Contents lists available at ScienceDirect

Physica D

journal homepage: www.elsevier.com/locate/physd

Data-driven inference of low order representations of observable dynamics for an airfoil model

Adharaa Neelim Dewanjee, Samir Sahyoun, Seddik Djouadi, Dan Wilson [*]

Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA

ARTICLE INFO

Communicated by Boumediene Hamzi

Keywords:
Airfoil model
Nonlinear dynamics
Vorticity
Isostable coordinates
Adaptive reduction strategy

ABSTRACT

Reduced order modeling of fluid flow systems can allow for the application of more sophisticated mathematical analysis and control techniques that would otherwise be computationally prohibitive. Existing reduction techniques often fail when nonlinear terms dominate the flow (e.g., at large Reynolds numbers) necessitating the development of new techniques. In this work, we implement an adaptive isostable reduction strategy to obtain a data-driven reduced order model that captures the dynamics of observables in a computational model for fluid flow over an airfoil at moderate Reynolds numbers. The resulting model characterizes the response to both time-varying inflow conditions and Dirichlet boundary conditions on the surface of the airfoil meant to represent suction or blowing through the action of surface jets. The resulting reduced order model behaviors agree well with the dynamics of the full order model simulations in response to both open loop and closed loop inputs. This study provides a proof of concept that reduced order modeling techniques using adaptive isostable coordinates can be successfully used in realistic fluid flow models using geometries with practical relevance.

1. Introduction

The steady growth of computational capabilities in recent decades has given rise to a sustained interest in the development of data-driven dynamical model identification techniques. Rather than relying on knowledge of underlying physical processes, data-driven strategies typically infer dynamical models from time-series measurements of observables [1,2]. Such techniques can readily be implemented when rich datasets are available and can be readily used across a wide variety of application domains. Data-driven model identification techniques have been particularly powerful for fluid flow applications [3–5] where time series measurements are often readily available but the mapping from the state (e.g., flow velocity, pressure) to the observables (e.g., drag, lift, and pixel intensity from flow imaging techniques) is not always known.

A wide variety of strategies have been developed for data-driven dynamical model identification purposes. For instance, proper orthogonal decomposition (POD) [3,4,6,7] can be used to yield a set of orthogonal modes arranged in descending order of their importance as gauged by the L^2 energy captured by each mode. Truncation of the less important modes yields a reduced order coordinate basis. Subsequently, Galerkin projection [6,8,9], can be applied to project the underlying model equations onto the reduced order POD basis to yield an associated set of low-order dynamical equations. Dynamic mode decomposition (DMD) is another widely used dynamical model identification procedure [1,10,11], whereby the temporal evolution of snapshot data can be represented as the superposition of linear eigenmodes. DMD shares a close relationship with Koopman operator theory [12–14], which can be used to characterize the dynamics of system observables in terms of a linear but generally infinite dimensional operator. DMD is often used to provide an approximation of Koopman eigenmodes from data and can be used in tandem with either lifting functions [15] or delayed embeddings [16] in order to yield a larger set of observables. Recent works have considered control inputs in concert with the DMD strategy. Associated model identification algorithms can yield linear [17,18], bilinear [19], and switched [20] control systems allowing for the use of well-established model predictive control techniques.

The central challenge of implementing the Koopman framework rests on finding a suitable finite basis of Koopman eigenmodes with which to approximate the infinite dimensional Koopman operator [3]. In this work, we will consider the isostable coordinate framework which encodes for level sets of the slowest decaying Koopman eigenmodes [21,22]. For systems that have a stable fixed point or periodic attractors, system behavior in response to inputs can often be accurately captured by considering the dynamics of only the slowest decaying Koopman eigenmodes [23–25], allowing for substantial dimension reduction. In situations where large magnitude inputs are

[*] Corresponding author.
E-mail address: dwilso81@utk.edu (D. Wilson).

https://doi.org/10.1016/j.physd.2023.133941
Received 6 April 2023; Received in revised form 28 August 2023; Accepted 29 September 2023
Available online 5 October 2023

necessary, Refs. [26,27] suggest the inclusion of an adaptive parameter set, defining extended isostable coordinates that are functions of both the state and the adaptive parameters and appropriately updating the adaptive parameter set in order to limit truncation errors associated with neglected nonlinear terms. Preliminary evidence [27–29], suggests that this is a promising approach for reduced order modeling of fluid flows, although these techniques have thus far only been applied to relatively simple models based on simulations of the Burgers' equation.

In this work, we implement the adaptive isostable reduction strategy for data-driven model identification and subsequent control of fluid flow over an airfoil in a regime with moderate Reynolds numbers. We use computational fluid dynamics (CFD) simulation data to identify a reduced order model that captures the dynamics of vorticity measurements in response to changes in the inflow conditions (meant to represent gusting winds) and to time-varying Dirichlet boundary conditions on the surface of the airfoil (meant to represent blowing or suction through the action of surface jets). We use the resulting reduced order model to design a simple controller to limit flow separation [30] which tends to reduce lift and increase drag. The resulting reduced order model behaviors agree well with the dynamics of the full order model obtained from CFD simulations when considering both open loop and closed loop inputs. Importantly, while the application considered here is still a prototype problem, the methodology described in this work is quite general and could be readily applied in other flow control applications.

The organization of this paper is as follows: Section 2 provides a description of the model equations, boundary conditions, and the control objective considered in this work. Section 3 gives necessary background about the reduced order modeling strategy and Section 4 describes the data-driven model identification procedure. Section 5 provides direct comparisons between full and reduced order model simulations. Section 5 also discusses results obtained for a control strategy designed using information about the reduced order model. Concluding remarks are given in Section 6.

2. Problem description

2.1. Model equations and boundary conditions

The aerodynamic features of two-dimensional fluid flows are investigated numerically using the turbulence closure model of the k-ϵ model to solve the two-dimensional steady incompressible Navier–Stokes equations. A NACA 0015 airfoil [31] was chosen as the physical model for this investigation. The dimensionless Navier–Stokes equations, a partial differential equation (PDE) that governs fluid velocity and pressure

$$\frac{\partial v_x}{\partial t} = \frac{1}{\text{Re}}\left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2}\right) - v_x\frac{\partial v_x}{\partial x} - v_y\frac{\partial v_x}{\partial y} - \frac{\partial p}{\partial x}, \tag{1}$$

$$\frac{\partial v_y}{\partial t} = \frac{1}{\text{Re}}\left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2}\right) - v_x\frac{\partial v_y}{\partial x} - v_y\frac{\partial v_y}{\partial y} - \frac{\partial p}{\partial y}, \tag{2}$$

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0, \tag{3}$$

is considered on the two-dimensional domain $\Omega(x, y)$. Here, x and y are coordinates in the horizontal and vertical direction, where v_x, and v_y are dimensionless velocity components in the x and y directions, respectively, p is the pressure and Re is the Reynolds number. Eqs. (1) and (2) comprise the momentum equations and Eq. (3) is the continuity equation. The Reynolds number is computed according to

$$\text{Re} = \frac{\rho u L}{\mu}, \tag{4}$$

where ρ is the fluid density, u is a characteristic velocity, L is the characteristic length taken to be the airfoil chord length, and μ is the fluid kinematic viscosity. In this model, we consider air at T = 293K at atmospheric pressure yielding a viscosity $\mu = 1.814 \times 10^{-5}$ kg/m-s and density ρ = 1.204 kg/m^3. The characteristic length and velocity are L = 2 m and u = 7 m/s, respectively, yielding Re $= 9.3 \times 10^5$.

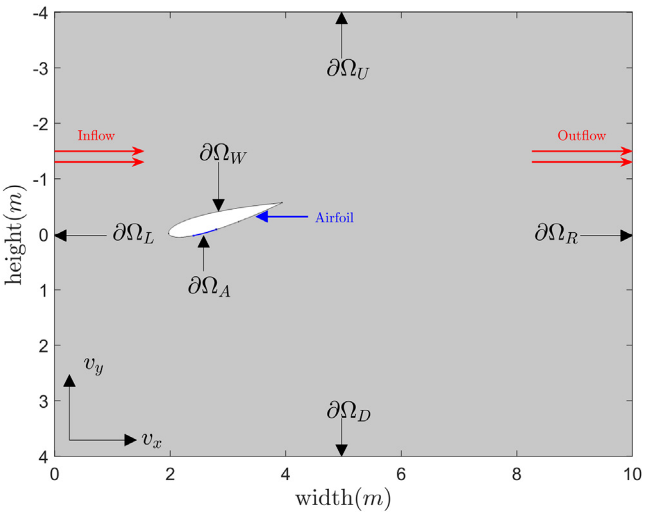

Fig. 1. Geometry of the airfoil model, Here, air flows in from the left boundary and leaves through the right boundary. Boundary conditions are summarized in Table 1.

Table 1
Boundary conditions associated with the airfoil geometry shown in Fig. 1. $n(x) \in \mathbb{R}^2$ denotes the unit normal to the surface boundary, the dot denotes the dot product, and the × denotes the cross product. In the development of a reduced order model, we consider $u_1(t)$ and $u_2(t)$ to be inputs.

