Aerospace Science and Technology 148 (2024) 109106

Contents lists available at ScienceDirect

Aerospace Science and Technology

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

Review

Data-driven controller and multi-gradient search algorithm for morphing airfoils in high Reynolds number flows

José M. Magalhães Júnior [a,*], Gustavo L.O. Halila [b], Kyriakos G. Vamvoudakis [a]

[a] Georgia Institute of Technology, GA, 30332, United States
[b] Embraer, Technology Development, São José dos Campos, SP, Brazil

ARTICLE INFO

Communicated by Raghvendra Cowlagi

ABSTRACT

In this paper, we propose a data-driven framework to control morphing airfoils in the subsonic flight regime, considering high Reynolds numbers with an efficient and safe way to reach a shape with improved values of the aerodynamic coefficients. The online solution is based on a data-driven controller combined with a surrogate model and a multi-gradient descent algorithm. Without full knowledge of the aerodynamic parameters (lift, drag, and pitching moment coefficients), the learning framework searches for an airfoil shape that minimizes a metric of performance associated to drag, lift, and pitching moment coefficients. The solution uses online data to improve the accuracy of the predictions of the aerodynamic coefficients provided by the surrogate model along the trajectory. The optimization framework focuses on subtle airfoil deformations to assure a smooth trajectory between the initial and the final shape. Finally, the efficacy and the robustness of our proposed solution was shown in numerical examples, resulting in a significant reduction in the prediction error.

1. Introduction

In aerospace design, drag reduction is a desired target due to fuel burn cutback, which has a positive impact on greenhouse gas emissions. To achieve reduced drag, aerodynamic shape optimization is currently used. Numerical optimization can enhance the consistent exploration of the design space and lead to trade-offs which would be difficult to achieve without resorting to numerical tools. One strategy to achieve reduced drag is the use of morphing configurations [1–4].

The advent of composite materials and advanced control techniques enable the design of morphing structures that can adapt to varying flight conditions with optimal performance. Moreover, morphing configurations eliminate the presence of gaps and steps which are typical in high-lift devices. Airplanes that employ morphing technologies can achieve optimal performance by minimizing drag over the entire flight envelope and leverage the use of laminar flow technologies [5].

Researchers have shown strong interest in morphing structures over the years. For instance, the work of [6] introduces a methodology of morphing which combines machine learning and adaptive dynamic model inversion control capable of learning the required shape of a morphing smart block for a given trajectory. The use of machine learning theory in morphing structures is also discussed by authors in [7] where they consider a Q-learning method combined with analytical aerodynamic calculations to determine the optimal shape by changing the maximum camber location, the airfoil incidence, thickness, and the camber. The investigation in [8] proposes a deep neural network and reinforcement leaning technique as a control strategy for shape-memory alloy actuators in the context of a morphing wing. A deep deterministic policy gradient (DDPG) algorithm based

Abbreviations: ADflow, Open-source computational fluid dynamics and shape optimization solver; ANK, Approximate Newton-Krylov; CFD, Computational Fluid Dynamics; CRM, Common Research Model; DDPG, Deep Deterministic Policy Gradient; DNN, Deep Neural Network; DOF, Degree of freedom; FFD, Free-form Deformation; HF-Model, High-Fidelity model; HPC, High-performance computing; LTI, Linear Time-Invariant; MAE, Mean absolute error; MGDA, Multi-gradient descent algorithm; MUSCL, Monotone Upstream-Centered scheme for conservation laws; RANS, Reynolds-Averaged Navier—Stokes; PTC, Pseudo-transient continuation; PyGeM, Python Geometrical Morphing; ReLU, Rectified Linear Unit; UAV, Unmanned Aerial Vehicle.

* Corresponding author.
E-mail address: jose.magalhaes@gatech.edu (J.M. Magalhães Júnior).

https://doi.org/10.1016/j.ast.2024.109106
Received 11 April 2023; Received in revised form 11 March 2024; Accepted 31 March 2024
Available online 5 April 2024

Nomenclature

α	angle of attack	$\overline{S}, \overline{T}, \overline{U}$	vectors on parallel-piped region used for FFD
$\alpha_1, \alpha_2, \alpha_3, \beta$	MGDA parameters	$u[k], y[k]$	input and output at time step k of a morphing airfoil system
B_L, B_U	lower and upper constraints on values of c_l, c_d, and c_m	u_r, y_r	reference trajectory (input and output)
c_l, c_d, c_m	lift, drag, and pitching moment coefficients	X_0, X_F	initial and final shapes
$f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, $f_{c_m}(\cdot)$	nonlinear functions of c_l, c_d and c_m	$X[k]$	states $(x_1, x_2, \ldots, x_{20})$ at time step k of a morphing airfoil system
m	number of steps to morph in every iteration	$x_i[k]$	state i at time step k of a morphing airfoil system
N	prediction horizon	(x_s, y_s)	ordered pairs that define an airfoil shape
$\Phi(\cdot)$	activation function of neural network		
Re	Reynolds number		

on the Actor-Critic framework is proposed in [9] to solve the problem of deep reinforcement learning morphing control in continuous action space applied to a bionic bird wing-foldable UAV model.

In our recent work [10], we developed a data-enabled predictive controller allied to a surrogate model and a simulated annealing search algorithm to safely actuate a morphing airfoil according to a lift constraint. The present work extends our previous results and incorporates a multi-gradient search algorithm allied to deep neural networks to find higher performance shapes in a multi-objective optimization problem with safety constraints related to lift, drag and pitching moment, and measurements collected online in a certain flight condition. Unlike the aforementioned studies, this work aims to search for shapes with better values for the three aerodynamic parameters and takes advantage of measurements collected online to predict trajectories between the initial and final shapes that minimize the effect of morphing on the aircraft stability characteristics for efficiency and safety.

Contributions. The contributions of the present paper are threefold. First, we develop a surrogate model, based on a deep neural network (DNN), to perform an offline prediction of the values of drag, lift, and pitching moment using the morphing airfoil model developed in [10]. Then, a multi-gradient descent search algorithm (MGDA) is used to find an improved shape and the resulting trajectory. Finally, we develop a data-based learning algorithm to prevent the system from reaching unsafe conditions during the shape morphing process.

Structure. The remainder of the paper is structured as follows. Section 2 formulates the problem and provides relevant mathematical background. In Section 3, we describe the surrogate model to predict the aerodynamic parameters, while Section 4 presents the multi-gradient search algorithm. We present results related to the data-driven shape controller used during the online learning in Section 5. The online learning algorithm used to prevent the system from unsafe conditions is introduced in Section 6. Finally, Section 7 concludes the present work and discusses future research directions.

2. Problem formulation

Consider an airfoil that can change shape in one dimension (1D), i.e., in the y-direction, starting from a baseline shape. We shall use the NACA 2412 airfoil as the baseline configuration since it is widely used in low-speed, general aviation wings. The shape and its corresponding computational fluid dynamics (CFD) surface mesh are defined by $1,102$ ordered pairs (x_s, y_s) distributed over the surface of the airfoil. In order to control the shape, instead of using all the $1,102$ pairs, we shall instead use 20 points by applying a technique called free-form deformation (FFD) [11,12].

2.1. Morphing airfoil modeling

Define the morphing airfoil as a discrete-time dynamical system of the form

$$\begin{aligned} x[k+1] &= x[k] + u[k] \\ y[k] &= f(x[k]) = [f_{c_l}(x[k]), f_{c_d}(x[k]), f_{c_m}(x[k])]^{\mathrm{T}}, \quad k = 0, 1, \ldots, L,\ x[0] = \bar{x}, \end{aligned} \tag{1}$$

where $x[k] \in \mathbb{R}^{20}$ is the state of the system, $u[k] \in \mathbb{R}^{20}$ is the control input that drives the position of the ordered pairs, and $y[k] \in \mathbb{R}^3$ is the output vector consisting of the lift, drag, and pitching moment coefficients at a given shape, i.e., $y = [f_{c_l}(\cdot), f_{c_d}(\cdot), f_{c_m}(\cdot)]^{\mathrm{T}}$, where $f_{c_l} : \mathbb{R}^{20} \to \mathbb{R}$, $f_{c_d} : \mathbb{R}^{20} \to \mathbb{R}$, and $f_{c_m} : \mathbb{R}^{20} \to \mathbb{R}$ denote the unknown nonlinear functions representing the lift, drag, and pitching moment coefficients, respectively. The following assumptions are now needed.

Assumption 1. The system (1) is stabilizable and $x[k]$ can be measured for all values of $k \in \{0, 1, \ldots, L\}$. □

Assumption 2. The nonlinear functions $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ are bounded and continuous. Moreover, small variations on the shape $x[k]$ for all values of $k \in \{0, 1, \ldots, L\}$ imply small variations on the values of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$. This assumption is directly related to the physical characteristics of the phenomenon and does not affect the size of the search space. □

Assumption 3. The flight condition, defined by the angle of attack (α), Reynolds (Re), and Mach numbers, remains unchanged during the whole morphing procedure. □

Let $\mathcal{X} \subseteq \mathbb{R}^{20}$ be the set of all the states reachable from the origin in k steps of the system (1), i.e., the set of all states $x[k]$ obtained starting from the initial condition $x[0]$. The purpose of this work is to find an improved shape denoted as $x^\star \in \mathcal{X}$ that has an increased value of c_l/c_d, an

increased value of c_l, a reduced value of c_d, and a desired $|c_m|$ not greater than the initial value. The improved shape can be found by solving the following multi-objective optimization problem:

$$\begin{aligned}
\underset{u\in\mathbb{R}^{20}}{\text{minimize}} \quad & \mathbf{F}(\mathbf{x}) = \left[-\frac{f_{c_l}(\mathbf{x}[L])}{f_{c_d}(\mathbf{x}[L])} \;,\; -f_{c_l}(\mathbf{x}[L]) \;,\; f_{c_d}(\mathbf{x}[L]) \right] \\
\text{subject to} \quad & x[k+1] = x[k] + u[k], \quad \forall k \in \{0, \cdots, L-1\} \\
& f_{c_l}(x[L]) \geqslant f_{c_l}(x[0]) \\
& f_{c_d}(x[L]) < f_{c_d}(x[0]) \\
& |\, f_{c_m}(x[L])\,| \leqslant |\, f_{c_m}(x[0])\,| \\
& x[0] = \bar{x}.
\end{aligned} \tag{2}$$

Since $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ are unknown nonlinear functions, a surrogate model must be constructed. This problem can be summarized as follows.

Problem 1. Find a shape with higher $\left(\frac{c_l}{c_d}\right)$, higher c_l, lower c_d, and feasible c_m that can be tractably estimated given that the functions $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ are unknown. □

Since morphing the wing airfoil can affect the stability of the aircraft, we will morph the shape while we guarantee efficiency and safety. Efficiency means that morphing occurs only when a new shape $x[\cdot]$ with improved values of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ exists and can be reached; while safety means that the trajectory from the baseline to the new shape will not reach an intermediate shape with "unsafe" values. Thus, the constraints can be mathematically described as follows:

$$\begin{aligned}
f_{c_l}(x[k]) &\geqslant 0.98 f_{c_l}(x[0]), \\
f_{c_d}(x[k]) &\leqslant 1.02 f_{c_d}(x[0]), \\
|\, f_{c_m}(x[k])\,| &\leqslant 1.05 |\, f_{c_m}(x[0])\,|, \\
\forall k &\in \{1, 2, \ldots, L\text{-}1\}.
\end{aligned} \tag{3}$$

It is known that a surrogate model has inherent errors on the prediction and thus in order to improve the prediction of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$, we shall use data gathered along the system's generated trajectories. This problem can be summarized as follows.

Problem 2. Estimate $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ in an online manner, using data gathered along the trajectory of (1) while preventing the system from reaching any "unsafe" values during the morphing procedure. □

3. Surrogate model and shape optimization

Here, we will deal with Problem 1. The approach used for modeling is based on a DNN where the training dataset is generated by using a CFD solver, for which more details are provided in Section 3.3. Once the surrogate model is obtained, an MGDA will be used to solve (2). Although the CFD simulations can provide accurate predictions of lift, drag, and pitching moment coefficients, i.e., c_l, c_d, and c_m, respectively, they still require a considerable computational cost. We are thus interested in developing a surrogate model based on a DNN to accurately predict the aerodynamic parameters. This development consists of a generation of 9,600 morphing shapes, from a baseline shape to train 3 fully connected DNNs.

3.1. Parametric method

The FFD technique has been used in shape optimization [5,13] and is based on the tensor product trivariate Bernstein polynomial [14–16] that inserts the shape representation into a rectangular box composed by the control points. A subsequent transformation recomputes the position of the solid points with respect to the modified points of the lattice.

The vectors $\overrightarrow{S}$, $\overrightarrow{T}$, and $\overrightarrow{U}$ define the size and the orientation of the control lattice. The lattice is uniformly layered by planes in their directions with several deformable cuboids to embed the object as shown in Fig. 1. Given several established basis functions (trivariate Bernstein polynomials), the changing positions of the vertexes of the cuboids (FFD control points) cause the inner model to deform [13].

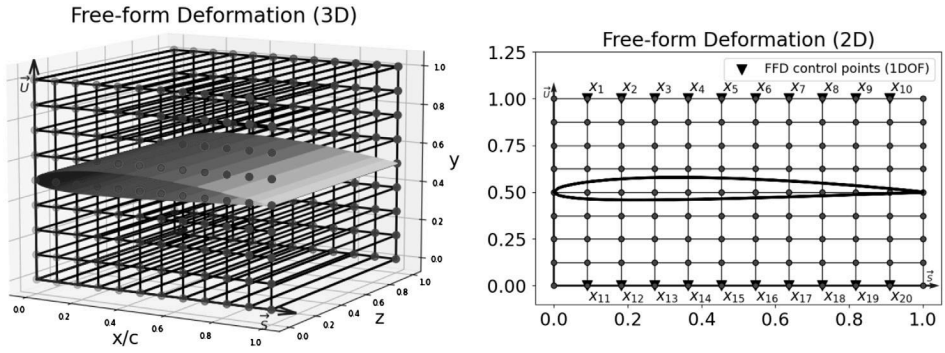
Every object point $X(x, y, z)$ interior to the control lattice has (s, t, u) coordinates in the coordinate system fixed to the FFD box as follows,

$$\overrightarrow{X} = \overrightarrow{X}_0 + s\overrightarrow{S} + t\overrightarrow{T} + u\overrightarrow{U}. \tag{4}$$

Let $P_{ijk}, i = 0, \cdots, l, j = 0, \cdots, m, k = 0, \cdots, n$ be the control points on the lattice. The deformation is represented by a movement of the control points P_{ijk} from their positions. The new position of an arbitrary point $X^\star(x(s), y(t), z(u))$ inside the lattice is computed as,

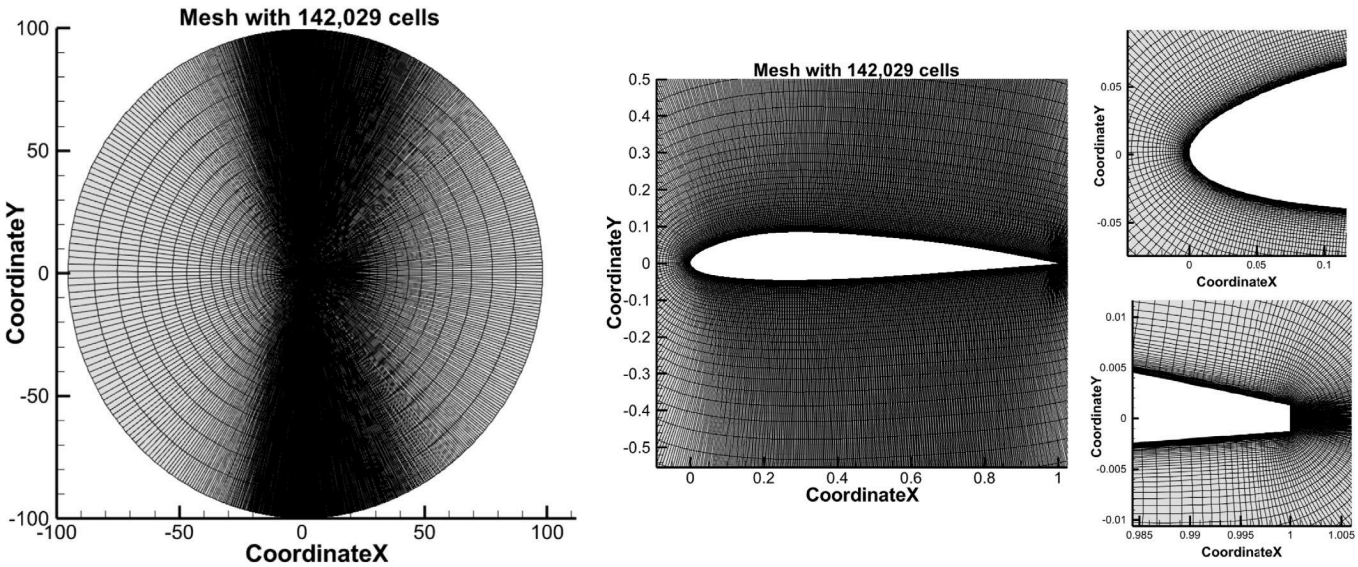
$$X^\star(x(s), y(t), z(u)) = \sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \left[\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \left(\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k P_{ijk} \right) \right]. \tag{5}$$

Note that, FFD can then be used to deform an object in 2D or 3D spaces, regardless of the representation of this object. Instead of manipulating the surface of the object directly, we will move the FFD control points to obtain a new shape. The displacement of a point inside the lattice is described by a third order Bézier tensor product. The control lattice of the parallel-piped FFD design used to morph the airfoil shape is shown in Fig. 1, which is implemented into a standalone Python package named PyGeM.



(a) Free-form deformation (3D). (b) Free-form deformation (2D).

Fig. 1. The parallel-piped lattice [10].



(a) Far field. (b) Airfoil wall.

Fig. 2. Computational mesh with 142,029 cells.

Using the NACA 2412 airfoil as a baseline configuration, a new airfoil shape is obtained by vertically moving the FFD control points as shown in Fig. 1b. The (x_s, y_s) coordinates of the new shape are computed using (5).

