex shedding eliminates the spatial oscillation in the clean mean flow. Based on these observations, we select $W = 1$ as the noise level for our estimation and control studies, using the modified mean flow depicted in figure 18(*b*) under the influence of the noisy free stream.

7.2.2. *Building estimation kernels with coloured-forcing statistics*

In § 7.1.1, we assumed the forcing CSD matrix $\hat{\boldsymbol{F}}_{nl}$ to be white noise (an identity matrix), resulting in kernels equivalent to a Kalman filter. To enhance the accuracy of the estimation kernels, the resolvent-based framework enables us to incorporate coloured forcing statistics via a non-identity $\hat{\boldsymbol{F}}_{nl}$, which can be obtained in § 6.3. Once the nonlinear terms $\hat{\boldsymbol{f}}_{nl}$, such as those shown in figure 36 in Appendix C, are available, we compute $\boldsymbol{R}_{yf}\hat{\boldsymbol{F}}_{nl}\hat{\boldsymbol{R}}_{yf}^{\dagger}$ and $\boldsymbol{R}_{zf}\hat{\boldsymbol{F}}_{nl}\hat{\boldsymbol{R}}_{yf}^{\dagger}$ during the two-stage run outlined in § 5.1.2.

Alternately, we can implement the data-driven approach (Martini *et al.* 2022) to obtain estimation kernels that automatically include the influence of the coloured-forcing statistics (Zare *et al.* 2017; Towne *et al.* 2020; Martini *et al.* 2022). The necessary sensor and target data are directly collected from DNS, and Welch's method (Welch 1967) is employed to obtain the CSD tensors required to construct the estimation kernels. The time series are divided into 64 blocks of length $tU_\infty/L_c = 5$, with 50 % overlap. All other parameters for building the kernels and Wiener–Hopf factorisation are consistent with those used in the operator-based approach.

The estimation kernels that account for coloured nonlinear forcing are shown in figure 19 for the same sensor and target configuration as was used for the white noise case, i.e. the sensor y_1 $[x/L_c = 0.5]$ and the targets z_1, z_2, z_3 and z_4 $[x/L_c = 1.2, 2.0, 3.0, 4.0]$, as shown in figure 15. The figure shows results for the noisy free stream case. As discussed in § 7.2.1 and shown in figure 36 in Appendix C, there is a higher degree of nonlinear interaction near the trailing edge. In this region, the estimation kernel with coloured forcing peaks at $\tau U_\infty/L_c < 0$, i.e. the kernel is highly non-causal. This can be alleviated by using the Wiener–Hopf method to optimally enforce causality. Further downstream in the wake in figures 19(*b*), 19(*c*) and 19(*d*), the kernels exhibit simpler behaviour and distinct peaks, which indicate the travel time of the fluctuations. The difference between the causal and non-causal kernels is small when the target is set further downstream in the wake, where the convective nature of the flow makes the kernels naturally (almost) causal. This trend is similar to the backward-facing step flow reported by Martini *et al.* (2022).

7.2.3. *Single-sensor estimation results for the nonlinear system*

We assess the estimation accuracy for the nonlinear system for a single sensor as a function of its location on the surface of the airfoil. Figures 20(*b*) (clean free stream) and 20(*c*) (noisy free stream) show the averaged estimation errors for four sets of targets (A, B, C and D) shown in figure 20(*a*). The errors are approximately an order of magnitude lower

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

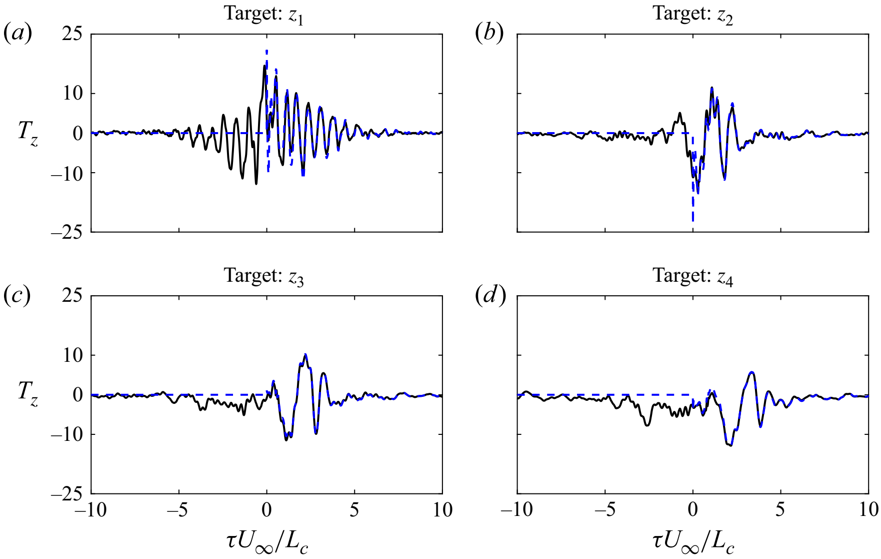


Figure 19. Estimation kernels with a coloured forcing statistics: non-causal (black solid line) and causal (blue dashed line) kernels between (*a*) y_1 [$x/L_c = 0.5$] and z_1 [$x/L_c = 1.2$], (*b*) z_2 [$x/L_c = 2.0$], (*c*) z_3 [$x/L_c = 3.0$], and (*d*) z_4 [$x/L_c = 4.0$].

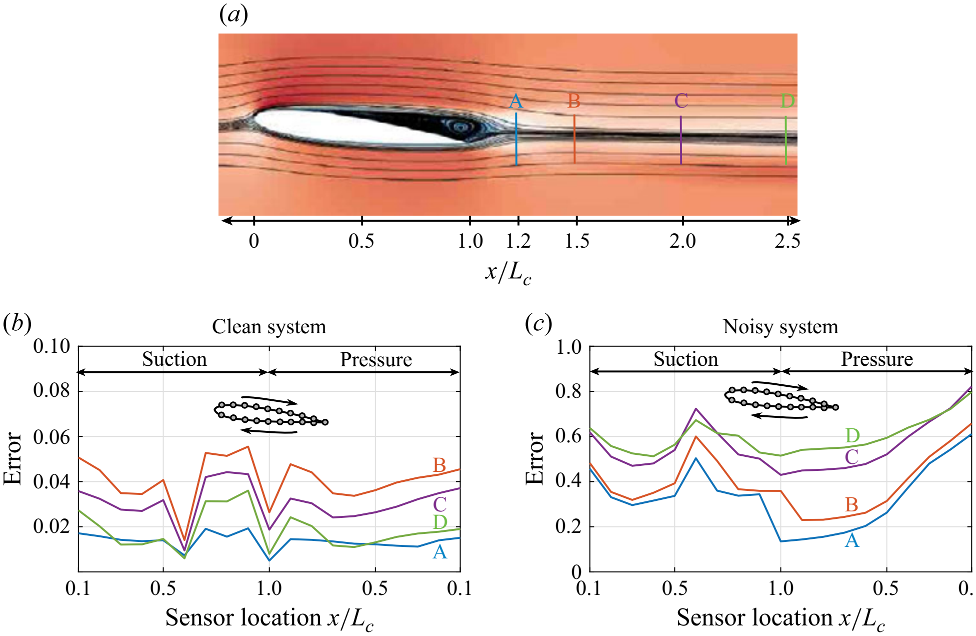


Figure 20. Streamline of the flow is shown along with four sets of targets: A ($x/L_c = 1.2$), B ($x/L_c = 1.5$), C ($x/L_c = 2.0$), and D ($x/L_c = 2.5$). Averaged estimation errors are reported for these four lines (A, B, C and D), which are based on sensor locations on the airfoil surfaces. Panel (*b*) illustrates the clean system, while panel (*c*) depicts the noisy system.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

for the clear free stream than for the noisy free stream. Imposing external forcing leads to higher-energy fluctuations in the wake, as illustrated in figure 35(*c*), which increases the impact of nonlinearity and deteriorates the estimation accuracy as the target is moved downstream (estimation error: $D > C > B > A$ in figure 20*b*). In contrast, the error for the clean free stream is non-monotonic with downstream distance: it is lowest for the nearest set of targets, increases and then decreases again. This latter decrease in error is likely due to the increasingly low-rank behaviour of the wake with increasing downstream distance for the clean inflow case.

The separation bubble impacts the sensors differently for the clean and noisy free stream cases. For the clean free stream results in figure 20(*b*), sensors positioned within the recirculation region $0.6 < x/L_c < 1$ (Marquet *et al.* 2022) show reduced accuracy. However, this effect is not as evident for the noisy free stream case in figure 20(*c*). This suggests that, whereas the separation bubble shields the sensors in the clean free stream case, incoming fluctuations from the noisy free stream instantaneously penetrate the bubble and later contribute to the wake dynamics, providing useful information to the sensors. Among the sensors on the suction surface for the noisy free stream in figure 20(*c*), those positioned before the separation bubble ($0.2 < x/L_c < 0.5$) and near the trailing edge ($0.8 < x/L_c < 1.0$) most effectively predict the wake dynamics. Notably, rear sensors on the pressure surface also show high estimation accuracy.

7.2.4. *Multi-sensor estimation results for the nonlinear system*

Next, we present estimation results for the nonlinear system with clean and noisy free stream conditions using multiple sensors. In Appendix D, we empirically evaluate six candidate sensor configurations guided by the single-sensor results from the previous section. Ultimately, we selected candidate 6, as defined in table 3. Since we are interested in vortex shedding, we report estimation results for both components of velocity that would be needed to compute vorticity.

Figures 21 and 22 present the causal resolvent estimation of u'_x (panels *a,c,e,g*) and u'_y (panels *b,d,f,h*) comparing with other methods and the true target reading (from DNS) for the clean and noisy free stream inflows at the target points ($x/L_c = 1.2, 2.0, 3.0, 4.0$) aligned with the trailing edge, shown in the top of each figure. The clean DNS system is well estimated using the causal resolvent-based approaches. The Kalman filter captures the dominant frequency's high-energy parts effectively, but lacks spatial and temporal detail due to treating the nonlinear terms as white noise. The target near trailing edge, where nonlinearity is strongest in figure 35 in Appendix C, is poorly estimated using the Kalman filter, shown in figure 22(*a*), while the poor estimates obtained using the TNC approach are due to the presence of substantial amplitude of the non-causal kernels in the non-causal part $\tau U_\infty L_c < 0$). However, the truncated non-causal kernels include the impact of the coloured forcing statistics, making them effective when the kernels are mostly naturally causal for the further downstream target locations, e.g. in the case of figures 22(*e*) and 22(*g*). Comparing left and right panels in figures 21 and 22, we observe that the cross-stream velocity is estimated more accurately than the streamwise velocity.