$n(\partial\Omega_{\text{A}}) \cdot f(t, \partial\Omega_{\text{A}}) = u_2(t)$	Dirichlet Control Input
$n(\partial\Omega_{\text{A}}) \times f(t, \partial\Omega_{\text{A}}) = 0$	Dirichlet Control Input
$v_x(\partial\Omega_{\text{W}}, t) = 0$	Dirichlet Flow Conditions (Wall)
$v_y(\partial\Omega_{\text{W}}, t) = 0$	Dirichlet Flow Conditions (Wall)
$v_x(\partial\Omega_{\text{U}}, t) = v_x(\partial\Omega_{\text{D}}, t)$	v_x Periodic Flow Conditions
$v_y(\partial\Omega_{\text{U}}, t) = v_y(\partial\Omega_{\text{D}}, t)$	v_y Periodic Flow Conditions
$p(\partial\Omega_{\text{U}}, t) = p(\partial\Omega_{\text{D}}, t)$	p Periodic Flow Conditions
$v_x(\partial\Omega_{\text{L}}, t) = u_1(t)$	v_x Inflow Conditions
$v_y(\partial\Omega_{\text{L}}, t) = 0$	v_y Inflow Conditions
$p(\partial\Omega_{\text{R}}, t) = 0$	p Outflow Conditions

Here, a rectangular area with a width of 10 m and a height of 8 m is considered as the domain of whole geometry. Six distinct boundaries are considered for the domain Ω taking $\partial\Omega = \partial\Omega_{\text{A}} \cup \partial\Omega_{\text{W}} \cup \partial\Omega_{\text{L}} \cup \partial\Omega_{\text{R}} \cup \partial\Omega_{\text{U}} \cup \partial\Omega_{\text{D}}$. The domain is shown in Fig. 1. Here, $\partial\Omega_{\text{A}}$ is the boundary on the bottom surface of the airfoil where control input is applied, $\partial\Omega_{\text{W}}$ is the remainder of the airfoil where no-slip boundary conditions are considered. $\partial\Omega_{\text{L}}$ is left of the airfoil where the inflow velocity condition is specified, $\partial\Omega_{\text{R}}$ is to the right of the airfoil where the outflow velocity condition is specified. $\partial\Omega_{\text{U}}$ and $\partial\Omega_{\text{D}}$ are the rectangle upper and lower boundaries where periodic flow conditions are used. All the boundaries are clearly depicted in Fig. 1 with boundary conditions summarized in Table 1. Note that $n(x) \in \mathbb{R}^2$ denotes the unit normal to the surface boundary at the location x and $f(t, x) \equiv \left[v_x(t, x) \quad v_y(t, x)\right]^T$. Numerical simulation is performed on a 12,014 node finite element mesh using COMSOL 5.6 with the turbulent flow k-ϵ model.

2.2. Flow separation

Boundary layer detachment from a solid surface [32–35], is commonly referred to as flow separation. Flow separation happens in a range of applications, with airfoils at high angles of attack being one of the most common instances. Separation can be caused, for instance, by adverse pressure gradients [36,37], or geometrical aberration on the flow surface [38,39]. Flow separation generally results in eddies and vortices, yielding a pressure differential between an object's front and back surfaces and ultimately resulting in reduced lift and elevated drag [40]. The design of aerodynamic and hydrodynamic surface

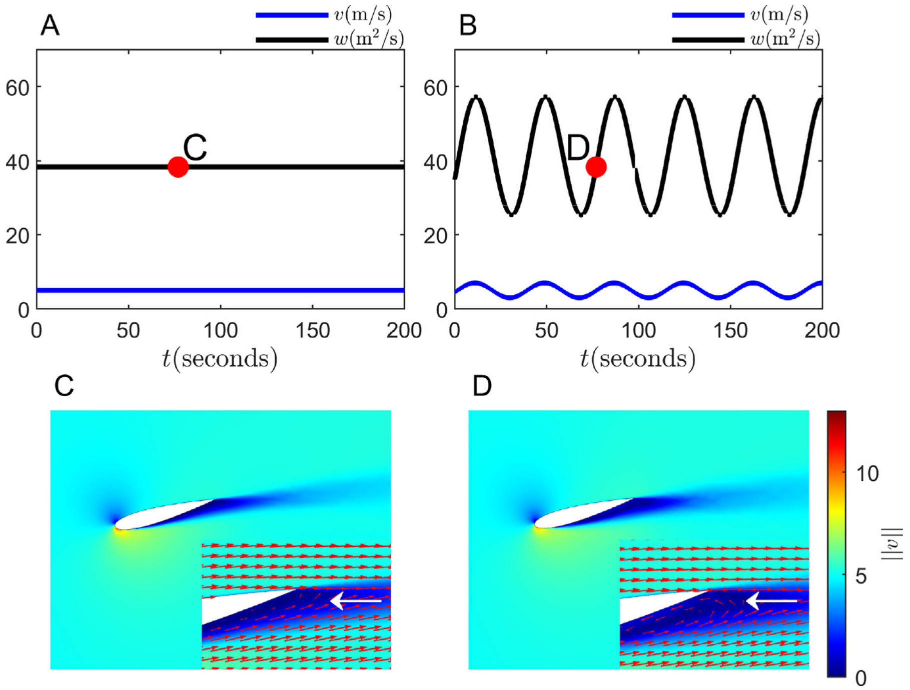

Fig. 2. Two representative simulations of the airfoil model taking $u_2(t) = 0$. Panel A shows the vorticity, $w(t)$ (black line) in response to constant flow of $u_1(t) = 5$m/s (blue line). Panel C shows a snapshot of the corresponding steady state flow; arrows on the inset indicate the direction of the fluid flow near the trailing edge of the airfoil and the white arrow indicates that there is no recirculating motion at the trailing edge of the airfoil. Panels B and D show the same information when taking $u_1(t) = 5 + 2\sin(2\pi t/50)$m/s. The inset from the snapshot in panel D clearly demonstrates a recirculating flow (indicated by the white arrow) near the trailing edge of the airfoil. Note that the instantaneous value of u_1 is identical for the snapshots from panels C and D.

contours can delay flow separation [41]. As shown in Fig. 2, flow separation develops in the model described in Section 2 in response to time-varying inflow in the boundary $\partial\Omega_L$ as demonstrated by the circulation that develops near the trailing edge of the airfoil. This recirculation does not develop when a comparable constant inlet velocity is applied.

2.3. General overview of the control objective

Control of flow separation is an important consideration for improving aerodynamic performance. Both active and passive flow management strategies have been used to avoid or mitigate separation-induced performance loss [42]. Previously, a variety of strategies have been used to control flow separation including periodic excitation [43] and acoustic excitation [44]. Other strategies have included the used of synthetic jets [45], plasma actuators [46], and electromagnetic forces [47].

In this work, we consider the vorticity as an observable of interest related to the formation of flow separation. The vorticity, $w(t)$ is defined as a surface integral of the curl of the flow velocity over the entire domain, i.e.,

$$w(t) = \iint_{\Omega} (\nabla \times v) \cdot dxdy. \tag{5}$$

In general, more pronounced flow separation will yield increased flow recirculation and consequently a larger value of the vorticity. As such, our control objective will be to maintain the vorticity at a constant low level in the presence of a time-varying inflow on the boundary $\partial\Omega_L$. To do this, we will consider the Dirichlet input at $\partial\Omega_A$ as a control input. The inflow velocity Ω_L will be considered as a time-varying disturbance.

We emphasize that the control objective described here is considered as a prototype problem. It characterizes a situation where we have limited access to observables which are not perfect representations of the behavior that we wish to control. In this case we assume that we only have access to a single observable which gives an aggregate view of the curl evaluated over the entire flow field, and want to use this information to influence the behavior that occurs at the boundary layer. In the model identification strategy and subsequent control design considered in this work, other observables could be used instead (for instance, pointwise velocity or pressure measurements).

As a first step for approaching this control problem, we will employ a reduced order modeling approach that considers coordinates that capture level sets of Koopman eigenfunctions [14,21] to yield an approximation for the dynamics of w of the form:

$$\dot{w} = F(w, u_1, u_2), \tag{6}$$

where $w \in \mathbb{R}$ is the vorticity observable defined according to Eqs. (5), $u_1 \in \mathbb{R}$ and $u_2 \in \mathbb{R}$ are time-varying inputs, and F sets the model dynamics. In principle, this could be done by directly taking the time derivative of Eq. (5) while considering the relationships on the x and y velocities governed by Eqs. (1)–(3). Of course, such an approach would generally require direct knowledge of the velocity and pressure as a function of time and would likely be too complicated to consider from a control theoretic perspective. In Section 3 below, we discuss the mathematical details underlying our approach; subsequent strategies for data-driven model identification are given in Section 4.

3. Reduced order modeling and control using an adaptive isostable coordinate reduction approach

3.1. The isostable coordinate basis

In order to identify a reduced order dynamical model for the vorticity from Eq. (5), we will consider a reduced order modeling approach using an adaptive parameter set as suggested by [27] (cf., [26]). Following the derivation from [27], we consider a general partial differential equation of the form

$$\frac{\partial}{\partial t} X(r,t) = F(X(r,t), u(t)), \tag{7}$$

where $X(r,t) \in \mathbb{R}^n$ is the state, r is some location on the domain Ω, F sets the dynamics, and $u \in \mathbb{R}^M$ is an input. Suppose that for some $u \in U \subset \mathbb{R}^M$ that Eq. (7) has a stationary solution $X_{\rm ss}(r,u)$, i.e., with $F(X_{ss}(r,u),u) = 0$. Suppose also that for all $u \in U$, the operator associated with the local linearization $J(r,u) \equiv \nabla F|_{X_{ss},u}$ exists and has a discrete spectrum. In this problem formulation we will also consider an observable $w \in \mathbb{R}$ defined according to

$$w(t) = G(X(r,t)), \tag{8}$$

where G maps the state to the observable. Near the steady state, solutions can be written as

$$\phi(t,X_0,u) - X_{\rm ss}(r,u) = \sum_{j=1}^{\infty} \psi_j(X_0,u)\zeta_j(r,u)e^{\lambda_j(u)t}, \tag{9}$$

where ϕ represents the flow of the vector field (7) taking u to be constant, X_0 is an initial condition, $\zeta_j(r,u)$ is an eigenfunction with associated decay rate $\lambda_j(u)$ of $J(r,u)$, and $\psi_j(X_0,u)$ maps the state to the eigenfunction basis. Eq. (9) is a linear, lowest-order approximation to the general expansion of the state of (7) using the Koopman eigenfunctions; the general Koopman expansion includes higher-order terms [12,48]. Specifically, each eigenfunction $\zeta_j(r,u)$ is a principal Koopman eigenfunction. Consequently, the coordinates $\psi_j(X,u)$ represent level sets of these eigenfunctions and can provide a reduced order coordinate basis that can be used to represent the dynamical behavior. To do so, we sort each $\lambda_1(u), \lambda_2(u), \ldots$ so that $0 > \max_{u\in U}(\mathrm{Re}(\lambda_j(u))) \geq \max_{u\in U}(\mathrm{Re}(\lambda_{j+1}(u)))$. Note that for a stable stationary solution, $\lambda_1(u)$ would correspond to the eigenfunction with the slowest decay rate. To a linear approximation, i.e., close to the stationary solution, one can write

$$\Delta X(r,u) \equiv X(r) - X_{\rm ss}(r,u) = \sum_{j=1}^{\infty} \psi_j(X,u)\zeta_j(r,u). \tag{10}$$

Here, $\psi_j(X,u) = \psi_j(X_0,u)e^{\lambda_j(u)t}$ because $\psi_j(X,u)$ is a Koopman eigenfunction with eigenvalue $\lambda_j(u)$.

