# A low-order nonlinear model of a stalled airfoil from data: Exploiting sparse regression with physical constraints

A. Leonid Heide*
*University of Minnesota, Minneapolis, MN 55455, USA*

Katherine J. Asztalos†
*Illinois Institute of Technology, Chicago, Illinois 60616, USA*

Scott T. M. Dawson‡
*Illinois Institute of Technology, Chicago, Illinois 60616, USA*

Maziar S. Hemati§
*University of Minnesota, Minneapolis, MN 55455, USA*

**This work uses data-driven sparsity-promoting methods to obtain low-order governing equations for the wake of a stalled airfoil. Direct numerical simulation data of a NACA-0009 airfoil at an angle of attack of $\alpha = 15^\circ$ is utilized in this study, with actuation being performed by injecting momentum into the flow near the airfoil's leading edge. Proper Orthogonal Decomposition (POD) is used to obtain a reduced order representation of the flow field. The Sparse Identification of Nonlinear Dynamics (SINDy) framework is then implemented to obtain low-order quadratic governing equations for the flow over the stalled airfoil. The SINDy model is constrained to preserve the energy-conserving property of the quadratic nonlinearity and associated triadic energy-transfer mechanisms. Low-order nonlinear models of the unsteady flow field associated with the stalled airfoil are obtained and cross-validated using off-design data. Furthermore, an output equation that predicts the lift coefficient is also identified and cross-validated. These low-order nonlinear models are expected to facilitate future developments in model-based analysis and control of separated flows.**

## I. Introduction

Unsteady aerodynamic effects greatly impact the performance of aircraft. Especially affected are small, highly-maneuverable, aeroelastic, and biological aircraft. Performance degradation is owed to the unsteady aerodynamic phenomenon of flow separation, which occurs when a laminar boundary layer becomes detached from a surface and transitions into a turbulent wake. Under certain conditions, the boundary layer may become fully separated from a surface, resulting in an effect known as stall. In the case of airfoils, the stall condition is met when the airfoil reaches a critical angle of attack, whereupon the fluid rolls up from the trailing edge to the leading edge until the flow becomes fully detached from the low pressure (suction) surface [1]. Stall can lead to the formation of an unstable shear layer and periodic vortex shedding, in addition to a reduction in lift and an increase in drag, which may result in the loss of control over an aircraft. Therefore, managing flow separation is necessary for ensuring safe and efficient flight.

Historically, flow separation has been addressed by restricting the operational envelope of aircraft to avoid situations where stall could occur. An alternative to this confining approach is active flow control, which offers a means to mitigate and control aerodynamic instabilities such as stall. One popular flow control method is to utilize synthetic jets embedded into a lifting surface to inject momentum into the crossflow [2–5]. By doing this, it is possible to manipulate the flow and temporarily re-attach a separated shear layer to the surface, leading to transient lift enhancement [1]. Active flow control significantly enhances aircraft performance by improving safety, flight efficiency, and robustness to aerodynamic disturbances.

*Graduate Student, Aerospace Engineering and Mechanics, AIAA Student Member.
†Graduate Research Assistant, MMAE Department, AIAA Student Member.
‡Assistant Professor, MMAE Department, AIAA Senior Member.
§Associate Professor, Aerospace Engineering and Mechanics, AIAA Associate Fellow.

However, the realization of real-time active flow control strategies is bottlenecked by the lack of reliable and computationally inexpensive flight controllers. Unsteady aerodynamic effects cause complex dynamic responses and multimodal instabilities that are difficult to control and model due to their high dimensionality and nonlinearity. This is especially true for flows where complex geometries are involved, which requires a large number of terms to be accurately modeled. Analytical methods that are based on the physics of flows provide accurate governing equations for a system, but are difficult to implement for complex systems. An alternative is to directly obtain models from representative flow data by implementing data-driven methods. However, without modification many data-driven methods do not provide governing equations that are representative of underlying physics of the flow. In this work, a data-driven method called "Sparse Identification of Nonlinear Dynamics" (SINDy) [6] is implemented and modified according to [7] by introducing physics-based constraints that take advantage of the energy-conserving property of the quadratic nonlinearity inherent to incompressible flows.

Constrained SINDy is leveraged to obtain low-order governing equations for a two-dimensional, incompressible NACA-0009 airfoil in stall (at $\alpha = 15°$) with short-duration momentum injection above the leading edge as described by Asztalos et al. [1]. In addition to obtaining governing equations for the flow, a method to determine output equations for flow-dependent derived quantities is implemented and used to determine the airfoil's lift-coefficient $C_\ell(t)$. The end result is a system of dynamic and output equations. It is demonstrated that constrained SINDy is an effective means of obtaining nonlinear reduced-order models (ROM) for stall over a two-dimensional airfoil. This establishes the approch as a viable candidate for modeling more complex multimodal flows in the future. This also creates opportunities for future studies on nonlinear analysis and control of such flows.

The paper proceeds as follows: In section II, the current state of the art of modeling unsteady flows is discussed, followed by an overview of the stalled airfoil system and the direct numerical simulation (DNS) approach in section III. Section IV then describes the data-driven approach taken to obtain reduced-order quadratic governing equations for the stalled NACA-0009 flow, including an overview of the physics that govern this type of flow. The results of the analysis are then presented in section V, and concluding remarks are given in section VI.

## II. State of the Art

Modeling unsteady flows poses unique challenges as the nonlinear dynamics result in complex systems where the tradeoff between accuracy and computational efficiency must be carefully balanced. A plethora of methods exist with which it is possible to accurately model unsteady aerodynamic effects such as flow separation. Unfortunately, many numerical approaches such as direct numerical simulation (DNS) have prohibitive computational cost that make them impractical for real-time control and prediction. Physics-based approaches provide accurate governing equations for systems, but may be difficult to obtain for complex flows. Conversely, many data-driven approaches lack the accuracy and flexibility to describe rapidly evolving dynamic systems, and resultant models may not reflect the underlying physics of the fluid system.

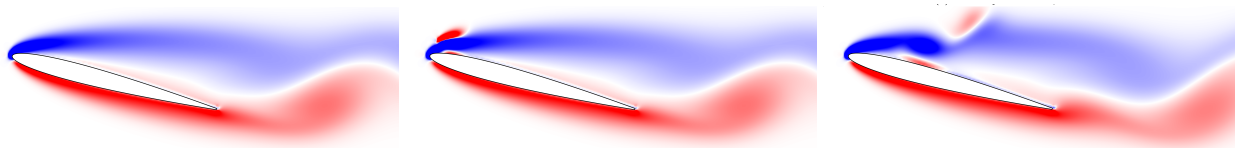
One means to generate low-order models for unsteady dynamics is by using dynamic mode decomposition (DMD) [8], after first applying proper orthogonal decomposition (POD) for dimensionality reduction. POD is used to identify basis functions that can describe the dynamic behavior of the system across some parameter space of interest [9, 10]. POD is especially desirable because it can be "trained" on a desired representation of the system dynamics through the use of so called "snapshots" [9–11]. These snapshots are obtained through the collection of observations about the solution at different time intervals and they are used to obtain a reduced basis that optimally spans the data. Using this reduced basis along with given time-series measurements of the system state and inputs, it is possible to obtain models by invoking DMD [12]. This DMD approach was successfully implemented by Asztalos et. al. [1] to describe the response of an airfoil to sudden momentum injection into the free-stream.

An alternative is to directly obtain a system's governing equations from datasets using Spare Identification of Nonlinear Dynamics (SINDy). SINDy obtains governing equations by taking advantage of sparsity-promoting techniques in a regression framework [6]. By assuming that there are only a few important parameters that govern the system dynamics, it is possible to establish sparsity, and thereby a reduced basis. To obtain the functions of interest, a time history of the state is collected from data, and the state's derivative is either measured or approximated. Candidate functions are then assembled into a matrix, and a sparse regression is implemented to the matrix to identify which nonlinearities are relevant. SINDy thereby obtains governing equations for the system in a reduced basis. The method has been successfully used in combination with other techniques such as POD to obtain accurate estimates of a canonical flow over a 2D cylinder [6]. Furthermore, the predictive capabilities of SINDy can be enhanced by utilizing a priori knowledge of the system's physics to introduce constraints that provide governing equations of a desired form. These

constraints have been implemented in [7] to obtain low-order models of an incompressible flow over a cylinder with $n = 3$ modes. The approach has recently been applied to systems with $n > 3$ modes in [13], whereby constrained SINDy was used to model and control instabilities in a Kolmogorov-like flow.

## III. System Description and Direct Numerical Simulation

This work examines a NACA-0009 airfoil at an angle of attack of $\alpha = 15^\circ$ and a Reynolds number of 500, as described in Ref. [1]. At these conditions, the airfoil generates an unstable wake which converges to a limit-cycle solution with periodic vortex shedding (when unactuated), as shown in figure 1a. The system is actuated using a "burst-type" momentum injection centered at a point that is $x = 0.05c$ downstream of the leading edge and $y = 0.0523c$ above the surface of the airfoil, where $c$ is the chord of the airfoil. This single impulse signal is modeled as a body force with a duration of $\Delta t_d = 1.2$, which generates a pair of counter-rotating vortices, as shown in figure 1b. The period of the vortex shedding $T \approx 1.6\frac{c}{U_{\text{inf}}}$ was computed from the unactuated case, and the onset of actuation was varied during the duration of one vortex-shedding period. A total of 17 tests were performed with the onset of actuation occurring every $\Delta t = 0.1$ from $(61.2 \leq t \leq 62.8)$, where $t = 0$ is the start of the simulation. Here and throughout, time is nondimensionalized by the convective time unit, $\frac{c}{U_{\text{inf}}}$. This work focuses on the unactuated case and the case where actuation occurs at $t = 62$, which corresponds to the the maximum lift ($C_{\ell,\text{max}}$) of the limit cycle. All subsequent results show time $t^+$ relative to $t = 60$. In other words, actuation at $t = 62$ corresponds to $t^+ = 2$. Furthermore, this work only examines the flow after the actuation has concluded, so that the energy conservation assumption—to be discussed in Sections IV.A and IV.E—can be made for the flow. For example, in the case of actuation at $t^+ = 2$, the analysis is started at $t^+ = 2.3$, which is one time step after actuation ends at $t^+ = 2.2$, as shown in figure 1c. The direct numerical simulations (DNS) were obtained by solving the two-dimensional Navier–Stokes equations using an immersed boundary projection method (IBM) [14, 15] incorporating a lattice Green function (LGF) formulation [16, 17]. Additional details are given in Ref. [1].