Figure 23 provides a more quantitative assessment of the estimation performance by showing the estimation error for each method as a function of the target position for both velocity components in the clean and noisy free stream cases. Notably, the causal resolvent-based approach estimates both velocity components in the clean system with high accuracy. In the noisy system, the estimation accuracy decreases as the distance between the sensor and the target increases downstream. The causal resolvent-based approach enhances estimation accuracy near the trailing edge, where the lack of causality

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

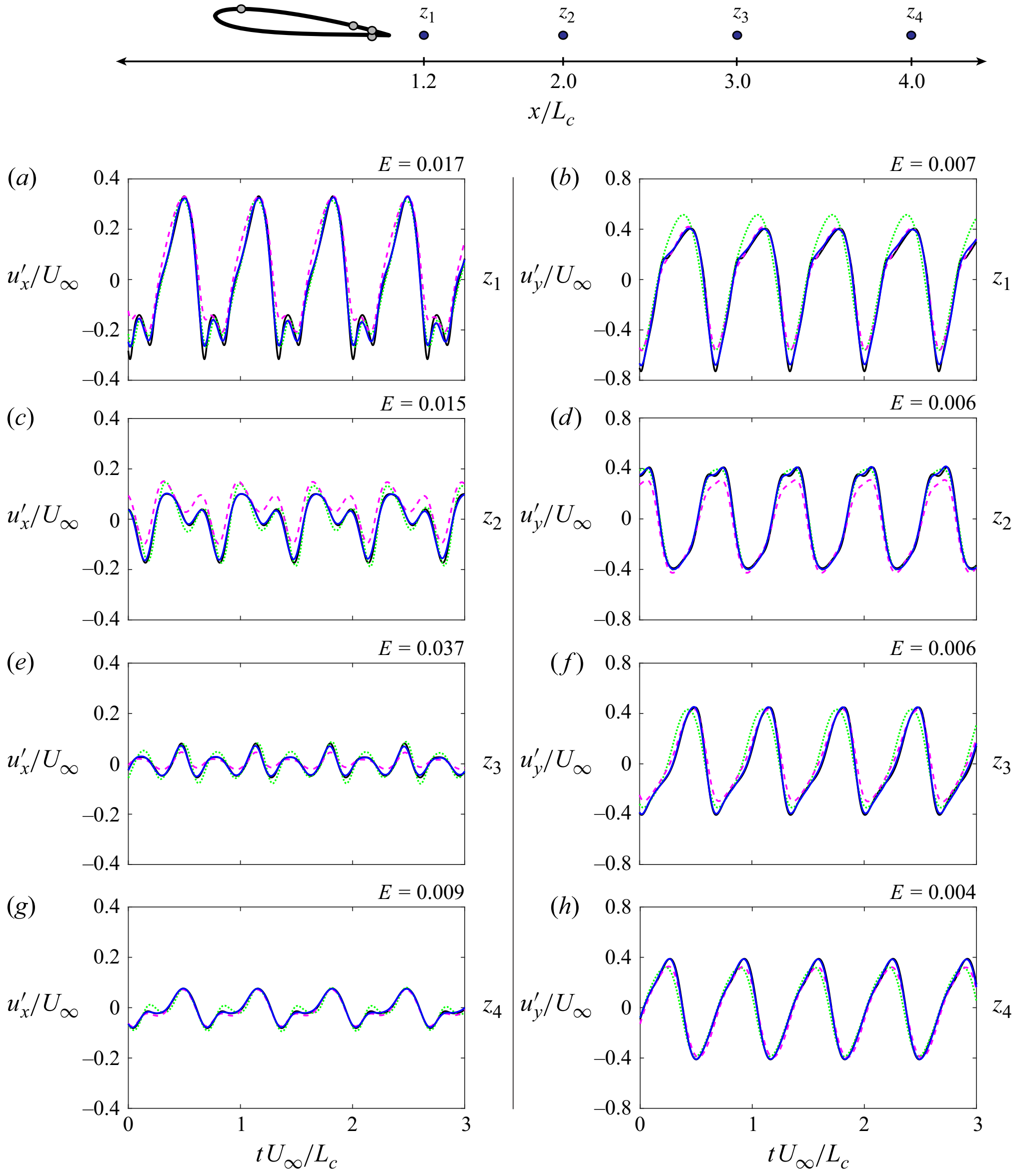


Figure 21. Estimation of (a,c,e,g) u'_x and (b,d,f,h) u'_y for the nonlinear system with a clean free stream for four targets. Lines: (black solid) true target from the DNS; (green dashed) Kalman filter estimates; (magenta dashed) truncated non-causal estimates; (blue solid) resolvent-based causal estimates. The target locations are: (a,b) $[z_1 = x/L_c,\ y/L_c] = [1.2, –0.11]$; ($c$,$d$) [2.0, –0.11]; ($e$,$f$) [3.0, –0.11]; and ($g$,$h$) [4.0, –0.11], as shown in the top figure and figure 15. The estimation errors for the resolvent-based method are noted in the top-right corner of each panel.

of the optimal non-causal estimator and the effects of nonlinearity are most significant. The causal resolvent-based estimator consistently outperforms the Kalman filter, highlighting the value of incorporating the space–time forcing statistics.

Using the same four shear stress sensors, we estimate the velocity fluctuations u'_x and u'_y for an extended region of the wake in both clean and noisy nonlinear systems.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

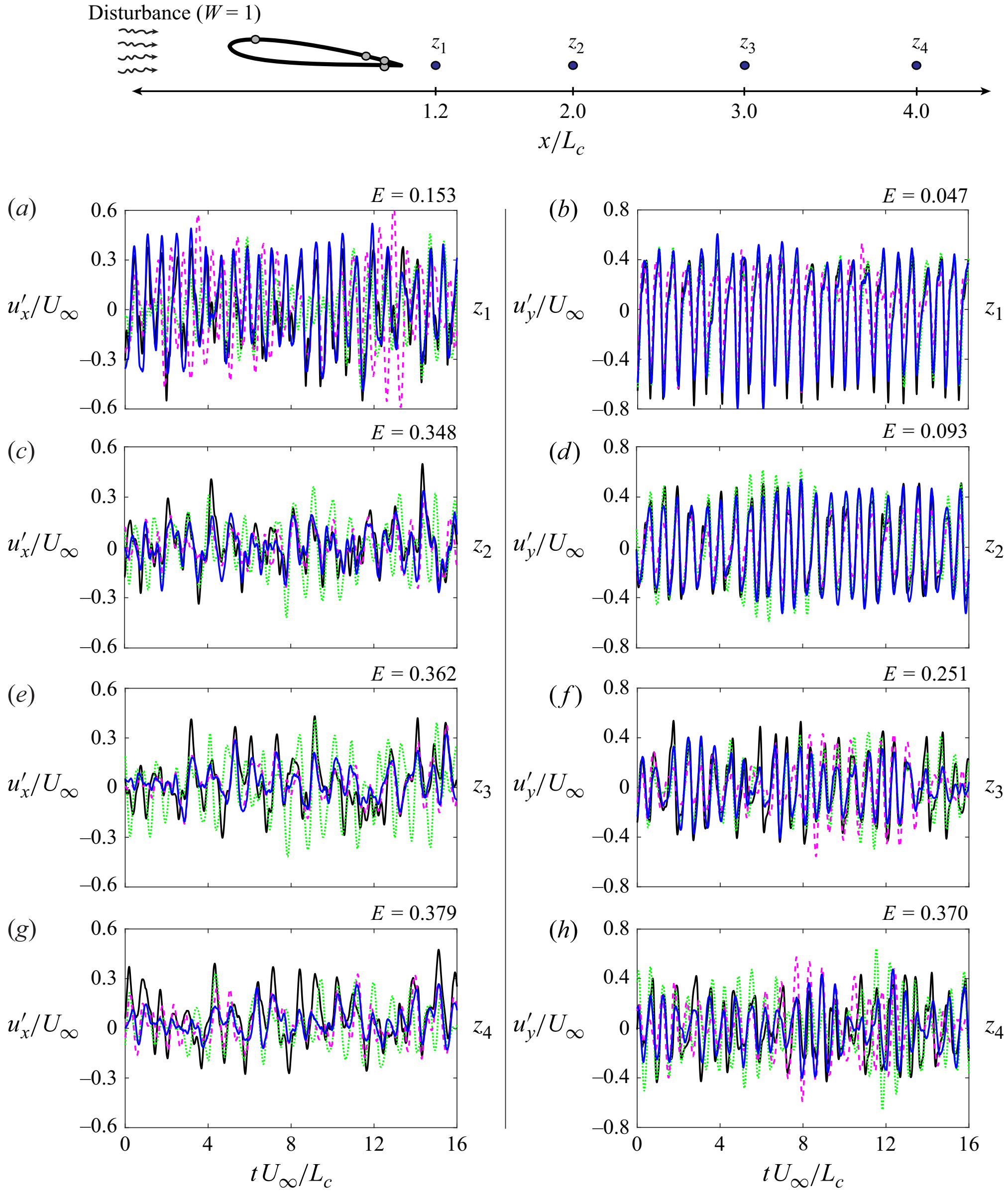


Figure 22. Estimation of (a,c,e,g) u'_x and (b,d,f,h) u'_y for the nonlinear system with the noisy free stream ($W = 1$). The black (solid) line shows the true DNS, while the other lines represent different methods: green (dashed) for Kalman filter; magenta (dashed) for truncated non-causal estimation; and blue (solid) for causal estimation (our method). The target locations are: (a,b) at [$z_1 = x/L_c$, y/L_c] = [1.2, –0.11]; (c,d) at [2.0, –0.11]; (e,f) at [3.0, –0.11]; and (g,h) at [4.0, –0.11], as shown in the top figure and figure 15. The estimation errors for the causal method are noted in the top right corner of each panel.

The sensors are positioned near the trailing edge, allowing us to estimate the region behind $x/L_c > 0.8$. We present two snapshots of the estimation, selected to represent different phases of the vortex shedding. For the clean system, our estimation results for u'_x, as shown in figure 24, are highly accurate. In the noisy nonlinear system, however, the estimation accuracy decreases due to perturbations that disrupt the vortex structure. Despite the

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

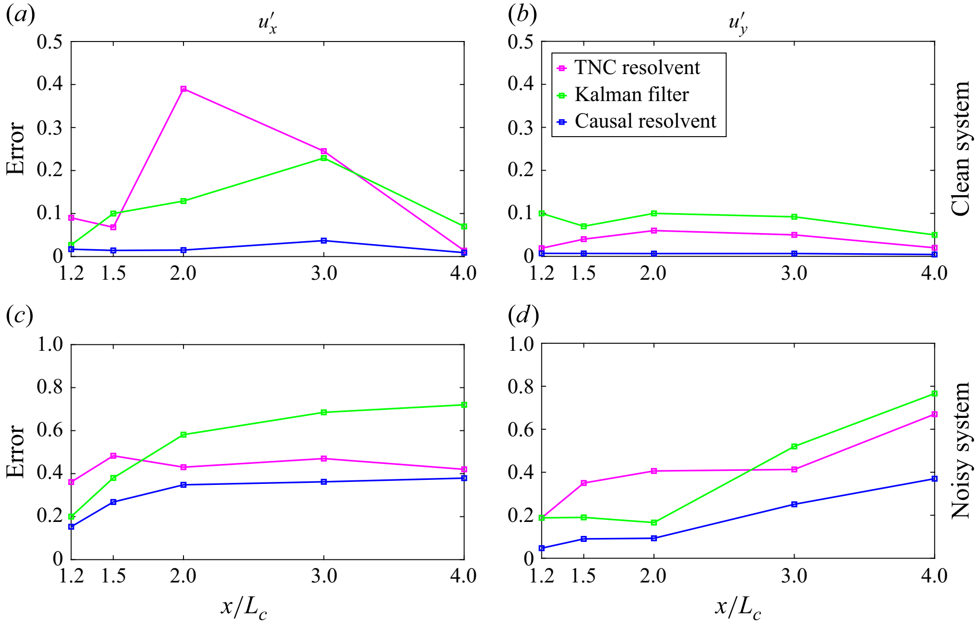


Figure 23. Estimation errors for nonlinear systems: (*a*) u'_x and (*b*) u'_y for clean free stream; (*c*) u'_x and (*d*) u'_y for noisy free stream. Blue lines represent causal resolvent-based estimation, while magenta and green lines denote truncated non-causal estimation using coloured forcing and a Kalman filter (white noise), respectively.

