Aerospace Science and Technology 121 (2022) 107388

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

Intelligent data-driven aerodynamic analysis and optimization of morphing configurations

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

[a] Georgia Institute of Technology, GA 30332, United States
[b] Instituto de Aeronáutica e Espaço, 12228-904 São José dos Campos, SP, Brazil

ARTICLE INFO

Article history:
Received 3 November 2021
Received in revised form 14 January 2022
Accepted 24 January 2022
Available online 29 January 2022
Communicated by Tsourdos Antonios

Keywords:
Aerodynamic shape optimization
Data-driven aerodynamic design
Neural networks
Model predictive control
Aerodynamic design

ABSTRACT

In this paper, we develop an online, data-based framework for the aircraft airfoil to be able to optimally morph vertically. The proposed framework combines data-driven analysis, optimization, and control-theoretic tools to optimally morph airfoils while guaranteeing efficiency and safety. It incorporates a surrogate model that is based on a deep neural network that is used to predict the aerodynamic coefficients while a meta-heuristic optimization algorithm is employed to find shapes with reduced value of drag coefficient that fulfill the geometric and lift constraints. Finally, a data-driven shape controller is used to morph the airfoil while following smooth trajectories and small aerodynamic coefficient variations. Experimental numerical results show the efficacy of the proposed framework for different flight conditions.

1. Introduction

In the aerospace industry [1,2], engineers are continuously exploring the design space to produce a more efficient and safe aircraft. One relevant performance metric can be fuel consumption, which is directly affected by the drag, so that a constrained drag minimization is a constant target in aeronautical design.

Related work

To perform aerodynamic analysis and shape optimization [3], one needs a flow solver that can account for the physical and environmental phenomena of interest. The increasing computational power allows aerodynamicists to use increasingly complex computational aerodynamics tools. The use of CFD became the norm in airplane design [4,5] but despite the availability of HPC facilities, traditional CFD computations are still time consuming and computationally inefficient. In the early design phases, the engineers investigate many configurations and, therefore, the time required to perform each aerodynamic analysis is relevant. To alleviate computational requirements, traditional CFD solvers can be adopted to generate high-fidelity aerodynamic data that will, in turn, be employed to calibrate potential surrogate models. The work of [6] proposes model-based convolutional neural networks to perform airfoil aerodynamic analysis. The authors in [7] use a B-spline-based generative adversarial network model for shape parameterization associated with a combination of multilayer perceptron, recurrent neural networks, and a mixture of experts for surrogate modeling to enable airfoil aerodynamic predictions over a range of Mach and Reynolds numbers. The investigation in [7] includes aerodynamic shape optimization and other data-driven aerodynamic simulations are developed in [8–14] and the references therein.

Since a variety of flight conditions is usually encountered in daily operations, a single optimal shape is, in general, incapable of minimizing the drag over the entire flight envelope. To this end, aircraft leveraging morphing technologies are better suited to attain optimal aerodynamic performance under various flight conditions. But determining the geometric parameters of an aerodynamically efficient airfoil and morphing is challenge. For instance, in [15], the authors propose a methodology to obtain the optimal shape of a morphing block by using machine learning and adaptive dynamic inversion control. With a numerical example, the authors show that the method can learn the commands that produce the optimal shape while in [16], they extend the methodology to an air-vehicle using Q-learning to obtain the optimal shape change policy. In [17], the authors use a Q-learning method combined with analytical aerodynamic calculations to determine the optimal

* Corresponding author at: Aerospace Engineering Department, Montgomery Knight Building, Georgia Institute of Technology, 270 Ferst Dr NW, Atlanta, GA 30313, United States.
E-mail address: jose.magalhaes@gatech.edu (J.M. Magalhães Júnior).

https://doi.org/10.1016/j.ast.2022.107388
1270-9638/

Nomenclature

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

Abbreviations

ADflow	Open-source computational fluid dynamics solver
ANK	Approximate Newton-Krylov
CFD	Computational Fluid Dynamics
CRM	Common Research Mode
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
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
SA	Simulated Annealing

shape by changing the parameters of maximum camber location, airfoil angle of attack, and thickness. The work of [18] developed an aerodynamic model and a dynamic model of a morphing flying wing aircraft by using a constant strength source doublet panel method. The developed model provides a versatile tool for calculating the aerodynamic forces on the morphing aircraft. Unlike the aforementioned studies, this work combines the advantages of deep neural networks with data-driven shape control to determine the optimal morphing shape for efficiency and safety. Furthermore, the optimal trajectory between the initial and optimal shapes is determined to minimize the effect of morphing on the aircraft stability characteristics.

Contributions. The contributions of the present paper are threefold. First, we use a surrogate model based on a deep neural network to perform an offline prediction of the aerodynamic parameters for a given airfoil shape under certain flight conditions. Then, a data-driven predictive controller is employed to predict the same aerodynamic parameters in an online manner. Finally, we develop an online learning algorithm to obtain, based on measured data, an airfoil shape with reduced drag for a given flight condition.

Structure. The remainder of the paper is structured as follows. Section 2 formulates the problem of the morphing airfoil along with some relevant mathematical background. In Section 3, we describe the surrogate model to predict the aerodynamic parameters while Section 4 designs the data-driven shape controller used during the online learning. In Section 5, we describe the framework used to obtain the optimal set of intermediate shapes. Finally, Section 6 concludes the present investigation and discusses future research directions.

2. Problem formulation

Consider an airfoil that can change the shape in one dimension, i.e., the y-direction, starting from a baseline shape. We use the NACA 2412 airfoil as the baseline configuration because it is widely used in low-speed, general aviation airfoils. It is the airfoil used for single engine Cessna aircraft, such as the Cessna 152, which attains maximum flight speeds. The shape is defined by $1,000$ ordered pairs (x_s, y_s) distributed over the surface of the airfoil. In order to control the shape, instead of using the $1,000$ pairs (x_s, y_s), we shall instead use 20 points by applying a technique called FFD [19, 20].