(a) Vorticity at $t^+ = 0.4$, baseline (b) Vorticity at $t^+ = 2.0$, actuation starts (c) Vorticity at $t^+ = 2.3$, actuation ends

**Fig. 1 Vorticity fields near the airfoil with actuation onset at $t^+ = 2.0$, actuation end at $t^+ = 2.2$, and the beginning of data-driven modeling one time-step later at $t^+ = 2.3$.**

## IV. Approach

In this work, governing equations for the unsteady aerodynamics of flow over a stalled NACA-0009 airfoil with momentum injection near the airfoil's leading edge are identified using constrained SINDy. The following attributes are desired from the resultant model: 1) it must be consistent with the physical equations that represent unsteady (incompressible) flows; 2) it must balance accuracy and complexity; and 3) it must be able to predict flow-dependent quantities of interest, such as the lift coefficient ($C_\ell$). This section describes the approach taken to satisfy these objectives for the 2-D stalled airfoil. The subsequent discussion is divided as follows:

- Section IV.A: The physics that govern the unsteady incompressible flow are examined. System dynamics and lift-coefficient output equations are defined.
- Section IV.B: Low-rank basis functions are obtained from DNS data using proper orthogonal decomposition (POD).
- Section IV.C: An overview of the Sparse Identification of Nonlinear Dynamics (SINDy) method is provided.
- Section IV.D: An outline of how SINDy can be modified with constraints is provided.
- Section IV.E: The constraints needed to satisfy the first objective are reviewed, and are imposed to SINDy to ensure that the quadratic nonlinearity in the identified model remains energy conserving.
- Section IV.F: A model-selection method is introduced to find a model that balances sparsity and accuracy.

### A. Incompressible Navier–Stokes Solution

The unsteady flow resulting from the stalled NACA-0009 airfoil system can be generalized as the following dynamic equation:

$$\dot{\tilde{\boldsymbol{u}}}(x, y, t) = f(\tilde{\boldsymbol{u}}(x, y, t)). \tag{1}$$

Here, $\tilde{\boldsymbol{u}}(x, y, t)$ is some state of the system at a given location and time (which will be denoted $\tilde{\boldsymbol{u}}$ for brevity), and $f(\tilde{\boldsymbol{u}})$ is the dynamics function that defines the governing equations for the system. To obtain the right hand side of (1) for the stalled airfoil, the Navier–Stokes equations of a perturbed velocity $\boldsymbol{u}$ evolving on top of a base flow $\boldsymbol{U}_b$ are considered:

$$\frac{\partial \boldsymbol{u}}{\partial t} = -(\boldsymbol{U}_b \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{U}_b - \nabla p + \frac{1}{\mathrm{Re}}\nabla^2 \boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \tag{2a}$$

$$0 = \nabla \cdot \boldsymbol{u}. \tag{2b}$$

Here, $p$ is the pressure and $Re$ is the Reynolds number. Projecting onto the basis of divergence-free fields, the pressure drops out and the incompressibility constraint will be satisfied automatically, yielding dynamics in the form of (1). These dynamics can further be decomposed as:

$$\frac{d\tilde{\boldsymbol{u}}}{dt} = \boldsymbol{Q}(\tilde{\boldsymbol{u}})\tilde{\boldsymbol{u}} + \boldsymbol{L}(\tilde{\boldsymbol{u}}) + \boldsymbol{C}_0, \tag{3}$$

where $\boldsymbol{Q}(\tilde{\boldsymbol{u}})\tilde{\boldsymbol{u}}$, $\boldsymbol{L}(\tilde{\boldsymbol{u}})$, and $\boldsymbol{C}_0$ are low-dimensional approximations of the Navier–Stokes nonlinear quadratic operator, linearized operator, and affine terms, respectively. A key component of the incompressible Navier-Stokes equations relevant to our investigation is the quadratic convective nonlinearity $-(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$. It is well known that this quadratic nonlinearity is lossless for incompressible flows:

$$\int_\Omega \boldsymbol{u} \cdot (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \, d\Omega = 0. \tag{4}$$

The quadratic ($\tilde{\boldsymbol{u}}_i^2$) and bilinear ($\tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{u}}_j$) terms must therefore be energy-conserving. In other words: the nonlinear terms can only redistribute the energy of the perturbation $\boldsymbol{u}$ among the other terms [7]. If the state of choice $\tilde{\boldsymbol{u}}$ is related directly to the kinetic energy of the perturbation, the constraint to make the quadratic nonlinear term energy-preserving can be written in the following form:

$$\tilde{\boldsymbol{u}} \cdot \boldsymbol{Q}(\tilde{\boldsymbol{u}})\tilde{\boldsymbol{u}} = 0. \tag{5}$$

The main objective for this work is therefore to obtain governing equations such that equation (1) has the form of equation (3) subject to the lossless constraint on the quadratic term. In addition, an output equation can be introduced to determine relevant derived quantities of interest, such as the lift coefficient $C_\ell$, moment coefficient $C_m$, or drag coefficient $C_D$. We note that these outputs can be determined from algebraic relations with the dynamic flow state, expressed generically as $\boldsymbol{y} = g(\tilde{\boldsymbol{u}})$. The model we seek will thus take the form:

$$\dot{\tilde{\boldsymbol{u}}} = f(\tilde{\boldsymbol{u}}) \tag{6a}$$

$$\boldsymbol{y}(t) = g(\tilde{\boldsymbol{u}}). \tag{6b}$$

As we will show later, we will extend the SINDy framework to identify an output equation for the lift response of a NACA-0009 airfoil, i.e., for the scalar output $y(t) = C_\ell(t) = g(\tilde{\boldsymbol{u}})$.

### B. Reducing Dimensionality of Simulation Data

This section describes the procedure used to obtain low-rank basis functions for the NACA-0009 system from DNS velocity data. For every simulated actuation case, the mean-subtracted velocity field data $(\boldsymbol{u}, \boldsymbol{v})$ is obtained from the DNS at $[n_x = 2000 \times n_y = 500]$ locations in $\boldsymbol{x} = (x, y)$, for 200 snapshots in time. Therefore, for each time series of

data containing $N$ snapshots and starting at time $t_j$ the following $2r \times N$ matrix is formed:

$$X^{(j)} = \begin{bmatrix} u(\boldsymbol{x}_1, t_j) & u(\boldsymbol{x}_1, t_j + \Delta t) & u(\boldsymbol{x}_1, t_j + 2\Delta t) & \dots & u(\boldsymbol{x}_1, t_j + N\Delta t) \\ u(\boldsymbol{x}_2, t_j) & u(\boldsymbol{x}_2, t_j + \Delta t) & u(\boldsymbol{x}_2, t_j + 2\Delta t) & \dots & u(\boldsymbol{x}_2, t_j + N\Delta t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u(\boldsymbol{x}_r, t_j) & u(\boldsymbol{x}_r, t_j + \Delta t) & u(\boldsymbol{x}_r, t_j + 2\Delta t) & \dots & u(\boldsymbol{x}_r, t_j + N\Delta t) \\ v(\boldsymbol{x}_1, t_j) & v(\boldsymbol{x}_1, t_j + \Delta t) & v(\boldsymbol{x}_1, t_j + 2\Delta t) & \dots & v(\boldsymbol{x}_1, t_j + N\Delta t) \\ v(\boldsymbol{x}_2, t_j) & v(\boldsymbol{x}_2, t_j + \Delta t) & v(\boldsymbol{x}_2, t_j + 2\Delta t) & \dots & v(\boldsymbol{x}_2, t_j + N\Delta t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v(\boldsymbol{x}_r, t_j) & v(\boldsymbol{x}_r, t_j + \Delta t) & v(\boldsymbol{x}_r, t_j + 2\Delta t) & \dots & v(\boldsymbol{x}_r, t_j + N\Delta t) \end{bmatrix}. \tag{7}$$

The singular value decomposition (SVD) is then taken for each time series of data (denoted by the subscript $j$):

$$X^{(j)} = U\Sigma V^*, \tag{8}$$

where $\Sigma$ is a matrix of singular values, and the columns of $U$ correspond to the proper orthogonal decomposition (POD) modes of the data. The associated coefficients for the POD modes are found by observing the rows of $\tilde{X}$:

$$\tilde{X} = U^* X^{(j)} = \Sigma V^*. \tag{9}$$

The state of the system as a function of position and time is therefore obtained in the POD basis as:

$$\boldsymbol{u}(x,t) = \bar{\boldsymbol{u}}(x) + \sum_{i=1}^{n} a_i(t)\phi_i(x), \tag{10}$$

where $\boldsymbol{u}(x,t)$ represents the velocity field, $a_i(t)$ are the POD-coefficients, $\phi_i(x)$ are the associated basis functions (POD modes), and $\bar{\boldsymbol{u}}(x)$ is the mean velocity field. The POD modes obtained in this way describe the vortex shedding observed as a result of the airfoil stall. In total, this work uses 200 POD modes over 200 snapshots for each simulation. After the POD coefficients are found, the state data matrix $A$ is defined for this work as:

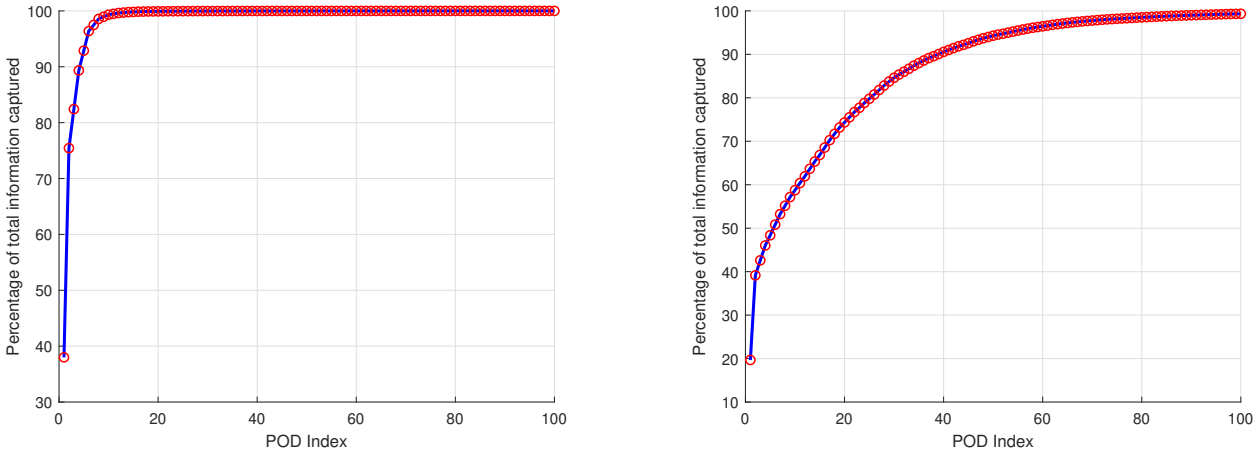
$$A = \begin{bmatrix} a_1(t_j) & a_2(t_j) & \dots & a_n(t_j) \\ a_1(t_j + \Delta t) & a_2(t_j + \Delta t) & \dots & a_n(t_j + \Delta t) \\ \vdots & \vdots & \ddots & \vdots \\ a_1(t_j + N\Delta t) & a_2(t_j + N\Delta t) & \dots & a_n(t_j + N\Delta t) \end{bmatrix}. \tag{11}$$

Although each time series contains 200 POD basis functions, the lower-order modes capture significantly more information than the higher order modes. In figure 2, the energy contributions of POD modes for the unactuated case are compared to the contributions of POD modes for the case where actuation occurred at $t^+ = 2$. The results demonstrate that the unactuated case requires significantly fewer POD modes to capture 99% of the information than the actuated case. This is attributed to the lower order modes carrying more information about global vortical structures and higher order modes characterizing local fluctuations [13, 18, 19]. As the actuated case exhibits more local phenomena, it requires a larger number of POD modes (approximately $n = 80$) to capture the same amount of information than with just $n = 10$ POD modes in the unactuated case.