3.2. Mesh generation

The computational meshes for the airfoils considered in this work consists of 1,102 points along the surface and 130 points in the off-wall direction. The meshes are generated using a cluster, a cutting edge high performance computing (HPC) resource at Georgia Tech equipped with Dual Intel Xeon Gold 6226 CPUs @2.7 GHz (24 cores/node) [17]. The mesh generation approach is implemented in the open-source module pyHyp [18], a hyperbolic mesh generator used to generate high-quality structured mesh. Fig. 2 shows the computational mesh for the NACA 2412 airfoil.

3.3. ADflow-CFD data generation

To obtain the values of c_l, c_d, and c_m for different airfoil shapes, we will use high-fidelity CFD simulations performed with ADflow. ADflow is an open source CFD solver [19] and has options to solve Euler, laminar Navier-Stokes, and Reynolds-Averaged Navier–Stokes (RANS) equations in steady, unsteady, and time-spectral modes, with multi-block structured and overset meshes. The governing equations are discretized using the finite volume method with first and second order stencils. The in-viscid fluxes are discretized by using 3 different numerical schemes: (i) the scalar artificial dissipation scheme [20], (ii) a matrix dissipation scheme [21], and (iii) a Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) based on the works of [22] and [23]. The viscous flux gradients are calculated by using the Green-Gauss approach while the residual equations can be converged with four distinct algorithms. An Approximate Newton–Krylov (ANK) solver is implemented and can be used as a globalization scheme for the full NK algorithm [24]. We use the Spalart-Allmaras (SA) turbulence model [25] as closure for all CFD simulations that are part of this work. The numerical methods and models in ADflow have been validated in [5,26] against experimental airfoil data.

For all the CFD simulations used in the data generation, we choose a 12th order decay in total residuals magnitude as a relative convergence criterion. The relative convergence is defined as $\eta_{\text{rel}}^{(n)} = \frac{||\mathcal{R}_0^{(n)}||_2}{||\mathcal{R}_0^{(\text{fs})}||_2}$, where the superscript fs refers to the free-stream properties. We observe that this

stopping criterion is reached, in general, after 2,000 nonlinear iterations for the test cases. For the SA turbulence model, used in this study, the convergence is proved in [24].

We follow best practices from the CFD literature [27,28] to assure that our CFD simulations are able to reproduce flight conditions as closely as possible. This is compatible with current aerospace industry standards and aerodynamic development practices [29]. Our meshes are composed of a number of cells and nondimensional distance from the first cell center to the wall such that the relevant physical phenomena are captured, including turbulence and separation effects.

3.4. Shape generation

For the CFD analysis, we discretize the airfoil using the 1,102 points distributed over the surface. This quantity of points is necessary to obtain accurate values of lift, drag, and pitching moment coefficients, c_l, c_d, and c_m, respectively, for the high Reynolds number flow considered in this work.

To generate the new shapes from the baseline NACA 2412, we consider 20 FFD control points able to independently move in the vertical direction, i.e., 1 degree of freedom (DOF) for each control point. The control points are identified as $(x_1, x_2, \cdots, x_{20})$ and are evenly distributed as shown in Fig. 1b. The control points may expand into small variations, resulting in small deformations around the original airfoil. The upper surface is defined by the control points $(x_1, x_2, \cdots, x_{10})$ and the lower surface is defined by $(x_{11}, x_{12}, \cdots, x_{20})$ by assuming values in the interval $[-2.5, 2.5]$, with pace 0.01, as described in Table 1. Fig. 3 presents the upper and lower bounds and the morphed shapes in a linear trajectory obtained using the FFD control points with the geometric constraint.

Table 1
Boundary of the FFD Control Points.

	max value	min value	upper bound shape	lower bound shape
$x_1, x_2, \cdots, x_{10}$	2.5	-2.5	2.5	-2.5
$x_{11}, x_{12}, \cdots, x_{20}$	2.5	-2.5	-2.5	2.5

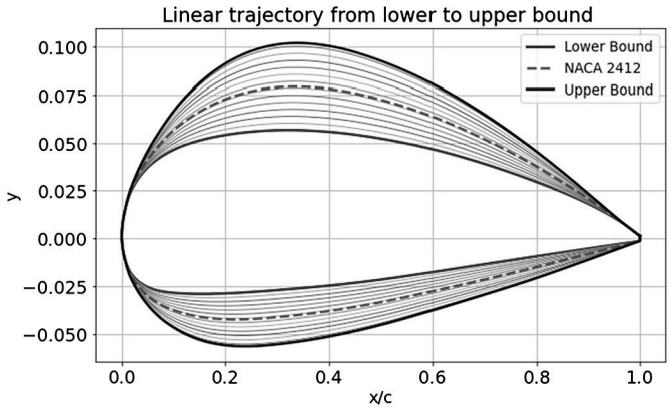


Fig. 3. Prediction of Trajectory 5. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.5. Data set

To train the DNNs used, we consider one flight condition defined as Re = 2e6, Mach = 0.2, and $\alpha = 1.0°$ and 9,600 shapes such that 7,680 shapes are used as training data, and 1,920 as validation data. Finally, we generate 5,100 more shapes to use as testing data and to evaluate the accuracy of the predictions. In the present work, we consider only one flight condition to perform a thorough evaluation of the efficiency of the proposed search algorithm. For future work, we will consider other flight conditions to build the DNNs as discussed in our previous work [10].

Table 2 summarizes the data set used to train the DNN. The numerical experiments to obtain c_l, c_d, and c_m were performed in the cluster [17]. ADFlow required, on average, 2 hours to simulate 50 morphed shapes using 8 GB memory, and 19 cores. This run-time corresponds to a 12th order decay in total residuals, which represents a standard engineering-level convergence for CFD analysis.

Table 2
Data set.

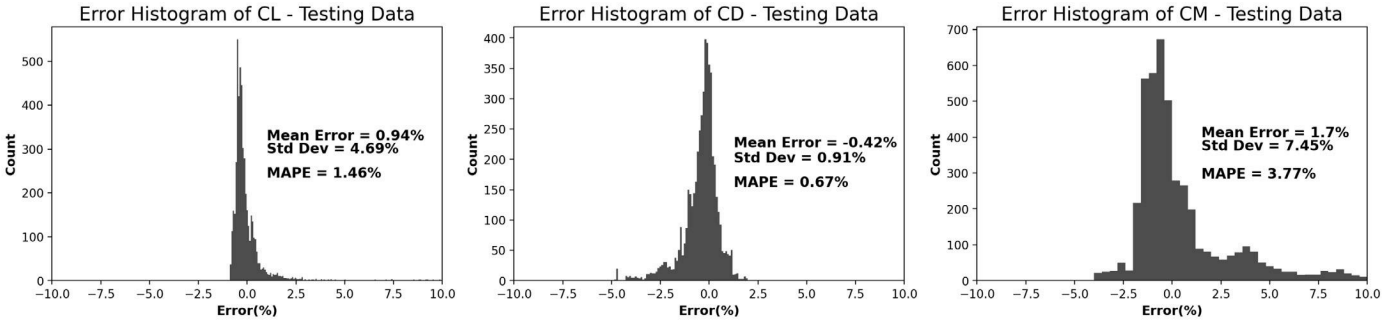
	Training data set	Validation data set	Testing data set
Data points size	7,680	1,920	5,100

3.6. DNN

Since the functions $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ are not known, one can approximate these functions by generating data points by using CFD. To estimate c_l, c_d, and c_m in a more efficient way, we propose to use three fully connected DNNs to approximate $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$. The input data of the DNN is the vector $X \in \mathbb{R}^{23}$ where the first 3 elements define the flight condition, i.e., the triple (Re, Mach, α) and the remaining 20

Table 3
DNN Architecture.

Layer	DNN (c_l)	DNN (c_d)	DNN (c_m)
Input	1x23	1x23	1x23
Hidden	5x25, 5x50, 5x25, 5x10	5x25, 5x50, 5x25, 5x10	5x25, 5x50, 5x25, 5x10
Output	1x1	1x1	1x1
MAPE	0.39%	0.04%	0.47%



(a) c_l (b) c_d (c) c_m

Fig. 4. Error histograms of c_l, c_d, and c_m with 5,100 shapes.

elements define a unique airfoil shape. The DNNs will be trained by using the training data-sets described in Table 2. The training is performed using Exponential Linear Unit (ELU) as activation function with input data being normalized as discussed in our previous work [10], the loss function considered is the Mean Absolute Percentage Error (MAPE). The architecture of the DNNs is described in Table 3 and the error histogram for each parameter using 5,100 shapes from the testing data set is shown in Fig. 4.

We use the DNNs to perform shape optimization to obtain, in seconds or in a few minutes, a safe trajectory to a given shape. A reasonable time of response is necessary to morph efficiently the shape during the flight. Using directly the ADflow solver to find the final shape, and generate a safe trajectory to this shape, will take many minutes or even hours since it takes 20 minutes on average to obtain the aerodynamic parameters of a single shape [10].

The efficiency of the DNN to predict the aerodynamic parameters is shown in our previous work [10] and in different related works. For instance, the authors in Ref. [30] used ADflow as CFD solver to obtain C_L, C_D, C_M, and C_P of wing shapes and trained deep neural networks using 135,108 and 47,967 CFD samples for training and validation, respectively. They performed CFD-based optimization using the efficient adjoint solver in ADflow, which costs 18.4 CPU hours using a 2.6 GHz processor.

The authors in [31] performed gradient-based aerodynamic shape optimization (ASO) of airfoils in subsonic and transonic flow conditions to investigate the benefits of including transition to turbulence effects into the optimization process. They performed aerodynamic shape optimization (ASO) framework, which uses ADflow as the flow solver, and the optimization runs took up to 72 hours running on 36 processors Intel Xeon Gold 6154, 3.0 GHz.

4. MGDA

With the developed surrogate models based on DNNs, we deal with the second part of Problem 1 that is to find the shape $x^\star$ the solve the optimization problem described in (2). To address this problem, we use the MGDA optimizer to reach a new shape with reduced drag and increased lift. To find the new shape, we solve an optimization problem with three objective functions that can be mathematically described as follows:

$$
\begin{aligned}
\underset{u\in\mathbb{R}^{20}}{\text{minimize}} \quad & \mathbf{F}(\mathbf{x}) = \left[\; -\frac{f_{c_l}(\mathbf{x}[L])}{f_{c_d}(\mathbf{x}[L])} \;\; , \;\; -f_{c_l}(\mathbf{x}[L]) \;\; , \;\; f_{c_d}(\mathbf{x}[L]) \; \right] \\
\text{subject to} \quad & \mathbf{x}[k+1] = \mathbf{x}[k] + u[k], \forall k \in \{0, \cdots, L-1\} \\
& f_{c_l}(\mathbf{x}[L]) > f_{c_l}(\mathbf{x}[0]) \\
& f_{c_d}(\mathbf{x}[L]) < f_{c_d}(\mathbf{x}[0]) \\
& ||f_{c_m}(\mathbf{x}[L])|| \leqslant ||f_{c_m}(\mathbf{x}[0])|| \\
& \mathbf{x}[0] = \bar{\mathbf{x}} \\
& x_i^{(L)} \leqslant x_i \leqslant x_i^{(U)} \quad i = 1, 2, \ldots, n \\
& f_{c_l}(x[k]) \geqslant 0.98 f_{c_l}(x[0]), \;\; \forall k \in \{1, 2, \; \ldots, L\text{-}1\} \\
& f_{c_d}(x[k]) \leqslant 1.02 f_{c_d}(x[0]), \;\; \forall k \in \{1, 2, \; \ldots, L\text{-}1\} \\
& | \, f_{c_m}(x[k]) \, | \leqslant 1.05 | \, f_{c_m}(x[0]) \, |, \;\; \forall k \in \{1, 2, \; \ldots, L\text{-}1\}.
\end{aligned}
\tag{6}
$$

The multi-objective optimization problem for the morphing airfoil involves three objective functions as described in (6) and it is solved by using the gradient vectors of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$, and $f_{c_m}(\cdot)$ represented by ∇f_{c_l}, ∇f_{c_d} and ∇f_{c_m} respectively; all gradient vectors are elements of $\mathbb{R}^{20}$.

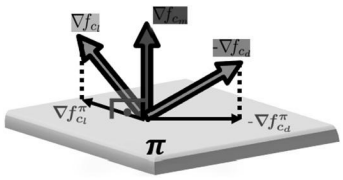
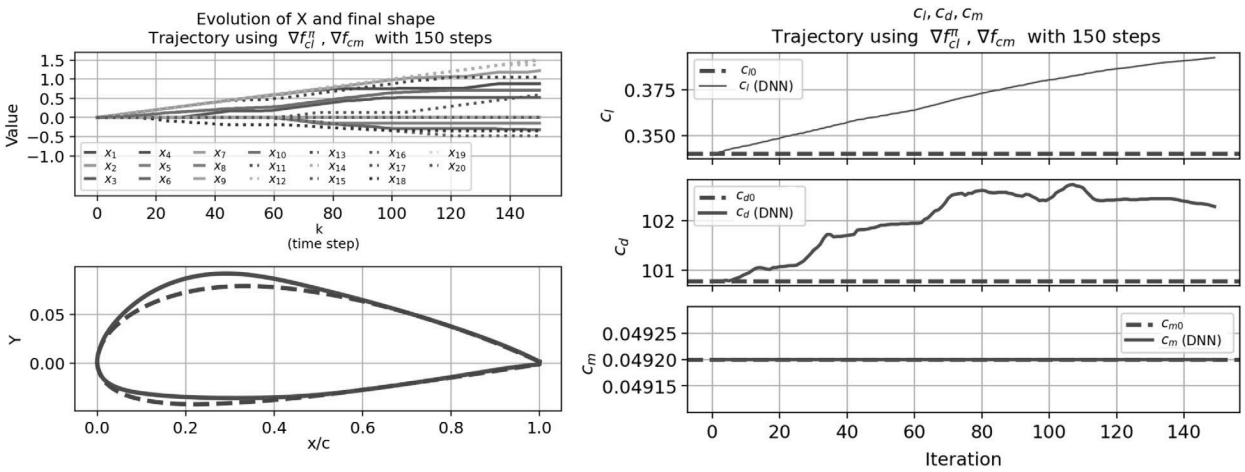


Fig. 5. Hyperplane π and gradient vectors.



(a) Evolution of state X and final airfoil shape.

(b) Evolution of c_l, c_d, and c_m.

Fig. 6. Example of a trajectory built by using gradient vectors.

Let π be a hyperplane defined as $\{v \in \mathbb{R}^n \mid \nabla f_{c_m}^{\mathrm{T}}.v = 0\}$. The orthogonal projections of ∇f_{c_l} and $-\nabla f_{c_d}$ on the hyperplane π can be expressed as

$$\nabla f^{\pi}_{c_l} = \mathrm{Proj}_{\pi}(\nabla f_{c_l}) = \nabla f_{c_l} - \frac{\nabla f_{c_m}^{\mathrm{T}}.\nabla f_{c_l}}{\left\|\nabla f_{c_m}\right\|_2}(\nabla f_{c_m})$$

$$-\nabla f^{\pi}_{c_d} = \mathrm{Proj}_{\pi}(-\nabla f_{c_d}) = -\nabla f_{c_d} + \frac{\nabla f_{c_m}^{\mathrm{T}}.\nabla f_{c_d}}{\left\|\nabla f_{c_m}\right\|_2}(\nabla f_{c_m}).$$

The gradient vectors are orthogonally projected on the hyperplane π and therefore $\nabla f_{c_m}^{\mathrm{T}}.\nabla f^{\pi}_{c_l} = 0$ and $\nabla f_{c_m}^{\mathrm{T}}.(-\nabla f^{\pi}_{c_d}) = 0$. Using the hyperplane π, we search for shapes that improve the value of c_l and c_d with small variations of c_m. Fig. 5 illustrates the hyperplane π and the gradient vectors as vectors of $\mathbb{R}^3$ without loss of generality since only the norms of the vectors and the angle between them do matter.

Fig. 6 shows a trajectory that, regardless the value of f_{cd}, moves toward the maximum value of f_{c_l} while keeping a minimum f_{c_m} variation.

It is worth mentioning that all gradient-based optimization techniques may converge to local minima. However, from an engineering perspective, achieving a local optimum that is better than the initial design is already a useful result [32]. Results in the literature [33–35] indicate that gradient-based algorithms are adequate for aerodynamic shape optimization. Furthermore, the real-time optimization process highlighted in this paper aims at finding, during flight, a shape that is compatible with flight stability characteristics and does not deteriorate the aerodynamic performance when compared with the initial, baseline shape. We suppose that the initial airfoil was optimized during the wing design process in a previous stage, such that we preserve the aerodynamic efficiency by requiring our morphed airfoil shapes to be at least as efficient as the original ones.

4.1. Morphing direction

The morphing direction $\nabla \mathbf{F}(\mathbf{x})$ used is,

$$\nabla \mathbf{F}(\mathbf{x}) = \alpha_1 \nabla f^{\pi}_{c_l}(\mathbf{x}) - \alpha_2 \nabla f^{\pi}_{c_d}(\mathbf{x}) - \alpha_3 \nabla f_{c_m}(\mathbf{x})$$

subject to:

$$\begin{aligned} &\alpha_1 + \alpha_2 = 1 \\ &\alpha_2 \geqslant 0.1 \\ &\alpha_3 \in [-0.01, 0, 0.01] \\ &\left\|\nabla f^{\pi}_{c_l}\right\|_2 = \left\|\nabla f^{\pi}_{c_d}\right\|_2 = \left\|\nabla f_{c_m}\right\|_2 = 1, \end{aligned} \tag{7}$$

where $\alpha_1 \in \mathbb{R}^+$ and $\alpha_2 \in \mathbb{R}^+$ are the weights of the gradient vectors used to balance the importance of $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ in (6) for every iteration when running the search algorithm. For the search algorithm, we do not directly use the gradient of the ratio $\left(\frac{f_{c_l}(\cdot)}{f_{c_d}(\cdot)}\right)$ as we evaluate each term

separately. The correction term $\alpha_3 \nabla f_{c_m}(\mathbf{x})$ is added to ensure that the constraint $||f_{c_m}(\mathbf{x}[L])|| \leqslant ||f_{c_m}(\mathbf{x}[0])||$ is fulfilled. The partial derivatives used to obtain the gradient vectors $\nabla(f_{c_l})$, $\nabla(f_{c_d})$, and $\nabla(f_{c_m})$ can be calculated using central difference approximations or automatic differentiation. In this work, we use the central difference approximations to obtain the gradient vectors.