2.1. Modeling morphing airfoil

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

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

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 $y[k] \in \mathbb{R}^2$ is the output vector consisting of the lift and drag coefficients at a given shape, i.e., $y = [f_{c_l}(\cdot), f_{c_d}(\cdot)]^{\mathrm{T}}$, where $f_{c_l} : \mathbb{R}^{20} \to \mathbb{R}$ and $f_{c_d} : \mathbb{R}^{20} \to \mathbb{R}$ denote the unknown nonlinear functions representing the lift and drag coefficients, respectively. The following assumptions are now needed.

Assumption 1. The system (1) is controllable and $x[\cdot]$ is known for all values of $k \in \{0, 1, \ldots, L\}$. □

Assumption 2. The functions $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ are bounded and continuous. Moreover, small variations on the shape x imply small variations on the values of $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$. □

Assumption 3. The flight condition, defined by the Reynolds, Mach, and Alpha numbers, remains unchanged during the whole morphing procedure. □

Assumption 4. Without loss of generality, assume that the shape deformations are no greater than 0.15% of the initial shape. □

2.2. Optimal shape

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 the optimal shape denoted as $x^\star \in \mathcal{X}$ that has the minimum drag while maintaining a sufficient lift. The optimal shape can be found by the solution of the following optimization problem:

$$\begin{aligned} \underset{u\in\mathbb{R}^{20}}{\text{minimize}} \quad & f_{c_d}(x[L]) \\ \text{subject to} \quad & x[k+1] = x[k] + u[k], \quad \forall k \in \{0, ..., L-1\} \\ & f_{c_l}(x[k]) \geqslant f_{c_l}(x[0]), \quad \forall k \in \{1, ..., L\} \\ & x[0] = \bar{x}. \end{aligned} \tag{2}$$

Thus, the goal is to find an optimal trajectory, namely $x^\star$, with lower drag without compromising the initial value of lift. As $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ are unknown nonlinear functions, a surrogate model for these two functions must be constructed.

The problem can be summarized as follows.

Problem 1. Find a morphing shape with lower drag and acceptable lift that could be tractably estimated given that functions $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ are unknown. □

2.3. Output estimation

Since morphing the wing can affect the stability of the aircraft, we aim to develop an algorithm to morph the shape in an efficient and safe way. Efficiency means that morphing occurs only when a new shape x with improved values of $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ exists and can be reached; while safety means that the trajectory from the baseline to the new shape x will not reach a shape with "unsafe" values of $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$. The constraints can be mathematically described as follows,

$$f_{c_l}(x[k]) \geqslant 0.9 f_{c_l}(x[0]);$$
$$f_{c_d}(x[k]) \leqslant 1.1 f_{c_d}(x[0]), \qquad \forall k \in \{1, 2, \ \ldots, L\}.$$

It is known that a surrogate model has inherent errors on the prediction and thus in order to improve the prediction of $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ we shall use data gathered along the system's trajectory.

The problem can be summarized as follows.

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

3. Surrogate model and shape optimization

In this section we will deal with Problem 1. The approach used for modeling is based on a DNN where the training dataset is generated from CFD techniques. Once the surrogate model is obtained the meta-heuristic optimization technique called SA is used to solve (2) for the optimal shape.

To perform aerodynamic analysis and shape optimization, one needs a flow solver that can account for the physical phenomena of interest. Although the CFD simulations can provide accurate predictions of lift and drag coefficients, c_l and c_d, respectively, they still require extensive computational power. This may potentially be an issue during the early design phases, during which many configurations must be explored. Thus, we propose the development of a surrogate model based on a DNN to accurately predict the aerodynamic parameters. This development consists of a generation of 2,000 morphing shapes from a baseline shape and the training of two fully connected neural networks.

We will focus on the aerodynamic shape optimization and we will constrain the maximum airfoil thickness so that it will not be smaller than the lower bound discussed in section 3.3. Additional considerations from other disciplines and from regulatory requirements should be included in the decision process considering best design practices.

3.1. ADflow-CFD data generation

To generate the training set, we use high-fidelity CFD simulations performed with ADflow. ADflow is an open source CFD solver [21] and has options to solve Euler, laminar Navier–Stokes, and RANS equations in steady, unsteady, and time-spectral modes, with multiblock structured and overset meshes. The governing equations are discretized using the finite volume method with first and second order stencils. A PTC strategy is used to convert the flow equations. The in-viscid fluxes are discretized by using three different numerical schemes: the scalar Jameson–Schmidt–Turkel [22] (JST) artificial dissipation scheme, a matrix dissipation scheme based on the work of Turkel and Vatsa [23], and a MUSCL based on the work of van Leer [24] and Roe [25]. The viscous flux gradients are calculated by using the Green–Gauss approach while the residual equations can be converged with four distinct algorithms. An ANK solver is implemented and can be used as a globalization scheme for the full Newton–Krylov algorithm [26]. We use the Spalart–Allmaras turbulence model [27] as closure for all CFD simulations used.

The numerical methods and models in ADflow have been validated in Refs. [28,29] that present validation against experimental airfoil data. ADflow has also been used to perform aerodynamic analysis and optimization of the CRM geometry [30,31,21].

For all the CFD simulations used in the data generation, we choose a 6-order decay in total residuals as a convergence criterion. We observe that this stopping criterion is reached, in general, after 100 nonlinear iterations for the test cases. For the Spalart-Allmaras (SA) turbulence model, used in this study, the convergence is possible due to the ANK solver robustness [26].

3.2. Parametric method

The adopted FFD technique [32] has been extensively [28,33]. This technique is based on the tensor product trivariate Bernstein polynomial [32,34,35]. The main idea is the insertion of the shape representation to the regular parallel-piped; subsequent transformation recomputes the position of the solid points with respect to the modified points of the lattice.

The vectors $\vec{S}$, $\vec{T}$ and $\vec{U}$ define the size and orientation of the control lattice and the lattice is uniformly layered by planes in their directions with several deformable cuboids to embed the object as shown in Fig. 1. Given an established basis function (trivariate Bernstein polynomials), the changing positions of the vertexes of the cuboids (FFD control points) cause the inner model to deform [33].