### C. Sparse Identification of Nonlinear Dynamics (SINDy)

By performing POD, the DNS data is projected onto a small set of orthogonal modes that describe the flow dynamics. The resultant basis functions are useful for performing system identification to obtain reduced-order models that may be used for control. This work utilizes SINDy to identify a set of governing equations from the reduced DNS data that describes the stalled NACA-0009 airfoil.

To describe the SINDy algorithm, the dynamic equation (1): $\dot{\tilde{\boldsymbol{u}}} = f(\tilde{\boldsymbol{u}})$ is considered. It can be assumed that the right hand side of (1) only has a few active terms, and is therefore sparse in the space of all possible functions that describe the dynamics [6]. Using this observation, a sparse-regression problem is defined, where the time-derivative of




(a) Unactuated

(b) Pulse actuation applied at $t = 2$

**Fig. 2 The cumulative energy content captured within each POD mode for (a) the unactuated flow and (b) the flow with pulse actuation applied at $t^+ = 2.0$. Without actuation, using $n = 10$ POD modes captures 99% of the mean-subtracted flow energy. In contrast, using $n = 10$ POD modes captures only 68% of the mean-subtracted flow energy for the the pulse actuation case. Note that nearly 100% of the energy is captured with $n = 100$ modes, even though $n = 200$ modes were obtained from the DNS.**

the state is regressed onto a set of candidate functions of the state. In this work, the state consists of the set of POD coefficients $a_i(t)$. Data for the time evolution of the state is stored in the matrix $A$, where each column describes the evolution of an individual state variable $a_i(t)$ in time. The associated matrix of time-derivatives used to represent the left-hand side of the dynamic system in (1) is thus written as:

$$\dot{A} = \begin{bmatrix} \dot{a}_1(t_j) & \dot{a}_2(t_j) & \dots & \dot{a}_n(t_j) \\ \dot{a}_1(t_j + \Delta t) & \dot{a}_2(t_j + \Delta t) & \dots & \dot{a}_n(t_j + \Delta t) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{a}_1(t_j + N\Delta t) & \dot{a}_2(t_j + N\Delta t) & \dots & \dot{a}_n(t_j + N\Delta t) \end{bmatrix}. \tag{12}$$

To determine the function $f(\tilde{\boldsymbol{u}})$ from sparse regression, the matrix $\Theta(A)$, consisting of nonlinear candidate functions of the columns of $A$ is defined. Candidate functions in this work are defined as polynomials, however SINDy allows for more complex functions, such as trigonometric functions, to be employed. In this work, the assumption that the flow is incompressible enforces that only polynomials up to the 2$^{\text{nd}}$ degree are used, the justification of which will come in the next section. The matrix of candidate functions thus takes the following form [6]:

$$\Theta(A) = \begin{bmatrix} \vdots & \vdots & \vdots \\ 1 & A & A^{P_2} \\ \vdots & \vdots & \vdots \end{bmatrix}. \tag{13}$$

The matrix (13) is formed by evaluating all the candidate functions at each instant of time up to $t_N$. For example, $A^{P_2}$ is defined in the following way [6]:

$$A^{P_2} = \begin{bmatrix} a_1^2(t_j) & a_1(t_j)a_2(t_j) & \dots & a_2^2(t_j) & a_2(t_j)a_3(t_j) & \dots & a_n^2(t_j) \\ a_1^2(t_j + \Delta t) & a_1(t_j + \Delta t)a_2(t_j + \Delta t) & \dots & a_2^2(t_j + \Delta t) & a_2(t_j + \Delta t)a_3(t_j + \Delta t) & \dots & a_n^2(t_j + \Delta t) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_1^2(t_j + N\Delta t) & a_1(t_j + N\Delta t)a_2(t_j + N\Delta t) & \dots & a_2^2(t_j + N\Delta t) & a_2(t_j + N\Delta t)a_3(t_j + N\Delta t) & \dots & a_n^2(t_j + N\Delta t) \end{bmatrix}. \tag{14}$$

The SINDy algorithm seeks to obtain a sparse set of the functions defined by $\Theta(A)$ that model the dynamics consistent with the data in $\dot{A}$ [6]. The objective is therefore to find a sparse matrix of coefficients $\Xi$ that indicates which linear combination of $\Theta$ "best fits" the DNS data. This allows for the original dynamic system (1) to be written as:

$$\dot{A} = \Theta(A)\Xi. \tag{15}$$

In other words, the columns of the matrix $\Xi$ (represented as $\xi_k$) indicate which terms are active within the candidate function matrix $\Theta(A)$. The product of $\Theta(A)$ and $\xi_k$ therefore constitutes the desired function $f(\boldsymbol{a})$ from the initial dynamic system (1). Simply performing a standard regression to find the columns $\xi_k$ would result in all elements being non-zero [6]. To obtain the desired sparse solution, it is necessary to add an $L_1$ regularization term to the regression such that:

$$\xi_k = \arg\min_{\xi_k} ||\dot{A}_k - \Theta(A)\xi_k||_2^2 + \lambda||\xi_k||_1. \tag{16}$$

Here, the second term of (16) promotes sparsity within $\Xi$ through adjustment of the sparsification variable $\lambda$. Increasing the value of $\lambda$ will increase the sparsity of $\Xi$ at the expense of accuracy (and vice-versa). Equation (16) can be solved in several ways, such as applying the LASSO method [6, 20], or the iteratively thresholded least squares method [6]. This work applies the latter approach within the SINDy framework to solve equation (16). This allows for a simple extension to problems with constraints, as will be described in the next section. Furthermore, the output equation (6b) $\boldsymbol{y} = g(\tilde{\boldsymbol{u}})$ is solved for the lift coefficient $C_\ell$ using the unconstrained version of SINDy. This is done by taking the POD coefficient model obtained from solving the constrained SINDy problem (1) and performing a linear least squares regression onto DNS lift coefficient data.

### D. Introducing Constraints to SINDy

The SINDy methodology described in section IV.C seeks sparse governing equations by individually regressing each state variable. While this approach has the benefit of being agnostic to the system in question, it does not take advantage of apriori knowledge of the system. Improvements can therefore be made by introducing physics-based constraints to the system. The problem described in (16) is thus recast to the following equality constrained least-squares problem over all the data [7, 13]:

$$\begin{aligned} \min_{\Xi} \quad & ||\Theta(A)\Xi - \dot{A}||_2^2 \\ \text{s.t.} \quad & C\Xi(:) = d, \end{aligned} \tag{17}$$

where $C$ is a constraint matrix that allows constraints to be imposed on the vectorized coefficient matrix $\Xi(:)$. Rather than solving a minimization problem for each state as described in (16), this approach creates a single problem to be solved. To obtain a sparse matrix $\Xi$ with this formulation, (17) is solved iteratively, where the values in $\Xi$ are compared to the predefined sparsification value $\lambda$. If the entries in $\Xi$ are smaller than $\lambda$, a row is added to $C$ and a corresponding zero is added to $d$ in order to force the entries to zero during the next iteration [13]. Iterations are performed until the $L_2$ norm of the difference between successive solutions of $\Xi$ are smaller than a predefined tolerance.

### E. Obtaining Physics-Based Constraints

In order to obtain reduced-order governing equations that are consistent with the Navier–Stokes form described in equation (3) the physical properties of the incompressible fluid (2) must be captured. On its own, SINDy would not be able to replicate these properties. However, the constrained approach described in section IV.D allow active enforcement of desired conditions. To enforce physics-based constraints, the assumption that the quadratic and bi-linear terms in the Navier–Stokes equation must be energy-conserving is made (as shown in equation (4)). As the POD modes are related directly to the kinetic energy of the perturbation, the constraint to make the quadratic nonlinear term energy-preserving can be written in the form of equation (5): $\tilde{\boldsymbol{u}} \cdot \boldsymbol{Q}(\tilde{\boldsymbol{u}})\tilde{\boldsymbol{u}} = 0$. With these considerations it is possible to use the POD-basis representation defined in (10) together with the orthonormal property of POD modes to rewrite (3) in terms of the POD coefficients [13]:

$$\dot{a}_i(t) = \sum_{j=1}^{n}\sum_{k=1}^{n} q_{ijk}a_j a_k + \sum_{j=1}^{n} l_{ij}a_j + c_i \qquad i = 1, 2, ..., n. \tag{18}$$

The tensors in (18) can be described as follows [13]:

$$q_{ijk} = \phi_i \cdot \boldsymbol{Q}(\phi_j)\phi_k, \tag{19a}$$
$$l_{ij} = (\phi_i \cdot \boldsymbol{L}(\phi_j)) + (\phi_i \cdot \boldsymbol{Q}(\bar{\boldsymbol{u}})\phi_j) + (\phi_i \cdot \boldsymbol{Q}(\phi_j)\bar{\boldsymbol{u}}), \tag{19b}$$
$$c_i = (\phi_i \cdot \boldsymbol{L}(\bar{\boldsymbol{u}})) + (\phi_i \cdot \boldsymbol{Q}(\bar{\boldsymbol{u}})\bar{\boldsymbol{u}}) + (\phi_i \cdot \boldsymbol{C}_0). \tag{19c}$$

Equations (18) and (19) can be interpreted in terms of SINDy as follows: each vector of coefficients $\xi_k$ is equivalent to the $C_i$, $l_{ij}$, and $q_{ijk}$ stretched into a column vector for each POD mode [13]. A more detailed derivation of using the terms in (19) to obtain constraints for $\Xi$ for a model with $n$ modes is described in [13] and is based on triadic interactions inherent to Galerkin projection methods [21, 22]. The approach takes advantage of the symmetric nature of the term $q_{ijk}$ and the fact that it retains the energy-preserving property of the non-linearity in the Navier–Stokes equations. The energy within each POD mode is expressed by $e_i(t) = \frac{1}{2}a_i^2(t)$ which is differentiated in time to obtain [21]:

$$\dot{e}_i(t) = \sum_{j=1}^{n}\sum_{k=1}^{n} q_{ijk}a_i a_j a_k + \sum_{j=1}^{n} l_{ij}a_i a_j + c_i a_i \qquad i = 1, 2, ..., n. \tag{20}$$