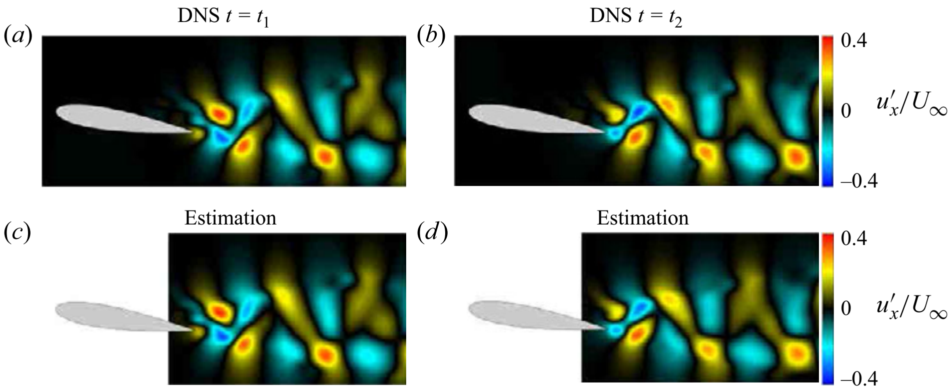


Figure 24. Estimation of the streamwise velocity fluctuation u'_x in an extended wake region for the nonlinear system with a clean free stream at two times: (*a,b*) DNS results, and (*c,d*) results from causal resolvent-based estimation.

challenges posed by chaotic fluctuations within the wake, the causal resolvent-based approach effectively estimates the wake flow, as demonstrated in figure 25.

8. Resolvent-based control results

In this section, we use the resolvent-based controller described in § 4.3 to suppress velocity fluctuations in the wake. Active flow control methods using blowing/suction, synthetic jets and plasma actuators have been successfully used to suppress laminar vortex shedding behind bluff bodies in both experimental (Ffowcs Williams & Zhao 1989;

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

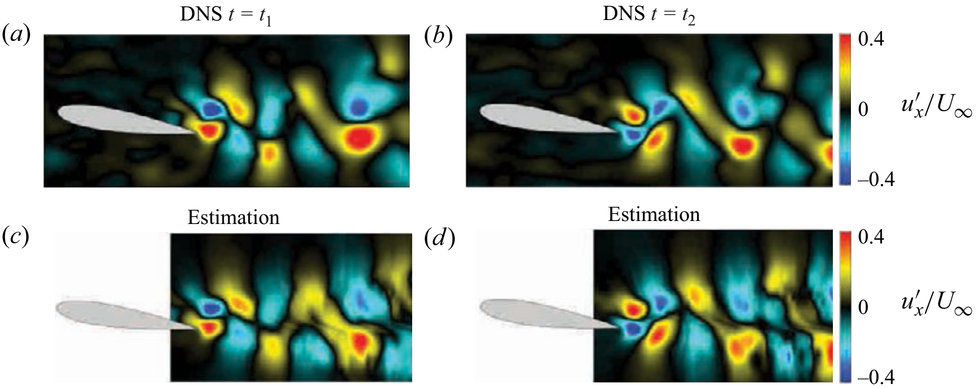


Figure 25. Estimation of the streamwise velocity fluctuation u'_x in an extended wake region for the nonlinear system with a noisy free stream at two times: (*a*,*b*) DNS results, and (*c*,*d*) results from causal resolvent-based estimation.

Strykowski & Sreenivasan 1990; Tao, Huang & Chan 1996; Min & Choi 1999; Fujisawa, Kawaji & Ikemoto 2001) and numerical studies (Roussopoulos & Monkewitz 1996; Lin & Tsai 2024). Suppressing vortex shedding significantly reduces lift and drag fluctuations, attracting considerable engineering interest. We seek to mitigate velocity fluctuations in the wake, which naturally includes the influence of vortex shedding.

Resolvent analysis has been fruitfully used to guide open-loop control efforts by suggesting effective forcing frequencies and actuator locations to which the flow is responsive (Yeh & Taira 2019; Gross, Marks & Sondergaard 2024) for open-loop control. Instead, we wish to design closed-loop controllers that react in real time to broadband frequency content within the flow. Using the estimated flow state from the resolvent-based estimator, implicitly included in the resolvent-based controller, the controller determines the actuation that most effectively cancels the target values, an approach also known as reactive control or the wave-canceling method (Sasaki *et al.* 2018*a*,*b*; Morra *et al.* 2020; Martini *et al.* 2022; Audiffred *et al.* 2023). As before, we first control the linear system before considering the actual nonlinear problem under noisy free stream conditions. We show that the optimal causal resolvent-based controller significantly outperforms a truncated non-causal controller, especially when using a pair of nested controllers to account for mean-flow modifications.

8.1. *Control set-up*

Our overall control architecture is shown schematically in figure 26. Some closed-loop controllers used to suppress wake fluctuations rely on impractically located sensors, e.g. behind the body in the wake itself. Our resolvent-based controller avoids this issue by implicitly using the resolvent-based estimator to predict the evolution of disturbances using shear-stress sensors on the airfoil surface. Our actuators mimic unsteady blowing and suction at discrete positions on the surface of the airfoil, modelled as compact source terms (again with Gaussian spatial support) in the momentum equations. As shown earlier in figure 16, the control signal is computed and applied online during the simulation.

In choosing the placement of our sensors and actuators, we account for the combined amplifier and oscillator characteristics of this flow (Schmid & Sipp 2016); the linear operator is globally stable, such that upstream disturbances are convectively amplified as they travel downstream, but it also contains a single lightly damped mode that generates oscillator-like vortex shedding. The sensor and actuator near the leading edge are designed

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

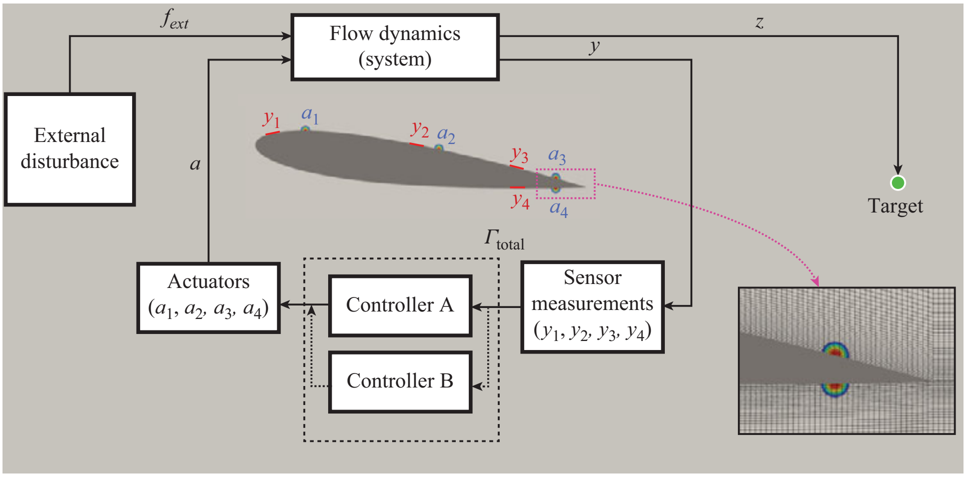


Figure 26. Control scheme for resolvent-based control of the flow around a NACA0012 airfoil. Red markers indicate shear-stress sensors, while contoured circles represent actuators in the form of momentum sources with Gaussian spatial support on the airfoil surface. The green circle marks the target location. For the control of the nonlinear system, we use a second nested controller (controller B) designed for the modified mean flow produced by the original controller (A). The insert highlights the grid resolution around the rear actuators.

to disrupt the separation bubble and suppress the global vortex-shedding behaviour (Broglia *et al.* 2018; Déda *et al.* 2023). The sensors and actuators near the trailing edge are designed to mitigate free stream fluctuations before they are amplified in the wake. We position the front sensors and actuators near the leading edge similarly to prior works (Colonius & Williams 2011; Broglia *et al.* 2018; Yeh & Taira 2019; Asztalos, Dawson & Williams 2021), while the positions of the rear sets are motivated by our estimation work.

For the nonlinear problem, the actuation modifies the mean flow via triadic interactions, even if the actuation itself is zero-mean. This presents a challenge since the controller is derived based on linearisation about the original (uncontrolled) mean flow and is designed to minimise fluctuations to that mean. However, our primary objective remains to minimise the flow unsteadiness at the targets, i.e. the velocity fluctuations about the modified (controlled) mean flow. To address this challenge, we use an iterative approach similar to that of Leclercq *et al.* (2019). After applying the controller based on the original (uncontrolled) mean flow (labelled as controller A in figure 26), we design a second controller (controller B) based on the new (controlled) mean flow and wrap it around the system that is still under the influence of controller A. The combined influence of the two controllers can be thought of as a single new nested controller that takes in the sensor measurements and drives the actuators. This iterative process can be repeated any number of times, but we achieve satisfactory results with just two nested controllers.

Following our previous work (Martini *et al.* 2022), we quantify the controller performance via the metric

$$\varepsilon_{con} = 1 - \frac{\sum_i \int (z_{i,con}(t))^2 \, \mathrm{d}t}{\sum_i \int (z_{i,uncon}(t))^2 \, \mathrm{d}t}, \tag{8.1}$$

which measures the reduction in fluctuation energy at the targets compared with the uncontrolled flow.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

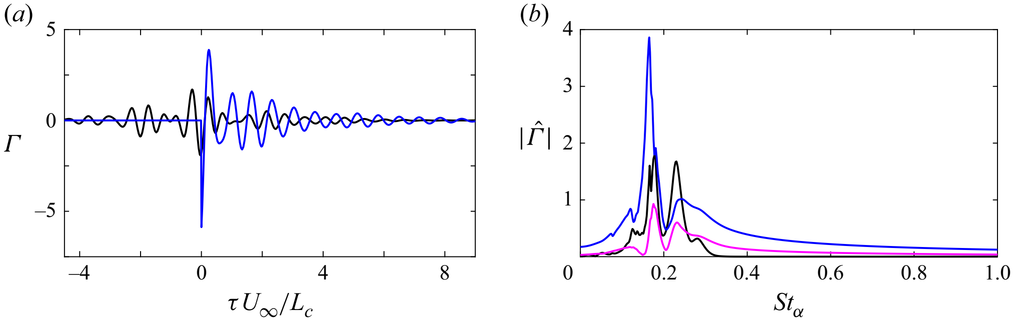


Figure 27. Control kernels with the sensor positioned near the trailing edge (y_3) and the target z located at $x/L_c = 1.5$ in the (*a*) time and (*b*) frequency domains. The black line represents the non-causal control kernel, the magenta line shows the truncated non-causal kernel, and the blue line depicts the causal control kernel computed using the Wiener–Hopf method.