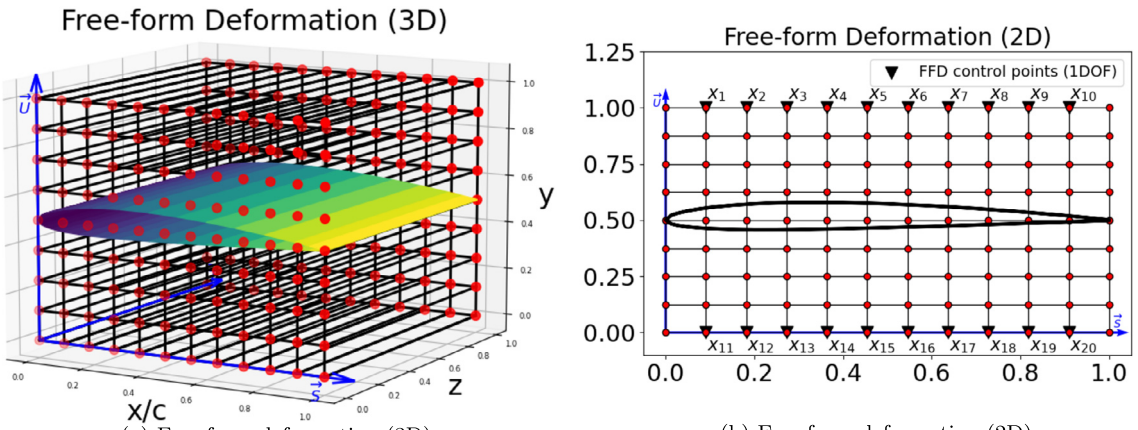
Every object point $X(x, y, z)$ interior to control lattice has (s, t, u) coordinates in the coordinate system on a parallel-piped region:

$$\vec{X} = \vec{X}_0 + s\vec{S} + t\vec{T} + u\vec{U}, \tag{3}$$

where s, t and u are the local coordinates of any point interior to the control lattice.

Let $P_{ijk}, i = 0, ..., l, j = 0, ...m, k = 0, ..., n$ be control points on a lattice (FFD control points). 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:

$$\begin{aligned} X^\star(x(s), y(t), z(u)) = \sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \Bigg[\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \\ \times \left(\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k P_{ijk} \right) \Bigg] \end{aligned} \tag{4}$$



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

Fig. 1. The parallel-piped lattice.

where (s, t, u) are the parameters in (3).

The pseudocode to obtain the renewed position of a point X inside the lattice using equation (4) can be written as:

Algorithm 1 Shape morphing.

Input: Parallel-piped points, parameters (s, t, u).
Output:

```
procedure
    for i = 0, ..., l do
        for j = 0, ..., m do
            for k = 0, ..., n do

            end for
        end for
    end for
end procedure
```

FFD can 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 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.

Using the airfoil NACA 2412 as baseline, a new airfoil shape is obtained by moving vertically the FFD control points presented in Fig. 1b. The (x_s, y_s) coordinates of the new shape are computed using the equation (4) implemented into a standalone Python package PyGeM.

3.3. Shape generation

We used the NACA 2412 airfoil as the baseline shape to generate morphed shapes. For the CFD analysis, we discretize the airfoil using $1,000$ points distributed over the surface. This quantity of points is necessary to obtain accurate values of c_l and c_d. Therefore, the airfoil is defined by $1,000$ coordinates (x_s, y_s) that represent $2,000$ design variables and variations on the values of (x_s, y_s) define a new shape.

To generate the new shapes from the baseline NACA 2412, we considered 20 FFD control points able to independently move on vertical direction, i.e., one DOF. These control points are identified as $(x_1, x_2, ..., x_{20})$ and evenly distributed as shown in Fig. 1b. The control points may retreat and expand small variations resulting in small deformations around the original NACA 2412 airfoil. We considered a geometric constraint that the deformation on morphing airfoil is no greater than 7.3% from the baseline shape. Fig. 2

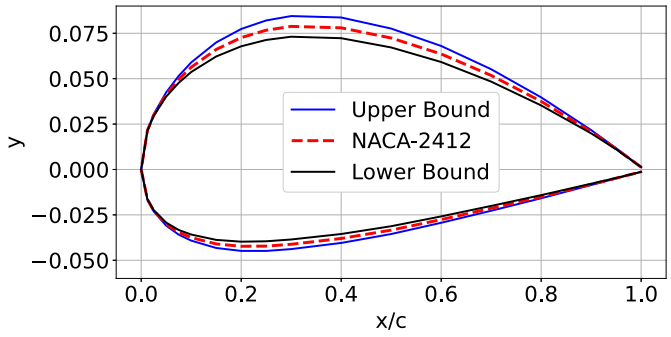


Fig. 2. Boundaries of the morphed shapes.

presents the upper and lower bounds of the morphed shapes obtained using the FFD control points with the geometric constraint.

The main goal of morphing the initial shape is to generate shapes with better values of aerodynamic parameters for a given flight condition. The constraint of maximal deformation prevents the generation of inappropriate shapes with high drag coefficient and low lift coefficient. Fig. 3 illustrates four shapes generated under the geometric constraints.

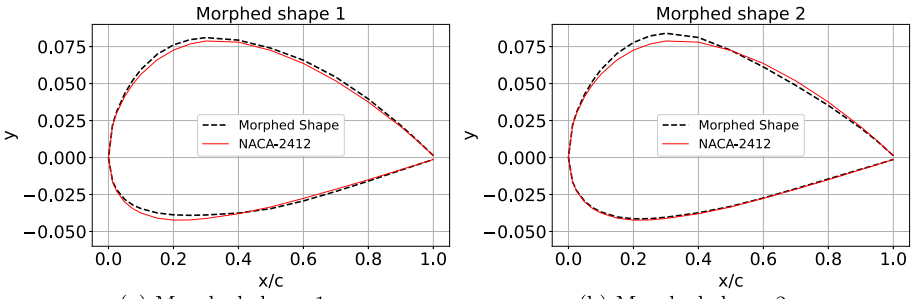
3.4. Data-set

We consider 3 parameters to describe a flight condition: Re, Mach, and α. To obtain an accurate surrogate model, it is important to use a dataset with many flight conditions.