Here, the constraint on $q_{ijk}$ means that the quadratic nonlinearity can only serve to exchange energy between modes and does not contribute to the rate of change of the total energy of the system [13]. Using this and the fact that the sum of the individual modes constitutes the total energy of the system $E(t) = \sum_{i=1}^{n} a_i^2$, the following expression is written:

$$\dot{E}_i(t) \propto \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} q_{ijk}a_i a_j a_k = 0. \tag{21}$$

Using the rules of index permutation and the symmetric property of $q_{ijk}$, the following energy-preserving constraint is obtained for use during SINDy:

$$q_{ijk} + q_{jki} + q_{kij} = 0. \tag{22}$$

As described in [13], equation (22) imposes three categories of constraints that are used to construct a constraint matrix $C$ and solve (17):

1) **Intrinsic constraint**: Occurs when $i = j = k$ and implies $q_{iii} = 0$ for any $i$. This is a result of a mode not being able to exchange energy with itself via the quadratic nonlinearity. This creates $n$ constraints, and $n$ rows are added to the matrix $C$.
2) **Binary constraint**: Occurs when $i \neq j = k$ or $i = j \neq k$, and implies that a term of the form $\dot{a}_i \propto a_j^2$ must be balanced by a term such as $\dot{a}_j \propto a_i a_j$. Intuitively, this constraint occurs when one mode interacts quadratically with one other mode. To impose this, $n(n-1)$ rows are added to the matrix $C$ (one for each possible binary constraint).
3) **Extrinsic constraint**: Occurs when $i \neq j \neq k$, and implies that a term $\dot{a}_i \propto a_j a_k$ involves interactions between mode $i$ and two other modes $(j, k)$. This constraint is fully triadic, and involves the energy exchange between three modes, and the energy exchange between two modes is mediated via a third mode. The complexity of these interactions result in the number of constraints added as rows to $C$ to grow rapidly: $\frac{n!}{3!(n-3)!} = \frac{1}{6}n(n-1)(n-2)$.

A total of $n + n(n-1) + \frac{1}{6}n(n-1)(n-2)$ rows in $C$ and elements in $d$ are therefore added upon the first solution of (17). Constraints for small entries in the coefficient matrix $\Xi$ are then added on subsequent iterations until a solution is reached [13]. This procedure results in a system of governing equations for which the quadratic nonlinearity conserves energy, consistent with the nonlinear physics of the incompressible Navier–Stokes equations.

### F. Model Selection

To find a satisfactory model, its accuracy and complexity must be balanced. An accurate model will be more complex and therefore more difficult to interpret while a sparse model will have higher errors. The complexity of a model is defined by model density $\rho(\lambda)$, which characterizes the number of active coefficients in the matrix $\boldsymbol{\Xi}$:

$$\rho(\lambda) = \frac{\text{card}(\boldsymbol{\Xi})}{fn}. \tag{23}$$

Here, $\rho$ ranges from 0 to 1, such that when $\rho = 0$ the system is fully sparse, $f$ is the number of candidate functions, $n$ is the number of states, and $card(\Xi)$ is the cardinality of the coefficient matrix. The accuracy of the model is defined by the model error $\epsilon(\lambda)$:

$$\epsilon(\lambda) = \frac{||A_{model} - A_{DNS}||_2}{||A_{DNS}||_2}, \tag{24}$$

where $A_{model}$ is obtained by integrating equation (15) from an initial condition, and $A_{DNS}$ contains the DNS results. Both the model density and error are functions of the sparsification constant $\lambda$, which was described in sections IV.C and IV.D. In other words, each $\lambda$ corresponds to a model error and density value, meaning it is the variable that must be optimized.

The Akaike Information Criterion (AIC) uses statistical methods to balance accuracy and complexity [23]. This method assigns an AIC score to every model, and the model corresponding to the lowest AIC score is chosen as the optimal one. The AIC scoring equation utilized in this work is defined as:

$$AIC = 2k + 2\log(\epsilon) - \frac{2(k+1)(k+2)}{k}, \tag{25}$$

where $k = \rho(\lambda) f n$ is the number of active terms in the model. The first right-hand term of equation (25) penalizes dense models, the second term penalizes inaccurate models, and the third term is a finite terms correction [23]. The sparsification constant $\lambda$ is swept through $O(10^2)$ values, and an AIC score is computed for each case. The resultant AIC scores are further re-scaled by the minimum of all AIC scores:

$$AIC_{relative} = AIC - AIC_{min}, \tag{26}$$

such that the best model has an AIC score $AIC_{relative} = 0$ [13, 23].

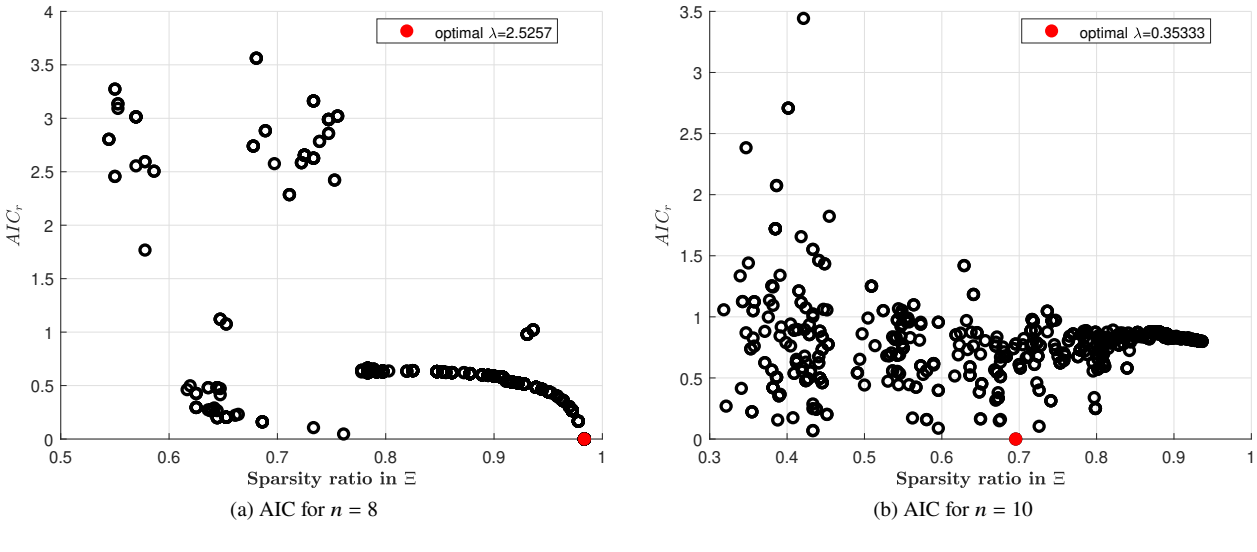
# V. Results

This section presents the results of implementing the constrained data-driven modeling approach to obtain a low-order nonlinear model for the stalled NACA-0009 airfoil system with actuation at $t^+ = 2.0$. POD modes are obtained for the $t^+ = 2.0$ flow, and reduced to a 10-mode model. An optimal $\lambda$ value is then found using the AIC model-selection method, and a quadratic 10-mode model that satisfies the physics constraints discussed in section IV.E is obtained. Linear and quadratic models for the lift-coefficient are also identified. The results are cross-validated by using the discovered governing equations to reconstruct a base flow and a flow actuated at $t^+ = 1.4$.

## A. Quadratic 10-Mode Model

To obtain a sparse set of governing equations for the stalled NACA-0009 system, only the first $n = 10$ POD modes were used to train the constrained SINDy algorithm. Using less then $n = 9$ modes results in high 2-norm errors, while model accuracy and sparsity drop rapidly if an excess of $n = 14$ modes are used. This phenomenon is shown in figure 4, where optimal models were obtained by implementing the AIC scoring procedure over a logarithmic range of $\lambda$ values for each case $n$. AIC scores for the $n = 8$, $n = 10$, $n = 11$ and $n = 15$ models are shown in figure 3. Although the selected $n = 11$ and $n = 12$ models have smaller 2-norm errors than the $n = 10$ mode model, they are more computationally intensive since the constraint matrix $C$ grows as $\frac{n!}{3!(n-3)!} = \frac{1}{6}n(n-1)(n-2)$. Furthermore, the $n = 10$ model corresponds to a higher sparsity in the coefficient matrix $\Xi$, and has only 201 active terms compared to 394 active terms in the $n = 11$ model and 440 active terms in the $n = 12$ model. The balance of lower complexity and lower error (relative to the other models) within the $n = 10$ mode model made it the superior choice for this work.

A disadvantage of using the $n = 10$ mode model is that it captures only 68% of the mean-subtracted flow energy, as shown in figure 2. A common approach in literature is to incorporate the dynamics described by higher-order POD modes is to introduce an additional mode, called the shift mode [19]. The shift mode captures the transients connecting the unstable steady state to the mean of the steady-state vortex shedding [6, 19]. Implementation of a shift mode was attempted for the $n = 10$ model, however it was found to not improve accuracy while reducing sparsity. Details of the shift-mode analysis can be found in the appendix section VIII.A.




(a) AIC for $n = 8$

(b) AIC for $n = 10$

(c) AIC for $n = 11$

(d) AIC for $n = 15$

**Fig. 3 Optimal model selection for $n = 8, 10, 11, 15$ mode models. Here, $(AIC_r)$ are the AIC relative scores and the sparsity ration in $\Xi$ is a function of the sparsification constant. The low system density (high sparsity) for the $n = 8$ and $n = 15$ are associated with very high 2-norm errors as is shown in figure 4. Due to these high errors, the optimal $n = 8$ and $n = 15$ models do not provide meaningful models of the stalled airfoil flow.**

An optimal sparsification constant of $\lambda = 0.3533$ was identified for the $n = 10$ mode model using the AIC model-selection method. Figure 5a shows the optimal point obtained from an AIC sweep over a logarithmic range of $\lambda$ values, resulting in a coefficient matrix ($\Xi$) that is 69.5% sparse (with 201 active terms). The governing equations for the low-order modes that correspond to higher energy contributions tended to be more sparse than those for the high-order modes, as is seen in 5b. Using the identified $\lambda$, constrained SINDy was invoked to find the left hand side of equation (15): $\dot{A} = \Theta(A)\Xi$.. The left hand side obtained from SINDy ($\dot{A}_{model}$) is shown plotted against the DNS state-derivative ($\dot{A}_{DNS}$) in figure 6. The error in $\dot{A}_{model}$ was found to be negligible, and the constrained SINDy algorithm accurately predicted the state-derivative evolution of the NACA-0009 system.

A Runge-Kutta solver was then implemented to integrate the ODEs described by equation (15) from an initial condition and obtain the POD coefficients from SINDy ($A_{model}$). The results of this integration were compared to the DNS ($A_{DNS}$), as seen in figure 7. Unlike the derivatives $\dot{A}_{model}$ presented in figure 6, the POD coefficients ($A_{model}$) exhibited a mismatch compared to the DNS solution as time increased. This issue also caused the variability in the 2-norm error plot in figure 5, as the error was computed by comparing the integrated model with data (see equation (24)). Several time-marching schemes were tested to solve the ordinary differential equation (15) with similar results.