4.2. Search algorithm

For every iteration of the search algorithm the new shape is given by,

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \beta \cdot \nabla \mathbf{F}(\mathbf{x}[k]), \forall k \in \{0, \cdots, L-1\} \tag{8}$$

where β is defined such that the maximum variation from $x[k]$ to $x[k+1]$ is no greater than 0.1, i.e., $\|\mathbf{x}[k+1] - \mathbf{x}[k]\|_\infty \leqslant 0.1$. The values of $\alpha_1, \alpha_2 \in (0.1, 0.9)$, $\alpha_3 \in [-0.01, 0, 0.01]$, and values of β are chosen such that $x[k+1]$ minimizes the objective function defined in (6). The MGDA can be described by the following pseudo-code.

Algorithm 1 Multi-Gradient Search Algorithm.

Input: DNNs: $f_{c_l}(\cdot), f_{cd}(\cdot), f_{c_m}(\cdot)$, flight condition: (Re, Mach, α) and initial shape $\mathbf{X}_0$.

Output: $x^\star$, shape with improved $\left(\frac{c_l}{c_d}\right)$, c_l, and c_d values.

procedure
  Set K, $A_1 \subseteq \{0.1, 0.15, \cdots, 0.85, 0.9\}$, $B_1 \subseteq \{0.01, 0.02, \cdots, 0.08, 0.09\}$
  Set $A_3 = \{0.01, 0.0, -0.01\}$
  Set target_cl, target_cd, target_ratio_cl_cd, MaxQty
  $k = 0$, $\mathbf{x} \leftarrow \mathbf{x}_0$
  stop_search = 0, stop_local_search = 0

  while $(k \leqslant K)$ **and** (stop_search == 0) **do**
    counter = 0, $x_{\text{list}} = [0]$
    Obtain $\nabla f_{c_l}(\mathbf{x}), \nabla f^{\pi}_{c_l}(\mathbf{x}), \nabla f_{c_d}(\mathbf{x}), \nabla f^{\pi}_{c_d}(\mathbf{x}), \nabla f_{c_m}(\mathbf{x})$
    for each $\beta_1 \in B$ **do**
      for each $\alpha_1 \in A_1$ **do**
        for each $\alpha_3 \in A_3$ **do**
          $\alpha_2 = 1 - \alpha_1$; $\beta = \frac{\beta_1}{\|\nabla \mathbf{F}(\mathbf{x})\|_\infty}$
          $\nabla \mathbf{F}(\mathbf{x}) = \alpha_1 \nabla f^{\pi}_{c_l}(\mathbf{x}) - \alpha_2 \nabla f^{\pi}_{c_d}(\mathbf{x}) + \alpha_3 \nabla f_{c_m}(\mathbf{x})$
          $\mathbf{x}[k+1] = \mathbf{x}[k] + \beta \cdot \nabla \mathbf{F}(\mathbf{x}[k])$
          if $\left(\frac{f_{c_l}(\mathbf{x}[k+1])}{f_{c_d}(\mathbf{x}[k+1])} \geqslant \frac{f_{c_l}(\mathbf{x}[k])}{f_{c_d}(\mathbf{x}[k])}\right)$ **and** $\left(f_{c_l}(\mathbf{x}[k+1]) > 0.98 f_{c_l}(\mathbf{x}[0])\right)$ **and**
            $\left(f_{c_d}(\mathbf{x}[k+1]) < 1.02 f_{c_d}(\mathbf{x}[0])\right)$ **and** $\left\|f_{c_m}(\mathbf{x}[k+1])\right\| \leqslant 1.05 \left\|f_{c_m}(\mathbf{x}[0])\right\|$ **then**
              $x_{\text{list}}[\text{counter}] = \mathbf{x}[k+1]$
              counter = counter + 1
          end if
          if counter == MaxQty **then**
              break
          end if
          if $\left(\frac{f_{c_l}(\mathbf{x}[k+1])}{f_{c_d}(\mathbf{x}[k+1])} \geqslant$ target_ratio_cl_cd$\right)$ **and** $\left(f_{c_l}(\mathbf{x}[k+1]) \geqslant \text{target\_cl}\right)$ **and**
            $\left(f_{c_d}(\mathbf{x}[k+1]) \leqslant \text{target\_cd}\right)$ **then**
              stop_search = 1
              break
          end if
        end for
      end for
    end for
    Evaluate and select cost function: Max $\left(\frac{c_l}{c_d}\right)$, Max($c_l$) or Min($c_d$).
    Obtain the best shape $x^\star_{\text{temp}}$.
    $\mathbf{x}[k+1] = x^\star_{\text{temp}}$; $k \leftarrow k+1$
  end while
  Evaluate the trajectory and obtain $x^\star$.
end procedure

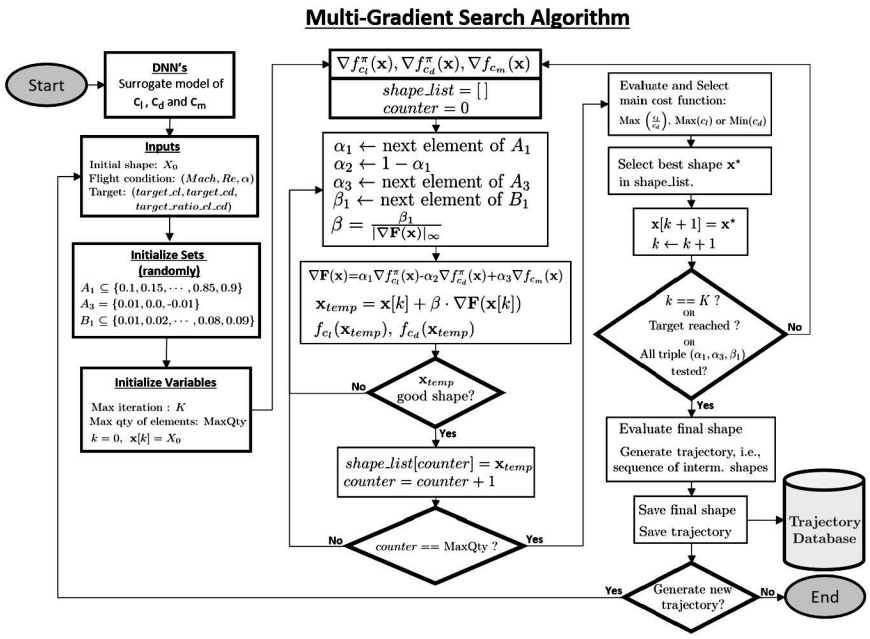
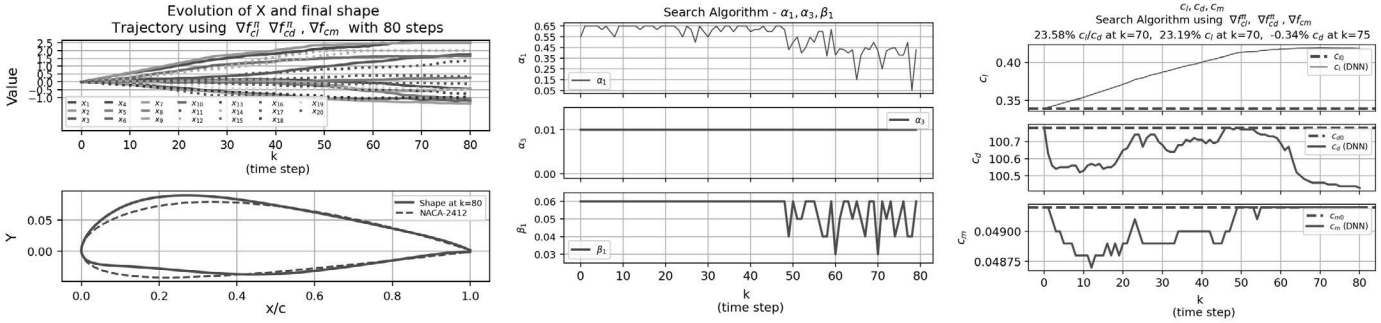


Fig. 7. Flow chart: multi-gradient search algorithm.

Table 4
Analysis of the three Trajectories generated using the DNN.

	Main function	Max $\left(\frac{c_l}{c_d}\right)$	Max c_l	Min c_d	Description
Trajectory 1	$\left(\frac{c_l}{c_d}\right)$	23.58% at k = 70	23.19% at k = 70	−0.34% at k = 75	Trajectory with focus to maximize $\left(\frac{c_l}{c_d}\right)$
Trajectory 2	c_l	23.27% at k = 77	22.88% at k = 77	−0.33% at k = 79	Trajectory with focus to maximize c_l
Trajectory 3	c_d, $\left(\frac{c_l}{c_d}\right)$	11.29% at k = 79	10.83% at k = 79	−0.67% at k = 22	Trajectory with focus to minimize c_d and then maximize $\left(\frac{c_l}{c_d}\right)$



(a) Evolution of state X and final shape. (b) α, β (c) c_l, c_d, c_m

Fig. 8. Search Algorithm - Trajectory 1 - Maximize $\frac{c_l}{c_d}$.

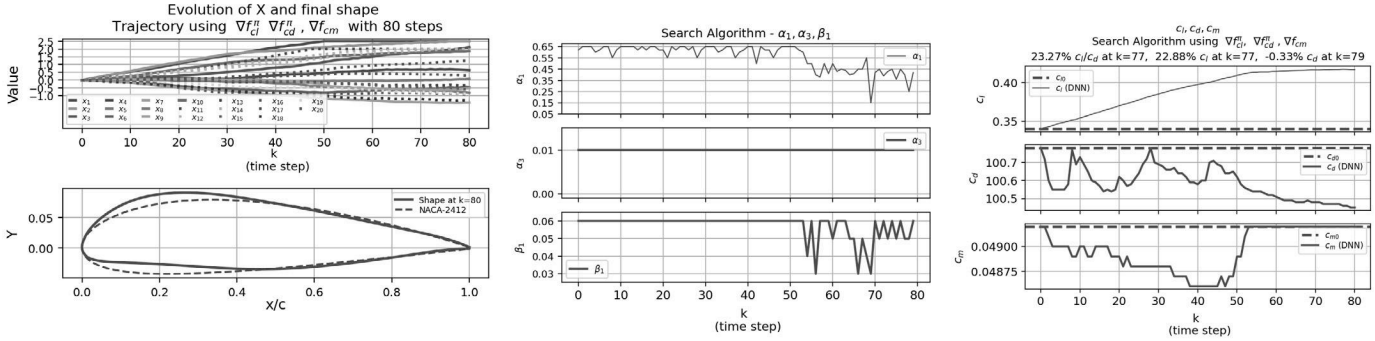
The subsets A_1 and B_1 can be chosen randomly, as shown in line 2, such that for every run of the search algorithm, the algorithm provides a different shape $x^\star$ that improves the aerodynamic parameters. Moreover, to avoid local minima, we perform an assessment of the cost functions in every iteration as shown in line 32 of the pseudo-code. It is important to mention that we aim to find shapes, with improved parameters, that are reachable using a trajectory that does not affect the safety conditions as described in (3). Fig. 7 shows the flow chart of the multi-gradient search algorithm.

To evaluate the efficiency of the search algorithm using the surrogate model, we generate three trajectories, each trajectory with a focus on an objective function, line 32 of the pseudo-code. For the three trajectories, we reach a final shape with significant improvement on $\left(\frac{c_l}{c_d}\right)$, c_l, and c_d as described in Table 4.

Figs. 8-10 present the trajectories described in Table 4 over three plots: plot **(a)** represents the evolution of the states x_i and the final shape after 80 steps; plot **(b)** depicts the evolution of α_1, α_3, and β_1; plot **(c)** shows the evolution of c_l, c_d, and c_m.

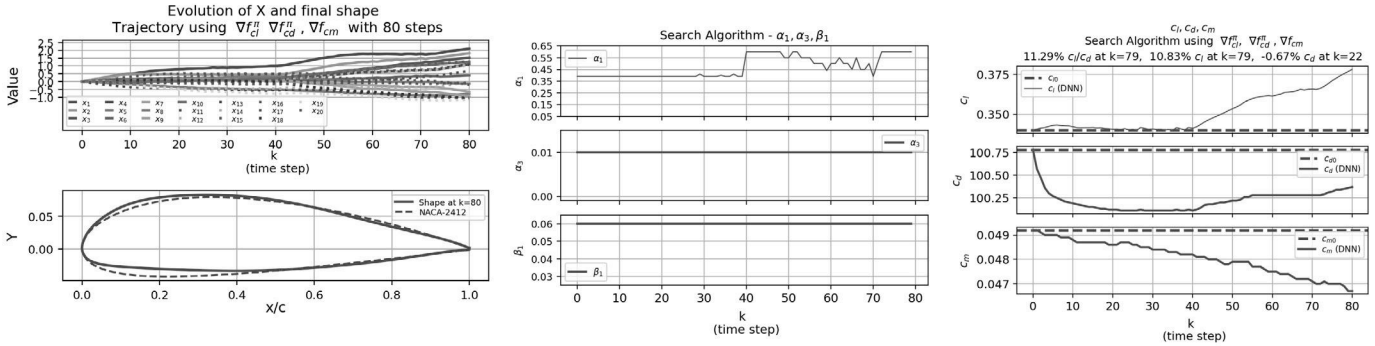
5. Data-enabled predictive shape algorithm

Although MGDA described in previous section and used to solve Problem 1 may provide the desired final shape, it only relies on the trained model, which does not consider real-time data that may exist during the morphing period from the initial shape to the final one. To this end, the data-enabled predictive shape algorithm proposed in [10] and based on behavioral system theory [36,37] must be used to solve Problem 2.



(a) Evolution of state X and the final shape. (b) α, β (c) c_l, c_d, c_m

Fig. 9. Search Algorithm - Trajectory 2 - Maximize c_l.



(a) Evolution of state X and the final shape. (b) α, β (c) c_l, c_d, c_m

Fig. 10. Search Algorithm - Trajectory 3 - Minimize c_d and then maximize $\frac{c_l}{c_d}$.

The algorithm has two components; the first is the time window $L = T_{\text{ini}} + N$, where T_{ini} refers to the length of data collected online that will be used to predict c_l, c_d, and c_m over the prediction horizon N; and the second is the matrix of trajectories, similar to the Hankel and Page matrices. Note that this matrix has as columns real trajectories with length L, and in this context, a trajectory is an input-output sequence provided as $\{u_k, y_k\}_{k=0}^{L-1}$. The matrix has full row rank, generated by a sufficiently rich input sequence to be used to recover all the trajectories.

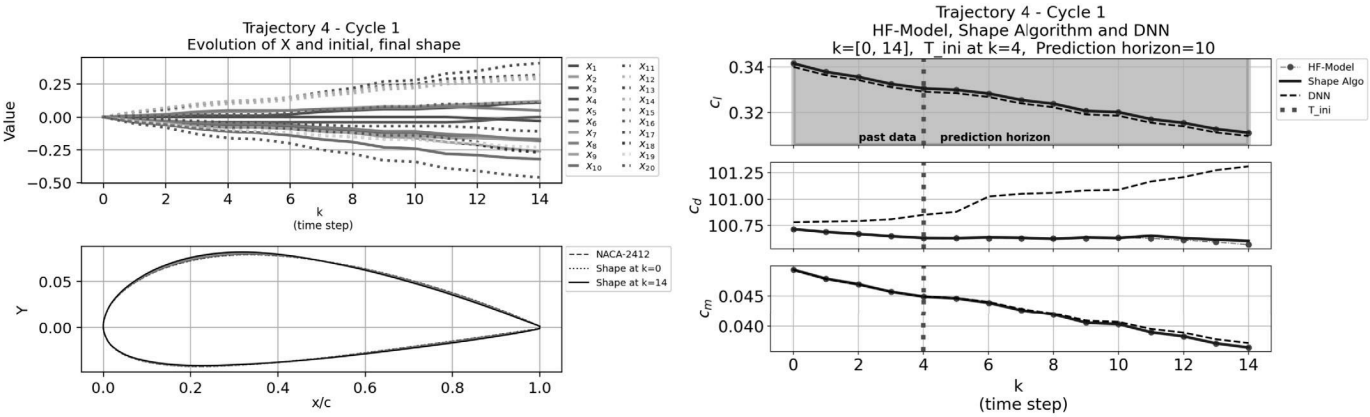
Using the high-fidelity model (HF-Model) based on CFD simulations, we generate T trajectories of length $L = T_{\text{ini}} + N$ to obtain the input data matrix $U_L = [U_{T_{\text{ini}}}^{\mathrm{T}}, U_N^{\mathrm{T}}]^{\mathrm{T}}$ and output data matrix $Y_L = [Y_{T_{\text{ini}}}^{\mathrm{T}}, Y_N^{\mathrm{T}}]^{\mathrm{T}}$. For a trajectory of length L to be predicted, we use the following notation, $y_{\text{ini}} = [y^{\mathrm{T}}[1], \dots, y^{\mathrm{T}}[T_{\text{ini}}]]^{\mathrm{T}}$, $y_N = [y^{\mathrm{T}}[T_{\text{ini}}+1], \dots, y^{\mathrm{T}}[T_{\text{ini}}+N]]^{\mathrm{T}}$, $u_{\text{ini}} = [u^{\mathrm{T}}[1], \dots, u^{\mathrm{T}}[T_{\text{ini}}]]^{\mathrm{T}}$, $u_N = [u^{\mathrm{T}}[T_{\text{ini}}+1], \dots, u^{\mathrm{T}}[T_{\text{ini}}+N]]^{\mathrm{T}}$.