A morphed shape is defined by 20 FFD control points namely $x_1, x_2, ...x_{20}$ where each control point can take values in the range $[-0.5, 0.5]$. To generate a new morphed shape, we select randomly the value of each FFD control point using uniform distribution. To illustrate how each control point is distributed, Fig. 4a shows the values of x_1 of the $1,000$ morphed shapes used to train the neural networks. Using a uniform distribution for each FFD control point, we can generate $1,000$ morphed shapes as shown in Fig. 4b.

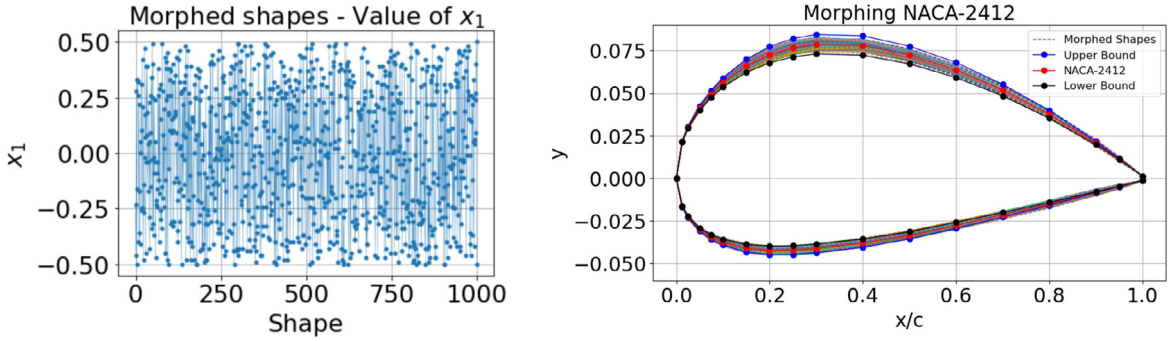
In order to have a representative database, we distributed the $1,000$ morphed shapes in 110 flight conditions such that we had at least 100 morphed shapes for each flight condition as described in Table 1.

To obtain accurate neural networks with low generalization error, we used cross-validation to prevent over-fitting [36,37], with a validation set comprised by datapoints with: flight conditions not used for training; morphed shapes not used for training; and a combination of flight conditions and morphed shapes not used



(a) Morphed shape 1. (b) Morphed shape 2. (c) Morphed shape 3. (d) Morphed shape 4.

Fig. 3. Examples of morphed shapes under geometric constraints.



(a) Values of x_1 (uniform distribution). (b) 1,000 morphed shapes uniformly distributed.

Fig. 4. The parallel-piped lattice. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 1
Data-set that is used to train and test the surrogate model.

	Flight conditions				
	Re (10^6)	Mach	Alpha	Shapes	Qty of datapoints
Train set1	[1.0, 2.0, 3.5]	[0.1, 0.2]	[0.0, 1.0, 1.5]	(0 to 199)	3600
Train set2	[1.5, 2.5]	[0.125, 0.175]	[0.5, 2.0]	(100 to 499)	3200
Train set3	[3.0]	[0.15, 0.1]	[0.0, 1.0]	(400 to 899)	2000
Train set4	[1.0, 2.0, 2.5, 3.0]	[0.1, 0.15, 0.175, 0.2]	[0.0, 0.5, 1.0, 1.5, 2.0]	(900 to 999)	8000
Test set	[1.25, 1.75, 2.25, 2.75]	[0.12, 0.14, 0.19]	[0.5]	(900 to 1149)	3000

for training. For the test set used to evaluate the accuracy of the neural networks, we selected 250 morphed shapes out of 2,000 to obtain 3,000 datapoints as described in Table 1. With this approach we achieved accurate neural networks to be used in the online learning solution presented in Section 5.

The data-set used to train the surrogate model is called *train set*. It is split in 4 groups and uses the first $1,000$ shapes, enumerated from 0 to 999. We also have a set of data to test the surrogate model to evaluate its accuracy to predict lift and drag coefficient. This set is called the test set and contains 250 shapes, enumerated from 900 to $1,149$. The test set is chosen such that there is no overlap between training and test sets and therefore we can accurately evaluate the neural network and detect over-fitting if it exists. The dataset used to train and test the surrogate model is described on Table 1.

We morph the baseline shape to obtain new airfoil shapes, called improved morphed shapes, with better values of aerodynamic parameters. We seek to minimize drag coefficient for given lift coefficient constraint, which corresponds to improving the lift-to-drag ratio, which is a direct metric of airfoil aerodynamic efficiency. Fig. 5 shows the values of c_l and c_d for the morphed shapes used in flight condition of *test set*. The clusters are grouped by the different flight conditions. The improved shape would be points within each cluster which has higher lift and lower drag coefficient. The morphed shapes are identified as blue dots while the

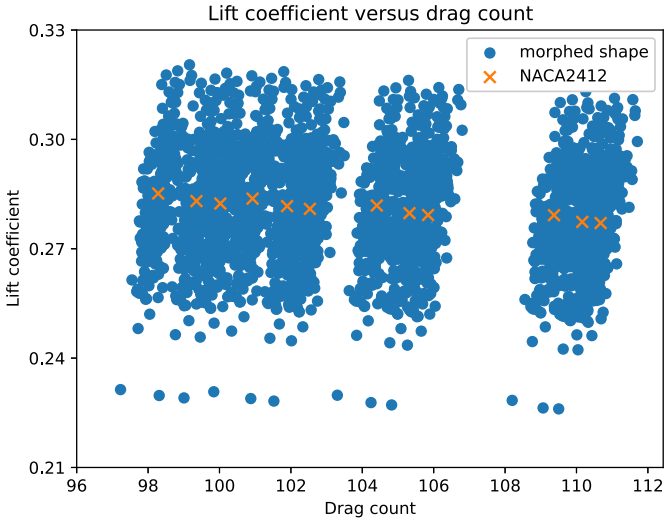


Fig. 5. Values of lift coefficient and drag count of the morphed shapes.

orange 'χ' represents the values of c_l and c_d for the baseline shape, the NACA 2412 airfoil.