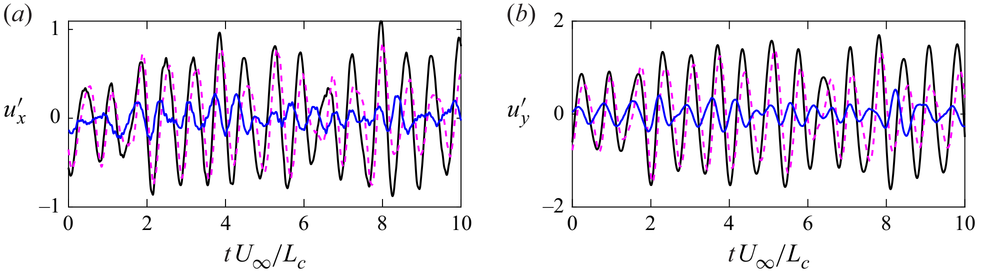


Figure 28. Time series of velocity fluctuations for the uncontrolled and controlled linear systems: (*a*) u'_x; (*b*) u'_y. Lines: uncontrolled (black line); truncated non-causal control (magenta dashed line); and causal control (blue line).

8.2. *Control of the linear system*

We first consider the linear system obtained by linearising the Navier–Stokes equations about the uncontrolled mean flow subject to external forcing, as in § 7.1. For the linear system, we use only controller A and just two actuators (a_3 and a_4). The resolvent-based control kernels are obtained using the operator-based approach described in § 5.1. The direct and adjoint runs are conducted over the interval $tU_\infty/L_c \in [-48, 48]$.

Figure 27 shows the non-causal and causal control kernels between the y_3 sensor and a_3 actuator for a target z at $x/L_c = 1.5$ in the time and frequency domains. In figure 27(*a*), due to the close proximity of the sensor and actuator, the non-causal control kernel contains relatively large values in the non-causal part $\tau U_\infty L_c < 0$. This issue is moderated using the Wiener–Hopf method, similar to the estimation kernels. For the causal kernel, the current measurement significantly impacts the actuation signal (Morra *et al.* 2020; Martini *et al.* 2022). Figure 27(*b*) presents the control kernels in the frequency domain. The truncated non-causal control kernel (magenta line) shows a considerable loss in magnitude, while the causal control kernel significantly amplifies the sensor measurement at the vortex-shedding frequency $St_\alpha = 0.17$. As expected, increasing the actuation cost $\boldsymbol{P}$ reduces the relative magnitude of the control kernel, leading to a smaller actuation force. We set $\boldsymbol{P} = \epsilon I$ with $\epsilon = 10^{-1}$ of the maximum value of $\boldsymbol{R}_{za}$.

Figure 28 presents the time-series data for the controlled and uncontrolled streamwise and cross-streamwise velocity fluctuations at the target. While the truncated non-causal

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

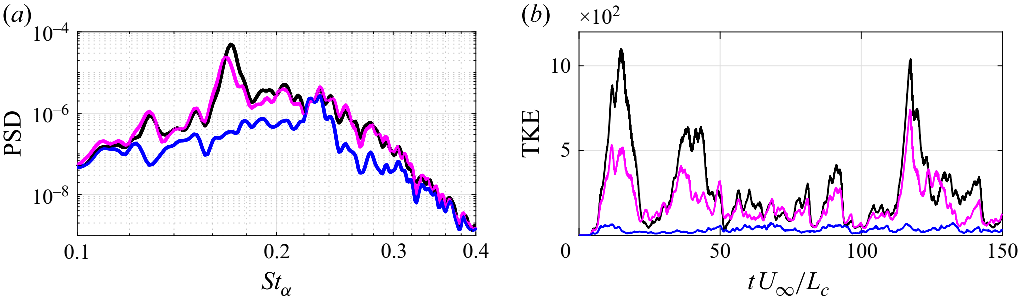


Figure 29. Control performance for the linear system: (*a*) PSD of the streamwise velocity fluctuations u_x' for the controlled (blue) and uncontrolled (black) cases, with the magenta line representing the truncated non-causal control approach; (*b*) turbulent kinetic energy (TKE) integrated over the wake region.

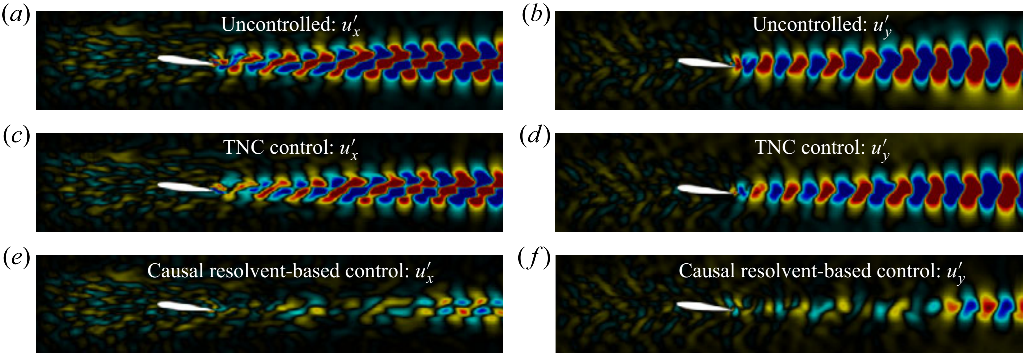


Figure 30. Snapshots of the streamwise and cross-streamwise velocity fluctuation fields for the linear system: (*a,b*) uncontrolled; (*c,d*) truncated non-causal (TNC) control; (*e,f*) causal resolvent-based control.

controller achieves modest improvements, the causal resolvent-based control significantly reduces both velocity components. The power spectral density (PSD) of the uncontrolled and controlled streamwise velocity is shown in figure 29(*a*). The causal resolvent-based controller effectively suppresses the dominant vortex-shedding frequency, while the truncated non-causal controller fails to accomplish this. Using the causal approach, the two actuators at the trailing edge reduce the turbulent kinetic energy of the velocity fluctuations at the target, as measured by (8.1), by 85 %. In contrast, the truncated non-causal approach achieves only a 27 % reduction. In terms of root-mean-square (r.m.s.) velocities, the causal controller achieves a 62 % decrease, compared with approximately 14 % for the truncated non-causal controller.

While the controller is designed to minimise the velocity fluctuations at the target, it also does so over an extended region of the wake. Figure 30 presents instantaneous snapshots of u_x' and u_y' for the controlled and uncontrolled systems. The wake modes excited by the upstream-generated external disturbance are effectively suppressed by the actuation at the trailing edge, as illustrated in figures 30(*e*) and 30(*f*). To make this reduction quantitative, the turbulent kinetic energy (TKE) integrated over the wake region ($x/L_c = [1.1, 5]$, $y/L_c = [-1, 1]$) is shown as a function of time in figure 29(*b*).

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

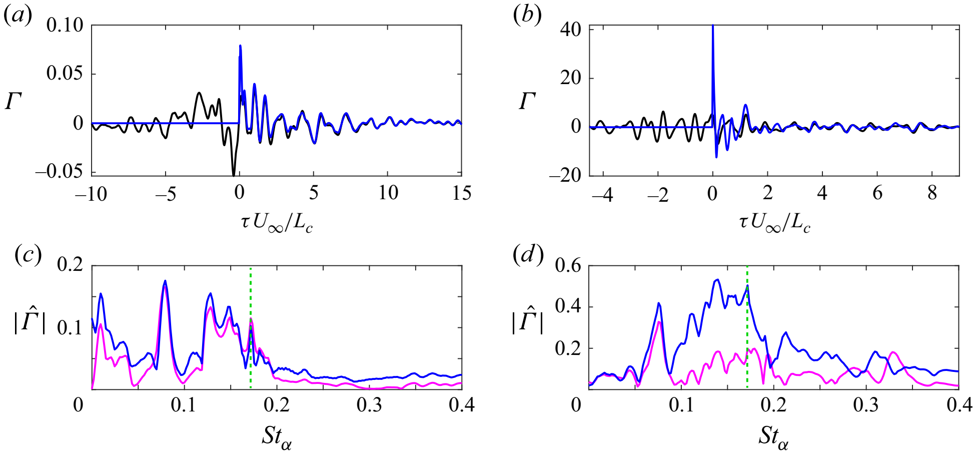


Figure 31. Control kernels for the nonlinear system: (*a,b*) kernels in the time domain; (*c,d*) kernels in the frequency domain. Specifically, panels (*a*) and (*c*) correspond to y_3 and a_3, and panels (*b*) and (*d*) correspond to y_4 and a_4, as shown in figure 26. The green dashed line in panels (*c*) and (*d*) indicates the vortex-shedding frequency.

8.3. *Control of the nonlinear system*

The ultimate goal of this work is to reduce the wake fluctuations in the nonlinear system (DNS) using the optimal linear resolvent-based controller. As discussed earlier, we use a second controller (controller B) to account for the impact of the first controller on the mean flow. After turning on the first controller, we wait until the flow achieves a new statistical steady state before turning on the second controller. The actuator outputs are determined by resolvent-based control kernels, with an additional constant forcing applied to the two front actuators to destroy the separation bubble. The steady forcing is not derived from the resolvent model, but is introduced to improve flow receptivity and enhance control effectiveness (Radespiel *et al.* 2016; Eggert & Rumsey 2017; Puri *et al.* 2018). The data-driven approach is used to build the resolvent-based kernels to account for the coloured nonlinear forcing. The operator approach is used to compute $\boldsymbol{R}_{ya}$ and $\boldsymbol{R}_{za}$, required in (5.19). Alternatively, these terms can be obtained using the data-driven approach by running a series of impulse response simulations, as discussed in § 5.2. We show results only for the noisy free stream, as our controller entirely stabilises the flow for the clean free stream case, resulting in a steady flow maintained by a steady actuation signal.