An isostable coordinate reduction only considers the β slowest decaying eigenmodes

$$\Delta X(r,u) \approx \sum_{j=1}^{\beta} \psi_j(X,u)\zeta_j(r,u). \tag{11}$$

In many applications, inputs that excite modes with fast decay die out rapidly so that Eq. (11) can be used to provide a good approximation for Eq. (10) [23,49,50]. Note that (10) considers the local linearization near the stationary solution $X_{\rm ss}(r,u)$. Isostable coordinates can also be defined in the fully nonlinear basin of attraction of $X_{\rm ss}(r,u)$ as level sets of Koopman eigenfunctions [14,21].

3.2. Adaptive isostable reduction

Following the general formulation proposed in [26], consider a rewritten version of (7)

$$\frac{\partial}{\partial t} X(r,t) = F(X,p) + U_e(X,u,p), \tag{12}$$

where

$$\begin{aligned} U_e(X,u,p) &= F(X,u) - F(X,p) \\ &= \sum_{j=1}^{M} \frac{\partial F}{\partial u_j}(u_j - p_j) + O(\|u-p\|^2), \end{aligned} \tag{13}$$

for $j = 1,\ldots,\beta$ where partial derivatives are evaluated at X taking $u = p$, $F(X,p)$ represents the dynamics that results when taking $u = p$ to be a constant, and U_e represents the effect of a time-varying input. The term $p \in \mathbb{R}^M$ is a time-varying parameter that will be used in the subsequent isostable reduction to mitigate the influence of truncation errors caused by nonlinear terms of (7). For convenience of notation above, dependence on r has been suppressed. Notice that Eq. (12) is identical to Eq. (7). Allowing for p from (12) to be nonstatic, changing to isostable coordinates using the chain rule for functional derivatives yields

$$\begin{aligned} \frac{d\psi_j}{dt} &= \left\langle \nabla\psi_j, \frac{\partial X}{\partial t} \right\rangle + \sum_{i=1}^{M} \frac{\partial \psi_j}{\partial p_i}\frac{dp_i}{dt} \\ &= \left\langle \nabla\psi_j, F(X,p) + U_e(X,u,p) \right\rangle - \sum_{i=1}^{M} \left\langle \nabla\psi_i, \frac{\partial X_{\rm ss}}{\partial p_i} \right\rangle \frac{dp_i}{dt} \\ &\approx \lambda_j(p)\psi_j + \sum_{i=1}^{M} \left\langle \nabla\psi_j, \frac{\partial F}{\partial u_i} \right\rangle (u_i - p_i) - \sum_{i=1}^{M} \left\langle \nabla\psi_i, \frac{\partial X_{\rm ss}}{\partial p_i} \right\rangle \frac{dp_i}{dt}, \end{aligned} \tag{14}$$

where $\nabla\psi_j$ is the gradient of the isostable coordinate ψ_j with respect to the state evaluated at $X_{\rm ss}(u)$, $\langle \cdot,\cdot \rangle$ denotes the inner product, $\partial X_{\rm ss}/\partial p_i = \lim_{a\to 0}[X_{\rm ss}(r,u)|_{p_i+a} - X_{\rm ss}(r,u)|_{p_i}]/a$, and the $O(\|u-p\|^2)$ terms have been neglected. In the final line above, the relation $\langle \nabla\psi_j, F(X,p) \rangle = \lambda_j(p)\psi_j$ is used and holds because isostable coordinates decay according to the relation $d\psi_j/dt = \lambda_j\psi_j$ for all j. Using the state approximation from (11), to leading order accuracy in the basis of isostable coordinates, the observable from (8) can be approximated according to

$$\begin{aligned} w &= G(X(r,p)) \\ &\approx G\left(X_{\rm ss}(r,p) + \sum_{j=1}^{\beta} \psi_j(X,p)\zeta_j(r,p) \right) \\ &\approx w_{\rm ss}(p) + \left\langle \frac{\partial G}{\partial X}, \sum_{j=1}^{\beta} \psi_j(X,p)\zeta_j(r,p) \right\rangle, \end{aligned} \tag{15}$$

where $w_{\rm ss}(p) = G(X_{\rm ss}(r,p))$. Considering Eqs. (14) and (15), the reduced order isostable-based model can be represented according to

$$\begin{aligned} \dot{\Psi} &= \Lambda(p)\Psi + I(p)(u-p) + B(p)\dot{p}, \\ w &= w_{\rm ss}(p) + \sum_{j=1}^{\beta} \eta_j(p)\psi_j, \end{aligned} \tag{16}$$

where $\Psi = \begin{bmatrix} \psi_1 & \ldots & \psi_\beta \end{bmatrix}$, $\Lambda(p) = \mathrm{diag}(\lambda_1(p),\ldots,\lambda_\beta(p)) \in \mathbb{C}^{\beta\times\beta}$, the ith row and jth column of $I(p) \in \mathbb{C}^{\beta\times M}$ is given by $\langle \nabla\psi_i, \partial F/\partial u_j \rangle$, the ith row and jth column of $B(p) \in \mathbb{C}^{\beta\times M}$ is given by $-\langle \nabla\psi_i, \partial X_{\rm ss}/\partial p_j \rangle$, and $\eta_j(p) = \langle \partial G/\partial X, \zeta_j(r,p) \rangle$.

Recall that Eq. (16) results after truncating $O(\psi_j^2)$ terms. As such, its accuracy begins to degrade as $\|\Psi\|$ increases. Noting that the parameter set p is free, one can mitigate these truncation errors by choosing an appropriate update rule

$$\dot{p} = G_p(\Psi, p), \tag{17}$$

to keep $\|\Psi\|$ small. General strategies for design of an appropriate update rule for p are discussed in [26,29]. Eqs. (16) with the parameter update rule (17) comprise the full form of the adaptive isostable reduction.

4. Data-driven model identification using the adaptive isostable reduction

We will investigate the model order reduction framework that yields Eqs. (16) with parameter update rule (17) in the context of the flow separation problem discussed in Section 2.2. As discussed in [27], the necessary terms of the reduced order Eq. (16) can be computed by finding the eigenfunctions of the operator $J(r,u) \equiv \nabla F(X_{\rm ss}(r,p),p)$ and its adjoint. The complexity of our domain precludes this computational approach and instead we consider a data-driven approximation.

To aid in this task, note that there is a direct relationship between $I(p)$ and $B(p)$. To see this, consider the response to a constant input $u = u + e_j\Delta u$ with $p = u$ held constant where e_j is the jth member of the standard unit basis. The steady state solution for each isostable coordinate is

$$\psi_{i,\rm ss} = -\frac{I_{i,j}(p)\Delta u}{\lambda_i(p)}, \tag{18}$$

for $i = 1, \dots, \beta$ where $I_{i,j}(p)$ is the ith row and jth column of $I(p)$. To linear order of accuracy considering Eq. (11), the steady state output in response to this static input is

$$X(r) = X_{\rm ss}(r,u) - \sum_{i=1}^{\beta} \frac{\zeta_i(r,p) I_{i,j}(p) \Delta u}{\lambda_i(p)}. \tag{19}$$

Considering the definition of $B(p)$ given below Eq. (16) and noting that $\langle \nabla \psi_i, \zeta_j \rangle = 1$ when $i = j$ and zero otherwise, starting with Eq. (19) it is straightforward to show that

$$I(p) = \Lambda(p) B(p). \tag{20}$$

As such, the only unknowns in Eq. (16) are $B(p)$, $\Lambda(p)$ and the collection $\eta_1(p), \dots, \eta_\beta(p)$. To infer these unknown equations, we adapt a model identification strategy proposed in [51].

4.1. Model identification procedure

In the context of the adaptive isostable reduction strategy, we let $X(r,t)$ be the velocity and pressure fields that govern the evolution of the Navier–Stokes Equations from (1)–(3) with geometry and boundary conditions shown in Fig. 1 and Table 1, respectively. We consider the two inputs $u_1(t) = w(\partial\Omega_L, t)$ and $u_2(t) = n(\partial\Omega_{\rm A}) \cdot f(t, \partial\Omega_{\rm A})$ as described in Table 1 so that $u \in \mathbb{R}^2$. Our observable is taken to be the vorticity defined according to Eq. (5). The model identification procedure below is adapted from [28].

Step (1) Choose some $u_{\rm nom} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ and $u_\Delta = \begin{bmatrix} (u_1 + \Delta u_1) & u_2 \end{bmatrix}^T$ so that both $u_{\rm nom}$ and u_Δ are taken from the subset $U \subset \mathbb{R}^2$ for which the Navier–Stokes equations have a stationary solution $X_{\rm ss}$. Here Δu_1 should be small. Apply the constant input $u = u_\Delta$ until the steady state solution $X_{\rm ss}(r, u_\Delta)$ is achieved with corresponding output $w_{\rm ss}(u_\Delta)$.

Step (2) At $t = t_0$, increment the input so that $u = u_{\rm nom}$, i.e., so that $u = u_\Delta$ when $t < t_0$ and $u = u_{\rm nom}$ for $t \geq t_0$. Record $w(t)$ during the recovery to its new steady state value $w_{\rm ss}(u_{\rm nom})$.

Step (3) Repeat Steps 1 and 2 taking $u_\Delta = \begin{bmatrix} (u_1 - \Delta u_1) & u_2 \end{bmatrix}^T$, $u_\Delta = \begin{bmatrix} u_1 & (u_2 + \Delta u_2) \end{bmatrix}^T$, and $u_\Delta = \begin{bmatrix} u_1 & (u_2 - \Delta u_2) \end{bmatrix}^T$. Note that Δu_2 should also be small.