3.5. Deep neural network

In (2), $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ are not known; one way to approximate these functions, is to generate data points using CFD. Fig. 5 presents the values of c_l and c_d provided by the CFD analysis. The numerical experiments were performed in a laptop with an Intel Pentium B950 processor (2.10 GHz, 1333, 2M cache) and the CFD tool requires, on average, 20 minutes to converge. This runtime corresponds to a 6th order decay in total residuals, which represents a standard engineering-level convergence for CFD analysis. To estimate c_l and c_d in a more efficient way, we propose to use two fully connected DNN to approximate $f_{c_l}(\cdot)$ and $f_{c_d}(\cdot)$ and predict each aerodynamic coefficient. The input data of the DNN is the vector $X \in \mathbb{R}^{23}$ where the first 20 elements define an unique airfoil shape and the remaining elements define the flight condition, i.e., values of Re, Mach and α. The approximation of these functions via neural network with N hidden layers can be mathematically expressed as follows,

$$z_m^{(1)} = \sum_{j=1}^{23} w_{jm}^{(0)} x_j + b_m^{(0)},\ m = 1, \cdots, M_1$$

$$a_k^{(i)} = \sum_{m=1}^{M_i} w_{mk}^{(i)} z_m^{(i)} + b_k^{(i)},\ k = 1, \cdots, K_i,\ i = 1, \cdots, N-1$$

$$z_m^{(i+1)} = \Phi(a_m^{(i)}),\ i = 1, \cdots, N-1,\ m = 1, \cdots, M_{i+1}$$

$$\tilde{f}_\lambda(\cdot) = \sum_{m=1}^{M_N} w_m^{(N)} z_m^{(N)},\ \lambda \in \{c_l, c_d\},$$

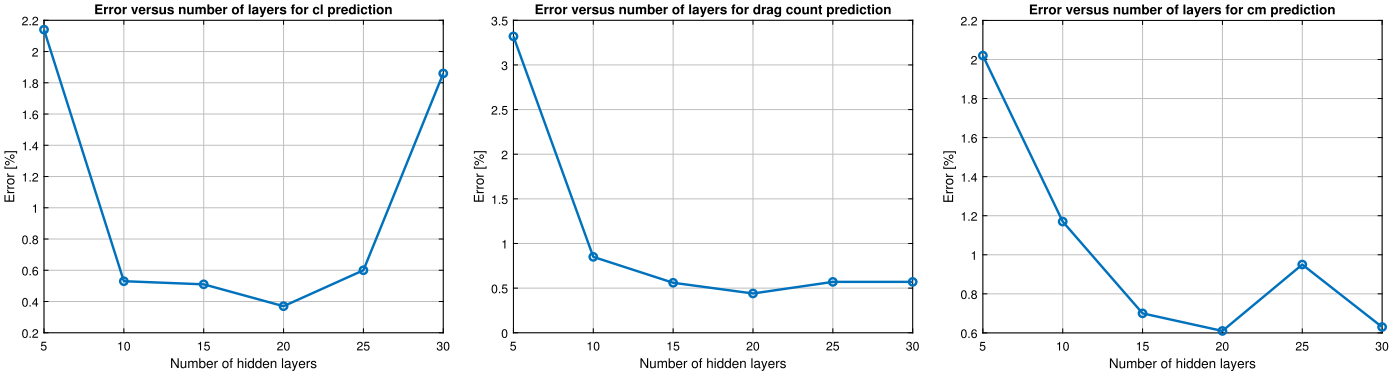
where x_j is the j^{th} element of the input data vector X, Φ is an activation function, M_i and K_i are the number of neurons and number of output respectively for the i^{th} hidden layer, $M_{i+1} = K_i$ holds since the two neural networks are fully-connected, $z_m^{(i)}$ is the output value of neuron m of i^{th} hidden layer while $w_{mk}^{(i)}$ and $b_k^{(i)}$ are weight and biases respectively. The goal of hyperparameter tuning and learning is to minimize the absolute value of the difference between the approximation from the neural network and the given data. There are various types of activation functions such as tangent hyperbolic (tanh), sigmoids, and ReLu. The DNNs will be trained by using the training datasets represented in Table 1. Each shape is defined by 20 different FFD control points and the flight condition is described by the triple (Re, Mach, α). To determine the optimal architecture of the neural networks, the architectures with various numbers of hidden layers, neurons, and activation functions are tested and the error is computed using the test set. The training is performed using $1,250$ epochs with a batch size of $4,000$ data points. Comparing the accuracy of the outputs from the neural network with different activation functions, ReLu gives the most accurate predictions. Fig. 6 shows the error versus the number of layers.

Figs. 6-8, show that the best architecture for accurately predicting lift, drag and moment coefficients is the neural network with 20 layers. This is explained by the fact that, during the training, both model loss and validation loss are low and the mean and variance prediction errors using the test set are also low.

Table 2 describes the architecture of the two developed neural networks and displays the high accuracy to predict the values of c_l and c_d. For both aerodynamic coefficients, the error is no larger than 0.7%.

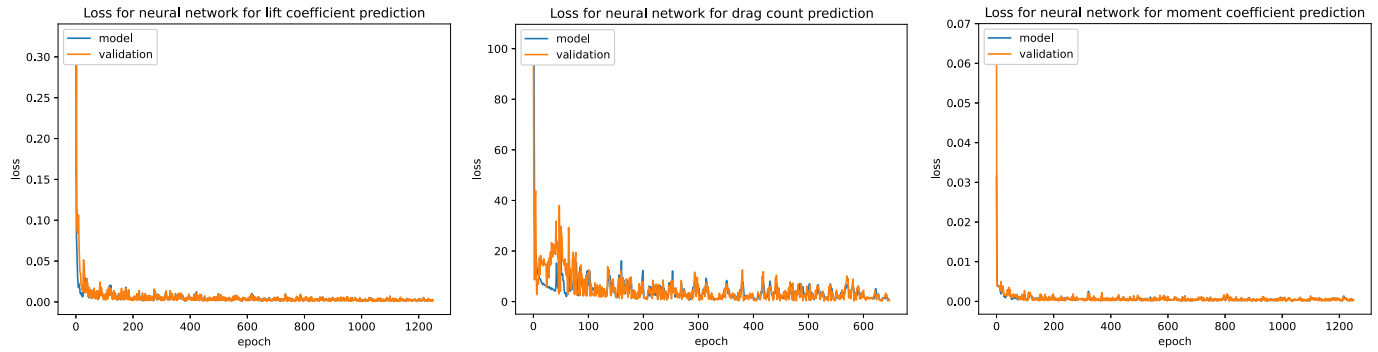
Table 2
DNN architecture.