We are interested in finding y for a given reference input, u_r. Thus, for a given final shape X_F, we generate a reference trajectory (u_r, y_r) from baseline shape X_0 toward X_F by using the surrogate model. To clarify, denote as state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, output $y \in \mathbb{R}^p$, and $U_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}m \times T}$, $U_N \in \mathbb{R}^{Nm \times T}$, $Y_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}p \times T}$ and $Y_N \in \mathbb{R}^{Np \times T}$.

The prediction of c_l, c_d, and c_m is given by the solution of the following optimization problem:

$$
\begin{aligned}
\underset{g,u,y,\sigma_y}{\text{minimize}} \quad & \sum_{k=1}^{N} \left(C(u[T_{\text{ini}}+k]) + \mathcal{W}(g, \sigma_y) \right) \\
\text{s.t.} \quad & \begin{bmatrix} U_{T_{\text{ini}}} \\ U_N \\ Y_{T_{\text{ini}}} \\ Y_N \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ u_N \\ y_{\text{ini}} \\ y_N \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \end{bmatrix} \\
& B_L[T_{\text{ini}}+k] \leqslant y[T_{\text{ini}}+k] \leqslant B_U[T_{\text{ini}}+k], \ \forall k \in \{1, \dots, N\} \\
& u[T_{\text{ini}}+k] \in \mathcal{U}, \ \forall k \in \{1, \dots, N\} \\
& y[T_{\text{ini}}+k] \in \mathcal{Y}, \ \forall k \in \{1, \dots, N\}
\end{aligned}
\tag{9}
$$

where, $C(u[.]) = \|u[.] - u_r[.]\|_R^2$, $R = \mathrm{w}_u I_{20}$, $\mathrm{w}_u \gg 1$, $\mathcal{W}(g, \sigma_y) = \mathrm{w}_g \|g\|_2^2 + \mathrm{w}_{\sigma_y} \|\sigma_y\|_2^2$, $\mathrm{w}_g, \mathrm{w}_{\sigma_y} > 0$, the weights $\mathrm{w}_u, \mathrm{w}_g$ and w_{σ_y} are defined using the data set described in section 3.5, B_U and B_L are upper and lower constraints used to guarantee small variations on output according to Assumption 2 and $I_{\cdot}$ is an identity matrix of appropriate dimensions. These constraints are based on online gathered data y_{ini}. Let $\Delta_{\max} = [\Delta_{c_l}, \Delta_{c_d}, \Delta_{c_m}]^{\mathrm{T}}$ be the maximum absolute variation on measured (c_l, c_d, c_m) for two consecutive intermediate shapes, we have,



(a) Evolution of state X and the final airfoil shape.

(b) Evolution of c_l, c_d, and c_m.

Fig. 11. Trajectory 4 - Cycle 1.

Table 5
Prediction error of c_l, c_d, and c_m for Trajectory 4.

		Cycle 1: y[5] to y[14]		Cycle 2: y[10] to y[19]		Cycle 3: y[20] to y[24]	
		Mean Abs Error	Max Abs Error	Mean Abs Error	Max Abs Error	Mean Abs Error	Max Abs Error
c_l	DNN	0.46%	0.47%	0.40%	0.47%	0.25%	0.42%
	Shape Algorithm	0.02%	0.03%	0.03%	0.04%	0.09%	0.12%
c_d	DNN	0.49%	0.74%	0.69%	0.85%	0.86%	1.0%
	Shape Algorithm	0.01%	0.04%	0.03%	0.06%	0.01%	0.05%
c_m	DNN	1.02%	2.01%	2.29%	4.58%	4.39%	6.62%
	Shape Algorithm	0.03%	0.05%	0.01%	0.02%	0.06%	0.11%

$$B_U[k] = y[T_{\text{ini}}] + \begin{bmatrix} k\mathrm{w}_\epsilon^{c_l} & 0 & 0 \\ 0 & k\mathrm{w}_\epsilon^{c_d} & 0 \\ 0 & 0 & k\mathrm{w}_\epsilon^{c_m} \end{bmatrix} \begin{bmatrix} \Delta_{c_l} \\ \Delta_{c_d} \\ \Delta_{c_m} \end{bmatrix}, \quad B_L[k] = y[T_{\text{ini}}] - \begin{bmatrix} k\mathrm{w}_\epsilon^{c_l} & 0 & 0 \\ 0 & k\mathrm{w}_\epsilon^{c_d} & 0 \\ 0 & 0 & k\mathrm{w}_\epsilon^{c_m} \end{bmatrix} \begin{bmatrix} \Delta_{c_l} \\ \Delta_{c_d} \\ \Delta_{c_m} \end{bmatrix}$$

where $y[T_{\text{ini}}]$ is the most recent measurement of (c_l, c_d, c_m) and $\mathrm{w}_\epsilon^{c_l}, \mathrm{w}_\epsilon^{c_d}, \mathrm{w}_\epsilon^{c_m} \in (0, 1]$ are correction factors whose values are equal to one as default but can be used to improve accuracy.

Remark 1. The data-enabled predictive shape algorithm uses a matrix of trajectories, randomly generated, for prediction which might be computed for every new flight condition. The size of this matrix depends on the horizon of prediction. □

Remark 2. We assume that the state $x[\cdot]$ is known for all values of $k \in \{0, 1, \ldots, L\}$ (Assumption 1) during the entire morphing process. □

Remark 3. The value of u[k] is bounded such as $\|\mathbf{x}[k+1] - \mathbf{x}[k]\|_\infty \leqslant 0.1$ so that only small shape variations arise and, therefore, small variation of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$ are produced (Assumption 2). This boundary was defined empirically by evaluating the data points used to build the DNN. □

Remark 4. Each element x_i, that defines a shape $\mathbf{x}$, can assume values in the set [-2.5, -2.49, ..., -0.01, 0.0, 0.01, ..., 2.49, 2.50] with 501 elements that are multiple of 0.01. There are $(501)^{20}$ states to be considered in our search algorithm. □

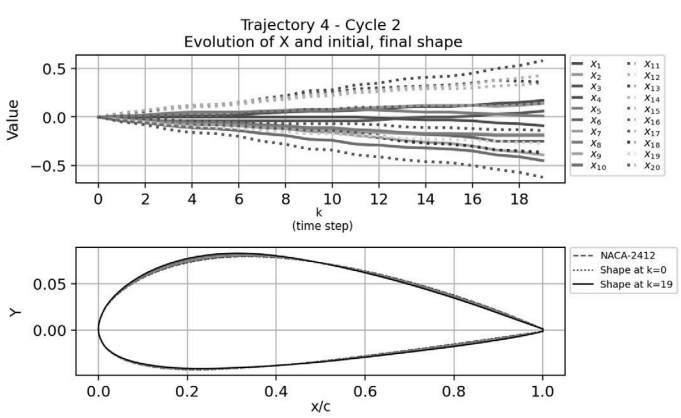
Remark 5. The constraints $(B_L[\cdot], B_U[\cdot])$ are based on online measured data and defined such as the variation of $f_{c_l}(\cdot)$, $f_{c_d}(\cdot)$ is no greater than 2%. □

To show the efficiency of the data-enabled predictive shape algorithm, we run the algorithm in a trajectory with 25 steps named as "Trajectory 4." Here we consider $L = 15$, $T_{\text{ini}} = 5$, $N = 10$, i.e., we use 5 time steps to predict 10 steps ahead. The matrix of trajectories $[U_L^{\mathrm{T}}, Y_L^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{345\times425}$, and the flight condition is defined as (Re = 2e6, Mach = 0.2, $\alpha = 1.0°$). The matrix of trajectories is compound by 425 trajectories, with each trajectory defined by a sequence of 15 shapes, in total we need 6,375 shapes to build the matrix $[U_L^{\mathrm{T}}, Y_L^{\mathrm{T}}]^{\mathrm{T}}$. The 6,375 shapes used to build the matrix is a subset of the training data set described in section 3.6.

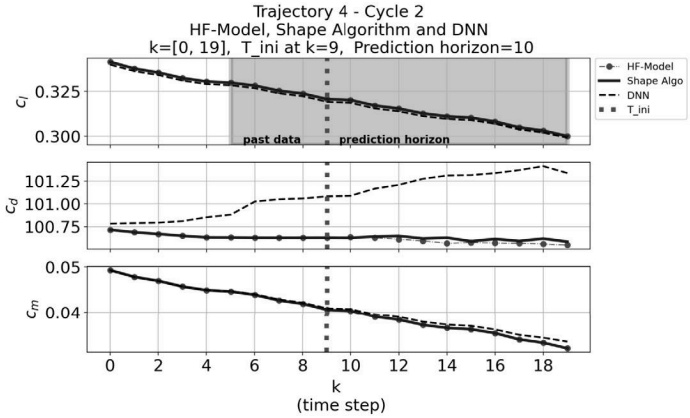
Figs. 11-13 show the use of the shape algorithm on "Trajectory 4" over 3 cycles: in cycle 1, we move the first 5 steps of the given trajectory and predict 10 steps forward; in cycle 2 we move other 05 steps and predict 10 steps forward; and cycle 3, the final cycle to cover the whole trajectory with 25 steps.

The figures above show that the prediction provided by the shape algorithm is more accurate than the one provided by the surrogate model based on DNN. At cycle 3, the maximum error on c_d prediction is about 1.0 drag-count. Table 5 summarizes the accuracy of the proposed shape algorithm.

It is important to point out that we morph the wing in flight only when the airplane is in a stable and trimmed cruise condition. This procedure is adopted due to safety reasons since morphing the wing during the flight is a disturbance to the system, affecting the stability of the airplane. For simplification, the flight condition defined by the triple (Re, Mach, α) remains unchanged during the morphing process as described by Assumption 3.

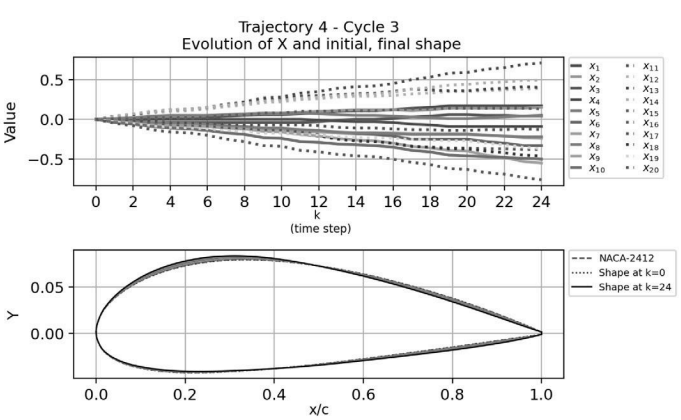


(a) Evolution of state X and the final airfoil shape.

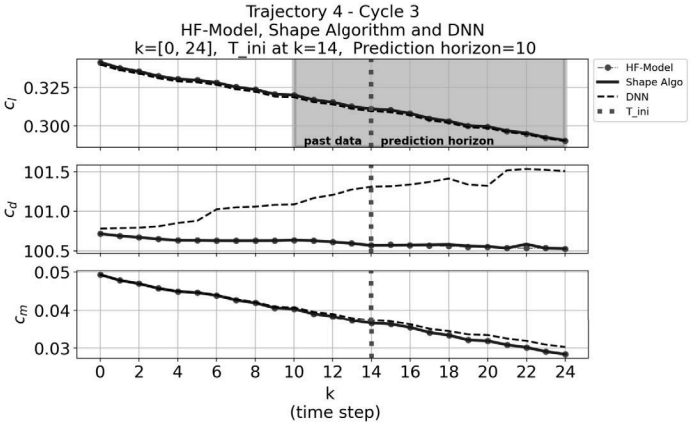


(b) Evolution of c_l, c_d, and c_m.

Fig. 12. Trajectory 4 - Cycle 2.



(a) Evolution of state X and the final airfoil shape.



(b) Evolution of c_l, c_d, and c_m.

Fig. 13. Trajectory 4 - Cycle 3.

6. Data-driven learning algorithm

The previous sections provided the tools to develop the online learning algorithm composed by two steps. The first step is performed offline by running the MGDA to obtain a candidate for the final shape and define a smooth linear trajectory to reach this candidate shape. The second step occurs online by running the data-enabled predictive shape algorithm to morph the shape in a safe and efficient way. During simulated flight, which is the second step of our online algorithm, we do not perform online CFD simulations. Instead, we suppose that a data acquisition system, installed in the aircraft, will provide us with real-time c_l, c_d, and c_m measures. In this paper, we emulate the real-time acquisition system mentioned above by taking the aerodynamic data generated by using CFD simulations corresponding to the first m steps in the morphing trajectory. The online learning algorithm can be described by the procedure shown in Fig. 14.

We will now show the efficiency of the proposed online solution in finding a shape with improved values of c_l, c_d, and c_m. Here, we consider $m = 5$ which means that for every 5 steps of morphing toward the final shape, we run the data-enabled predictive shape algorithm to predict the c_l, c_d, and c_m values for the remaining 10 steps. For this numerical example, we choose "Trajectory 1" that provides an increase of 23.58% in $\left(\frac{c_l}{c_d}\right)$.

We morph the airfoil over the first 5 steps following "Trajectory 1." Then, using the data-enabled predictive shape algorithm, we predict the values of c_l, c_d, and c_m for the following 10 steps of "Trajectory 1." This first movement called Cycle 1, as discussed in Section 5, is described in Fig. 15.

Evaluating Cycle 1 of "Trajectory 1," it is possible to conclude that this trajectory is not appropriate for the given flight condition. After 5 steps of morphing, there is no improvement on c_d. Moreover, the prediction provided by the data-enabled predictive shape algorithm indicates that the value of c_d will not be improved over "Trajectory 1." At this point, to prevent the system from reaching unsafe condition as described in (3), we use the collected data to evaluate other potential trajectories given by MGDA. Using the data-enabled predictive shape algorithm, we evaluate other trajectories to search for a shape with improved valued of c_l, c_d, and c_m as shown in Fig. 16.

The assessment of the 3 trajectories shown in Fig. 16a indicates that "Trajectory 5" provides the shape with the best values of c_l, c_d, and c_m. We keep moving every 5 steps ($m = 5$) over "Trajectory 5," at $k = 44$ we have a reduction of 0.7% in c_l and after 82 steps we reach a shape that provides a gain of 6.17% on $\left(\frac{c_l}{c_d}\right)$ with actual reduction in c_d as shown in Fig. 17.

Fig. 18 presents the prediction of the entire "Trajectory 5" provided by the surrogate model and the predictive shape algorithm. The accuracy of the proposed shape algorithm shown in Table 6 is higher than the accuracy of the DNNs.

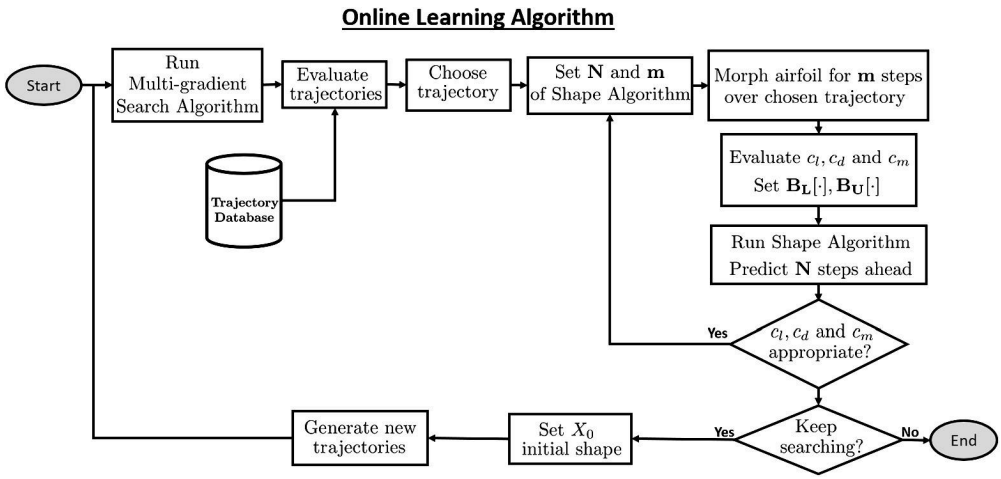
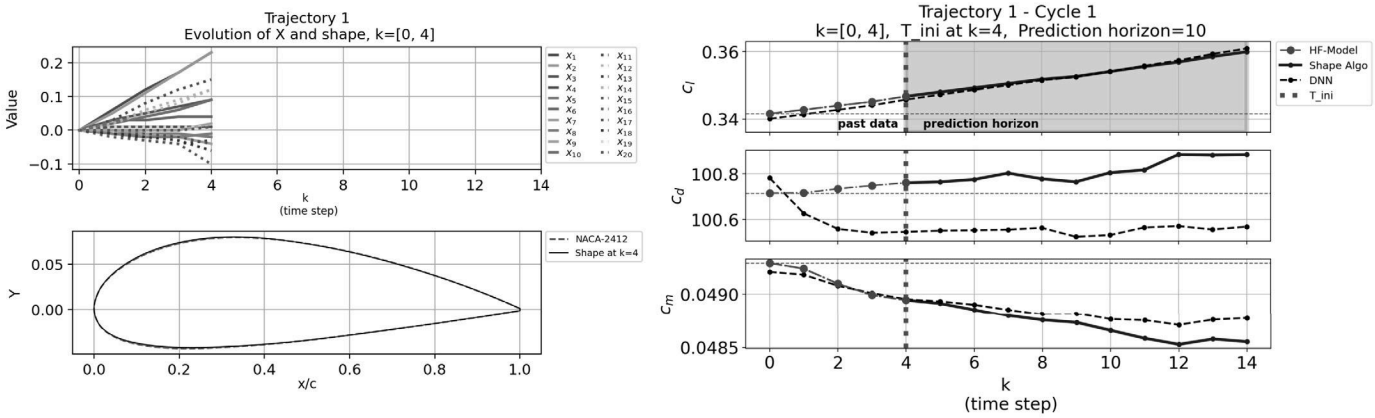


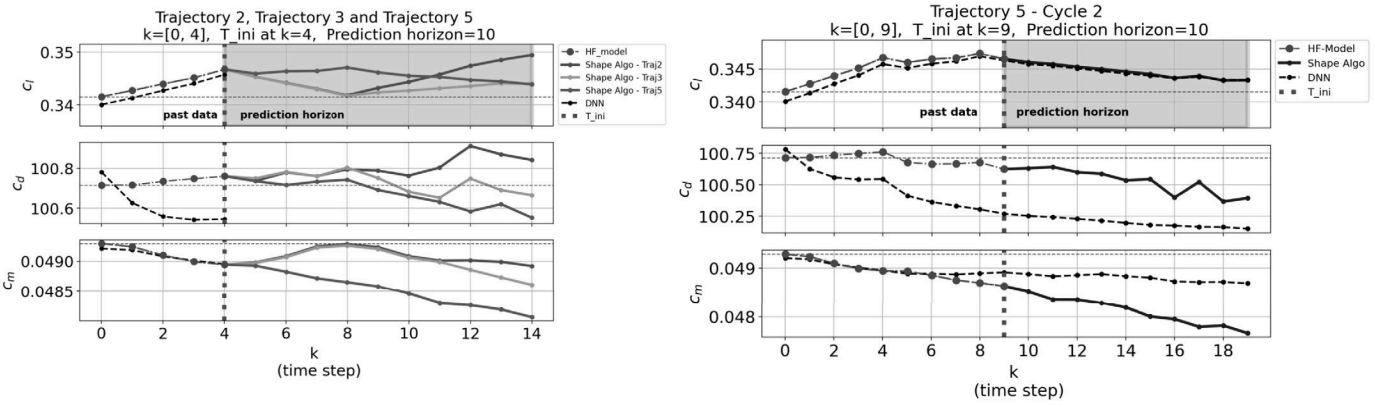
Fig. 14. Online learning algorithm.