Step (4) The data from Steps 1–3 can be used to identify the necessary dimension of Ψ, and the associated values of $\lambda(u)$, $B(u)$ and each $\eta_1, \dots, \eta_\beta$. In order to infer the terms associated with multiple isostable coordinates, we employ a delayed embedding approach [52,53], splitting each recording $w(t) - w_{\rm ss}(u_{\rm nom})$ from Step 2 into separate intervals $\hat{w}_t, \hat{w}_{2t}, \dots$ where $\hat{w}_t$ contains data recorded over the interval $t = [t, t+T)$ that are discretized by a timestep Δt. See Fig. 4 for a description of this delayed embedding process.

Step (5) Arrange the delayed embeddings from Step 4 into a matrix $Y \in \mathbb{R}^{a,b}$ where $a = \frac{T}{\Delta t}$ is the length of each delayed embedding and b is the total number of delayed embeddings. Proper orthogonal decomposition (POD) [3,6] is performed by finding the eigenvalues and eigenvectors of the matrix YY^T which will be denoted $\lambda_j^{\rm POD}(u_{\rm nom})$ and $\xi_j(u_{\rm nom})$, respectively. Note that while $w(t) \in \mathbb{R}$, the delay embeddings lift the observable space to a higher dimension, making the POD useful here to reduce dimensionality. POD eigenvectors with the largest corresponding POD eigenvalues capture more of the temporal fluctuations in the data. Here, we only consider a single POD mode and truncate the rest, however, it is straightforward to implement the fitting strategy when considering multiple POD modes.

Step (6) The coefficient μ_1 of the POD basis for a given embedding can be determined according to

$$\mu_1(t) = \xi_1^T(u_{\rm nom}) \hat{w}_t. \tag{21}$$

A least squares fitting approach is used to infer the relationship $\mu_1(t+T) = A\mu_1(t)$. This is accomplished by taking pairs of measurements, arranging them into matrices $Z_1 = \begin{bmatrix} \mu_1(t_1) & \dots & \mu_1(t_d) \end{bmatrix}$ and $Z_2 = \begin{bmatrix} \mu_1(t_1+T) & \dots & \mu_1(t_d+T) \end{bmatrix}$, and taking $A(u_{\rm nom}) = Z_2 Z_1^\dagger$ where $\dagger$ denotes the Moore–Penrose pseudoinverse. Note that Step 6 can be viewed as DMD applied to the time series $\mu_1(t)$ [1]. Here μ_1 is scalar so that $A(u_{\rm nom})$ is assumed to give an approximation of a single, representative Koopman eigenvalue. Other strategies such as Hankel DMD [16,53] could be applied to give a similar result for this step.

Step (7) The estimate $A(u_{\rm nom})$ from Step 6 gives a finite time mapping from $\mu_1(t)$ to $\mu_1(t+T)$ when taking $u = u_{\rm nom}$ to be a constant. In continuous time, this relationship is approximated by

$$\begin{aligned} \dot{\psi}_1 &= \lambda_1(u_{\rm nom})\psi_1, \\ w(t) &= w_{\rm ss}(u_{\rm nom}) + \eta_1(u_{\rm nom})\psi_1, \end{aligned} \tag{22}$$

where $\lambda(u_{\rm nom}) = \log(A(u_{\rm nom}))/T$, $\eta_1(u_{\rm nom}) = \xi_1^T(u_{\rm nom}) e_1$, and e_1 is the first component of the standard unit basis and is used to select for the component of ξ_1 (obtained from the delayed embedding) that corresponds to the state at time t. Note that if multiple POD modes were considered, a slightly different transformation would need to be used, for instance, the decay rates in (22) would be given by the eigenvalues of $\log(A(u_{\rm nom}))/T$.

Step (8) Comparing to the form of the model to be fit from Eq. (16), Eq. (22) gives the unperturbed evolution of the reduced order system. The terms of $B(p)$ can be inferred by noting that the initial stepwise shift from $u_\Delta = \begin{bmatrix} (u_1 + \Delta u_1) & u_2 \end{bmatrix}$ to $u_{\rm nom} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$ occurring at t_0 from Step 1 will yield a shift in the isostable coordinate $\psi_1(t_0) = -B_1(u_{\rm nom})/\Delta u_1$, where $B_1(u)$ is the first component of $B(u)$. Because ψ_1 is identical to μ_1, one can use Eq. (21) to write

$$\psi_1(t_0) = \exp(-\lambda_1(t - t_0)) \xi_1^T(u_{\rm nom}) \hat{w}_t, \tag{23}$$

for some $\hat{w}_t$ obtained during the relaxation to the steady state from Step 4. Samples of $B_1(u_{\rm nom})$ can be averaged over multiple trials. Data from trials for which $u_\Delta = \begin{bmatrix} u_1 & (u_2 + \Delta u_2) \end{bmatrix}^T$ can be used to obtain samples of $B_2(u_{\rm nom})$. Once $B(p)$ is obtained, $I(p)$ can be obtained according to the relation (20).

Step (9) Steps 1–8 yields a data-driven estimate for all terms in (16) for a static value of $p = u_{\rm nom}$. These steps can be repeated for an adequate sampling of $u_{\rm nom} \in U$ to allow for interpolation. Note that when the data is noisy, it may be necessary to obtain multiple estimates of the terms $\Lambda(u), B(u)$, and each $\eta_j(u)$ in order to average out the effects of noise. In the application considered in this work, there is no such noise or uncertainty.

4.2. Model fitting using data collected from numerical simulations

We implement the model identification procedure described in Section 4.1 taking the input $u = \begin{bmatrix} u_1(t) & u_2(t) \end{bmatrix}$, i.e., that describes the time-varying boundary conditions from Table 1. We consider the vorticity, $w(t)$ defined according to Eq. (5) to be the sole observable. A single isostable coordinate, ψ_1 is considered in the reduced order model of the form (16). Correspondingly, the terms $\lambda_1(u)$, $I^T(u) = \begin{bmatrix} I_1(u) & I_2(u) \end{bmatrix}$, $B^T(u) = \begin{bmatrix} B_1(u) & B_2(u) \end{bmatrix}$, $w_{\rm ss}(u)$, and $\eta_1(u)$ must be inferred from data. Fig. 3 shows a subset of the data used in the model fitting procedure. Inputs $u_1 \in [2.5, 7.5]$ and $u_2 \in [-0.5, 0.5]$ are changed in stepwise increments, the relaxation to the new steady state $w_{\rm ss}(u_1, u_2)$ is observed, and this data is used to fit the model of the form (16). Fig. 3 shows simulation data when sweeping through allowable values of u_1 and u_2, primarily holding u_1 constant while incrementing u_2. This

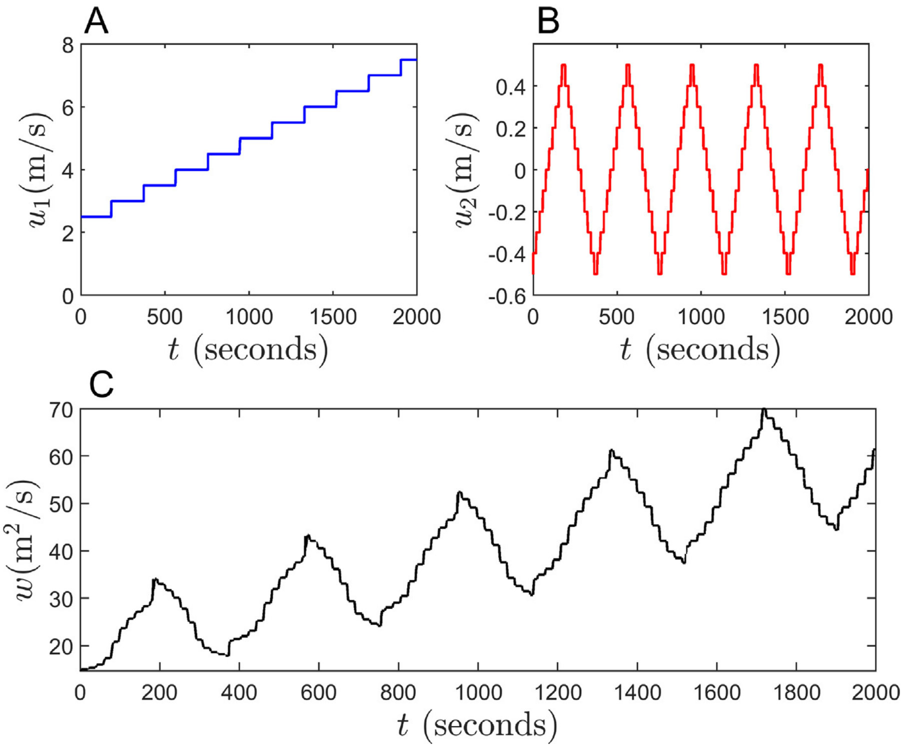

Fig. 3. This figure shows a subset of the data used for fitting the model of the form (16) as required by steps 1–3 of the model fitting procedure from Section 4.1. Panels A and B show the timecourse of u_1 and u_2. The vorticity (i.e., the observable) is shown in Panel C during the course of this simulation. This data provides a sweep through allowable parameters with stepwise increments while primarily making changes to u_2.