Layer	DNN (c_l)	DNN (c_d)	DNN (C_M)
Input	1x23	1x23	1x23
Hidden	10x200, 10x150	10x200, 10x150	10x200, 10x150
Output	1x1	1x1	1x1
Error	0.37%	0.69%	0.61%



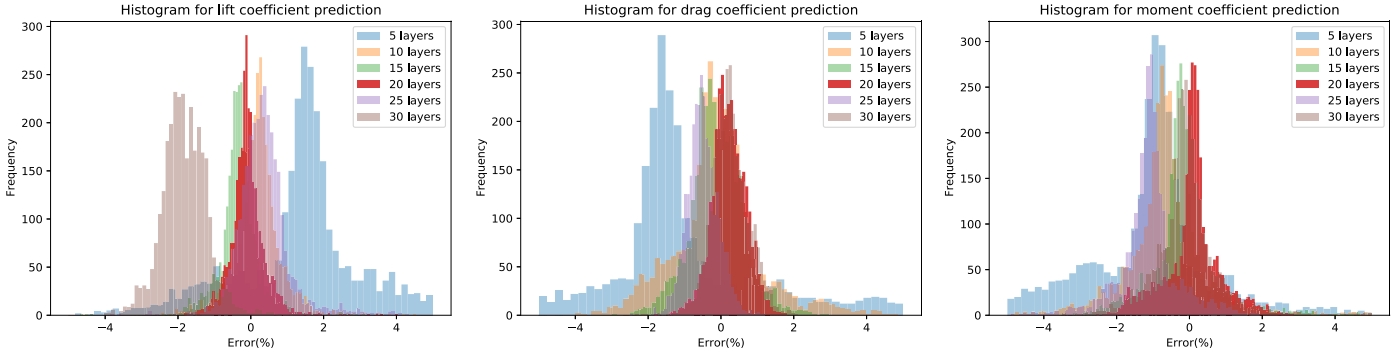
(a) Lift coefficient prediction error. (b) Drag count prediction error. (c) Moment coefficient prediction error.

Fig. 6. Mean absolute percentage error versus the number of hidden layers for neural network prediction.



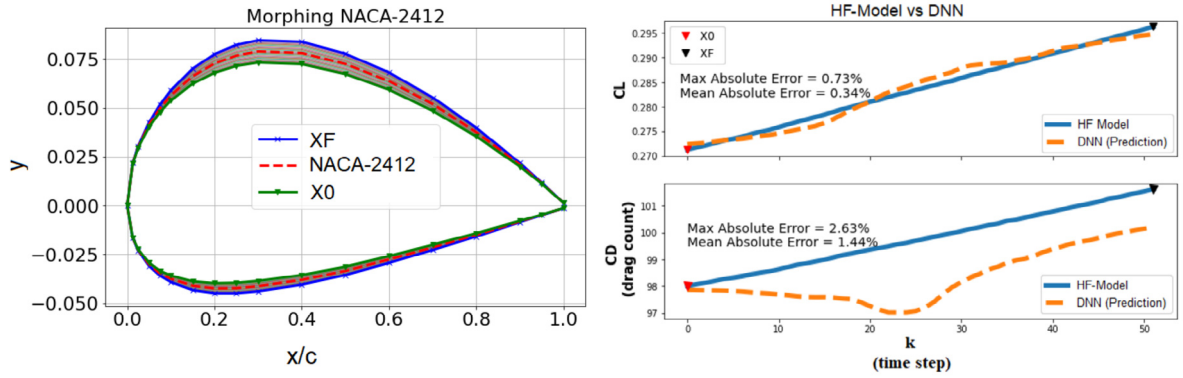
(a) Lift coefficient prediction loss. (b) Drag count prediction loss. (c) Moment coefficient prediction loss.

Fig. 7. Loss for neural network for 20 hidden layers during training.



(a) Lift coefficient. (b) Drag count. (c) Moment coefficient.

Fig. 8. Histograms for the prediction error.



(a) The evolution of 50 intermediate shapes. (b) Values of c_l and c_d.

Fig. 9. Morphing from X_0 to X_F.

3.5.1. Shape transition

The DNN developed in this section will be used to predict the values of c_l and c_d during the shape transition between initial ($X_0 = x[0]$) and final ($X_F = x[L]$) shapes in the morphing process. The prediction might be accurate enough to guarantee that it is safe to morph to a new shape and the new shape provides better values for c_l and c_d. There are infinite intermediate shapes between X_0 and X_F, we consider fifty intermediate shapes obtained by performing small deformations over the original shape. From Assumption 2, small deformations on the shape represent small variation on the values of c_l and c_d.

Remark 1. Regarding the assumption of small deformations on the shape, we can relax the geometric constraint to achieve expressive drag coefficient reductions and increase accordingly the number of intermediate shapes in order to guarantee a pace with small shape deformation for every iteration. □

The maximum deformation of the airfoil takes place when we directly move between the lower and upper bounds. Fig. 9 presents the value of the aerodynamic coefficients for the fifty intermediate shapes during the shape transition in the morphing process with maximum deformation for the high-order result (CFD analysis) and the one predicted by the DNN's. Fig. 9 shows that the prediction of c_l is accurate with mean absolute error less than 0.35% and the prediction of c_d is biased with maximum absolute error of 2.63%.