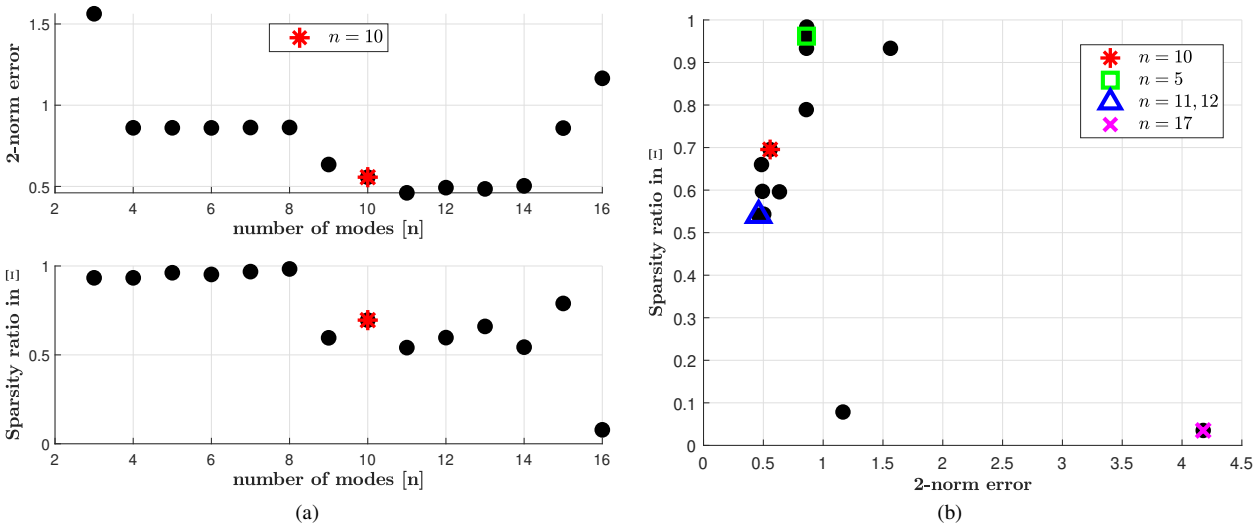



**Fig. 4 Optimal quadratic models for $n = 3$ through $n = 17$ POD modes. Although the $n = 11$ mode result corresponds to the lowest 2-norm error, the $n = 10$ mode provides a more favorable complexity-misfit trade-off.**

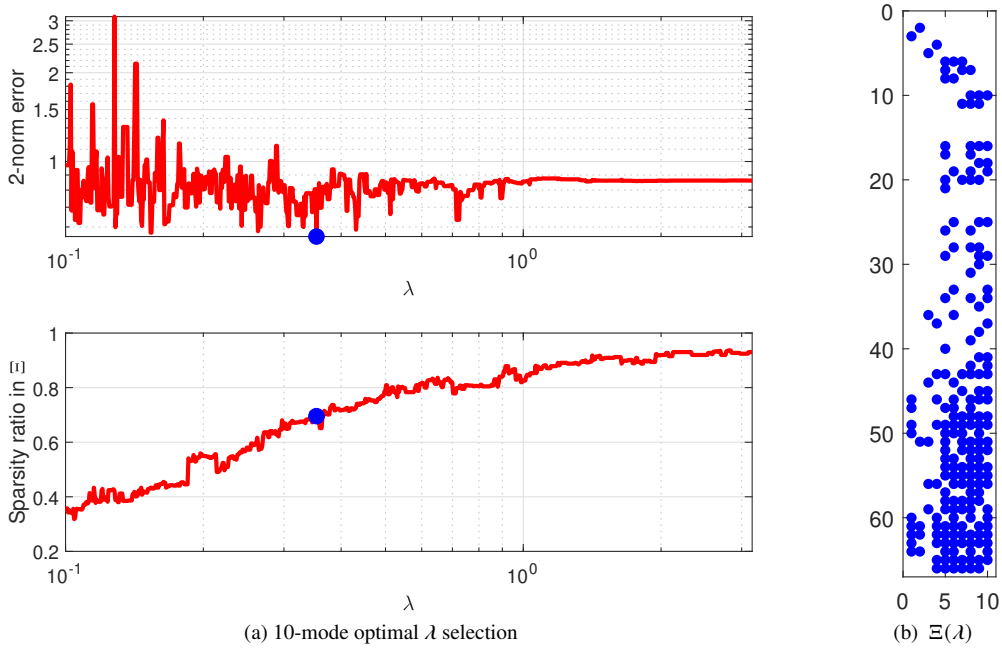



**Fig. 5 Optimal model for $n = 10$ POD modes. The optimal sparsification value was found to be $\lambda = 0.3533$ using the AIC model-selection method, and is shown as a blue point on both the top and bottom left plots. The right-hand plot shows the active terms within the coefficient matrix $\Xi$ for the optimal 10-mode SINDy model. Here, the columns correspond to the states and the rows correspond to the various quadratic combinations of the state variables.**

The modal phase-drift observed in figure 7 is possibly the consequence of either inefficiencies within the Runge-Kutta solver, or the sensitivity of the quadratic terms in the discovered governing equations to errors or perturbations.

The effects of the phase-drift observed in the model can also be observed in figure 8, where the DNS data was projected onto $n = 200$ modes and $n = 10$ modes to find the vorticity fields. The resulting vorticity fields were compared with the quadratic SINDy model with $n = 10$ modes at several instances in time. Although the initial vorticity field agrees closely with the DNS data, the model predictions become less accurate at the later time steps.

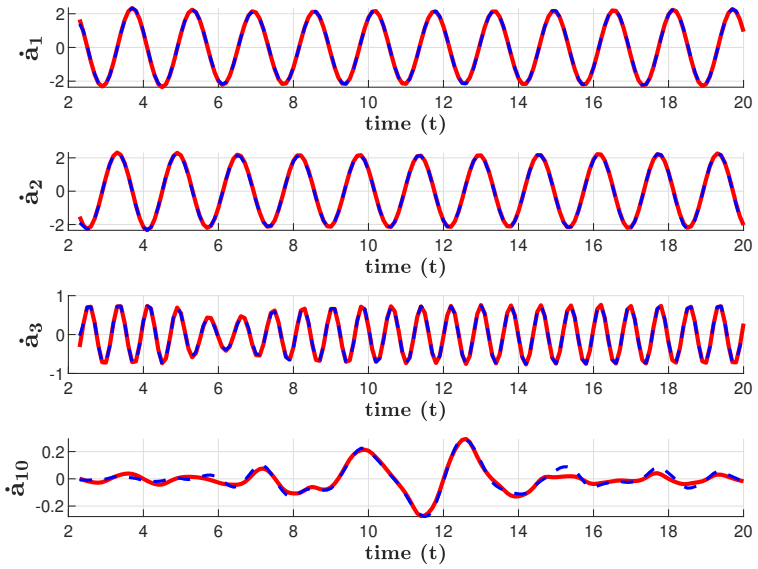



**Fig. 6 In red: the derivatives ($\dot{A}_{model}$) of the POD coefficients obtained from SINDy using equation (15). In blue: the derivatives of the DNS POD coefficients ($\dot{A}_{DNS}$). The SINDy model is quadratic and consists of 10 POD modes with a sparsification value of $\lambda = 0.3533$.**

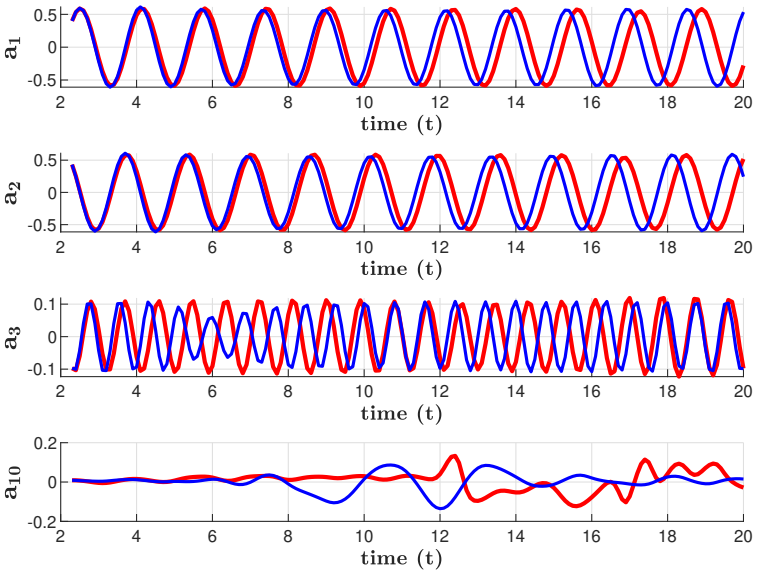
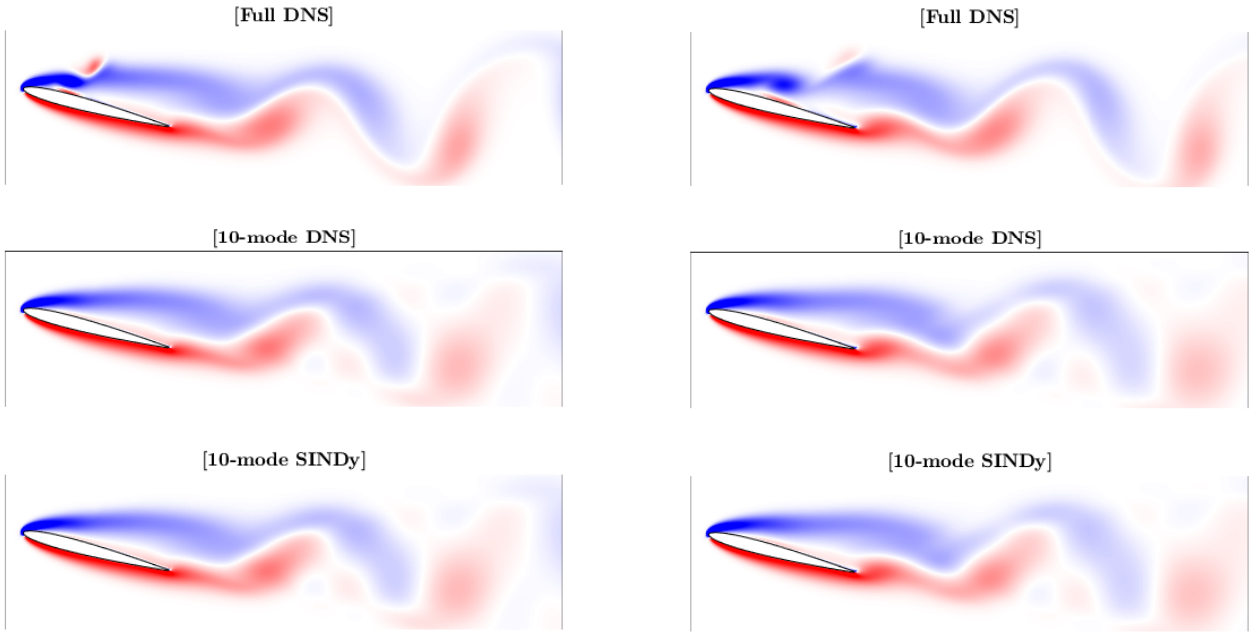