Figure 31 shows the control kernels for two sensor–actuator pairs in the time and frequency domains. The sensor, actuator and target combination of figures 31(*a*) and 31(*c*) are equivalent to those considered for the linear system in figure 27. Since our sensor and actuator locations are positions close to each other, the peaks of the non-causal kernels are near $\tau U_\infty/L_c = 0$. A notable difference between the kernels for the linear and nonlinear systems (which are different because of the coloured nonlinear forcing) is that the vortex-shedding frequency is considerably less prevalent in the nonlinear case, presumably due to the disruption of the vortex shedding by the noisy free stream. Accordingly, the control kernels will amplify a wide range of frequencies in the sensor readings, producing broadband actuation signals.

Figure 32 shows the PSD of the streamwise velocity at the target for the uncontrolled and controlled flows. Both the single and nested controllers significantly reduce the energy of velocity fluctuations for $St_\alpha < 1$. The control performance, measured by the metric

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

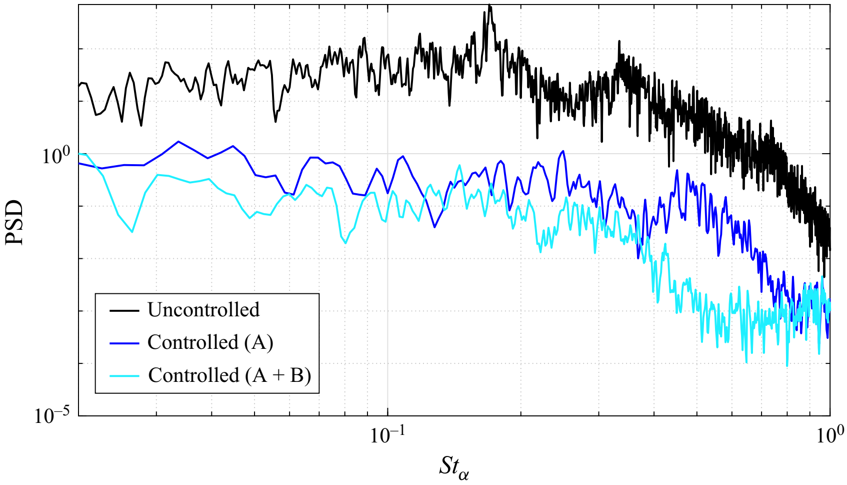


Figure 32. PSD of the streamwise velocity fluctuation u'_x at the target z located at $[x/L_c, y/L_c] = [1.5, -0.11]$. The black solid line represents the uncontrolled flow; the blue line shows the controlled flow using Controller A; the cyan line depicts the controlled flow using both controllers (Controller A + Controller B).

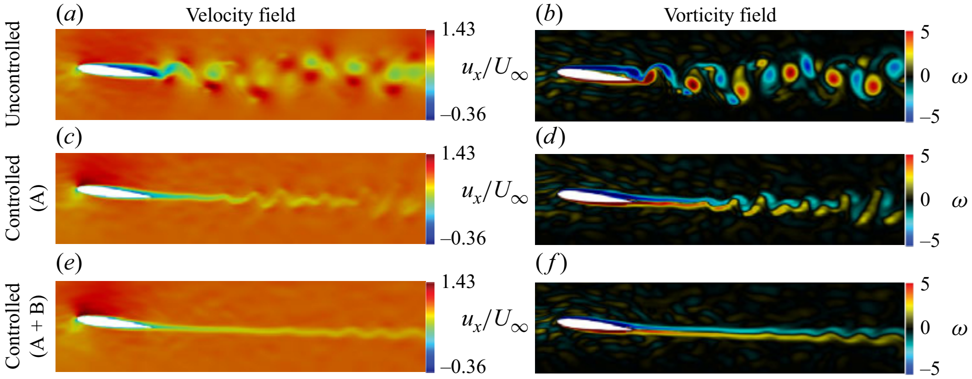


Figure 33. Velocity (u_x) and vorticity (ω) fields for the system with noisy free stream inflow. (*a,b*) Uncontrolled flows; (*c,d*) controlled flows using Controller A; (*e,f*) controlled flows using both controllers (Controller A + Controller B).

in (8.1) representing the reduction in velocity fluctuation energy, reaches approximately 94 % with Controller A. By incorporating a second controller (Controller A + B), the performance improves further, achieving 98 % reduction in fluctuation energy at the target. The reduction in r.m.s. velocity is 78 % using Controller A, and 90 % using both Controllers A and B. As shown in figure 32, this improvement further mitigates target fluctuations at higher frequencies ($0.4 < St_\alpha < 0.7$), which originate further downstream, as we can verify in figure 33.

Figure 33 presents both uncontrolled and controlled snapshots of the streamwise velocity and vorticity. As shown in figures 33(*a*) and 33(*b*), chaotic vortex shedding is prominent in the wake of the uncontrolled flow. However, this can be significantly mitigated by Controller A. Introducing Controller B further suppresses the oscillating flow downstream ($x/L_c > 2$). In the controlled flow, the mean flow changed, which is essential as the wake fluctuations from vortex shedding originate from the separated flow.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

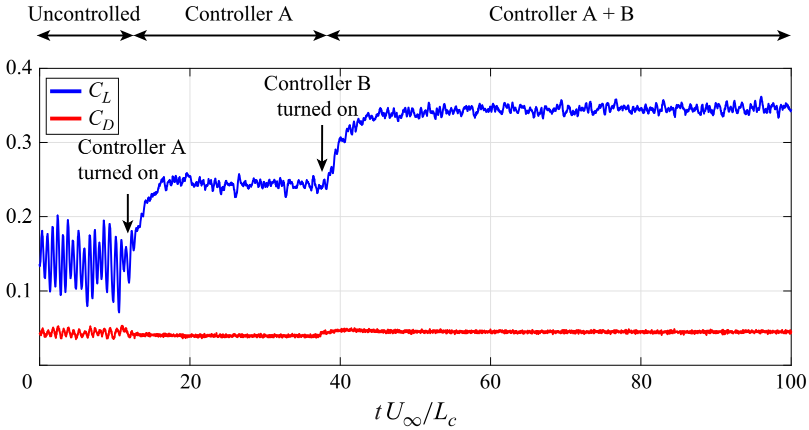


Figure 34. Lift and drag coefficients for the uncontrolled and controlled flow over time.

Finally, figure 34 demonstrates the impact of the controllers on the aerodynamic coefficients. The time-averaged lift coefficient ($\bar{C}_L$) is improved by 143 % with the use of Controllers A and B, while the time-averaged drag coefficient ($\bar{C}_D$) remains largely unchanged. The fluctuations in both coefficients are effectively suppressed by the controllers. Although improving the aerodynamic coefficients was not the primary objective, mitigating wake fluctuations improved the lift coefficient as a welcome byproduct.

9. Conclusions

We have demonstrated the successful application of resolvent-based estimation and control to a two-dimensional NACA 0012 airfoil at $Ma_\infty = 0.3$, $Re_{L_c} = 5000$ and $\alpha = 6.5^\circ$ immersed in both clean and noisy free streams. This represents an advance in the methodology, implementation and application of the resolvent-based framework to vortex shedding in aerodynamic wakes.

The resolvent-based framework, recently introduced by Martini *et al.* (2022), offers significant advantages over standard methods. Under similar assumptions, our estimator and controller converge to the Kalman filter and LQG controller. However, our approach can incorporate the nonlinear terms of the Navier–Stokes equations using coloured-in-time statistics, leading to significantly higher estimator accuracy and improved controller performance. Second, to construct the estimator and controller, we employed two computational approaches: an operator-based approach and a data-driven approach. The operator-based approach is computationally efficient, does not require *a priori* model reduction and accounts for the coloured statistics of nonlinear terms from the Navier–Stokes equations that act as a forcing on the linear dynamics. The data-driven approach, which circumvents the need to construct linearised Navier–Stokes operators, naturally incorporates these coloured statistics of the nonlinear terms. We used the Wiener–Hopf formalism to enforce causality, enabling the evaluation of only available measurements, which is an optimal strategy for real-time estimation and control.

This work represents the first application of resolvent-based estimation and control within a compressible flow solver. Specifically, it demonstrates the integration and execution of resolvent-based estimation and control approaches within a compressible flow solver designed for high-performance computing for large-scale problems. First, the

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

linearised Navier–Stokes operator developed in this work is accurate and applicable for parallel computing, making it a powerful tool for large-scale applications. Second, using a parallel time-stepping approach significantly accelerates computations when handling large-scale linearised Navier–Stokes operators, including using a parallel adjoint solver efficiently. Additionally, the streaming Fourier transform within a solver is valuable for saving memory for constructing cross-spectral densities, which are necessary for building kernels. Another key feature of the tool is its ability to extract the nonlinear terms of the Navier–Stokes equations, offering valuable insights for other methodologies. Lastly, we solve the Wiener–Hopf problems directly within the solver to minimise reliance on post-processing tools such as MATLAB and to enhance computational efficiency by enabling faster routines with reduced memory usage.

Before applying estimation and control to the laminar airfoil, we obtained the mean flow through direct numerical simulation and performed global stability and resolvent analysis around this mean flow. Random upstream perturbations were introduced to disrupt the periodic limit cycle caused by vortex shedding, inducing chaotic fluctuations. We then conducted resolvent-based estimation and control on both linear and nonlinear systems under these conditions.

Our results demonstrate that resolvent-based kernels are effective in estimating and controlling chaotic fluctuations in the wake of an airfoil, thus demonstrating their applicability to vortex shedding in a wake for the first time. The performance of both estimation and control is enhanced when sensors and actuators are strategically placed in effective locations. To determine these effective placements, we investigated estimation errors and employed a streamline strategy. While we addressed the estimation across the entire wake region, we found that controlling the entire region yields results similar to targeting a single point. In the linear system, the estimation error is approximately 8 % with two sensors and the control performance reaches 85 % using two actuators. For the nonlinear system, the estimation error increases to 30 % with four sensors, while the control performance improves to 98 % using four actuators and two controllers. These accomplishments are significant, as they demonstrate the feasibility of applying a new closed-loop control method in a numerical set-up and achieving reliable estimation. This work provides a solid foundation for extending the resolvent-based framework to turbulent wakes and other turbulent flow scenarios.

Future work will focus on optimising sensor and actuator placements, as well as applying control strategies to turbulent wakes behind an airfoil. In this study, we evaluated effective sensor placement based on estimation error, leading to satisfactory estimation and control results. However, the placement and the number of sensors and actuators have not yet been optimised. Addressing this will require formulating and solving a mathematical optimisation problem, which will be considered in future research. Additionally, applying estimation and closed-loop control strategies to turbulent wakes is another area for future exploration. The positive effects of reducing turbulent wakes in terms of aerodynamic performance should be further investigated. Finally, incorporating concepts from robust control theory to the resolvent-based control framework could broaden its applicability to wider classes of flows.

Supplementary movies. Supplementary movies are available at https://doi.org/10.1017/jfm.2025.10423.

Funding. J.J. and A.T. were supported by the Air Force Office of Scientific Research (AFOSR) grant no. FA9550-20-1-0214.