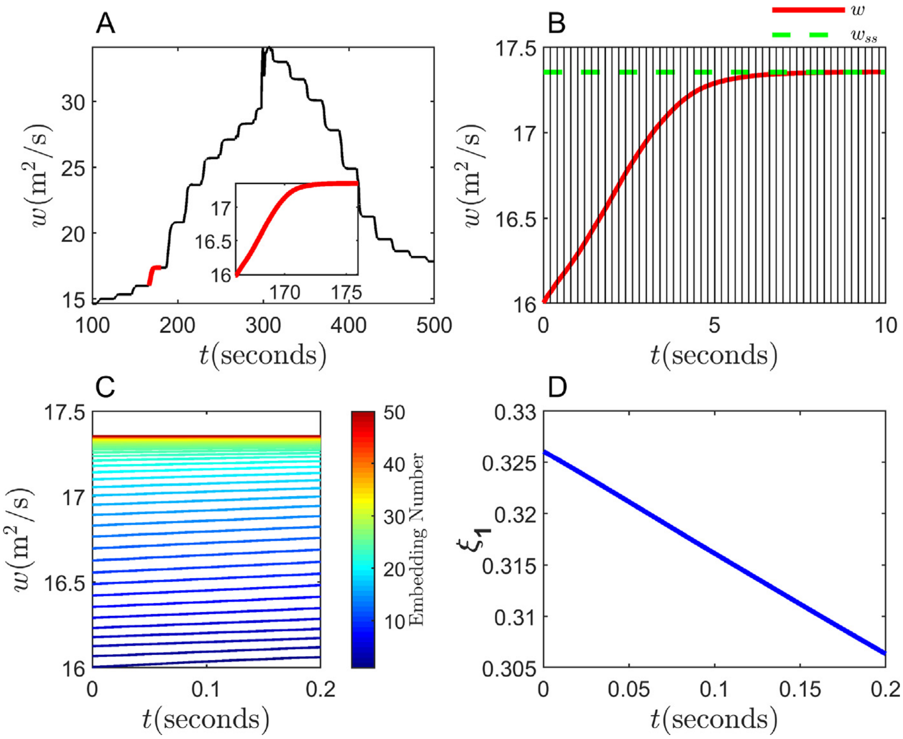

Fig. 4. This figure describes the delayed embedding and subsequent least squares fitting procedure as required by steps 4 and 5 of the model fitting procedure from Section 4.1. Panel A shows highlights the vorticity following a stepwise increase in u_2. This recording is split into 50 delayed embeddings as shown in Panel B. These embeddings are plotted on the same axis in panel C. Panel D shows the first POD mode associated with this data as defined in step 5 of the model fitting procedure.

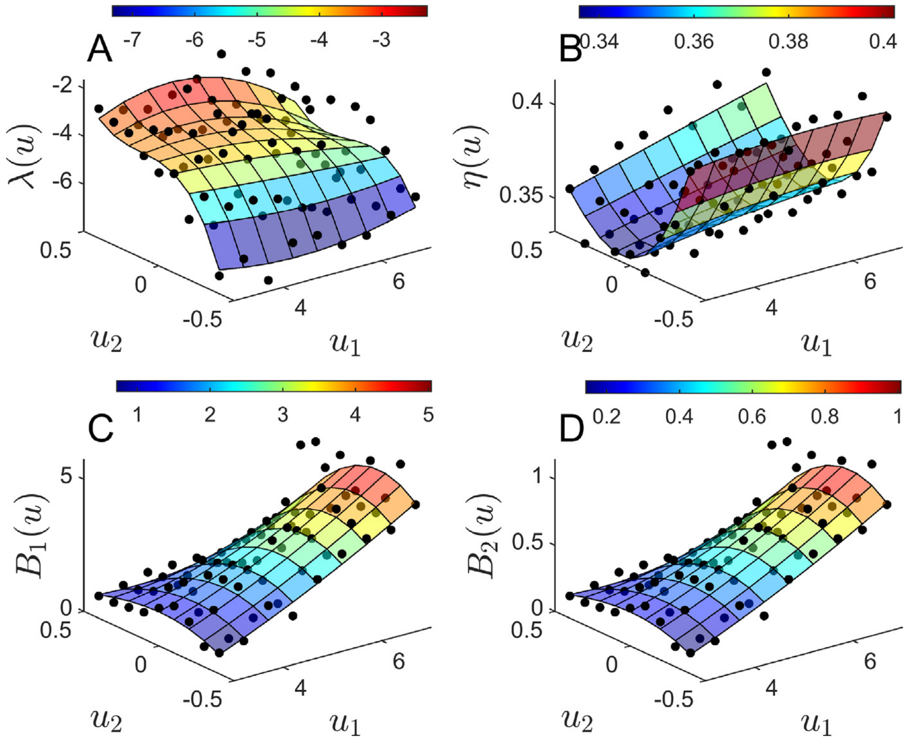

Fig. 5. For the allowable set of inputs $u = [u_1, u_2]$, colormaps in panels A, B, C, and D show the terms $\lambda(u)$, $\eta(u)$ $B_1(u)$, and $B_2(u)$ that comprise Eq. (16). These terms are identified as part of steps 6–8 of the model fitting procedure from Section 4.1. Dots represent discrete measurements and the surface plots are fitted to the data. Note that the term $I(p)$ is determined from the relation (20).

information is used to infer $B_2(u)$. Additional data (not shown) holds p_2 constant while incrementing p_1 and is used to infer $B_1(p)$. Fig. 4 gives a visual representation of how the vorticity data is used in the model fitting procedure. Panel A highlights the vorticity measurement where $u_{\text{nom}} = [2.5, 0]^T$ m/s and $u_\Delta = [2.5, -0.1]$. The ensuing 10 s of decay towards the new fixed point is highlighted in panel B. As described in Step 4 of the procedure from Section 4.1, this signal is split into 50 separate delayed embeddings, each lasting $T = 0.2$ seconds. Each of these embeddings is plotted in panel C, and POD is subsequently applied to the data as described in Step 5 of the model fitting procedure. Steps 6 through 8 are implemented to infer the associated reduced order dynamics. For each value of u_{nom} considered, the minimum and average value of $\lambda_1^{\text{POD}} / \sum_{j=1}^{a} \lambda_j^{\text{POD}}$ is 0.9839 and 0.9945, respectively, indicating that one POD mode provides an adequate representation for the data from the delayed embeddings. The resulting inferred terms of the reduced order models are shown in Fig. 5. The dots represent terms from individual trials that are computed from Steps 7 and 8 of the model fitting procedure from Section 4.1. The surface plots provide a polynomial fit of order two and three in the variables u_1 and u_2, respectively.

5. Results

5.1. Reduced order model validation

Terms of the reduced order model (16) are computed as described in Section 4. The resulting model takes two inputs u_1 and u_2. As such there are two adaptive parameters $p = [p_1, p_2]$ in the adaptive isostable reduction. The parameter update rule (i.e., G_p from Eq. (17)) is chosen to be

$$\dot{p_1} = \dot{u}_1,$$
$$\dot{p_2} = -10\psi_1 B_2(p). \tag{24}$$

Here, p_1 tracks u_1, and p_2 is adjusted in order to limit the magnitude of ψ_1. With this choice of the update rule, the adaptive isostable reduction governed by Eqs. (16) and (17) becomes

$$\dot{\psi}_1 = (\lambda_1(p) - 10(B_2(p))^2)\psi_1 + I_2(p)(u_2 - p_2) + B_1(p)\dot{u}_1,$$
$$\dot{p}_1 = \dot{u_1},$$
$$\dot{p_2} = -10\psi_1 B_2(p), \tag{25}$$

with output

$$w(t) = w_{\text{ss}}(p) + \eta_1(p)\psi_1. \tag{26}$$

In Eq. (25) above, the parameter p_2 is modulated in order to limit the magnitude of ψ_1 while the contributions from the terms $I_2(p)(u_2 - p_2)$ and $B_1(p)\dot{u}_1$ will generally drive ψ_1 to larger magnitude values. Note that $u_1 = p_1$ in this formulation so that $I_1(p)$ does not appear in Eq. (25). The reduced order model described by Eq. (25) with output (26) is compared to full model simulations of Eqs. (1)–(3) with output (5) using purely sinusoidal inputs $u_1(t) = 5 + \sin(2\pi/80)$ m/s and $u_2(t) = 5 + \sin(2\pi/50)$ m/s. Results are shown in Fig. 6. Panel A (resp., B) show $u_1(t)$ and $p_1(t)$ (resp., $u_2(t)$ and $p_2(t)$) over the course of the simulation. Panels C shows the time course of ψ_1 indicating that the parameter update rule keeps the isostable coordinates small as desired. Panel D shows the error between the true model output and the reduced order model output. Panel E shows the model output for the full and reduced order models. Note that while the inputs u_1 and u_2 are purely sinusoidal, we emphasize that the model was inferred using step inputs and not sinusoidal inputs, i.e., we are not testing on the same inputs used for training. For additional comparison, we also implement the Hankel DMD algorithm [52] which applies the DMDc algorithm [18] taking the state to be a delay embedding of the system observable. Here a delay embedding of length 100 data points is used to define the state, with u_1 and u_2 comprising the input. The prediction when using the DMDc algorithm is shown in yellow for comparison and is substantially worse than the proposed strategy.

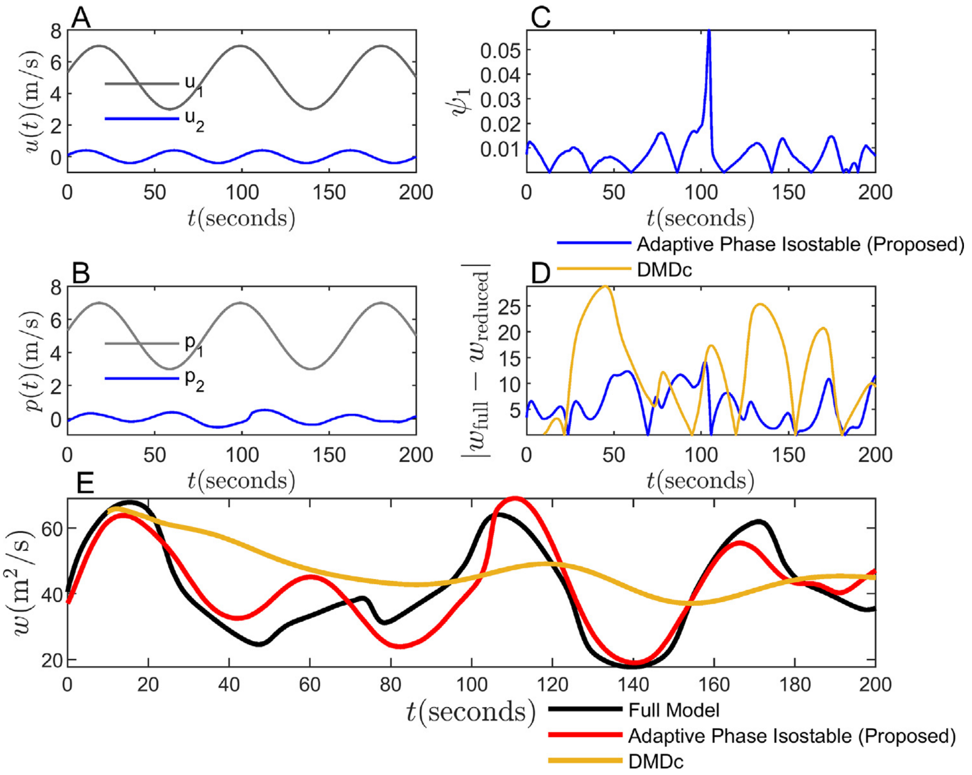

Fig. 6. Comparisons are shown among the full order model (Eqs. (1)–(3) with output (5)), reduced order model (Eq. (25) with output (26)) and DMDc output (Eq. (A.4)). Panel A shows the applied inputs u_1 and u_2. Panel B shows the adaptive parameters on p_1 and p_2. Panel C shows the resulting isostable coordinates of the reduced order model. Panel D shows the error between the full and reduced order models and full and DMDc models. Panel E gives a direct comparison among the full, reduced order model and DMDc outputs.