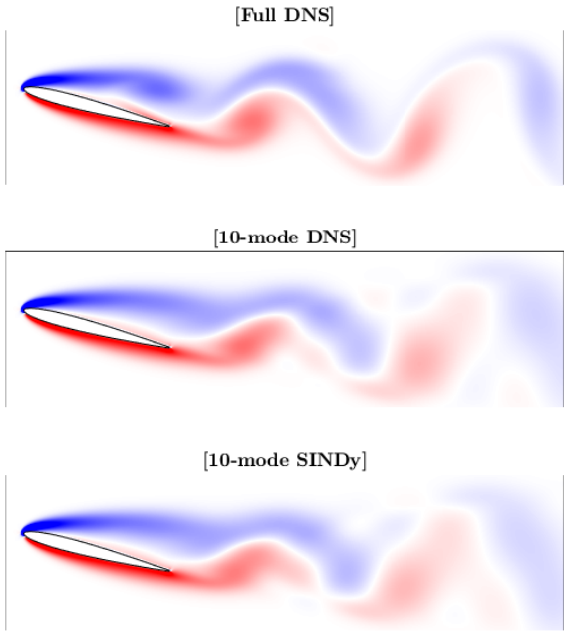

**Fig. 7 The POD coefficients ($A_{model}$) obtained by integrating the result from figure 6 (shown in red) vs. the the POD coefficients obtained from DNS ($A_{DNS}$) shown in blue. The SINDy model is quadratic and consists of 10 POD modes with a sparsification value of $\lambda = 0.3533$.**



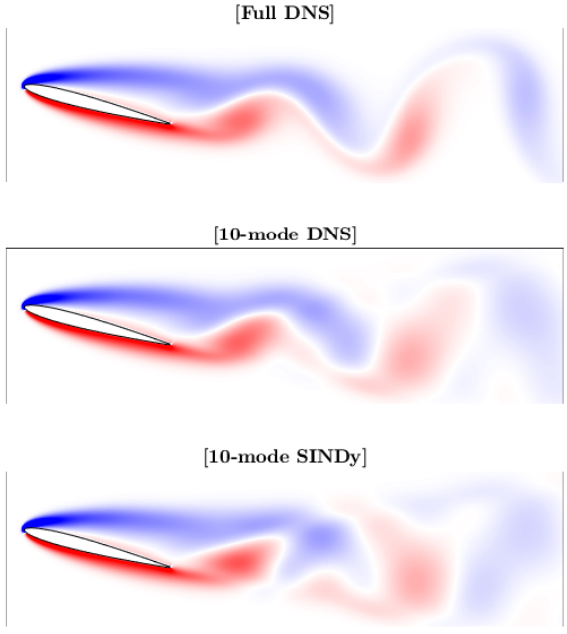

(a) $t^+ = 2.3$, end of actuation




(b) $t^+ = 2.7$, lift reversal ($\Delta C_{l,min}$)




(c) $t^+ = 3.6$, lift enhancement ($\Delta C_{l,max}$)




(d) $t^+ = 10$ return to baseline flow

**Fig. 8 Vorticity fields obtained at various times. The top plots shows the vorticity fields obtained from the DNS data projected onto $n = 200$ modes, the middle plots shows DNS data projected onto $n = 10$ modes, and the bottom plots are the SINDy results with $n = 10$ modes. The initial three figures are consistent with the DNS model, however the phase-drift observed in figure 7 reduces the accuracy of the prediction over time.**

### B. Quadratic and Linear Output Equation Models

A model for the lift coefficient $C_\ell$ was also obtained by implementing unconstrained SINDy (as discussed in section IV.C) to solve the output equation (6b): $\boldsymbol{y} = g(A)$. Two sets governing equations were discovered by regressing the POD-coefficient data onto the DNS lift coefficient data. The first model was obtained by training SINDy on the DNS POD-coefficient data $A_{DNS}$, so as to obtain a system without the phase-shift effects that resulted from the integration of $\dot{A}_{model}$, as discussed in section V.A. The second model was discovered by training SINDy onto the modeled POD coefficients ($A_{model}$) shown in figure 7. This resulted in the output equation models $\boldsymbol{y} = g_{model}(A)$ and $\boldsymbol{y} = g_{DNS}(A)$, where the subscripts on the function $g$ indicate whether the system was trained on DNS or the discovered dynamic-equation output (modeled) data.

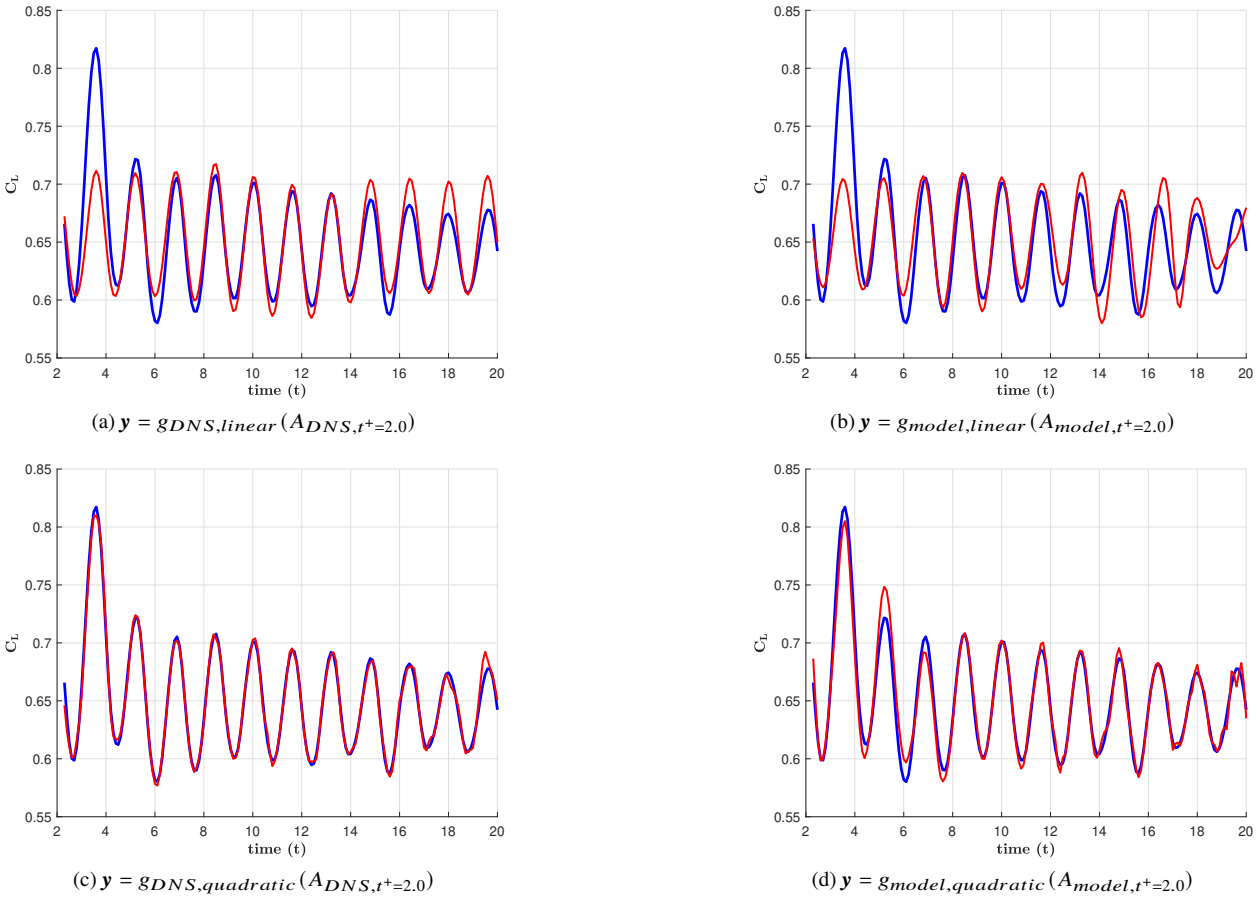
Both linear and quadratic fits were investigated for the two aforementioned cases, as is shown in figure 9. The sparsification constant $\lambda$ was set to zero to preserve accuracy, resulting in non-sparse models. It was found that the quadratic models more accurately modeled the training data, though they were more sensitive to errors and off-design inputs which may indicate over-fitting. The linear models did not accurately capture the maxima and minima in the lift, but produced smoother functions in general. For the $\boldsymbol{y} = g_{DNS,linear}(A_{DNS,t^+=2.0})$ system, a test was performed with $n = 40$ modes, which did result in a close fit with the DNS $C_\ell$ data. However, since the dynamic model was found to be optimal for $n = 10$ modes, this result was omitted in this work. A cubic fit was also implemented for both the $A_{model}$ and $A_{DNS}$ cases. Although the accuracy of these models was better than both the linear and quadratic cases (as detailed in Appendix VIII.B), this resulted in an definitively over-fit model that did not provide meaningful results when off-design datasets were used as inputs.

### C. Model Cross-Validation with off-Design Input Data

The governing equations discovered by training the constrained SINDy algorithm on the $t^+ = 2.0$ flow were cross-validated with DNS data from a case with actuation at $t^+ = 1.4$ and an unactuated flow. This was done by using the model discussed in the previous section and solving equation (15) for inputs and initial conditions corresponding to the $t^+ = 1.4$ and unactuated flows.

Modes $a_1$ and $a_2$ of both the unactuated and $t^+ = 1.4$ models follow the DNS data for the first few timesteps before diverging (figure 10), which is consistent with the phase-drift that was observed for the training model. Mode $a_3$ for the unactuated model exhibits immediate out-of-phase behavior before temporarily converging with the DNS result at times $t = 5$ and $t = 14$, while mode $a_4$ exhibits the same initial accuracy as modes $a_1$ and $a_2$. The case actuated at $t^+ = 1.4$ also shows agreement at initial time steps, although higher-order modes (including $a_3$ and $a_4$) have a different frequency from the DNS data. The results shown in figure 10 may imply the same problems with model sensitivity to perturbations as the training case.