(a) Evolution of state X and the airfoil shape.

(b) Evolution of c_l, c_d, and c_m.

Fig. 15. Cycle 1.



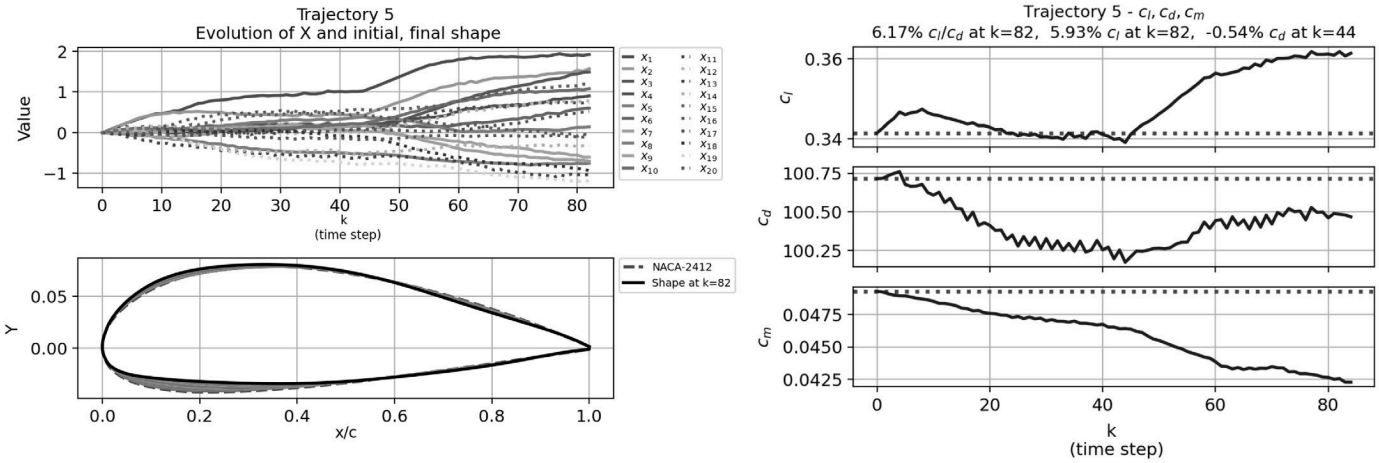
(a) Evaluation of Trajectories 2, 3, and 5.

(b) Trajectory 5: evolution of c_l, c_d, and c_m.

Fig. 16. Evaluation of other trajectories.

This numerical example shows the efficiency of the online learning algorithm in preventing the system from reaching undesired/unsafe conditions by improving the prediction of c_l, c_d, and c_m for a given trajectory. Moreover, we can use the data gathered online to evaluate, in an offline way, alternative trajectories that provide improvement on the aerodynamic coefficients.

For a further understanding of the efficiency of the proposed solution, Fig. 19 presents the values of c_l, c_d, and c_m of HF-Model and DNN for the entire "Trajectory 1." Although the final shape of "Trajectory 1" provides an increase of 21.56% in c_l, this shape violates the safe constraint defined in (3) with an increase of 2.36% in c_d and therefore it must be avoided.



(a) Evaluation of Trajectory 5.

(b) Trajectory 5: evolution of c_l, c_d, and c_m.

Fig. 17. The reached final shape of Trajectory 5.

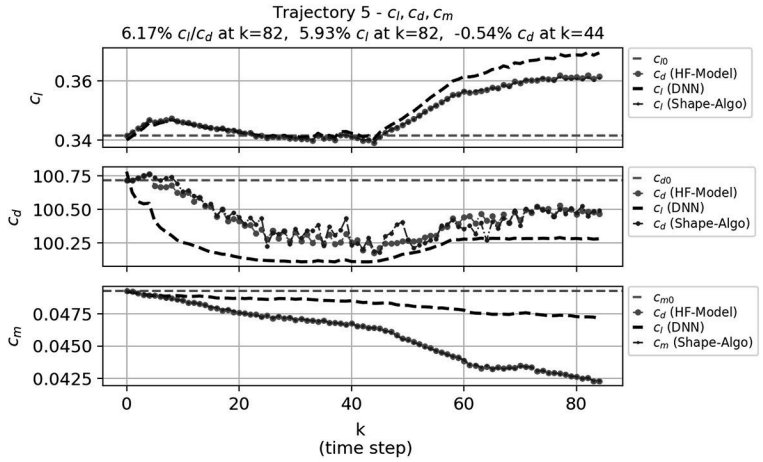


Fig. 18. Prediction of Trajectory 5.

Table 6
Prediction error of c_l, c_d, and c_m for Trajectory 5.

		Max Abs Error	Mean Abs Error
c_l	DNN	2.21%	2.07%
	Shape Algorithm	0.18%	0.14%
c_d	DNN	0.24%	0.20%
	Shape Algorithm	0.03%	0.02%
c_m	DNN	11.71%	10.87%
	Shape Algorithm	0.16%	0.05%

6.1. Robustness analysis

The matrix of trajectories in equation (9), used to predict any trajectory of the controllable and observable system defined in (1), is generated by using the HF-model; this matrix might be generated for every new flight condition. As discussed in our previous work [10], real values of c_l, c_d, and c_m are not always available and therefore the matrix of trajectories is built by using the developed surrogate model.

For the robustness analysis, we run the data-driven shape algorithm over twenty trajectories using the matrix of trajectories provided by the surrogate model, modeled with inherent errors on the parameters' prediction. Fig. 20 presents the histograms of these errors for the given flight condition.

Table 7 presents the robustness analysis of the proposed algorithm for twenty different trajectories. For every trajectory, we consider the use of five and ten real-time data points for prediction. Fig. A.1 in Appendix A presents the evolution of c_l, c_d, and c_m for each trajectory.

The present robustness analysis shows that, even with the error inherent to a poorly trained DNN, we can reduce the error of the prediction when we use the real-time collected data of the proposed solution.

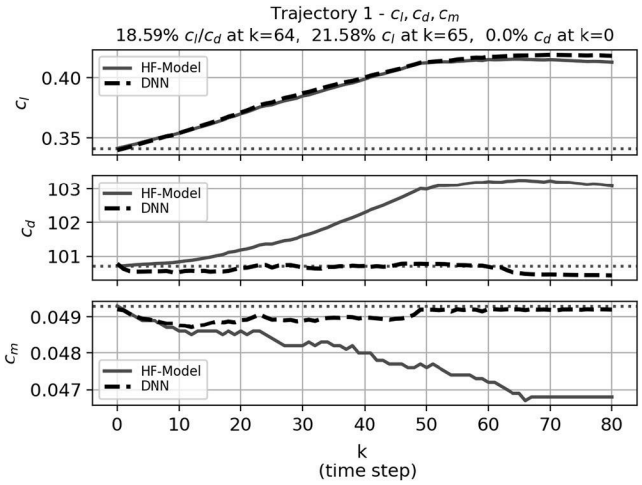
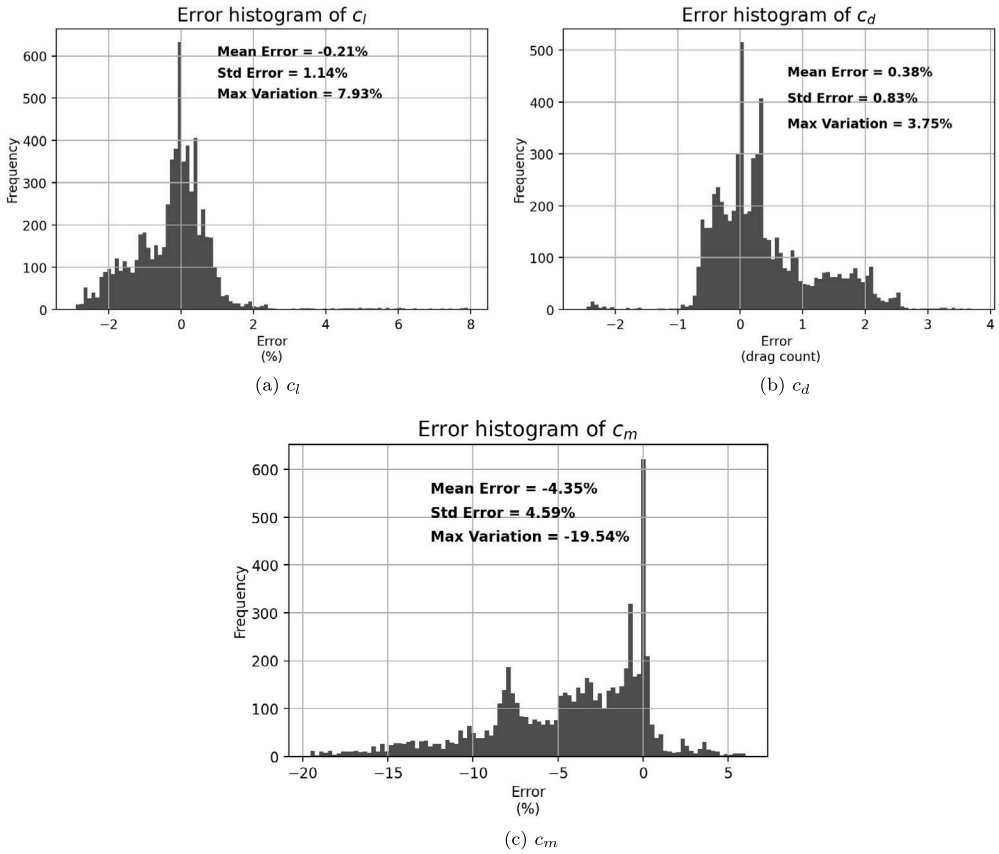


Fig. 19. Final shape of Trajectory 1.



(a) c_l (b) c_d

(c) c_m

Fig. 20. Error histograms of c_l, c_d, and c_m - Matrix of trajectories.

6.2. Sensitivity analysis

The proposed online learning is compounded by two main parameters: m, the number of steps to morph in every iteration; and N, the length of the prediction horizon. Since we use a fixed time window of 15, the value of N is defined by the number of real-time data points used for prediction.

It is important to evaluate how the performance of the shape algorithm varies as we increase the number of collected real-time data points. For this sensitivity analysis, we analyzed 33 different trajectories. For all scenarios, the shape algorithm provides a significant reduction on error prediction when compared to the surrogate model. Table 8 presents the sensitivity analysis of the "Trajectory 19" to the number of real-time data points.

As shown in Table 8, the accuracy of the shape algorithm is directly affected by the quantity of real-time data-points used for prediction. Based on these results, we can conclude that for "Trajectory 19" just two real-time data points are necessary to achieve a significant reduction of error to predict c_l in comparison to the surrogate model. In the case of c_d and c_m, we need to use at least 5 real-time data points to achieve a good reduction of prediction error. Fig. B.1 in Appendix B presents the evolution of c_l, c_d, and c_m for "Trajectory 19" for different values of real-time data points.

Table 7
Robustness analysis of the shape algorithm.

		Shape Algorithm w/ noise			DNN		
	Data points	Max Abs Error in $\mathbf{c_l}$	Max Abs Error in $\mathbf{c_d}$	Max Abs Error in $\mathbf{c_m}$	Max Abs Error in $\mathbf{c_l}$	Max Abs Error in $\mathbf{c_d}$	Max Abs Error in $\mathbf{c_m}$
Traj 6	**5**	1.42%	0.10 (drag count)	0.63%	12.94%	0.10 (drag count)	15.83%
	10	0.37%	0.10 (drag count)	0.26%			
Traj 7	**5**	0.92%	0.24 (drag count)	4.07%	14.83%	0.22 (drag count)	31.22%
	10	0.22%	0.11 (drag count)	2.30%			
Traj 8	**5**	0.32%	0.09 (drag count)	1.27%	0.72%	0.08 (drag count)	1.28%
	10	0.26%	0.03 (drag count)	1.27%			
Traj 9	**5**	0.06%	0.78 (drag count)	0.73%	0.83%	1.06 (drag count)	1.80%
	10	0.04%	0.31 (drag count)	0.48%			
Traj 10	**5**	0.40%	1.17 (drag count)	0.42%	0.32%	2.13 (drag count)	2.32%
	10	0.15%	0.54 (drag count)	0.16%			
Traj 11	**5**	0.09%	0.31 (drag count)	1.99%	0.47%	0.74 (drag count)	2.0%
	10	0.05%	0.17 (drag count)	1.99%			
Traj 12	**5**	0.49%	0.32 (drag count)	8.03%	0.58%	1.07 (drag count)	8.07%
	10	0.41%	0.23 (drag count)	0.65%			
Traj 13	**5**	0.47%	0.15 (drag count)	3.91%	0.94%	1.27 (drag count)	60.35%
	10	0.19%	0.08 (drag count)	2.97%			
Traj 14	**5**	0.70%	0.29 (drag count)	10.53%	2.83%	1.56 (drag count)	102.57%
	10	0.22%	0.39 (drag count)	6.75%			
Traj 15	**5**	0.58%	0.05 (drag count)	3.40%	5.49%	2.09 (drag count)	107.68%
	10	0.30%	0.05 (drag count)	3.74%			
Traj 16	**5**	0.13%	0.09 (drag count)	1.82%	7.04%	0.97 (drag count)	113.81%
	10	0.23%	0.14 (drag count)	1.12%			
Traj 17	**5**	0.23%	0.35 (drag count)	0.22%	0.65%	0.11 (drag count)	0.24%
	10	0.18%	0.11 (drag count)	0.21%			
Traj 18	**5**	0.35%	0.21 (drag count)	1.65%	1.19%	0.35 (drag count)	1.65%
	10	0.18%	0.18 (drag count)	0.88%			
Traj 19	**5**	0.11%	0.44 (drag count)	1.66%	2.81%	0.78 (drag count)	3.54%
	10	0.19%	0.24 (drag count)	0.47%			
Traj 20	**5**	0.21%	0.74 (drag count)	2.22%	4.56%	0.97 (drag count)	5.36%
	10	0.14%	0.66 (drag count)	0.22%			
Traj 21	**5**	0.24%	0.49 (drag count)	0.18%	0.28%	0.65 (drag count)	0.15%
	10	0.18%	0.35 (drag count)	0.18%			
Traj 22	**5**	0.12%	1.36 (drag count)	0.43%	0.15%	0.93 (drag count)	0.39%
	10	0.15%	0.66 (drag count)	0.43%			
Traj 23	**5**	0.49%	0.99 (drag count)	0.17%	0.49%	3.72 (drag count)	0.18%
	10	0.51%	0.22 (drag count)	0.18%			
Traj 24	**5**	0.33%	0.25 (drag count)	0.39%	0.34%	0.25 (drag count)	0.32%
	10	0.32%	0.28 (drag count)	0.37%			
Traj 25	**5**	0.14%	1.12 (drag count)	0.24%	0.19%	1.19 (drag count)	0.25%
	10	0.12%	0.53 (drag count)	0.18%			

Table 8
Sensitivity analysis for Trajectory 19.

		Shape Algorithm		
		Max Abs Error in $\mathbf{c_l}$	**Max Abs Error in $\mathbf{c_d}$**	**Max Abs Error in $\mathbf{c_m}$**
qty real-time data points	**2**	0.23%	0.50 (drag count)	3.22%
	3	0.21%	0.43 (drag count)	3.07%
	4	0.15%	0.46 (drag count)	2.23%
	5	0.11%	0.44 (drag count)	1.66%
	6	0.16%	0.41 (drag count)	1.28%
	7	0.21%	0.37 (drag count)	0.88%
	8	0.23%	0.32 (drag count)	0.46%
	9	0.23%	0.28 (drag count)	0.53%
	10	0.19%	0.24 (drag count)	0.47%
	11	0.13%	0.20 (drag count)	0.47%
	12	0.14%	0.16 (drag count)	0.24%
	13	0.11%	0.13 (drag count)	0.11%
	14	0.07%	0.04 (drag count)	0.19%

7. Conclusion and future work

In this paper, we performed a study on morphing airfoil technology in the subsonic flight regime, considering high Reynolds numbers and using the SA turbulence model included in the high-fidelity CFD simulations. We proposed a data-driven framework to control the morphing process with an efficient and safe way to reach a shape with improved values of the aerodynamic coefficients. The online solution is based on a data-driven controller combined with a surrogate model and an MGDA. Without full knowledge of the aerodynamic parameters (lift, drag, and pitching moment coefficients), the proposed learning framework searches for an airfoil shape that minimizes a metric of performance associated to drag, lift, and pitching moment coefficients. The solution uses online data to improve the accuracy of the predictions of the aerodynamic coefficients provided by the surrogate model along the trajectory. The optimization framework focuses on subtle airfoil deformations to assure a smooth trajectory between the initial and the final shape. The efficiency of our proposed solution was shown in numerical examples, resulting in a significant reduction in prediction error. The proposed online learning algorithm successfully prevented the system from reaching unsafe conditions and predicted, with high accuracy, a trajectory toward a shape that provided an improvement of 6.17% of the ratio $\left(\frac{c_l}{c_d}\right)$ with truly reduction of c_d and increase of c_l.