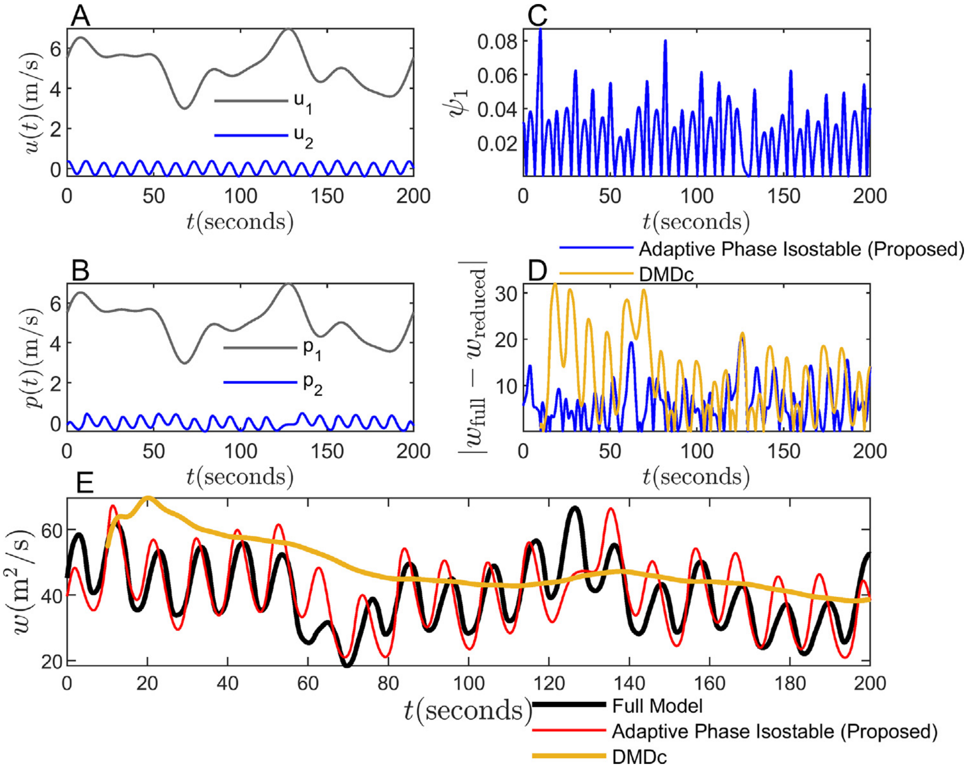

Fig. 7. Additional comparisons are shown among the full order model (Eqs. (1)–(3) with output (5)), reduced order model (Eq. (25) with output (26)) and DMDc output (Eq. (A.4)), using nonperiodic and more rapidly varying inputs (relative to the results from Fig. 6). Panel A shows the applied inputs u_1 and u_2. Panel B shows the adaptive parameters on p_1 and p_2. Panel C shows the resulting isostable coordinates of the reduced order model. Panel D shows the error between the full and reduced order models and full and DMDc models as a function of time. Panel E gives a direct comparison among the full, reduced order model DMDc outputs.

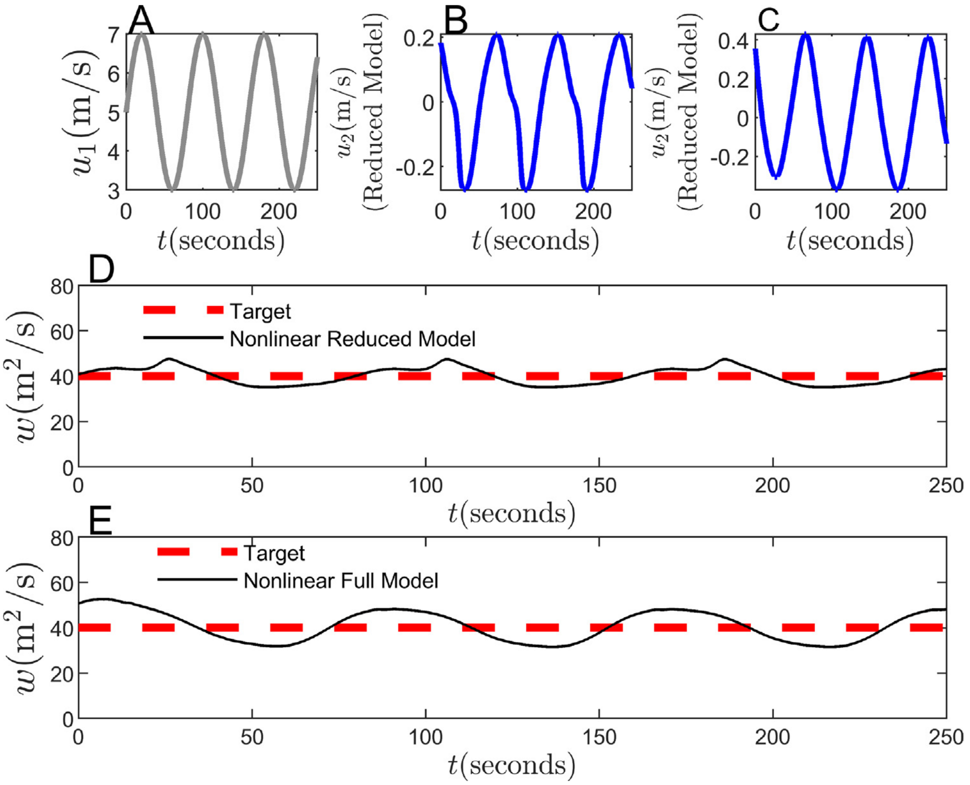

Fig. 8. Comparison of the PI controller (27) applied to both the full and nonlinear reduced order models. Panel A shows a sinusoidally varying input velocity, $u_1(t)$. The PI controller is set to have a constant target of 40 m^2/s for the vorticity measurements. The resulting input u_2 is shown in panels B and C when applying the PI controller when using the reduced and full order models, respectively. Panels D and E show the actual outputs from each simulation relative to the target value. The reduced order model provides a reliable representation of the model behavior when the PI controller is implemented.

Results for a second comparison between full and reduced order model simulations are shown in Fig. 7. Here, we take $u_1(t) = 1.1\sin(0.02\pi t){+}0.3\sin(0.08\pi t){+}0.7\sin(0.051\pi t)$ and $u_2(t) = 0.33\sin(0.195\pi t) - 0.02\sin(0.054\pi t) + 0.08\sin(0.051\pi t)$. In contrast to the previous illustration, u_2 varies slightly faster and both inputs are not periodic. Once again, panels A and B show the model inputs and adaptive parameters, respectively. Panels C and D show the isostable coordinate for the reduced order simulation and the difference between the full and reduced order models, respectively. Panel E provides a direct comparison between full and reduced order model outputs. Once again, DMDc is markedly worse than the proposed model identification strategy.

Considering the results in Figs. 6 and 7, the reduced order model given by Eqs. (25) and (26) performs well, especially considering that it requires only three state variables. When considering more rapidly varying inputs, as in the case of Fig. 7, The full and reduced order models do not always agree well, especially in moments for which $u_1(t)$ is near its maximum and minimum values. The primary reason for the discrepancy is that we are representing the dynamics with a single isostable coordinate and that the input is changing fast enough for other, unmodeled isostable coordinates to contribute to the dynamics. In principle, it would be possible to consider multiple isostable coordinates, however, this runs the risk of overfitting to the available data. Additionally, in a scenario with multiple isostable coordinates, G_p would need to be updated accordingly to keep their magnitudes low.