Declaration of interests. The authors report no conflict of interest.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

Appendix A. Resolution study of the baseflow in terms of eigenvalues

Here, we study the grid convergence of the frequency of the vortex-shedding frequency in the DNS and the eigenvalue of the linear operator $\boldsymbol{A}$. Here, N_s is the number of grid points along the suction surface and N_w is the number of grid points along the streamwise axis in the wake. Due to the need for a fine grid at the downstream target locations, we use a finer grid in the wake region compared with the typical grids with fewer points used in previous studies (Kojima *et al.* 2020; Marquet *et al.* 2022). The Strouhal number (based on the angle of attack) $St_{\alpha,DNS}$ is the vortex-shedding frequency observed in the DNS and $St_{\alpha,\boldsymbol{A}}$ is the frequency of the least-stable eigenvalue of the operator $\boldsymbol{A}$ linearised about the corresponding DNS mean flow. Based on our convergence study, we selected Mesh 5 for this work.

Appendix B. Wiener–Hopf method

B.1. *Theoretical Wiener–Hopf decomposition*

The Wiener–Hopf method (Noble 1958) is a mathematical technique extensively used in applied mathematics. It enables the decomposition of arbitrary functions into components corresponding to the upper and lower halves of the complex plane. This paper leverages the Wiener–Hopf method to impose causality on estimation and control kernels, following methodologies outlined by Daniele & Lombardi (2007), Martinelli (2009) and Martini *et al.* (2022).

First, we define the Fourier transform as

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-i\omega t}\, \mathrm{d}t, \tag{B1}$$

where $\hat{f}(\omega)$ represents an arbitrary function in the frequency domain. This function can be decomposed into $\hat{f}_+(\omega)$ and $\hat{f}_-(\omega)$, which are analytic functions in the lower and upper complex half-planes, respectively. These can also be analysed in the time domain as

$$\hat{f}_+(\omega) = \int_{0}^{+\infty} f(t) e^{-i\omega t}\, \mathrm{d}t, \tag{B2}$$

$$\hat{f}_-(\omega) = \int_{-\infty}^{0} f(t) e^{-i\omega t}\, \mathrm{d}t, \tag{B3}$$

where the $(+)$ subscript indicates that the function contains values only in the positive time domain, while the $(-)$ subscript denotes that the function contains values only in the negative time domain. Thus, the subscripts $(+)$ and $(-)$ impose causality and non-causality on the function, respectively.

Now, we consider the two Wiener–Hopf problems (Martini *et al.* 2022) related to this paper's work,

$$\hat{\boldsymbol{H}}(\omega)\hat{\boldsymbol{\Gamma}}_+(\omega) = \hat{\boldsymbol{\Lambda}}_-(\omega) + \hat{\boldsymbol{G}}(\omega), \tag{B4}$$

$$\hat{\boldsymbol{K}}(\omega)\hat{\boldsymbol{\Gamma}}_+(\omega)\hat{\boldsymbol{H}}(\omega) = \hat{\boldsymbol{\Gamma}}_-(\omega) + \hat{\boldsymbol{L}}(\omega)\hat{\boldsymbol{G}}(\omega), \tag{B5}$$

where $\omega = i2\pi f$ with frequency f, and $\hat{\boldsymbol{H}}$, $\hat{\boldsymbol{G}}$, $\hat{\boldsymbol{K}}$, $\hat{\boldsymbol{L}}$ are known matrices (or functions), while $\hat{\boldsymbol{\Gamma}}_+(\omega)$ and $\hat{\boldsymbol{\Lambda}}_-(\omega)$ are the unknown matrices (or functions). The objective of the Wiener–Hopf problem is to determine $\hat{\boldsymbol{\Gamma}}_+(\omega)$ and $\hat{\boldsymbol{\Lambda}}_-(\omega)$.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

Mesh	N_s	N_w	Total cells	$St_{\alpha,DNS}$	$St_{\alpha,\boldsymbol{A}}$
1	100	90	29 704	0.169650	0.168319
2	100	180	43 924	0.168541	0.167249
3	150	135	67 354	0.169124	0.168933
4	200	180	1 20 204	0.169525	0.169140
5	200	268	1 48 188	0.169288	0.169241
6	200	300	1 58 364	0.169211	0.169222
7	200	360	1 77,444	0.169246	0.169207

Table 2. Grid convergence of the DNS vortex shedding frequency and the least-stable eigenvalue of the linear operator $\boldsymbol{A}$.

To solve the two Wiener–Hopf problems, we employ two types of factorisations. The additive factorisation decomposes the matrix into two ± components, separated only through the addition process

$$\hat{\boldsymbol{\Gamma}}(\omega)=[\hat{\boldsymbol{\Gamma}}(\omega)]_-+[\hat{\boldsymbol{\Gamma}}(\omega)]_+, \tag{B6}$$

indicated by square brackets $[\cdot]_\pm$. The multiplicative factorisation, which is not commutative, is performed as

$$\hat{\boldsymbol{\Gamma}}(\omega)=(\hat{\boldsymbol{\Gamma}}(\omega))_-(\hat{\boldsymbol{\Gamma}}(\omega))_+, \tag{B7}$$

using the parentheses $(\cdot)_\pm$. The solutions of the first Wiener–Hopf problems in (B4) is given by

$$\hat{\boldsymbol{\Gamma}}_+(\omega)=\left[\hat{\boldsymbol{G}}(\omega)(\hat{\boldsymbol{H}}(\omega))_-^{-1}\right]_+(\hat{\boldsymbol{H}}(\omega))_+^{-1}, \tag{B8}$$

where $(\hat{\boldsymbol{H}})_-$ is obtained from the reverse multiplicative factorisation. For (B5), the solutions is

$$\hat{\boldsymbol{\Gamma}}_+(\omega)=(\hat{\boldsymbol{K}}(\omega))_+^{-1}\left[(\hat{\boldsymbol{K}}(\omega))_-^{-1}\hat{\boldsymbol{L}}(\omega)\hat{\boldsymbol{G}}(\omega)(\hat{\boldsymbol{G}}(\omega))_-^{-1}\right]_+(\hat{\boldsymbol{H}}(\omega))_+^{-1}. \tag{B9}$$

In (B8) and (B9), the inverse operation is applied following the completion of the Wiener–Hopf factorisation.

B.2. *Numerical Wiener–Hopf decomposition*

To ensure a, seemly, fully parallelised workflow, we solve the Wiener–Hopf problems (Noble 1958) used to enforce causality within the CFD solver. While the additive factorisation is straightforward to solve numerically, the multiplicative factorisation is more complicated. A numerical solution for the multiplicative factorisation was provided by Martini *et al.* (2022). The solution of the multiplicative factorisation can be independently formed as

$$\hat{\boldsymbol{G}}(\omega)\hat{w}_{i,+}(\omega)=\hat{w}_{i,-}(\omega), \tag{B10}$$

where

$$(\hat{\boldsymbol{G}})_-(\omega)=[\hat{w}_{1,-}(\omega),\,\hat{w}_{2,-}(\omega),\,\ldots,\,\hat{w}_{n_{freq},-}(\omega)], \tag{B11a}$$

$$(\hat{\boldsymbol{G}})_+(\omega)=[\hat{w}_{1,+}(\omega),\,\hat{w}_{2,+}(\omega),\,\ldots,\,\hat{w}_{n_{freq},+}(\omega)]^{-1}, \tag{B11b}$$

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

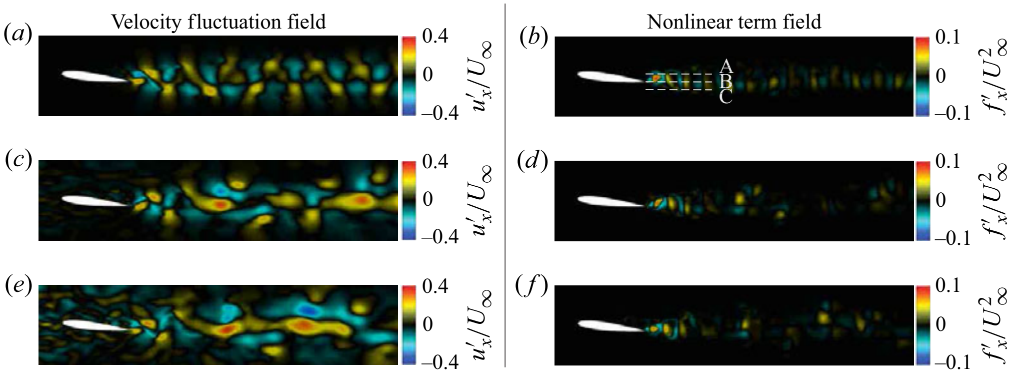


Figure 35. (*a,c,e*) Instantaneous velocity fluctuation field u'_x and (*b,d,f*) the corresponding nonlinear terms f'_x computed from (6.3): (*a,b*) no forcing ($W = 0$); (*c,d*) $W = 1$; and (*e,f*) $W = 3$.

where $\hat{\boldsymbol{G}} \in \mathbb{C}^{n_y \times n_y \times n_{freq}}$ or $\hat{\boldsymbol{G}} \in \mathbb{C}^{n_a \times n_a \times n_{freq}}$. To solve (B10), a Fredholm integral equation of the second kind (Daniele & Lombardi 2007) is derived as

$$\hat{\boldsymbol{x}}_i(\omega) + \frac{1}{2\pi \mathrm{i}} \int_{-\infty}^{\infty} \frac{\hat{\boldsymbol{G}}^{-1}(\omega)\hat{\boldsymbol{G}}(u) - 1}{u - \omega} \hat{\boldsymbol{x}}_i(u)\,\mathrm{d}u = \hat{\boldsymbol{G}}^{-1}(\omega) \frac{\hat{\boldsymbol{w}}_{i,-}(\omega_0)}{\omega - \omega_0}. \tag{B12}$$

Due to the difficulty of the formation of the integration path, Martini *et al.* (2022) constructed a linear problem with the size $\hat{\boldsymbol{G}} \in \mathbb{C}^{n_y \times n_y \times n_{freq}}$, given by

$$\hat{\boldsymbol{x}}_i(\omega) + \frac{1}{2\mathrm{i}} \mathcal{H}\left(\hat{\boldsymbol{x}}_i\right)(\omega) - \frac{1}{2\mathrm{i}} \hat{\boldsymbol{G}}^{-1}(\omega) \mathcal{H}\big(\hat{\boldsymbol{G}}\hat{x}_i\big)(\omega) = \hat{\boldsymbol{G}}^{-1}(\omega) \frac{\hat{\boldsymbol{w}}_{i,-}(\omega_0)}{\omega - \omega_0}, \tag{B13}$$

where

$$\mathcal{H}(\hat{\boldsymbol{x}}) = P.V. \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{1}{\omega - u} \hat{\boldsymbol{x}}(u)\mathrm{d}u, \tag{B14}$$

represents the Hilbert transform of $\hat{\boldsymbol{x}}(\omega)$. Equation (B14) can be solved using the generalised minimal residual (GMRES) iterative method (Saad & Schultz 1986). We solve (B14) directly within the solver to minimise reliance on post-processing tools such as MATLAB and to enhance computational efficiency by enabling faster routines with reduced memory usage.