A similar procedure was repeated to obtain output equations for both the unactuated and $t^+ = 1.4$ cases. As in section V.B, both the integrated coefficients ($A_{model}$) shown in figure 10 and DNS POD Coefficients ($A_{DNS}$) were used as inputs to the RHS of the output equation (6b) to find the lift coefficients $C_\ell$. The quadratic models shown in figure 9 did not provide meaningful results when $A_{DNS}$ and $A_{model}$ were used as inputs. However, the linear models performed better, as shown in figure 11. The increasing discrepancy between the results and data as time increases for the $A_{model}$ cases is consistent with the phase drift observed in figures 10 and 7. The cross-validation for the $t^+ = 1.4$ also showed immediate out-of-phase behavior, since the training data set is out of phase with the $t^+ = 1.4$ data. These results indicate that the output equation model obtained in section V.B is not sufficiently accurate for use in analysis of off-design flows.



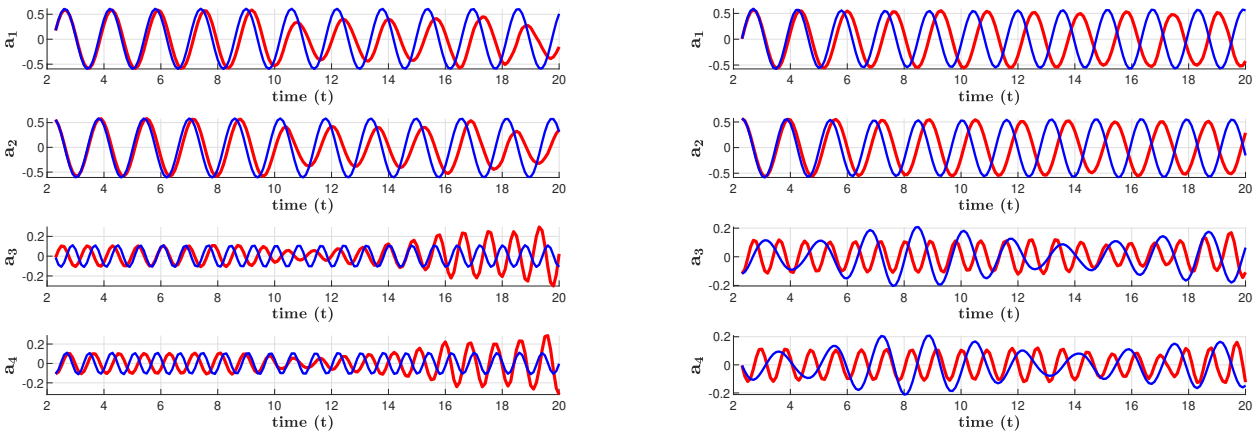

(a) $\boldsymbol{y} = g_{DNS,linear}(A_{DNS,t^+=2.0})$

(b) $\boldsymbol{y} = g_{model,linear}(A_{model,t^+=2.0})$

(c) $\boldsymbol{y} = g_{DNS,quadratic}(A_{DNS,t^+=2.0})$

(d) $\boldsymbol{y} = g_{model,quadratic}(A_{model,t^+=2.0})$

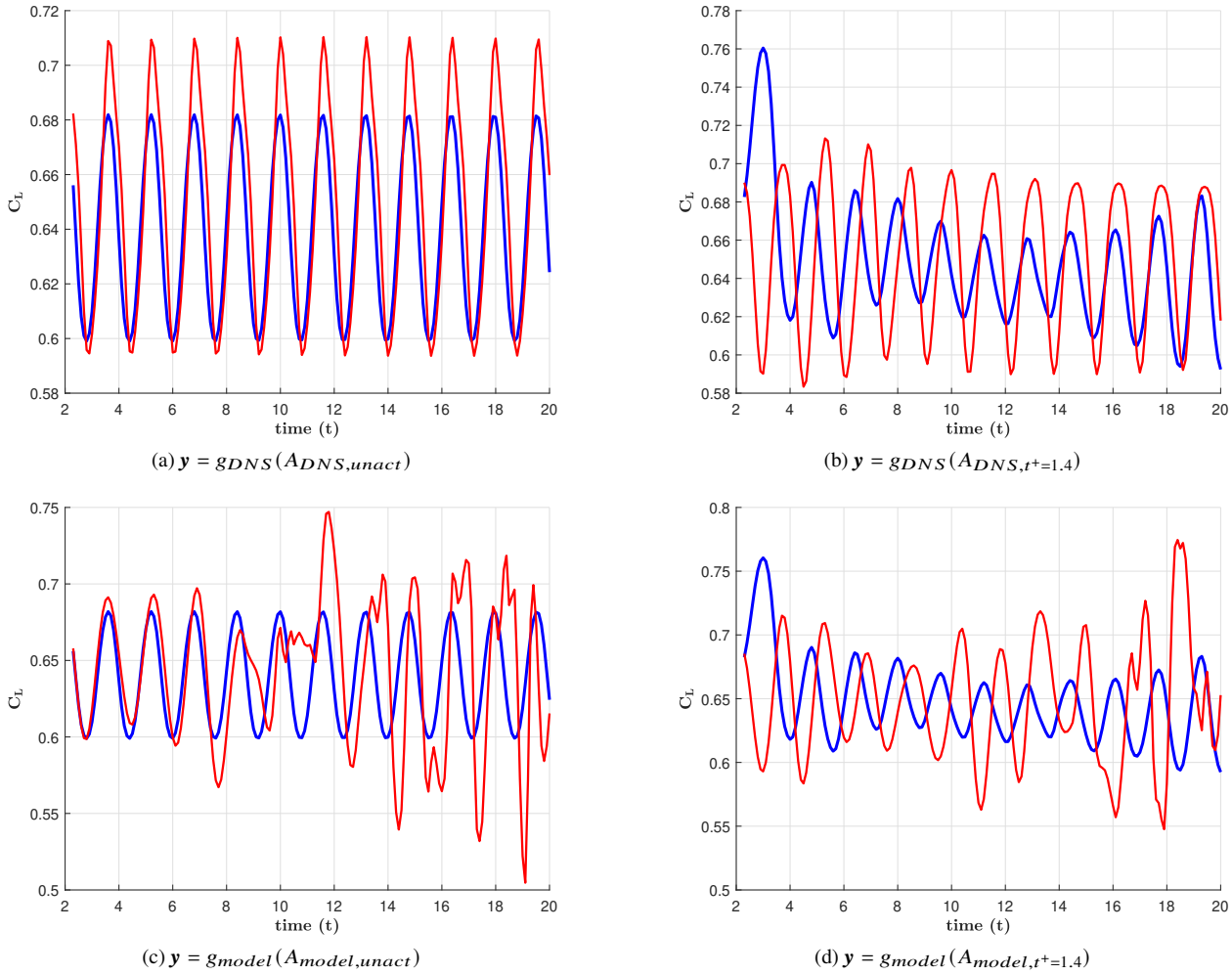
**Fig. 9 Left: Linear and quadratic output equations ($\boldsymbol{y}$) for $n = 10$ modes, trained on either DNS or modeled data, and with $A_{DNS}$ or $A_{model}$ inputs. The subscript $t^+ = 2.0$ on $A_{DNS}$ emphasizes that the dataset employed for the rest of the analysis was used. The blue lines represent the DNS $C_\ell$ data and red lines the identified models. The linear output models fail to capture the lift enhancement that occurs at $t^+ = 3.6$, but are locally less sensitive to perturbations and errors. Conversely, the quadratic models better capture the lift oscillations but are more sensitive.**




(a) Cross-validation without actuation

(b) Cross-Validation with actuation at $t^+ = 1.4$

**Fig. 10 Cross-validation performed using the training dataset model (with actuation at $t^+ = 2.0$) on an unactuated flow and a flow with actuation at $t^+ = 1.4$. The DNS data is shown in blue.**




(a) $\boldsymbol{y} = g_{DNS}(A_{DNS,unact})$ (b) $\boldsymbol{y} = g_{DNS}(A_{DNS,t^+=1.4})$

(c) $\boldsymbol{y} = g_{model}(A_{model,unact})$ (d) $\boldsymbol{y} = g_{model}(A_{model,t^+=1.4})$

**Fig. 11 Cross-validation of linear output equations ($\boldsymbol{y}$) for $n = 10$ modes, trained on either DNS or modeled data, and with $A_{DNS}$ or $A_{model}$ inputs. The subscript $t^+1.4$ on the input corresponds to actuation at $t^+ = 1.4$, and *unact* subscript indicates the unactuated flow. $C_\ell$ DNS data is shown in blue.**

## VI. Conclusion

Governing equations were obtained from data for a two-dimensional, incompressible NACA-0009 airfoil in stall (at $\alpha = 15°$) with short-duration momentum injection above the leading edge. Direct numerical simulation data of the NACA-0009 flow [1] was reduced using proper orthogonal decomposition. A constrained version of "Sparse Identification of Nonlinear dynamics" (SINDy) [6, 7] was implemented to take advantage of the lossless property of the quadratic nonlinearity in the incompressible Navier–Stokes equation. This allowed for the discovered governing equations to be consistent with the physical equations that represent unsteady, incompressible flows. A model selection algorithm was implemented so that the resultant equations balanced accuracy and complexity, and a framework to determine output equations for flow-dependent parameters was implemented to obtain the lift coefficient ($C_\ell$). A quadratic ten-mode dynamic model with 201 active terms was found. Linear and quadratic output equations that described the lift coefficient as a function of the flow-state were also discovered. The model was then cross-validated using off-design datasets.

It is demonstrated that constrained SINDy is an effective means of obtaining nonlinear reduced-order models (ROM) for complex, multimodal flows such as a two-dimensional airfoil system. Furthermore, it is shown that it is possible to obtain output equations for relevant parameters using the same framework. However, because unsteady flows with complex geometries feature tightly coupled dynamics and multi-modal instabilities, the equations were found to be

extremely sensitive to noise and small perturbations. Although constrained SINDy correctly predicted the derivatives of the state, a phase-shift was observed in the results when integration of the discovered ODEs was performed. Furthermore, by reducing the model to ten modes, many local flow phenomena were not captured. Similarly, the output equation models suffered from issues with over-fitting and a loss of generality. These issues became more apparent for both the dynamic and output equations when cross-validation was performed with off-design data.

The results of this work motivate the development of future model-based analysis methods using the SINDy framework. Although the described approach can be used for model-based analysis, robustness to small perturbations must be improved to obtain models that can be used to control separated flows in real-time. Nonlinear stability analysis methods such as those described in [24, 25] can be used to identify instabilities within the model. Global stability can also be enforced for SINDy models by again leveraging the energy conserving property of the quadratic nonlinearity to define regions to which the system must converge [26, 27]. These efforts will allow for the development of stable, robust data-driven models with more modes.

## VII. Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0434.

## References

[1] Asztalos, K. J., Dawson, S. T., and Williams, D. R., "The sensitivity of leading-edge momentum injection response to instantaneous flow state for an airfoil in deep stall," American Institute of Aeronautics and Astronautics Inc, AIAA, 2020, pp. 1–17. https://doi.org/10.2514/6.2020-0577.