Future work includes the improvement on online search algorithm to find efficient trajectories toward the final shape; the development of a metric to select efficiently new trajectories to be considered for the morphing process during the flight; the use of data gathering online to improve the accuracy of the surrogate model prediction; and inclusion of the transition-to-turbulence effects on CFD simulations. Finally, we plan to extend the current analysis to a 3D morphing wing and evaluate the effect of morphing on airplane stability.

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.

Acknowledgements

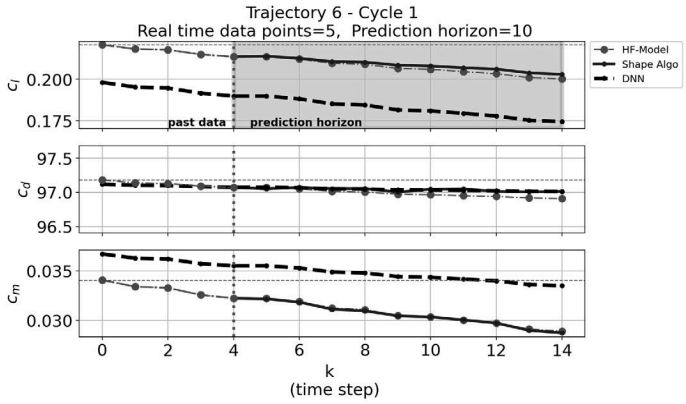
The authors gratefully acknowledge the partial support for this research provided by NSF S&AS-1849198 and by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, Brazil, under the Research Grant No. 99999.005340/2015-02.

Appendix A. Robustness analysis

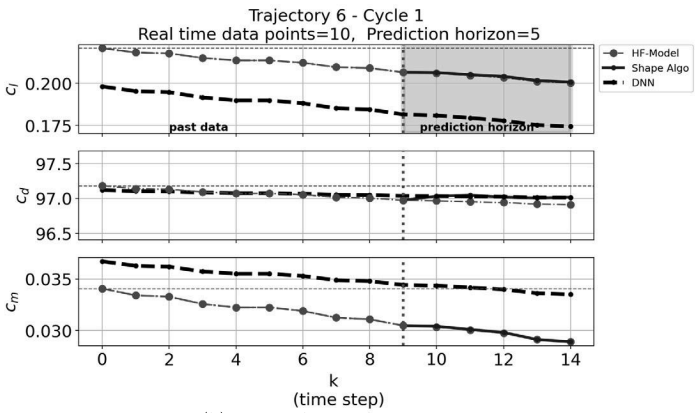
This appendix includes the analysis of 20 different trajectories used to perform the robustness analysis of the data-driven learning algorithm discussed in Section 6.1. The figures below describe the evolution of c_l, c_d, and c_m for each trajectory considering the use of five and ten real-time data points for prediction.

Appendix B. Sensitivity analysis

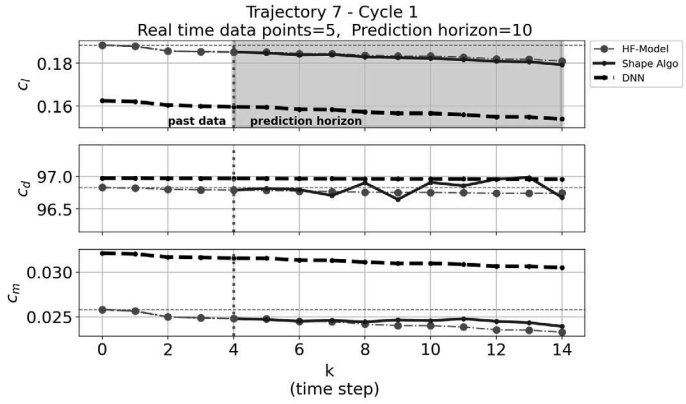
This appendix contains the result of the sensivity analysis of the data-driven learning algorithm discussed in Section 6.2. The figures below describe the evolution of c_l, c_d, and c_m for “Trajectory 19” considering different quantities of real-time data points.



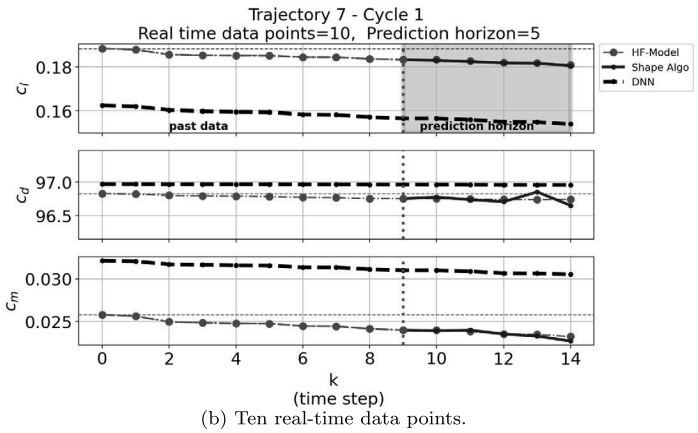
(a) Five real-time data points.



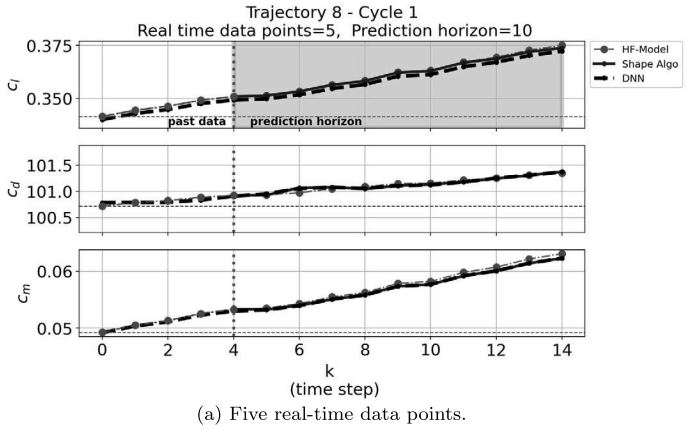
(b) Ten real-time data points.



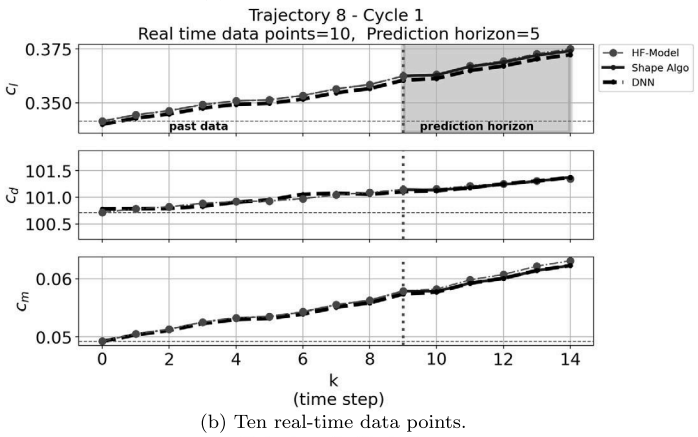
(a) Five real-time data points.



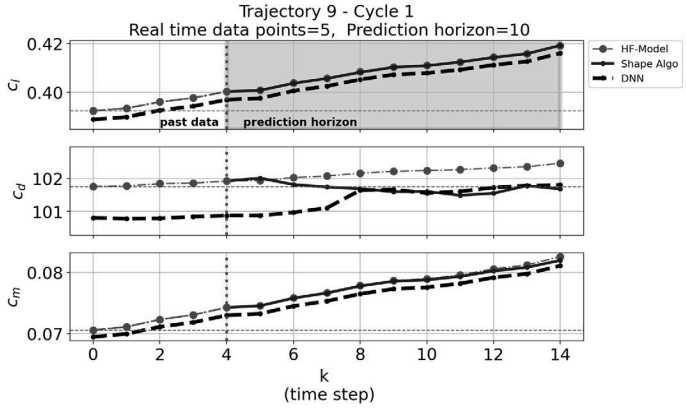
(b) Ten real-time data points.



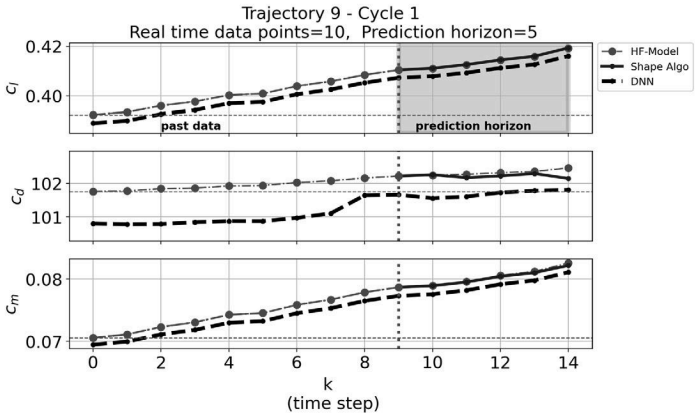
(a) Five real-time data points.



(b) Ten real-time data points.

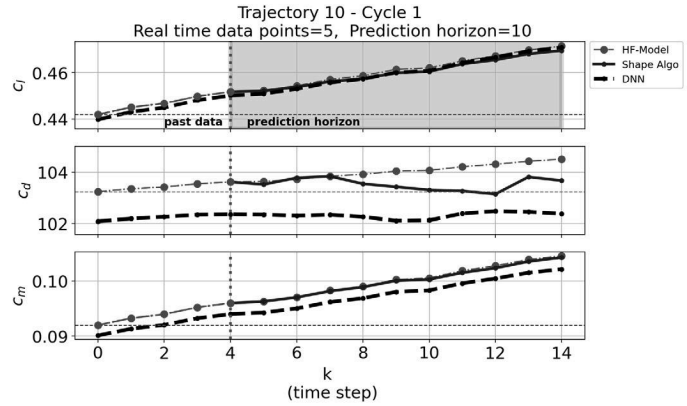


(a) Five real-time data points.

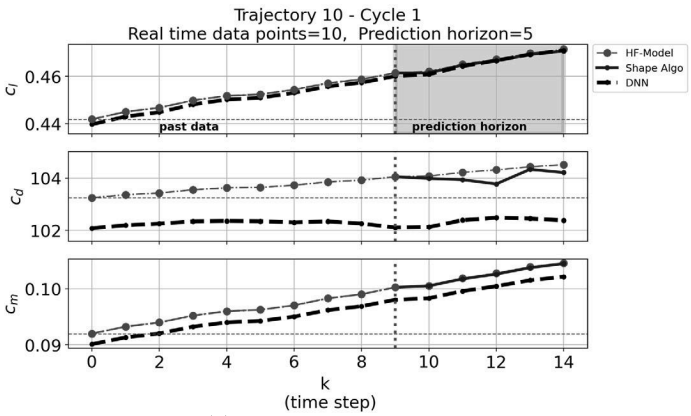


(b) Ten real-time data points.

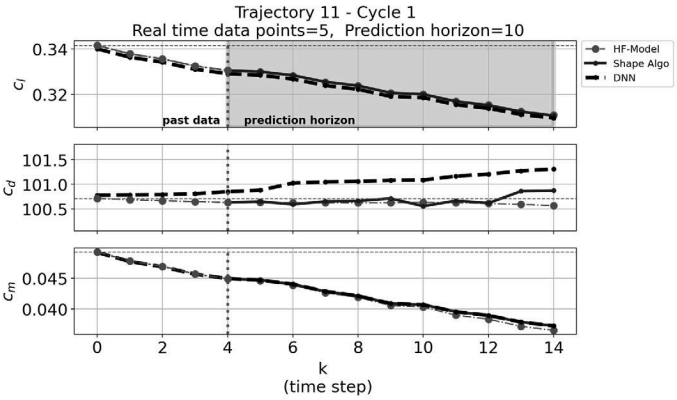
Fig. A.1. Robustness analysis using 20 different trajectories.



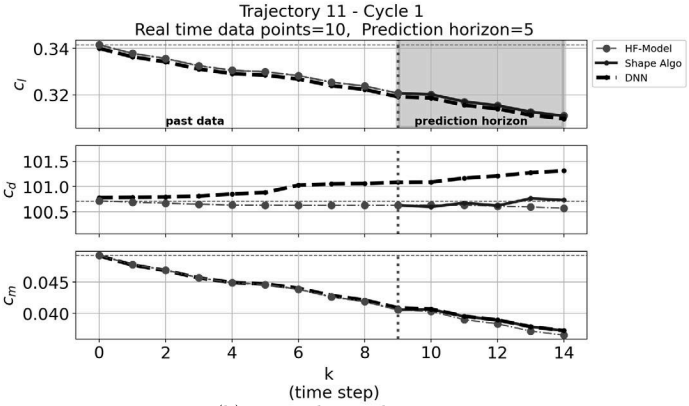
(a) Five real-time data points.



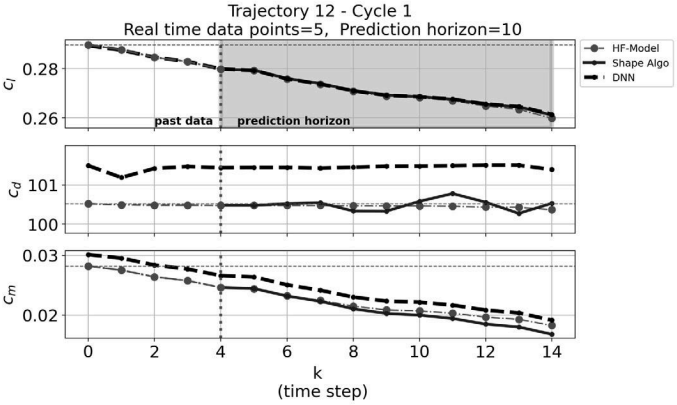
(b) Ten real-time data points.



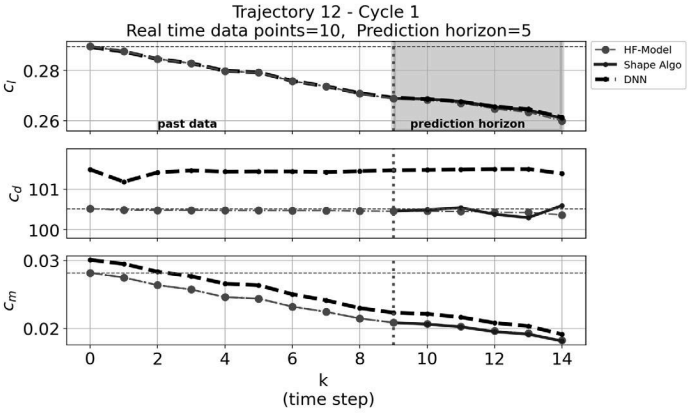
(a) Five real-time data points.



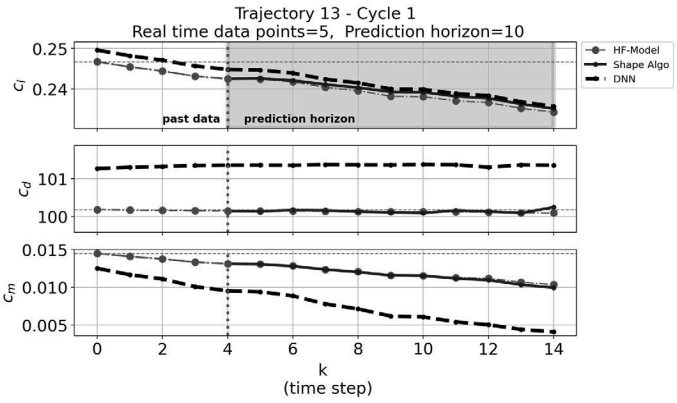
(b) Ten real-time data points.



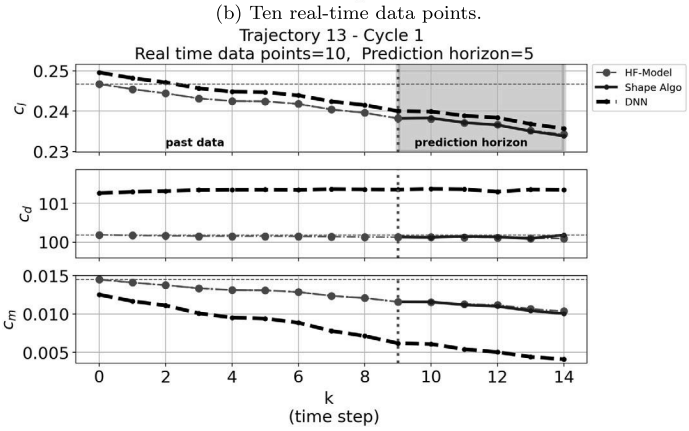
(a) Five real-time data points.



(b) Ten real-time data points.

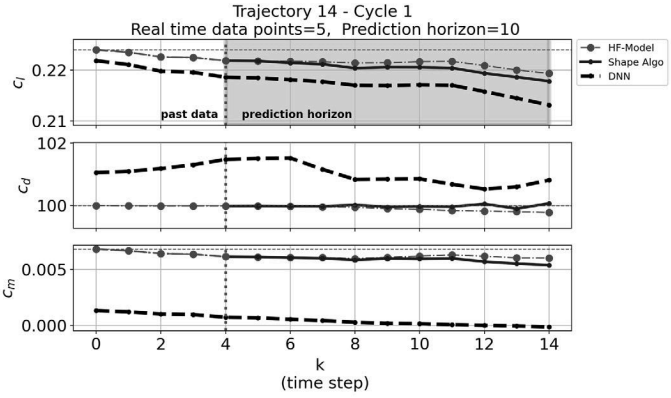


(a) Five real-time data points.