Appendix C. Statistics of the nonlinear terms

We explore how much the nonlinearity interaction is developed from the external forcing $\boldsymbol{F}_{ext}$ for the laminar airfoil flow with the intensity of the external disturbance. If the intensity of the external forcing is sufficiently small, (5.13) is approximately equivalent to (5.14). Figure 35 illustrates the instantaneous streamwise velocity fluctuations u'_x field with $W = [0, 1, 3]$ in panels (*a*), (*c*) and (*e*), and the corresponding same time step's nonlinear terms in panels (*b*), (*d*) and (*f*). The nonlinear terms are extracted using our application in § 6.3. The clean system ($W = 0$) shown in figures 35(*a*) and 35(*b*) and the noisy systems ($W = [1, 3]$) depicted in figures 35(*c*)–35(*f*) demonstrate that nonlinear terms predominantly exist in the wake near the trailing edge and in regions where the gradients of the velocity component are significant.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

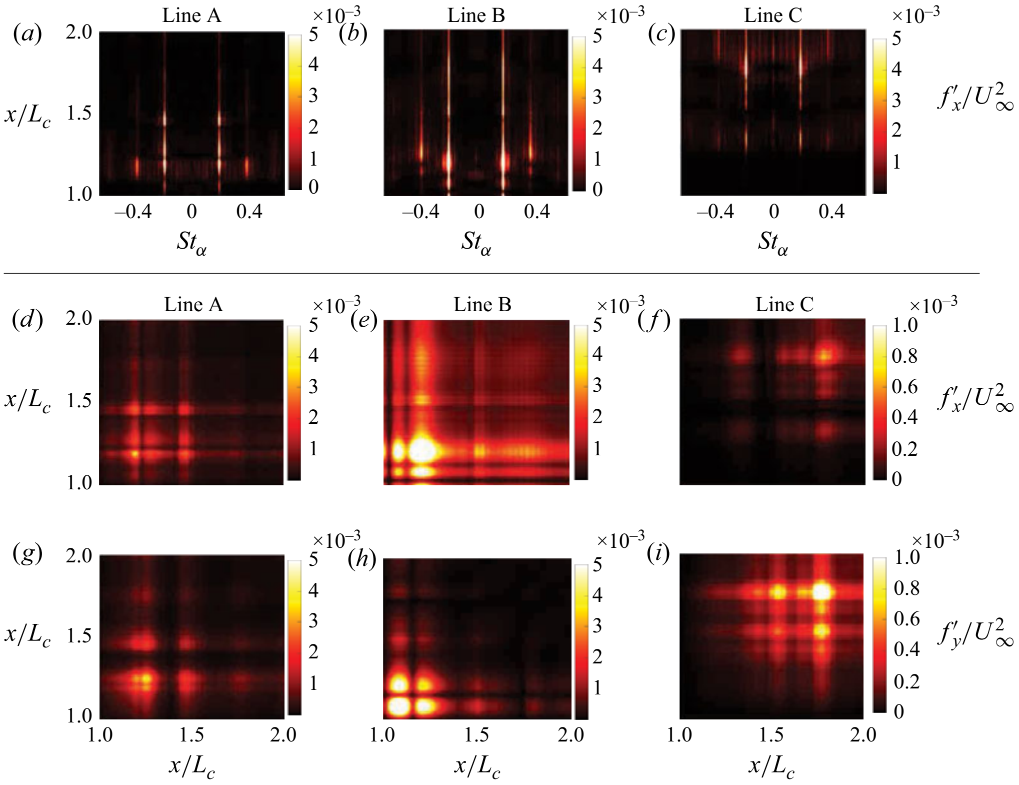


Figure 36. PSDs and CSDs of the nonlinear terms. (*a*)–(*c*) PSDs along lines A, B, and C in figure 35(*b*). (*d*)–(*f*) CSDs of f_x', and (*g*)–(*i*) CSDs of f_y' for each corresponding line.

Figure 36 presents the PSDs and the CSDs at vortex-shedding frequency ($St_\alpha = 0.17$) of the nonlinear terms along lines A ($y/L_c = 0.01$), B ($y/L_c = -0.11$) and C ($y/L_c = -0.21$) behind the airfoil, shown in figure 35(*b*). In figures 36(*a*)–36(*c*), the nonlinear terms exhibit significant energy at the vortex-shedding frequencies. Strong energy is observed near the trailing edge ($x/L_c < 1.5$) along the top (A) and middle (B) lines, as shown in figures 36(*a*) and 36(*b*), while the bottom line (C) shows energetic regions further downstream, as seen in figure 36(*c*). This pattern arises because vortex shedding primarily originates from the separated flow on the suction surface, located slightly above the trailing edge. In figures 36(*d*)–36(*i*), the nonlinear terms for both velocity components exhibit statistically significant strength in the region $1 < x/L_c < 1.2$ along line B, consistent with the observed deterioration in the estimation performance of the linear system. Examining the CSD tensors $\hat{\boldsymbol{F}}_{nl}$ is meaningful, as they are closely associated with coherent structures (Towne *et al.* 2018). By constructing the forcing CSD matrix $\hat{\boldsymbol{F}}_{nl}$, we can account for the nonlinear effects of the flow in the estimation kernels, potentially improving estimation accuracy in turbulent flows.

Appendix D. Sensor placement for estimation

Based on the single-sensor results reported in § 7.2.3, we propose several candidates for the most effective sensor placement or multiple sensors, shown in table 3. The targets

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

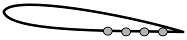
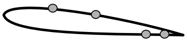
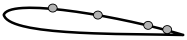
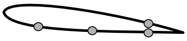
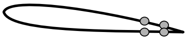
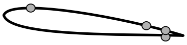
	Placement candidates	Errors	
		Clean system	Noisy system
1		0.04	0.54
2		0.03	0.40
3		0.02	0.36
4		0.01	0.35
5		0.001	0.33
6		0.001	0.33

Table 3. Sensor placement candidates for causal resolvent-based estimation.

are averaged points of interest for both clean and noisy nonlinear systems. We verified that the estimator can improve accuracy when the sensor is at the trailing edge. However, this configuration is impractical for the control problem, which will be discussed in the upcoming section. Ultimately, we adopted candidate 6 in table 3, where sensors are located upstream of the separation bubble on the suction surface and near the trailing edge on both the suction and pressure surfaces for estimation. The mathematical formulations for determining the optimal sensor placement are beyond the scope of this study and will be addressed in future research.

References

Agrawal, B.R., Rosenberg, A. & Sharma, A. 2015 Towards identifying contribution of wake turbulence to inflow noise from wind turbines. *E3S Web Conf.* **5**, 01002.

Alam, M.M., Zhou, Y., Yang, H.X., Guo, H. & Mi, J. 2010 The ultra-low Reynolds number airfoil wake. *Exp. Fluids.* **48** (1), 81–103.

Amaral, F.R., Cavalieri, A.V.G., Martini, E., Jordan, P. & Towne, A. 2021 Resolvent-based estimation of turbulent channel flow using wall measurements. *J. Fluid Mech.* **927**, A17.

An, X., Wiliams, D.R., Eldredge, J.D. & Colonius, T. 2021 Lift coefficient estimation for a rapidly pitching airfoil. *Exp. Fluids.* **62** (1), 11.

Asztalos, K.J., Dawson, S.T.M. & Williams, D.R. 2021 Modeling the flow state sensitivity of actuation response on a stalled airfoil. *AIAA J.* **59** (8), 1–15.

Audiffred, D.B.S., Cavalieri, A.V.G., Brito, P.P.C. & Martini, E. 2023 Experimental control of Tollmien–Schlichting waves using the Wiener–Hopf formalism. *Phys. Rev. Fluids* **8** (7), 073902.

Bagheri, S., Brandt, L. & Henningson, D.S. 2009 Input–output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mech.* **620**, 263–298.

Balay, S. *et al.* 2019 *Petsc users manual*. Argonne National Laboratory.

Barkley, D. 2006 Linear analysis of the cylinder wake mean flow. *Europhys. Lett.* **75** (5), 750–756.

Bhagwat, R. 2020 Development of stability analysis tools for high speed compressible flows. *PhD thesis*, North Carolina State University, USA.

Broglia, R., Choi, K.-S., Houston, P., Pasquale, L. & Zanchetta, P. 2018 Output feedback control of flow separation over an aerofoil using plasma actuators. *Intl J. Numer. Anal. Model.* **15** (6), 864–883.

Brunton, S.L. & Noack, B.R. 2015 Closed-loop turbulence control: progress and challenges. *ASME Appl. Mech. Rev.* **67** (5), 050801.

Brès, G.A., Ham, F.E., Nichols, J.W. & Lele, S.K. 2017 Unstructured large-Eddy simulations of supersonic jets. *AIAA J.* **55** (4), 1164–1184.

Chang, J., Zhang, Q., He, L. & Zhou, Y. 2022 Shedding vortex characteristics analysis of NACA 0012 airfoil at low Reynolds numbers. *Energy Rep.* **8**, 156–174.

Chu, B.-T. 1965 On the energy transfer to small disturbances in fluid flow (Part I). *Acta Mechanica* **1** (3), 215–234.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

COLBURN, C.H., CESSNA, J.B. & BEWLEY, T.R. 2011 State estimation in wall-bounded flow systems. Part 3. The ensemble Kalman filter. *J. Fluid Mech.* **682**, 289–303.

COLONIUS, T. & TOWNE, A. 2025 *Chapter 2 – Modal Decomposition*. Academic Press.

COLONIUS, T. & WILLIAMS, D.R. 2011 Control of vortex shedding on two- and three-dimensional aerofoils. *Phil. Trans. R. Soc. A* **369** (1940), 1525–1539.

COOK, D.A., THOME, J., BROCK, J.M., NICHOLS, J.W. & CANDLER, G.V. 2018 *Understanding Effects of Nose-Cone Bluntness on Hypersonic Boundary Layer Transition Using Input-Output Analysis*.

DANIELE, V. & LOMBARDI, G. 2007 Fredholm factorization of Wiener–Hopf scalar and matrix kernels. *Radio Sci.* **42** (6), 2007RS003673.

DÉDA, T., WOLF, W. & DAWSON, S. 2023 Neural networks in feedback for flow analysis, sensor placement and control. arXiv:2308.13029.

EGGERT, C.A. & RUMSEY, C.L. 2017 CFD study of NACA 0018 airfoil with flow control.

FABBIANE, N., BAGHERI, S. & HENNINGSON, D.S. 2017 Energy efficiency and performance limitations of linear adaptive control for transition delay. *J. Fluid Mech.* **810**, 60–81.

FABBIANE, N., SEMERARO, O., BAGHERI, S. & HENNINGSON, D.S. 2014 Adaptive and model-based control theory applied to convectively unstable flows. *Appl. Mech. Rev.* **66** (6), 060801.

FABBIANE, N., SIMON, B., FISCHER, F., GRUNDMANN, S., BAGHERI, S. & HENNINGSON, D.S. 2015 On the role of adaptivity for robust laminar flow control. *J. Fluid Mech.* **767**, R1.