3.6. Shape optimization

With the developed surrogate model, 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 SA algorithm. SA is a meta-heuristic approach for approximating a global optimal solution of a nonlinear optimization problem, which is (2) in this case. The algorithm is summarized in Algorithm 1.

Algorithm 2 Online learning.

```
1: procedure SA(T_0, x_0, f, α)
2:     L ← f(x_0), x ← x_0, T ← T_0
3:     loop
4:         loop
5:             x_new ← N(x), L_new ← f(x_new)
6:             if L_new < L then
7:                 x ← x_new
8:             else
9:                 u ~ U(0, 1)
10:                if u ⩽ e^((L_new − L)/T) then
11:                    x ← x_new
12:                end if
13:            end if
14:        end loop
15:        T ← αT
16:    end loop
17: end procedure
```

At Line 1, Algorithm 1 takes an initial artificial temperature (T_0), initial states (x_0), a nonlinear function that will be optimized, and a cooling factor α as inputs. Line 2 performs an initialization of the objective value, states, and the temperature. The first loop (Line 3) iterates over the temperature value which is cooled at Line 12. The second loop (Line 4) performs an optimization for a given value of the temperature. Line 5 randomly picks a new state within the neighborhood of the previous one and evaluates the new objective value. The state always gets updated if the new objective value is better (Lines $6-7$); otherwise, the state gets updated based on a probability function (Lines $8-11$).

3.6.1. Numerical example

The final shape is defined by the SA algorithm that uses the surrogate model to identify, in offline way, candidates for final shapes for a given flight condition. These candidates are shapes with $c_l \geqslant C_L[0]$ and $c_d \leqslant C_d[0]$, where $C_L[0]$ and $C_D[0]$ are the values of c_l and c_d of the shape x at $k = 0$. Since the algorithm is running offline, the framework avoids unnecessary shape morphing. It is up to decision maker to choose the final shape among the candidates. The decision maker may choose the shape with minimum value of c_d, the shape with minimum shape deformation or even a shape with a good balance between shape deformation and reduction of c_d. Fig. 10 presents three candidates for final shape, with minimum value of c_d, obtained by the SA algorithm for the flight condition $\text{Re} = 2.5 \times 10^6$, $\text{Mach} = 0.175$, and $\alpha = 0.5°$.

4. Data-enabled predictive shape

In this section we deal with Problem 2. Although, the optimizer used to solve Problem 1 may provide the desired final shape, it only relies on the trained model, which does not consider real-time feedback data that may exist during the morphing period from the initial shape to the final one. To this end, the data-enabled predictive shape must be applied to solve Problem 2 and improve the system performance. The remaining of this section provides a brief mathematical background of this approach based on behavioral system theory [38,39].

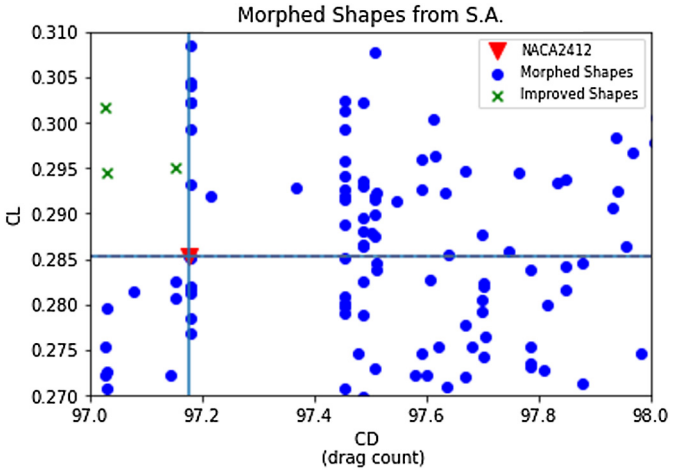


Fig. 10. Three candidates for the final shape.

Definition 1. Let $T, L \in \mathbb{Z}_{\geqslant 0}$ such that $T \geqslant mL$, then a sequence $\{u_k\}_{k=1}^{T}$ where $u_k \in \mathbb{R}^m$ is persistently exciting of order L if the Hankel matrix

$$\mathcal{H}_L(u) = \begin{bmatrix} u_1 & u_2 & \dots & u_{N-L} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{L-1} & \dots & u_T \end{bmatrix}$$

is full row rank, i.e., $\text{rank}(\mathcal{H}_L(u)) = mL$. □

Definition 2. An input-output sequence $\{u_k, y_k\}_{k=0}^{N}$ is a trajectory of an LTI system G if there exists an initial condition $\bar{x} \in \mathbb{R}^n$ with a state sequence $\{x_k\}_{k=0}^{N}$ such that

$$G = \begin{cases} x_{k+1} = Ax_k + Bu_k, \\ y_k = Cx_k + Du_k, \\ x_0 = \bar{x},\ k \in \{0, \dots, N\} \end{cases} \tag{5}$$

where $x \in \mathbb{R}^n$ are the states, $u \in \mathbb{R}^m$ are the control inputs, $y \in \mathbb{R}^p$ are the outputs, and system matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. Define with the tuple (A, B, C, D) as the minimal realization of G. □

The following theorem is fundamental for developing a data-enabled predictive shape control.

Theorem 1. *Suppose that $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ is a trajectory of an LTI system G where u^d is persistently exciting of order $L + n$. Then, $\{u_k, y_k\}_{k=1}^{L}$ is a trajectory of G if and only if there exists $g \in \mathbb{R}^{N-L+1}$ such that*

$$\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} u_{1:L} \\ y_{1:L} \end{bmatrix}.$$

Proof. The proof follows from [40,41]. □

According to [38], Theorem 1 provides the sufficient condition to capture all system trajectories based on a known trajectory and initial state. The full proof of Theorem 1 is omitted here for brevity.

Define the page matrix of depth L as,

$$\mathcal{P}_L(u_T) = \begin{bmatrix} u_1 & u_{L+1} & \dots & u_{T-L+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_L & u_{2L} & \dots & u_T \end{bmatrix},$$

which after being generated by a sufficiently rich input sequence, this matrix can be used to recover the whole space of trajectories

by replacing the Hankel matrix in Theorem 1. The page matrix has no repeated entries, and this fact has important implications when the entries of the matrix are corrupted. Moreover, it is observed that certain robustness and optimality guarantees become tight for Page matrices [42,43].