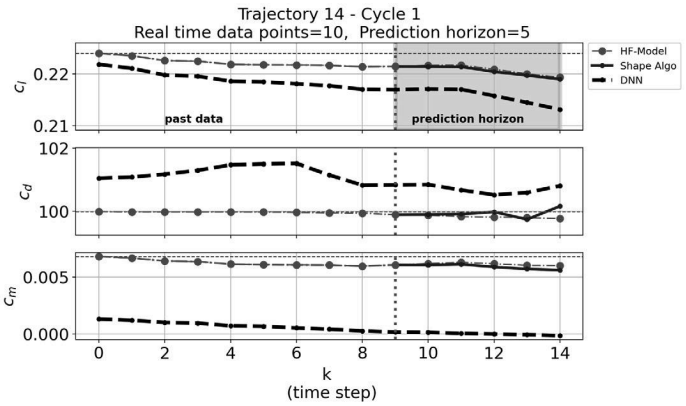


(b) Ten real-time data points.

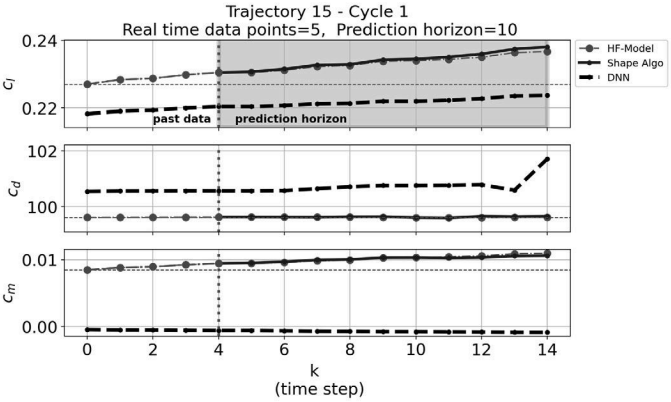
Fig. A.1. (*continued*)



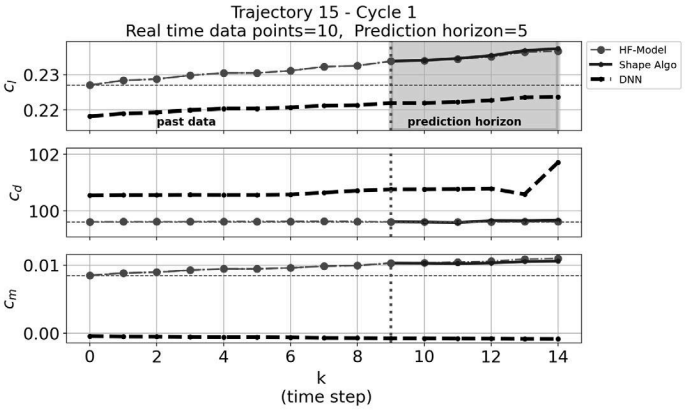
(a) Five real-time data points.



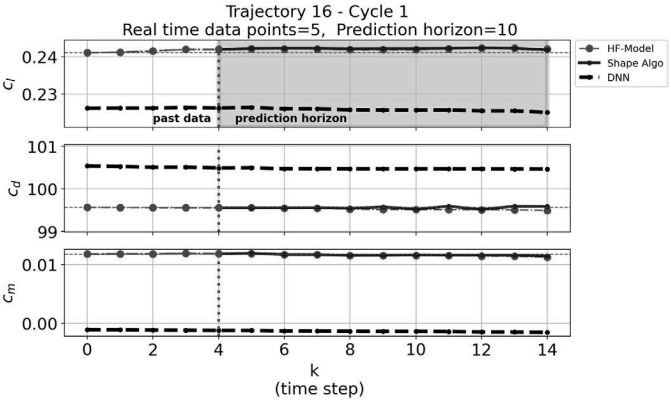
(b) Ten real-time data points.



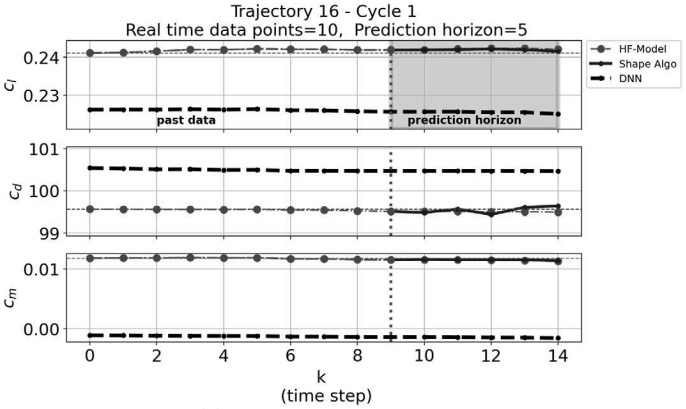
(a) Five real-time data points.



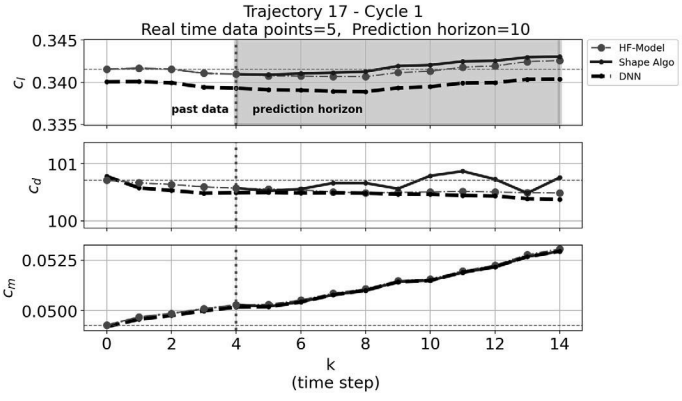
(b) Ten real-time data points.



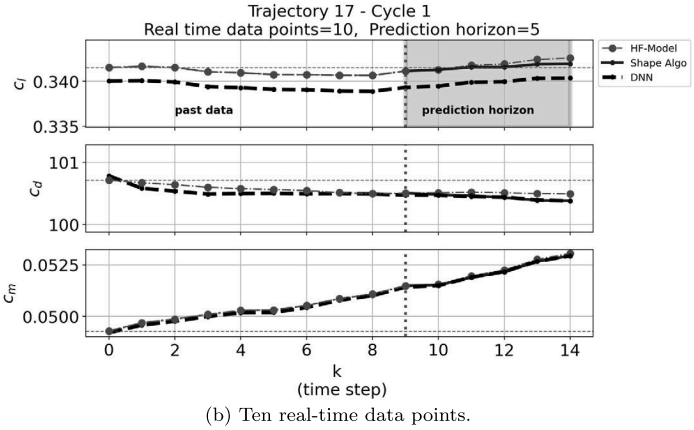
(a) Five real-time data points.



(b) Ten real-time data points.

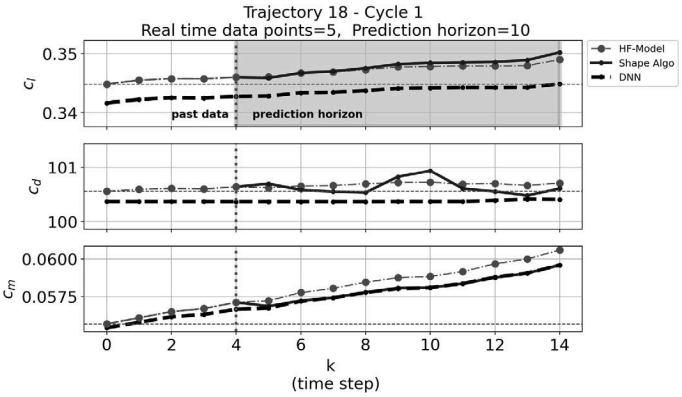


(a) Five real-time data points.

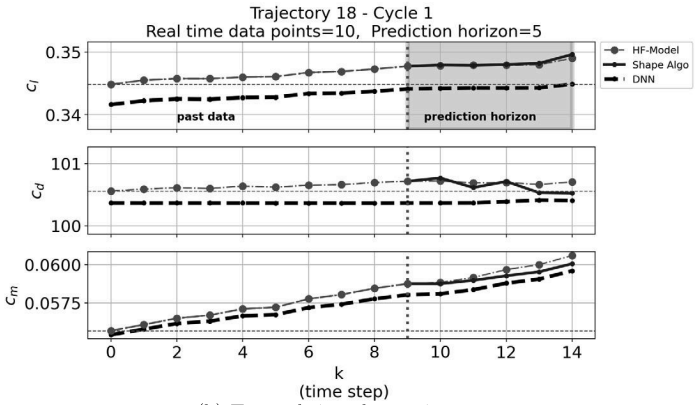


(b) Ten real-time data points.

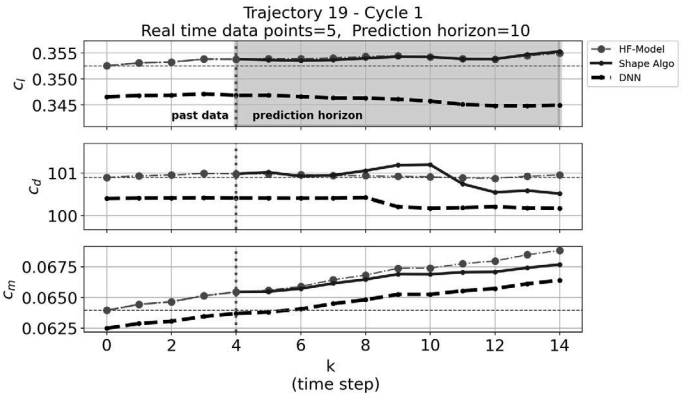
Fig. A.1. (*continued*)



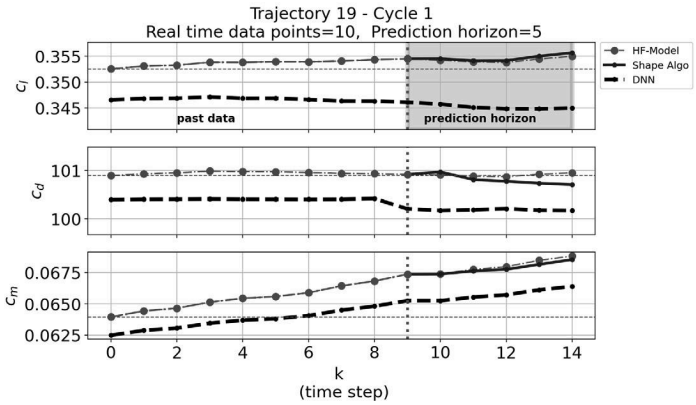
(a) Five real-time data points.



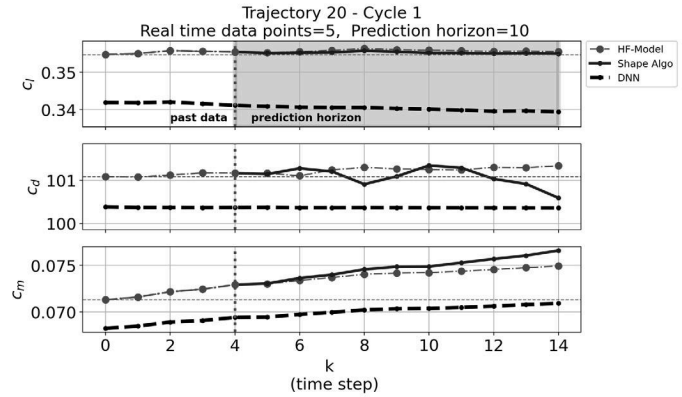
(b) Ten real-time data points.



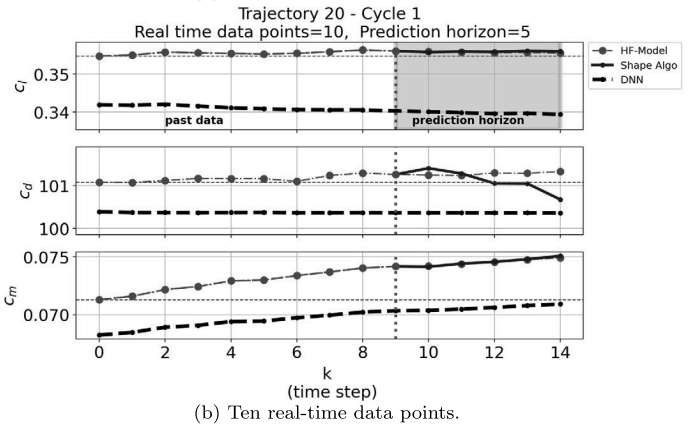
(a) Five real-time data points.



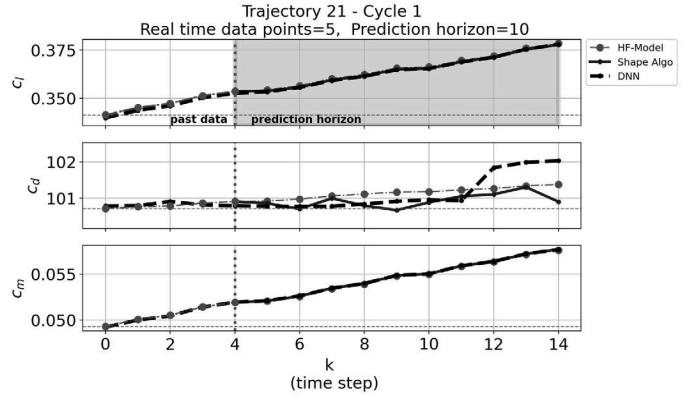
(b) Ten real-time data points.



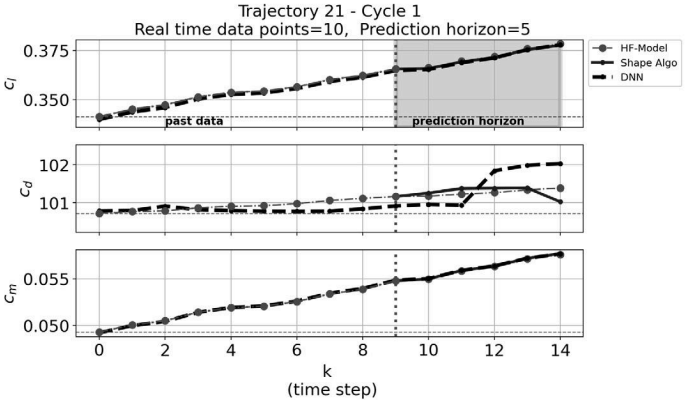
(a) Five real-time data points.



(b) Ten real-time data points.

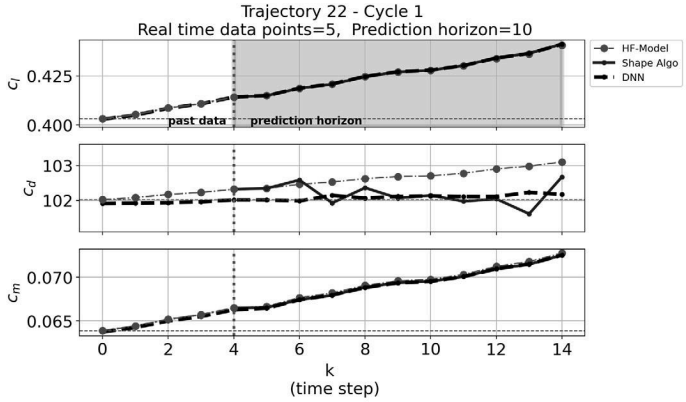


(a) Five real-time data points.

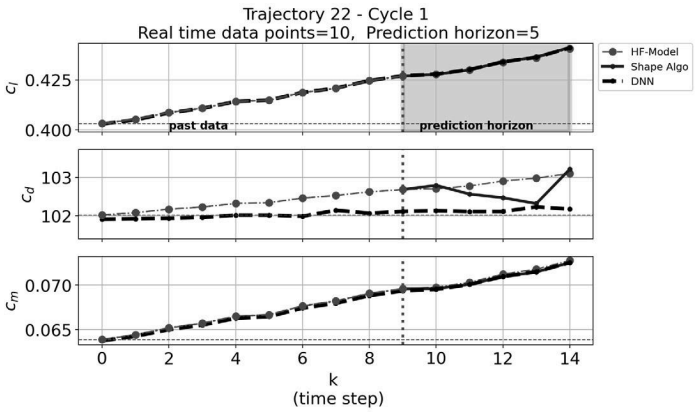


(b) Ten real-time data points.

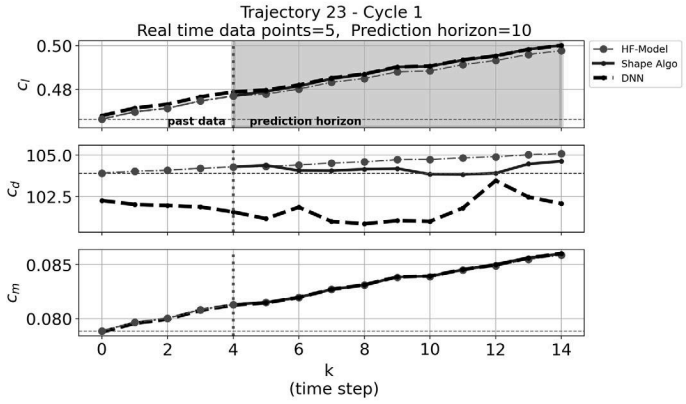
Fig. A.1. (*continued*)



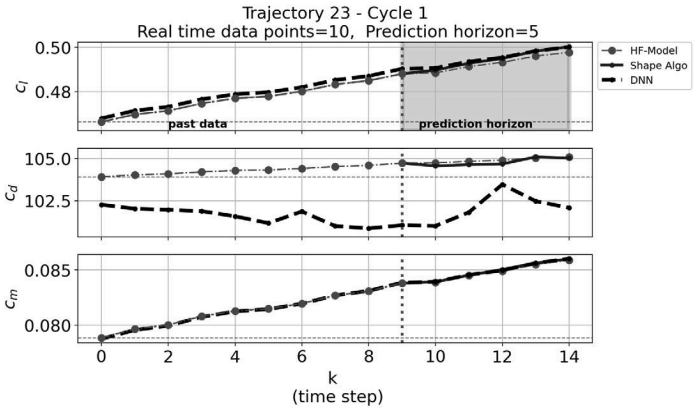
(a) Five real-time data points.