[2] Raju, R., Mittal, R., and Cattafesta, L., "Dynamics of airfoil separation control using zero-net mass-flux forcing," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3103–3115. https://doi.org/https://doi.org/10.2514/1.37147.

[3] An, X., Williams, D. R., da Silva, A. F., Colonius, T., and Eldredge, J., "Response of the Separated Flow over an Airfoil to a Short-Time Actuator Burst," *47th AIAA Fluid Dynamics Conference*, 2017, p. 3315. https://doi.org/https://doi.org/10.2514/6.2017-3315.

[4] Deem, E., Cattafesta, L., Hemati, M., Zhang, H., Rowley, C., and Mittal, R., "Adaptive Separation Control of a Laminar Boundary Layer using Online Dynamic Mode Decomposition," *Journal of Fluid Mechanics*, Vol. 903, No. A21, 2020.

[5] Bhattacharjee, D., Klose, B., Jacobs, G., , and Hemati, M., "Data-driven selection of actuators for optimal control of airfoil separation," *Theoretical and Computational Fluid Dynamics*, Vol. 34, 2020.

[6] Brunton, S. L., Proctor, J. L., Kutz, J. N., and Bialek, W., "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 113, 2016, pp. 3932–3937. https://doi.org/10.1073/pnas.1517384113.

[7] Loiseau, J. C., and Brunton, S. L., "Constrained sparse Galerkin regression," *Journal of Fluid Mechanics*, Vol. 838, 2018, pp. 42–67. https://doi.org/10.1017/jfm.2017.823.

[8] Schmid, P. J., "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, Vol. 656, 2010, pp. 5–28.

[9] Lucia, D. J., Beran, P. S., and Silva, W. A., "Aeroelastic system development using proper orthogonal decomposition and Volterra theory," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 509–518. https://doi.org/10.2514/1.2176.

[10] Berkooz, G., Holmes, P., and Lumley, J. L., "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.

[11] Hemati, M. S., Dawson, S. T. M., and Rowley, C. W., "Parameter-Varying Aerodynamics Models for Aggressive Pitching-Response Prediction," *AIAA Journal*, Vol. 55, No. 3, 2017, pp. 693–701. https://doi.org/10.2514/1.J055193.

[12] Dawson, S. T. M., Schiavone, N. K., Rowley, C. W., and Williams, D. R., "A Data-Driven Modeling Framework for Predicting Forces and Pressures on a Rapidly Pitching Airfoil," *45th AIAA Fluid Dynamics Conference*, 2015, p. 2767. https://doi.org/10.2514/6.2015-2767.

[13] Conlin, N., Tithof, J., and Hemati, M. S., "Data-Driven Modeling and Control of Oscillatory Instabilities in a Kolmogorov-like Flow," *manuscript in preparation*, 2022.

[14] Taira, K., and Colonius, T., "The immersed boundary method: a projection approach." *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 2118–2137.

[15] Colonius, T., and Taira, K., "A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, 2008, pp. 2131–2146.

[16] Liska, S., and Colonius, T., "A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains," *Journal of Computational Physics*, Vol. 316, 2016, pp. 360–384. https://doi.org/https://doi.org/10.1016/j.jcp.2016.04.023.

[17] Liska, S., and Colonius, T., "A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions," *Journal of Computational Physics*, Vol. 331, 2017, pp. 257–279. https://doi.org/https://doi.org/10.1016/j.jcp.2016.11.034.

[18] Rubini, R., Lasagna, D., and Ronch, A. D., "The $l_1$-based sparsification of energy interactions in unsteady lid-driven cavity flow," *Journal of Fluid Mechanics*, 2020, p. 905:A15.

[19] Noack, B. R., Afanasiev, K., Morzynski, M., Tadmor, G., and Thiele, F., "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *Journal of Fluid Mechanics*, Vol. 497, 2003, pp. 335–363.

[20] Tibshirani, R., "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, 2005, pp. 267–288.

[21] Rempfer, D., and Fasel, H. F., "Dynamics of three-dimensional coherent structures in a flat-plate boundary layer," *Journal of Fluid Mechanics*, Vol. 275, 1994, pp. 257–283.

[22] Balajewicz, M. J., Dowell, E. H., and Noack, B. R., "Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation," *Journal of Fluid Mechanics*, Vol. 729, 2013, pp. 285–308.

[23] Mangan, N. M., Askham, T., Brunton, S. L., Kutz, J. N., and Proctor, J. L., "Model selection for hybrid dynamical systems via sparse regression," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 475, 2019. https://doi.org/10.1098/rspa.2018.0534.

[24] Kalur, A., Seiler, P., and Hemati, M. S., "Nonlinear stability analysis of transitional flows using quadratic constraints," *Phys. Rev. Fluids*, Vol. 6, 2021, p. 044401. https://doi.org/10.1103/PhysRevFluids.6.044401, URL https://link.aps.org/doi/10.1103/PhysRevFluids.6.044401.

[25] Kalur, A., Mushtaq, T., Seiler, P., and Hemati, M. S., "Estimating Regions of Attraction for Transitional Flows Using Quadratic Constraints," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 482–487. https://doi.org/10.1109/LCSYS.2021.3081382.

[26] Schlegel, M., and Noack, B. R., "On long-term boundedness of Galerkin models," 2013. URL http://arxiv.org/abs/1310.0053.

[27] Kaptanoglu, A. A., Callaham, J. L., Aravkin, A., Hansen, C. J., and Brunton, S. L., "Promoting global stability in data-driven models of quadratic nonlinear dynamics," *Physical Review Fluids*, Vol. 6, 2021. https://doi.org/10.1103/PhysRevFluids.6.094401.

# VIII. Appendix

## A. Shift Mode Implementation

POD provides a low-rank basis that is optimal in the $L_2$ sense, and the modes capture most of the energy contained within the original system. For this work, $n = 10$ POD modes were used, which captured approximately 68% of the energy for the actuated system as shown in figure 2. It is common in low-order modeling of fluids to introduce an additional mode, called the shift mode, to capture the transient dynamics connecting the unstable steady state to the mean of the steady-state vortex shedding [6]. A more detailed derivation of the shift mode can be found in [19]. The shift mode is constructed starting from the mean-field correction $u_\Delta^a$:

$$u_\Delta^a = u_0 - u_{eq}, \tag{27}$$

where $u_0$ is the mean flow and $u_{eq}$ is the equilibrium solution. Using the mean field correction, it is possible to compute a mode $\phi_\Delta^b$ that is orthogonal to the POD modes $\phi_i$:

$$\phi_\Delta^b = u_\Delta^a - \sum_{i=1}^{N} (u_\Delta^a \cdot \phi_i)\phi_i. \tag{28}$$

Normalizing $\phi_\Delta^b$ by its 2-norm yields the shift mode $\phi_\Delta$:

$$\phi_\Delta = \frac{u_\Delta^b}{||\phi_\Delta^b||_\Omega}. \tag{29}$$

Finally, to obtain a shift mode coefficient $a_\Delta$ that can be appended to the POD coefficient matrix, the shift mode $\phi_\Delta$ is projected onto the velocity field:

$$a_\Delta = X(:)^T \phi_\Delta, \tag{30}$$

where $X(:)^T$ is a matrix with columns consisting of the velocity field ($u$ and $v$ components) at every snapshot. The resultant shift coefficient can then be appended as a column to the matrix (12), resulting in an additional mode that is used in the analysis.

Although the shift mode is very effective in capturing transient dynamics for unsteady flows, it was found that in the case of the NACA-0009 system the shift mode did not increase the accuracy and reduced the sparsity of the model. Furthermore, it was found that the shift-mode's energy contribution was negligible relative to the other modes, as shown in 12. The minor contribution of the shift mode is possibly a result of the actuating signal's magnitude being very small. Due to this, the shift mode was excluded from the analysis of the NACA-0009 system.

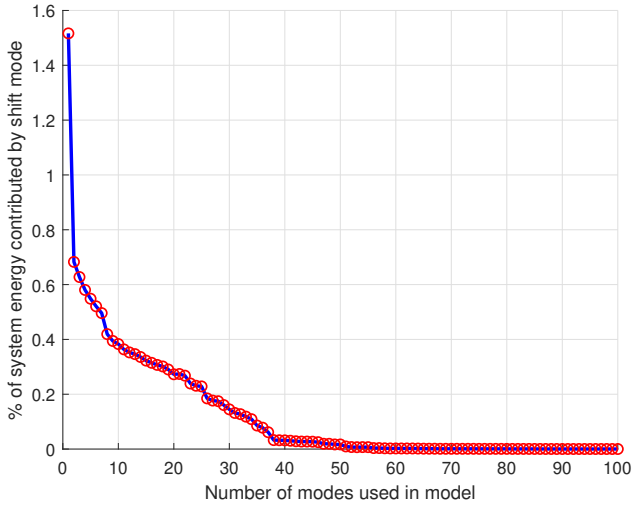



**Fig. 12 Energy contribution of the shift vs. the number of POD-modes used in the model. For the $n = 10$ mode model, the shift mode captured less than $0.4\%$ of the total energy within the flow.**

### B. Additional Output Equation Models

A cubic model for the output equations trained on the $t^+ = 2.0$ dataset is presented in figure 13. This model yielded the lowest error for the training dataset, but did not provide meaningful results for the cross-validation cases.

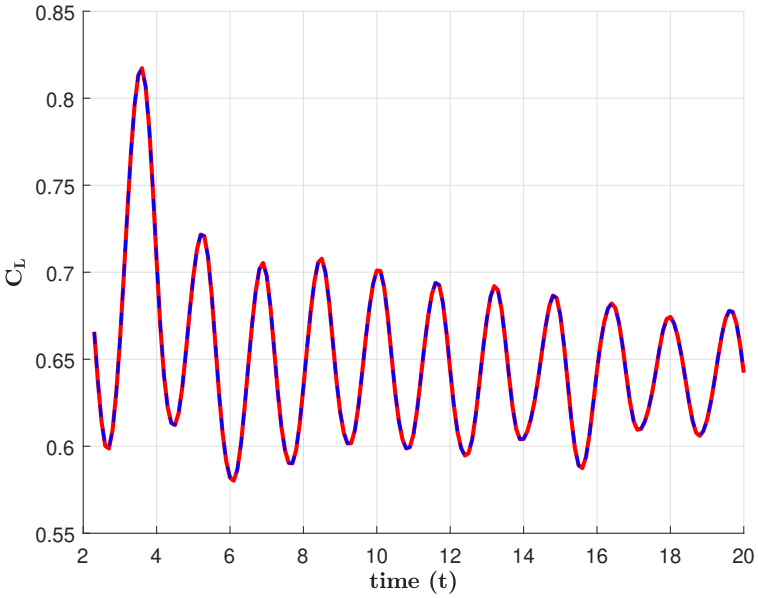



**Fig. 13 Cubic output equation model obtained from performing a linear least squares regression of the 10-mode SINDy system onto DNS lift coefficient data. Here blue represents the DNS data and red the discovered model.**