5.2. Comparison when using a proportional integral controller

As discussed in Section 2.3, our control objective for this prototype problem is to maintain a constant vorticity despite a time-varying inlet velocity $u_1(t)$. This will be accomplished using a proportional–integral (PI) controller that dictates the control input $u_2(t)$:

$$u_2(t) = k_p(y_{\text{targ}} - w(t)) + k_i \int_0^t (w_{\text{targ}} - w(t))dt. \tag{27}$$

Here, k_p and k_i are proportional and integral gains, respectively. The purpose of implementing this controller is two-fold: first, we illustrate that the reduced order model from Eqs. (25) and (26) can be used to choose appropriate gains when considering the full order model; second, we will illustrate that the reduced order and full order models have qualitatively similar behavior when the PI control is applied. To determine appropriate gains for the controller, the system is first linearized in reference to the steady state solution $\begin{bmatrix}\psi_{1,\text{ss}} & p_{1,\text{ss}} & p_{2,\text{ss}}\end{bmatrix} = \begin{bmatrix}0 & 5 & 0\end{bmatrix}$ that results when taking the constant inputs $\begin{bmatrix}u_1 & u_2\end{bmatrix} = \begin{bmatrix}5 & 0\end{bmatrix}$. The steady state vorticity is $w_{\text{ss}} = 40.76$ m^2/s. For this linearization, we consider u_2 as the control and let $\dot{u}_1$ be a disturbance. The linearized version of Eqs. (25) and (26) neglecting disturbances is

$$\frac{d}{dt}\begin{bmatrix}\Delta\psi_1\\ \Delta p_1\\ \Delta p_2\end{bmatrix} = \begin{bmatrix}\lambda_1(p) - 10B_2(p) & 0 & -I_2(p)\\ 0 & 0 & 0\\ -10B_2(p) & 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\psi_1\\ \Delta p_1\\ \Delta p_2\end{bmatrix} + \begin{bmatrix}I_2(p)\\ 0\\ 0\end{bmatrix}u_2(t),$$

$$\Delta w(t) = \begin{bmatrix}\eta_1(p) & \frac{\partial w_{\text{ss}}}{\partial p_1} & \frac{\partial w_{\text{ss}}}{\partial p_2}\end{bmatrix}\begin{bmatrix}\Delta\psi_1\\ \Delta p_1\\ \Delta p_2\end{bmatrix}. \tag{28}$$

where $\Delta\psi_1 = \psi_1 - \psi_{1,\text{ss}}$, $\Delta p_1 = p_1 - p_{1,\text{ss}}$, and $\Delta p_2 = p_2 - p_{2,\text{ss}}$ and $\lambda_1(p)$, $I_2(p)$, $B_1(p)$ and $B_2(p)$ are evaluated at $p_1 = 5$ m/s and $p_2 = 0$ m/s. The gains of the PI controller are obtained for the linearized system after the reduced order nonlinear model is linearized. Gains for the PI controller are chosen to be $k_p = 0.0240$ and $k_i = 0.0029$ in order to balance the trade off between overshoot and settling time for the linearized model. As an additional consideration, the reduced order model performs worse when using faster varying inputs (as displayed in results from Fig. 7); the gains are also chosen so that the resulting model does not yield rapid fluctuations in $p_2(t)$. When implementing the PI controller in the results below, the target value is set to 40 m^2/s.

Using the PI controller (27), comparisons between the full order model (Eqs. (1)–(3) with output (5)) and reduced order model (Eq. (25)

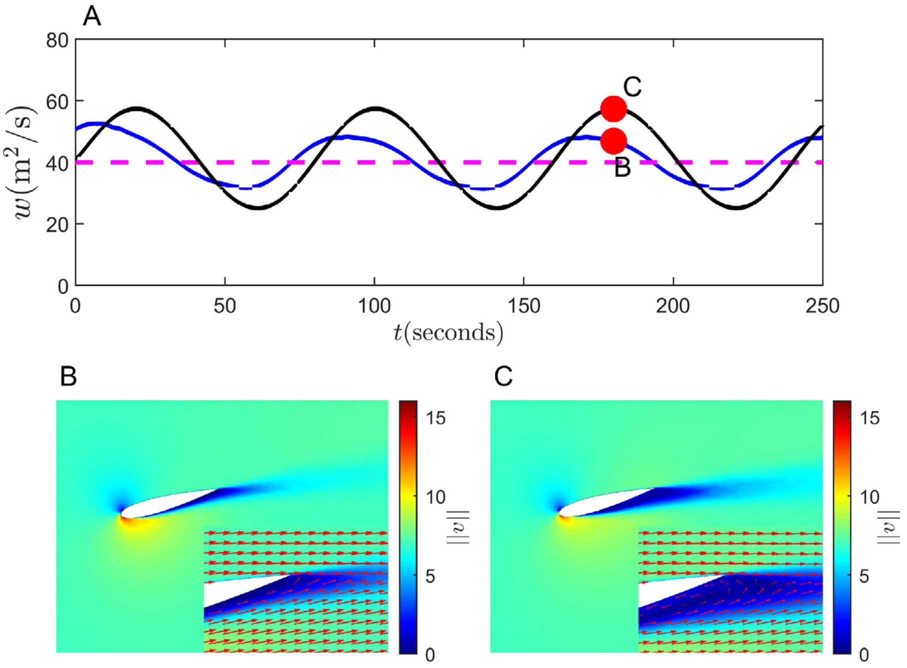

Fig. 9. In full order model simulations, panel A shows resulting vorticities both with (blue line) and without (black line) the application of the PI controller from Eq. (27). The dashed line highlights the target value for the PI controller and is shown for reference. Red dots indicate the observable values corresponding to the velocity field snapshots shown in panels B and C. In the simulation where the PI controller is not applied (panel C), velocity flow lines highlight flow separation and an associated region of recirculating flow. In the simulation where the PI controller is applied (Panel B). This region of recirculating flow is not present. While the PI controller does not fully suppress oscillations in the vorticity, it is sufficient to reduce flow separation.

with output (26)) are shown in Fig. 8. Panel A shows the profile of $u_1(t)$. The resulting input $u_2(t)$ is shown in panels B and C when using the nonlinear reduced order model and full order models, respectively. Likewise, panels D and E show the controlled output for the full and reduced order models. On balance, the reduced order model provides an accurate representation for the full order model dynamics when using the PI controller.

Finally, we consider the ability of the PI controller to suppress flow separation in the full order model with results shown in Fig. 9. Here we consider the same sinusoidal input for $u_1(t)$ that was used in Fig. 8. Panel A compares the uncontrolled vorticity (black line) to the controlled vorticity (blue line). The dashed magenta line highlights the target value for the PI controller. While the controller does not perfectly maintain the prescribed reference value, the magnitude of the oscillations is substantially diminished. Panels B and C highlight velocity snapshots at $t = 180$ s in simulations of the full order model (Eqs. (1)–(3) with output (5)) with and without the application of the PI controller, respectively. In simulations where the PI controller is not applied, $u_2(t)$ is held constant at 0 m/s. Arrows from the insets in panels B and C highlight the flow direction near the trailing end of the airfoil. In the uncontrolled simulation from panel C, flow separation results in a large recirculation zone. This recirculation is not seen in the controlled simulations from panel B. While the PI controller does not perfectly maintain the 40 m^2/s target value for vorticity, it is sufficient to suppress flow separation in these simulations.

6. Conclusion

In this work, we have implemented an adaptive isostable reduction strategy for data-driven model identification and control of fluid flows over an airfoil with moderate Reynolds numbers. Using data taken from simulations of the dimensionless Navier–Stokes equations (1)–(3), we are able to identify a low dimensional model that accurately captures the dynamics of the vorticity from Eq. (5) in response to time-varying boundary conditions. A linearized version of this model is then used to identify appropriate gains for a PI controller that attempts to keep the vorticity at a constant low level in the presence of time-varying inflow conditions. While the controller does not perfectly control the vorticity, it is able to suppress flow separation as illustrated in the results from Fig. 9.

Previous work [27,29] using the adaptive isostable reduction approach has been limited to prototype problems involving the Burgers' equation on simple 1 and 2-dimensional domains. Results presented in this work provide proof of concept that this framework can be successfully used in realistic fluid flow models using geometries with practical relevance. Because the model identification process does not require any information about the underlying model equations, it could be implemented in an experimental setting. In this work, we considered a single observable (5) indirectly related to the emergence of flow separation. In an experimental setting, other observables such as lift and drag generated by the airfoil, pointwise velocity flow velocity measurements, or data from flow visualization techniques would be more feasible to obtain. The underlying methodology applied in this work is quite general and could be used with any combination of these observables provided time-series measurements can be reliably obtained. While the control problem considered in this work is a straightforward implementation of a PI controller, the low dimensionality of the resulting reduced order model would allow for the implementation more sophisticated nonlinear optimal control strategies. Given the low-dimensionality of the resulting representation for the system dynamics, it might be possible to use sparse fitting techniques such as those described in [54–56] to learn a representation for the system dynamics. However, an appropriate basis would need to be determined for the fitting and there is no guarantee that one would arrive at an adequate realization for the dynamics.

We emphasize that the present study is still a prototype problem meant to illustrate the feasibility of implementing data-driven isostable-based reduced order modeling strategies in a practically relevant fluid flow model. We have not directly considered other relevant questions such as the placement of the inlet on the airfoil, the angle of attack of the airfoil or variations in the airfoil geometry. There are many other limitations of the present study. For instance, we did not consider the influence of measurement noise on the observable data which would likely degrade the accuracy of the inferred reduced order model. The influence of noise could be mitigated by performing multiple identical trials and averaging the results, but this approach would likely require substantially more data to implement the model fitting procedure. As an additional limitation, we consider the k-ϵ model to obtain numerical solutions of the Navier–Stokes equation. This is a common model that includes two extra transport equations in order to account for the flow turbulent properties [57]. It is a numerically stable model with fast convergence, but it is not always accurate for flows containing large adverse pressure gradients and hence not preferred for separated flows [58]. Future work will consider the simulation of different turbulence models such as the sheer stress transport (SST) model [59] or the k-ω model [60] that may be able to yield results that would agree better with experimental data. As an additional limitation, the present reduced order modeling approach considers solutions relative to a stable fixed point attractor. The k-ϵ model used in this work averages out the effect of turbulent features in the flow data, effectively yielding stationary solutions for the model considered in this work. For more finely resolved simulations and for experimental data, however, modifications to the proposed method would need to be made in order to handle applications where the observables cannot be analyzed in reference to a fixed point attractor.

CRediT authorship contribution statement

Adharaa Neelim Dewanjee: The model, Computational framework, Data analysis, Numerical computations, Writing – review & editing. **Samir Sahyoun:** Writing – review & editing. **Seddik Djouadi:** Writing – review & editing. **Dan Wilson:** The model, Computational framework, Data analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This material is based upon the work supported by the National Science Foundation, United States grant CMMI-2024111.