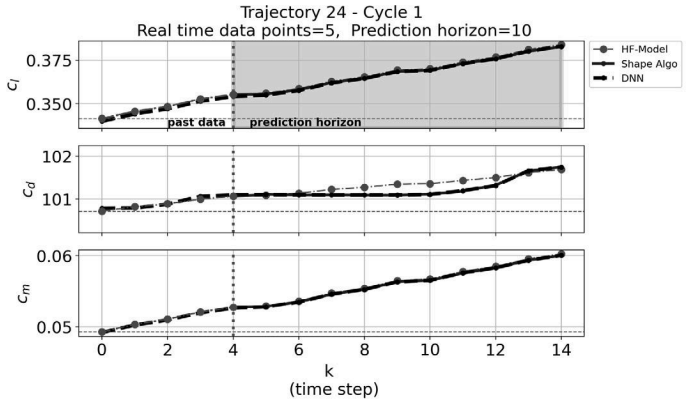
(b) Ten real-time data points.



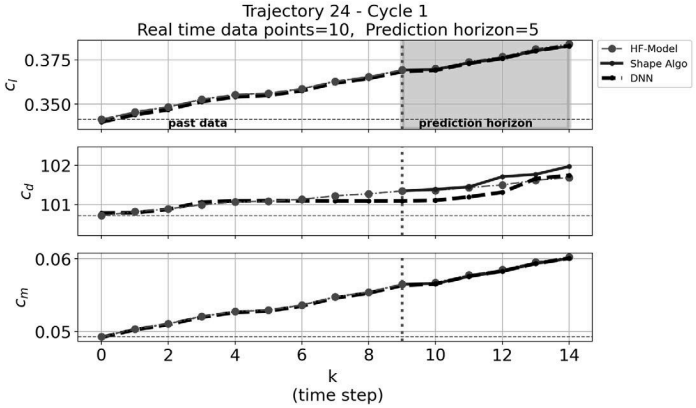
(a) Five real-time data points.



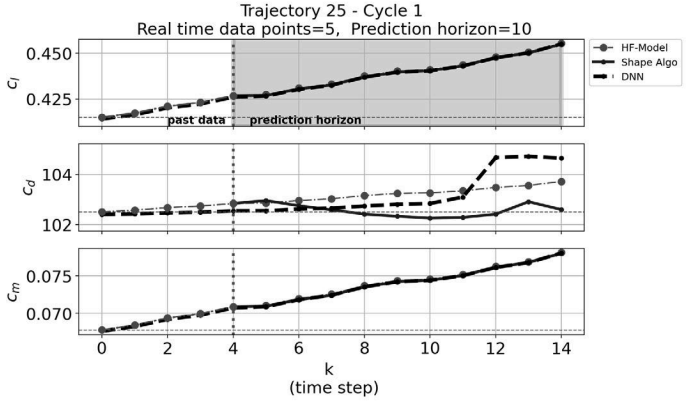
(b) Ten real-time data points.



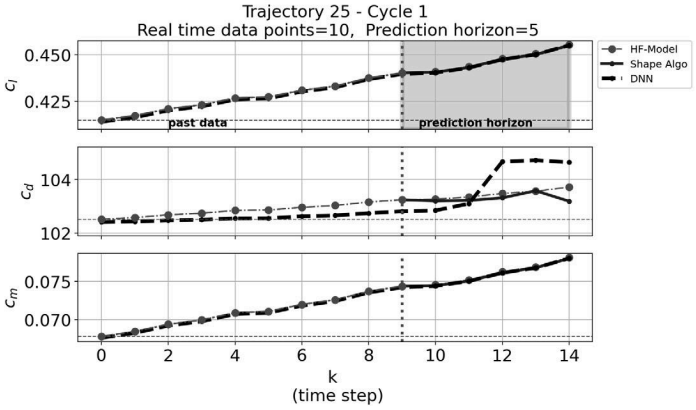
(a) Five real-time data points.



(b) Ten real-time data points.



(a) Five real-time data points.



(b) Ten real-time data points.

Fig. A.1. (*continued*)

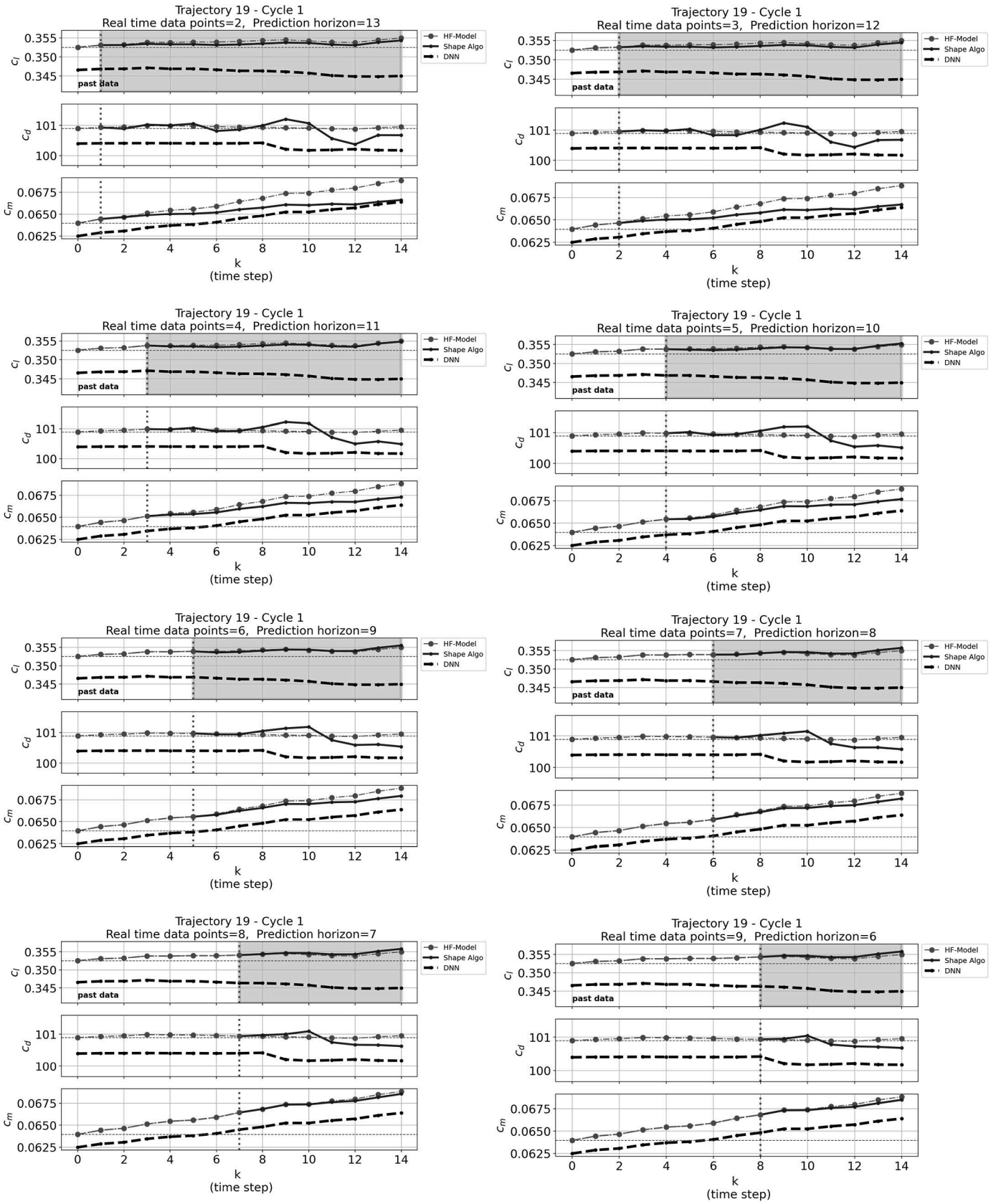


Fig. B.1. Sensitivity analysis for Trajectory 19.

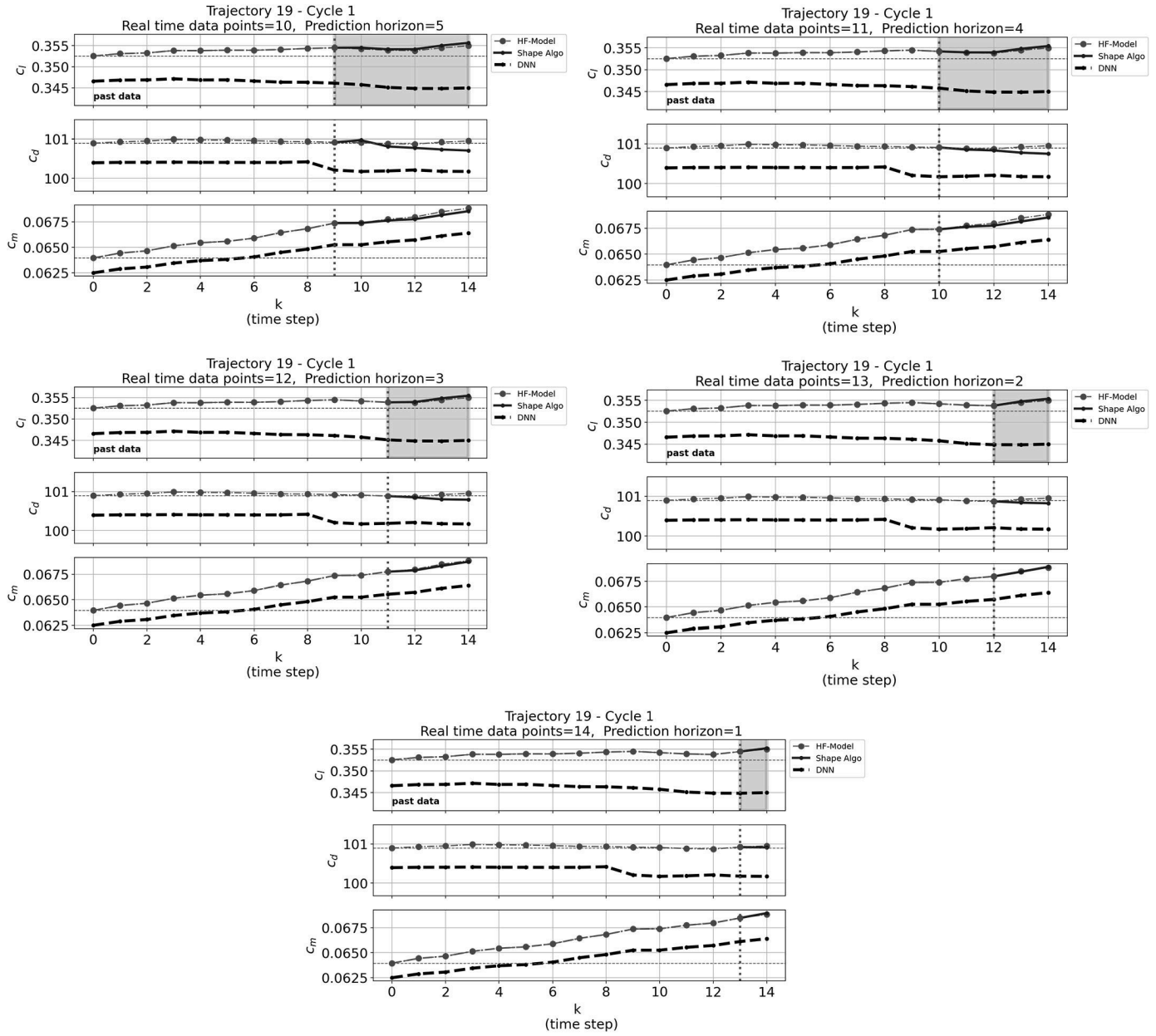


Fig. B.1. (*continued*)

References

[1] F. Afonso, J. Vale, F. Lau, A. Suleman, Performance based multidisciplinary design optimization of morphing aircraft, Aerosp. Sci. Technol. 67 (2017) 1–12.

[2] A. Koreanschi, O.S. Gabor, J. Acotto, G. Brianchon, G. Portier, R.M. Botez, M. Mamou, Y. Mebarki, Optimization and design of an aircraft's morphing wing-tip demonstrator for drag reduction at low speeds, part ii - experimental validation using infra-red transition measurement from wind tunnel tests, Chin. J. Aeronaut. 30 (2017) 164–174.

[3] L. Chu, Q. Li, F. Gu, X. Du, Y. He, Y. Deng, Design, modeling, and control of morphing aircraft: a review, Chin. J. Aeronaut. 35 (2022) 220–246.

[4] J.P. Eguea, P.D. Bravo-Mosquera, F.M. Catalano, Camber morphing winglet influence on aircraft drag breakdown and tip vortex structure, Aerosp. Sci. Technol. 119 (2021) 107148.

[5] G.L.O. Halila, J.R.R.A. Martins, K.J. Fidkowski, Adjoint-based aerodynamic shape optimization including transition to turbulence effects, Aerosp. Sci. Technol. 107 (2020) 106243.

[6] J. Valasek, M. Tandale, J. Rong, A reinforcement learning - adaptive control architecture for morphing, J. Aerosp. Comput. Inf. Commun. 2 (2005) 174–195.

[7] A. Lampton, A. Niksch, J. Valasek, Morphing airfoils with four morphing parameters, in: Guidance, Navigation and Control Conference and Exhibit, AIAA, 2008, p. 7282.

[8] V.G. Goecks, P.B. Leal, T. White, J. Valasek, D.J. Hartl, Control of morphing wing shapes with deep reinforcement learning, in: Proceedings of the 2018 SciTech Forum, AIAA, 2018, pp. 1–12.

[9] D. Xu, Z. Hui, Y. Liu, G. Chen, Morphing control of a new bionic morphing uav with deep reinforcement learning, Aerosp. Sci. Technol. 92 (2019) 232–243.

[10] J.M. Magalhães Júnior, G.L.O. Halila, Y. Kim, T. Khamvilai, K.G. Vamvoudakis, Intelligent data-driven aerodynamic analysis and optimization of morphing configurations, Aerosp. Sci. Technol. 121 (2022) 107388.

[11] A.H. Barr, Global and local deformations of solid primitives, in: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH-84, vol. 378, 1984, pp. 21–30.

[12] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models, in: Proceedings of the SIGGRAPH, 1986, pp. 151–160.

[13] Y. Shen, W. Huang, L. Yan, T. Tian Zhang, Constraint-based parameterization using ffd and multi-objective design optimization of a hypersonic vehicle, Aerosp. Sci. Technol. 100 (2020) 105788.

[14] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models, SIGGRAPH Comput. Graph. 20 (1986) 151–160.

[15] J. Prochazkova, Free form deformation methods - the theory and practice, in: 16th Conference on Applied Mathematics, APLIMAT 2017 - Proceedings, 2017, pp. 1276–1282.

[16] J.A. Samareh, Aerodynamic shape optimization based on free-form deformation, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004, p. 4630.
[17] PACE, Partnership for an advanced computing environment (PACE), Available at http://www.pace.gatech.edu, 2017.
[18] N. Secco, G.K.W. Kenway, P. He, C.A. Mader, J.R.R.A. Martins, Efficient mesh generation and deformation for aerodynamic shape optimization, AIAA J. (2021).
[19] C.A. Mader, G.K.W. Kenway, A. Yildirim, J.R.R.A. Martins, ADflow: an open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization, J. Aerosp. Inform. Syst. 17 (2020) 508–527.
[20] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes, in: 14th Fluid and Plasma Dynamics Conference, AIAA, 1981, p. 1, 1981-1259.
[21] E. Turkel, V.N. Vatsa, Effects of artificial viscosity on three-dimensional flow solutions, AIAA J. 32 (1994) 39–45.
[22] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[23] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[24] A. Yildirim, G.K.W. Kenway, C.A. Mader, J.R.R.A. Martins, A Jacobian-free approximate Newton–Krylov startup strategy for RANS simulations, J. Comput. Phys. 397 (2019) 108741.
[25] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, Rech. Aérosp. 1 (1994) 5–21.
[26] G. Halila, A. Yildirim, C.A. Mader, K.J. Fidkowski, J.R.R.A. Martins, Linear stability-based smooth Reynolds-averaged Navier-Stokes transition model for aerodynamic flows, AIAA J. 60 (2022) 1077–1090.
[27] R.M. Cummings, W.H. Mason, S.A. Morton, D.R. McDaniel, Applied Computational Aerodynamics: A Modern Engineering Approach, Cambridge Aerospace Series, Cambridge University Press, New York, NY, 2015.
[28] A. Jameson, Computational Aerodynamics, Cambridge Aerospace Series, Cambridge University Press, 2022.
[29] F. Johnson, E. Tinoco, J. Yu, Thirty years of development and application of cfd at Boeing Commercial Airplanes, Seattle, in: AIAA Paper No. 2003-3439, in: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, AIAA, Oralndo, FL, 2003.
[30] J. Li, M. Zhang, Data-based approach for wing shape design optimization, Aerosp. Sci. Technol. 112 (2021) 106639.
[31] G.L. Halila, J.R. Martins, K.J. Fidkowski, Adjoint-based aerodynamic shape optimization including transition to turbulence effects, Aerosp. Sci. Technol. 107 (2020) 106243.
[32] J.R.R.A. Martins, A. Ning, Engineering Design Optimization, Cambridge University Press, Cambridge, UK, 2022, https://doi.org/10.1017/9781108980647.
[33] Z. Lyu, Z. Xu, J.R.R.A. Martins, Benchmarking optimization algorithms for wing aerodynamic design optimization, in: Proceedings of the 8th International Conference on Computational Fluid Dynamics, ICCFD8-2014-0203, Chengdu, Sichuan, China, 2014.
[34] Z. Lyu, G.K.W. Kenway, J.R.R.A. Martins, Aerodynamic shape optimization investigations of the common research model wing benchmark, AIAA J. 53 (2015) 968–985.
[35] Y. Yu, Z. Lyu, Z. Xu, J.R.R.A. Martins, On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization, Aerosp. Sci. Technol. 75 (2018) 183–199.
[36] J.W. Poldeman, J.C. Willems, Introduction to Mathematical Systems Theory: A Behavioral Approach, vol. 3, Springer, 1998.
[37] J. Coulson, J. Lygeros, F. Dörfler, Data-enabled predictive control: in the shallows of the DeePC, in: Proceedings of the 18th European Control Conference, 2019.