FARGHADAN, A., JUNG, J., BHAGWAT, R. & TOWNE, A. 2024 Efficient harmonic resolvent analysis via time stepping. *Theor. Comput. Fluid Dyn.* **38** (3), 331–353.

FARGHADAN, A., MARTINI, E. & TOWNE, A. 2023 Scalable resolvent analysis for three-dimensional flows.

FFOWCS WILLIAMS, J.E. & ZHAO, B.C. 1989 The active control of vortex shedding. *J. Fluids Struct.* **3** (2), 115–122.

FOSAS, D.P.M., SIPP, D. & SCHMID, P.J. 2012 Efficient evaluation of the direct and adjoint linearized dynamics from compressible flow solvers. *J. Comput. Phys.* **231** (23), 7739–7755.

FRIGO, M. & JOHNSON, S.G. 2005 The design and implementation of FFTW3. *Proc. IEEE* **93** (2), 216–231.

FUJISAWA, N., KAWAJI, Y. & IKEMOTO, K. 2001 Feedback control of vortex shedding from a circular cylinder by rotational oscillations. *J. Fluids Struct.* **15** (1), 23–37.

GOMEZ, D.F., LAGOR, F., KIRK, P.B., LIND, A., JONES, A.R. & PALEY, D.A. 2019 Unsteady dmd-based flow field estimation from embedded pressure sensors in an actuated airfoil. In *AIAA SCITECH Forum*.

GOTTLIEB, S. & SHU, C. 1998 Total variation diminishing Runge–Kutta schemes. *Maths Comput.* **67** (221), 73–85.

GROSS, A., MARKS, C. & SONDERGAARD, R. 2024 Laminar separation control for eppler 387 airfoil based on resolvent analysis. *AIAA J.* **62** (4), 1487–1502.

HUERRE, P. & MONKEWITZ, P.A. 1990 Local and global instabilities in spatially developing flows. *Annu. Rev. Fluid Mech.* **22** (1), 473–537.

JEUN, J., NICHOLS, J.W. & JOVANOVIĆ, M.R. 2016 Input-output analysis of high-speed axisymmetric isothermal jet noise. *Phys. Fluids* **28** (4), 047101.

JIN, B., ILLINGWORTH, S.J. & SANDBERG, R.D. 2020 Feedback control of vortex shedding using a resolvent-based modelling approach. *J. Fluid Mech.* **897**, A26.

JOVANOVIĆ, M.R. & BAMIEH, B. 2005 Componentwise energy amplification in channel flows. *J. Fluid Mech.* **534**, 145–183.

JUNG, J. 2024 Resovlent-based estimation and control of aerodynamic flows. PhD thesis, University of Michigan.

JUNG, J., MARTINI, E., CAVALIERI, A., JORDAN, P., LESSHAFFT, L. & TOWNE, A. 2020 Optimal resolvent-based estimation for flow control. *APS Div. Fluid Dyn.* **G06**, 004.

KALMAN, R.E. 1960 A new approach to linear filtering and prediction problems. *J. Basic Engng* **82** (1), 35–45.

KOJIMA, Y., YEH, C., TAIRA, K. & KAMEDA, M. 2020 Resolvent analysis on the origin of two-dimensional transonic buffet. *J. Fluid Mech.* **885**, R1.

LECLERCQ, C., DEMOURANT, F., POUSSOT-VASSAL, C. & SIPP, D. 2019 Linear iterative method for closed-loop control of quasiperiodic flows. *J. Fluid Mech.* **868**, 26–65.

LIN, C. & TSAI, H. 2024 Feedback flow control on a plunging circular cylinder. *Phys. Fluids* **36** (4), 047126.

MARQUET, O., LEONTINI, J.S., ZHAO, J. & THOMPSON, M.C. 2022 Hysteresis of two-dimensional flows around a NACA0012 airfoil at $Re = 5000$ and linear analyses of their mean flow.. *Intl J. Heat and Fluid Flow* **94**, 108920.

MARTINELLI, F. 2009 Feedback control of turbulent wall flows. *PhD thesis*.

MARTINI, E., CAVALIERI, A.V.G., JORDAN, P., TOWNE, A. & LESSHAFFT, L. 2020 Resolvent-based optimal estimation of transitional and turbulent flows. *J. Fluid Mech.* **900**, A2.

MARTINI, E., JUNG, J., CAVALIERI, A., JORDAN, P. & TOWNE, A. 2022 Resolvent-based tools for optimal estimation and control via the Wiener–Hopf formalism. *J. Fluid Mech.* **937**, A19.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

McKeon, B.J. & Sharma, A.S. 2010 A critical-layer framework for turbulent pipe flow. *J. Fluid Mech.* **658**, 336–382.

Min, C. & Choi, H. 1999 Suboptimal feedback control of vortex shedding at low Reynolds numbers. *J. Fluid Mech.* **401**, 123–156.

Morra, P., Nogueira, P.A.S., Cavalieri, A.V.G. & Henningson, D.S. 2021 The colour of forcing statistics in resolvent analyses of turbulent channel flows. *J. Fluid Mech.* **907**, A24.

Morra, P., Sasaki, K., Hanifi, A., Cavalieri, A.V.G. & Henningson, D.S. 2020 A realizable data-driven approach to delay bypass transition with control theory. *J. Fluid Mech.* **883**, A33.

Nielsen, E.J. & Kleb, W.L. 2006 Efficient construction of discrete adjoint operators on unstructured grids using complex variables. *AIAA J.* **44** (4), 827–836.

Noack, B.R., Afanasiew, K., Morzynski, M., Tadmor, G. & Thiele, F. 2003 A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.* **497**, 335–363.

Noble, B. 1958 Methods based on the Wiener-Hopf technique for the solution of partial differential equations. In *Integral Series on Monograph Pure Applied Math*, vol. **7**. Pergamon Press.

Pasquale, L., Broglia, R., Choi, K.-S., Durante, D. & Zanchetta, P. 2017 Robust control of flow separation over a pitching aerofoil using plasma actuators. *IFAC-PapersOnLine* **50** (1), 11120–11125.

Puri, K., Laufer, M., Müller-Vahl, H., Greenblatt, D. & Frankel, S.H. 2018 Computations of active flow control via steady blowing over a NACA-0018 airfoil: implicit LES and RANS validated against experimental data.. In Proc. AIAA Aerospace Sciences Meeting.

Radespiel, R., Burnazzi, M., Casper, M. & Scholz, P. 2016 Active flow control for high lift with steady blowing. *Aeronaut. J.* **120** (1223), 171–200.

Rafiee, M., Wu, Q. & Bayen, A.M. 2009 Kalman filter based estimation of flow states in open channels using Lagrangian sensing. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC), pp. 8266–8271. IEEE.

Ribeiro, J.H.M., Yeh, C. & Taira, K. 2020 Randomized resolvent analysis. *Phys. Rev. Fluids* **5** (3), 033902.

Roussopoulos, K. & Monkewitz, P.A. 1996 Nonlinear modelling of vortex shedding control in cylinder wakes. *Physica D: Nonlinear Phenom.* **97** (1–3), 264–273.

Saad, Y. & Schultz, M.H. 1986 GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.* **7** (3), 856–869.

Sasaki, K., Morra, P., Fabbiane, N., Cavalieri, A.V.G., Hanifi, A. & Henningson, D.S. 2018*a* On the wave-cancelling nature of boundary layer flow control. *Theor. Comput. Fluid Dyn.* **32** (5), 593–616.

Sasaki, K., Tissot, G., Cavalieri, A.V.G., Silvestre, F.J., Jordan, P. & Biau, D. 2018*b* Closed-loop control of a free shear flow: a framework using the parabolized stability equations. *Theor. Comput. Fluid Dyn.* **32** (6), 765–788.

Schmid, P.J. & Sipp, D. 2016 Linear control of oscillator and amplifier flows. *Phys. Fluids* **1** (4), 040501.

Schmidt, O.T. & Towne, A. 2019 An efficient streaming algorithm for spectral proper orthogonal decomposition. *Comput. Phys. Commun.* **237**, 98–109.

Schmidt, O.T., Towne, A., Rigas, G., Colonius, T. & Brès, G.A. 2018 Spectral analysis of jet turbulence. *J. Fluid Mech.* **855**, 953–982.

Sharma, A., Limebeer, D., McKeon, B. & Morrison, J. 2006 Stabilising control laws for the incompressible Navier–Stokes equations using sector stability theory. In *3rd AIAA Flow Control Conf*. AIAA.

Sipp, D., Marquet, O., Meliga, P. & Barbagallo, A. 2010 Dynamics and control of global instabilities in open-flows: a linearized approach. *Appl. Mech. Rev.* **63** (3), 030801.

Stengel, R.F. 1994 *Optimal Control and Estimation*. Dover Publishing.

Strykowski, P.J. & Sreenivasan, K.R. 1990 On the formation and suppression of vortex 'shedding' at low Reynolds numbers. *J. Fluid Mech.* **218**, 71–107.

Symon, S., Sipp, D. & McKeon, B.J. 2019 A tale of two airfoils: resolvent-based modelling of an oscillator versus an amplifier from an experimental mean. *J. Fluid Mech.* **881**, 51–83.

Tao, J.S., Huang, X.Y. & Chan, W.K. 1996 A flow visualization study on feedback control of vortex shedding from a circular cylinder. *J. Fluids Struct.* **10** (8), 965–970.

Thomareis, N. & Papadakis, G. 2018 Resolvent analysis of separated and attached flows around an airfoil at transitional Reynolds number. *Phys. Rev. Fluids* **3** (7), 073901.

Towne, A., Bhagwat, R., Zhou, Y., Jung, J., Martini, E., Jordan, P., Audiffred, D., Maia, I. & Cavalieri, A. 2024 Resolvent-based estimation of wavepackets in turbulent jets. In *30th AIAA/CEAS Aeroacoustics Conf, 2024*. AIAA.

Towne, A., Brès, G.A. & Lele, S.K. 2017 A statistical jet-noise model based on the resolvent framework. *AIAA Paper* 2017–3406.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press

TOWNE, A., LOZANO-DURÁN, A. & YANG, X. 2020 Resolvent-based estimation of space–time flow statistics. *J. Fluid Mech.* **883**, A17.

TOWNE, A., SCHMIDT, O.T. & COLONIUS, T. 2018 Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *J. Fluid Mech.* **847**, 821–867.

WAGNER, S., BAREISS, R. & GUIDATI, G. 1996 *Wind Turbine Noise*. Springer-Verlag.

WELCH, P. 1967 The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust.* **15** (2), 70–73.

YEH, C. & TAIRA, K. 2019 Resolvent-analysis-based design of airfoil separation control. *J. Fluid Mech.* **867**, 572–610.

ZARE, A., JOVANOVIĆ, M.R. & GEORGIOU, T.T. 2017 Colour of turbulence. *J. Fluid Mech.* **812**, 636–680.

https://doi.org/10.1017/jfm.2025.10423 Published online by Cambridge University Press