Appendix. Dynamic mode decomposition with control

The DMDc algorithm is described in [18] and provided here for convenience. When the system state is comprised of a delay embedding of time series measurements, this procedure is often called Hankel DMD [16]. The goal is to approximate a linear operator (from data) that relates a future system measurement $x_{k+1} \in \mathbb{R}^n$ to the current measurement x_k and control $u_k \in \mathbb{R}^l$, i.e.

$$x_{k+1} \approx Ax_k + Bu_k. \tag{A.1}$$

Here, $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times l}$. The measurements of the system and the control input throughout time can be captured in temporal snapshots that can be used to create data matrices. The measurement snapshot matrices, X and X', are organized as

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_{m-1} \\ | & | & & | \end{bmatrix},$$

$$X' = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \dots & x_m \\ | & | & & | \end{bmatrix}, \tag{A.2}$$

where m is the total number of snapshots. The sequence of collected control input snapshots are arranged as

$$Y = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_{m-1} \\ | & | & & | \end{bmatrix}. \tag{A.3}$$

To incorporate the new data matrices, Eq. (A.1) can be rewritten in matrix form as follows

$$X' = AX + BY,$$

$$X' = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{A.4}$$

Estimates for the A and B matrix can be obtained using least squares fitting

$$\begin{bmatrix} A & B \end{bmatrix} = X' \begin{bmatrix} X \\ Y \end{bmatrix}^{\dagger}, \tag{A.5}$$

where † denotes the pseudoinverse. The output of Eq. (6) can be predicted using Eq. (A.4) in response to general inputs once estimates of A and B have been determined.

References

[1] J.N. Kutz, S.L. Brunton, B.W. Brunton, J.L. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[2] S.L. Brunton, J.N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, Cambridge University Press, New York, 2019.

[3] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, L.S. Ukeiley, Modal analysis of fluid flows: an overview, AIAA J. (2017) 4013–4041.

[4] C.W. Rowley, S.T.M. Dawson, Model reduction for flow analysis and control, Annu. Rev. Fluid Mech. 49 (2017) 387–417.

[5] K. Taira, M.S. Hemati, S.L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S.T.M. Dawson, C.A. Yeh, Modal analysis of fluid flows: Applications and outlook, AIAA J. 58 (3) (2020) 998–1022.

[6] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, New York, 1996.

[7] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Sci. (2000) 808–817.

[8] C.W. Rowley, T. Colonius, R.M. Murray, Model reduction for compressible flows using POD and Galerkin projection, Physica D 189 (1–2) (2004) 115–129.

[9] B.R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, J. Fluid Mech. 497 (2003) 335–363.

[10] P. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.

[11] C.W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, D.S. Henningson, Spectral analysis of nonlinear flows, J. Fluid Mech. 641 (1) (2009) 115–127.

[12] M. Budišić, R. Mohr, I. Mezić, Applied Koopmanism, Chaos 22 (4) (2012) 047510.

[13] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annu. Rev. Fluid Mech. 45 (2013) 357–378.

[14] I. Mezić, Spectrum of the Koopman operator, spectral expansions in functional spaces, and state-space geometry, J. Nonlinear Sci. 30 (5) (2020) 2091–2145.

[15] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlinear Sci. 25 (6) (2015) 1307–1346.

[16] H. Arbabi, I. Mezic, Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator, SIAM J. Appl. Dyn. Syst. 16 (4) (2017) 2096–2126.

[17] M. Korda, I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, Automatica 93 (2018) 149–160.

[18] J.L. Proctor, S.L. Brunton, J.N. Kutz, Dynamic mode decomposition with control, SIAM J. Appl. Dyn. Syst. 15 (1) (2016) 142–161.

[19] S. Peitz, S.E. Otto, C.W. Rowley, Data-driven model predictive control using interpolated Koopman generators, SIAM J. Appl. Dyn. Syst. 19 (3) (2020) 2162–2193.

[20] S. Peitz, S. Klus, Koopman operator-based model reduction for switched-system control of PDEs, Automatica 106 (2019) 184–191.

[21] A. Mauroy, I. Mezić, J. Moehlis, Isostables, isochrons, and Koopman spectrum for the action–angle representation of stable fixed point dynamics, Physica D 261 (2013) 19–30.

[22] D. Wilson, J. Moehlis, Extending phase reduction to excitable media: Theory and applications, SIAM Rev. 57 (2) (2015) 201–222.

[23] D. Wilson, Phase-amplitude reduction far beyond the weakly perturbed paradigm, Phys. Rev. E 101 (2) (2020) 022220.

[24] D. Wilson, Data-driven inference of high-accuracy isostable-based dynamical models in response to external inputs, Chaos 31 (6) (2021) 063137.

[25] Y. Park, D.D. Wilson, High-order accuracy computation of coupling functions for strongly coupled oscillators, SIAM J. Appl. Dyn. Syst. 20 (3) (2021) 1464–1484.

[26] D. Wilson, An adaptive phase-amplitude reduction framework without $\mathcal{O}(\epsilon)$ constraints on inputs, SIAM J. Appl. Dyn. Syst. 21 (1) (2022) 204–230.

[27] D. Wilson, S.M. Djouadi, Adaptive isostable reduction of nonlinear PDEs with time varying parameters, IEEE Control Syst. Lett. 5 (1) (2021) 187–192.

[28] D. Wilson, Degenerate isostable reduction for fixed-point and limit-cycle attractors with defective linearizations, Phys. Rev. E 103 (2) (2021) 022211.

[29] D. Wilson, Data-driven identification of dynamical models using adaptive parameter sets, Chaos 32 (2) (2022) 023118.

[30] H. Dong, T. Xia, L. Chen, S. Liu, Y.D. Cui, B.C. Khoo, A. Zhao, Study on flow separation and transition of the airfoil in low Reynolds number, Phys. Fluids 31 (10) (2019) 103601.

[31] American Bureau of Shipping, NACA 0015 Wing Pressure and Trailing Vortex Measurements, NASA Technical Paper 3151, ABS American Bureau for Shipping, 1991.

[32] E.C. Maskell, Flow Separation in Three Dimensions, RAE Report No. Aero 2565, Ministry of Supply, Royal Aircraft Establishment, RAE Farnborough, 1955.

[33] P.K. Chang, Control of Flow Separation, McGraw-Hill, New York, 1976.

[34] D. Telionis, Unsteady boundary layers, separated and attached, J. Fluids Eng. 101 (1) (1979) 29–43.

[35] M. Gad-el Hak, D.M. Bushnell, Separation control, J. Fluids Eng. 113 (1) (1991) 5–30.

[36] R.L. Simpson, Turbulent boundary-layer separation, Annu. Rev. Fluid Mech. 21 (1) (1989) 205–232.

[37] R.L. Simpson, Aspects of turbulent boundary-layer separation, Prog. Aerosp. Sci. 32 (5) (1996) 457–521.

[38] P. Bradshaw, F.Y.F. Wong, The reattachment and relaxation of a turbulent shear layer, J. Fluid Mech. 52 (1) (1972) 113–135.

[39] J. Kim, S.J. Kline, J.P. Johnston, Investigation of a reattaching turbulent shear layer: flow over a backward-facing step, J. Fluids Eng. 102 (3) (1980) 302–308.

[40] J.C. Lin, Control of turbulent boundary-layer separation using micro-vortex generators, in: 30th AIAA(99-3404) Fluid Dynamics Conference, 1999, pp. 1–16.

[41] J. Béra, M. Michard, M. Sunyach, G. Comte-Bellot, Changing lift and drag by jet oscillation: experiments on a circular cylinder with turbulent separation, Eur. J. Mech. B Fluids 19 (5) (2000) 575–595.

[42] H. Shan, L. Jiang, C. Liu, M. Love, B. Maines, Comput. & Fluids 37 (8) (2008) 975–992.

[43] D. Greenblatt, I.J. Wygnanski, The control of flow separation by periodic excitation, Prog. Aerosp. Sci. 36 (7) (2000) 487–545.

[44] M. Nishioka, M. Asai, S. Yoshida, Control of flow separation by acoustic excitation, AIAA J. 28 (11) (1990) 1909–1915.

[45] D. You, P. Moin, Active control of flow separation over an airfoil using synthetic jets, J. Fluids Struct. 24 (8) (2008) 1349–1357.

[46] T.C. Corke, P.O. Bowles, C. He, E.H. Matlis, Sensing and control of flow separation using plasma actuators, Phil. Trans. R. Soc. A 369 (1940) (2011) 1459–1475.

[47] T. Weier, G. Gerbeth, G. Mutschke, O. Lielausis, G. Lammers, Control of flow separation using electromagnetic forces, Flow Turbul. Combust. 71 (1) (2003) 5–17.

[48] I. Mezić, Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry, J. Nonlinear Sci. (2019) 1–55.

[49] D. Wilson, Analysis of input-induced oscillations using the isostable coordinate framework, Chaos 31 (2) (2021) 023131.

[50] D. Wilson, B. Ermentrout, Augmented phase reduction of (not so) weakly perturbed coupled oscillators, SIAM Rev. 61 (2) (2019) 277–315.

[51] D. Wilson, A data-driven phase and isostable reduced modeling framework for oscillatory dynamical systems, Chaos 30 (1) (2020) 013121.

[52] H. Arbabi, I. Mezić, Study of dynamics in post-transient flows using Koopman mode decomposition, Phys. Rev. Fluids 2 (12) (2017) 124402.

[53] S.L. Brunton, B.W. Brunton, J.L. Proctor, E. Kaiser, J.N. Kutz, Chaos as an intermittently forced linear system, Nature Commun. 8 (1) (2017) 1–9.

[54] S.L. Brunton, B.W. Brunton, J.L. Proctor, J.N. Kutz, Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control, PLoS One 11 (2) (2016).

[55] N.M. Mangan, T. Askham, S.L. Brunton, J.N. Kutz, J.L. Proctor, Model selection for hybrid dynamical systems via sparse regression, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 475 (2223) (2019) 20180534.

[56] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, Proc. R. Soc. A 473 (2197) (2017) 20160446.

[57] B. Launder, D. Spalding, The numerical computation of turbulent flows, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 3 (2) (1974) 269–289.

[58] J. Bardina, P. Huang, T. Coakley, Turbulence modeling validation, in: 28th Fluid Dynamics Conference, 1997, p. 2121.

[59] F. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (8) (1994) 1598–1605.

[60] D. Wilcox, Formulation of the k-w turbulence model revisited, AIAA J. 46 (11) (2008) 2823–2838.