From the unknown system (5), we collect T sequences of input/output data of length $L = T_{\text{ini}} + N$, where T_{ini} denotes the length of the initialization sequence and N denotes the length of the prediction horizon. Define the following matrices,

$$U_L := \begin{bmatrix} U_{T_{\text{ini}}} \\ U_N \end{bmatrix} = [u_L^1, u_L^2, \dots, u_L^T], \qquad Y_L := \begin{bmatrix} Y_{T_{\text{ini}}} \\ Y_N \end{bmatrix} = [y_L^1, y_L^2, \dots, y_L^T]. \tag{6}$$

Lemma 1. *Let the augmented matrix $[X_0^{\mathrm{T}}, U_L^{\mathrm{T}}]^{\mathrm{T}}$ and $X_0 := [x_0^1, x_0^2, \dots, x_0^T]$ be full row rank matrices and let (6) be the collected input/output data of system (5). Any sequence $y_{\text{ini}} = [y_1^{\mathrm{T}}, \dots, y_{T_{\text{ini}}}^{\mathrm{T}}]^{\mathrm{T}}$, $y_N = [y_{T_{\text{ini}}+1}^{\mathrm{T}}, \dots, y_{T_{\text{ini}}+N}^{\mathrm{T}}]^{\mathrm{T}}$, $u_{\text{ini}} = [u_1^{\mathrm{T}}, \dots, u_{T_{\text{ini}}}^{\mathrm{T}}]^{\mathrm{T}}$, $u_N = [u_{T_{\text{ini}}+1}^{\mathrm{T}}, \dots, u_{T_{\text{ini}}+N}^{\mathrm{T}}]^{\mathrm{T}}$ is a trajectory of system (5) if and only if, there exists $g \in \mathbb{R}^T$, such that:*

$$\begin{bmatrix} U_{T_{\text{ini}}} \\ Y_{T_{\text{ini}}} \\ U_N \\ Y_N \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_N \\ y_N \end{bmatrix}. \tag{7}$$

Proof. The proof follows from [44]. □

To solve Problem 2, we propose to use a data-enabled predictive shape framework with appropriate regularization and bounded constraints. Using the surrogate model, we generate T trajectories of length $L = T_{\text{ini}} + N$ for the unknown system to obtain the data matrices $U_L = [U_{T_{\text{ini}}}^{\mathrm{T}}, U_N^{\mathrm{T}}]^{\mathrm{T}}$ and $Y_L = [Y_{T_{\text{ini}}}^{\mathrm{T}}, Y_N^{\mathrm{T}}]^{\mathrm{T}}$. For a given flight condition, these matrices remain unchanged.

For a trajectory of length L to be predicted, we shall use the following notation, $y_{\text{ini}} = [y^{\mathrm{T}}[1], \dots, y^{\mathrm{T}}[T_{\text{ini}}]]^{\mathrm{T}}$, $y_N = [y^{\mathrm{T}}[T_{\text{ini}} + 1], \dots, y^{\mathrm{T}}[T_{\text{ini}} + N]]^{\mathrm{T}}$, $u_{\text{ini}} = [u^{\mathrm{T}}[1], \dots, u^{\mathrm{T}}[T_{\text{ini}}]]^{\mathrm{T}}$, $u_N = [u^{\mathrm{T}}[T_{\text{ini}} + 1], \dots, u^{\mathrm{T}}[T_{\text{ini}} + N]]^{\mathrm{T}}$, and T_{ini} in this sequence refers to the length of data collected online that will be used to predict c_l and c_d over the prediction horizon N.

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 the notation used in this work, we have state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, output $y \in \mathbb{R}^p$, $U_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}m \times T}$, $U_N \in \mathbb{R}^{Nm \times T}$, $Y_{T_{\text{ini}}} \in \mathbb{R}^{T_{\text{ini}}p \times T}$ and $Y_N \in \mathbb{R}^{Np \times T}$.

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

$$\begin{aligned} \underset{g,u,y,\sigma_y}{\text{minimize}} \quad & \sum_{k=1}^{N} \big(\mathcal{C}(u[T_{\text{ini}}+k], y[T_{\text{ini}}+k]) + \mathcal{W}(g, \sigma_y)\big) \\ \text{s.t.} \quad & \begin{bmatrix} U_{T_{\text{ini}}} \\ Y_{T_{\text{ini}}} \\ U_N \\ Y_N \end{bmatrix} g = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u_N \\ y_N \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_y \\ 0 \\ 0 \end{bmatrix} \\ & B_L[k] \leqslant y[k] \leqslant B_U[k], \ \ \forall k \in \{1, \dots, N\} \\ & u[T_{\text{ini}}+k] \in \mathcal{U}, \ \ \forall k \in \{1, \dots, N\} \\ & y[T_{\text{ini}}+k] \in \mathcal{Y}, \ \ \forall k \in \{1, \dots, N\} \end{aligned} \tag{8}$$

where, $\mathcal{C}(u[.], y[.]) = \|y[.] - y_r[.]\|_Q^2 + \|u[.] - u_r[.]\|_R^2$. $R = \lambda_u \mathbb{I}_{20}$, $Q = \begin{bmatrix} \lambda_{c_l} & 0 \\ 0 & \lambda_{c_d} \end{bmatrix}$, $\lambda_u \gg \lambda_{c_l} > \lambda_{c_d} > 0$. $\mathcal{W}(g, \sigma_y) = \lambda_g \|g\|_2^2 + \lambda_{\sigma_y} \|\sigma_y\|_2^2$. $\lambda_g, \lambda_{\sigma_y} > 0$, B_U and B_L are upper and lower constraints used to guarantee small variations on output according to Assumption 2. These constraints are based on online gathered data y_{ini} and on predicted values given by the surrogate model. Let $\Delta_{\max} = [\Delta_{c_l}, \Delta_{c_d}]^{\mathrm{T}}$ be the maximum absolute variation on measured (c_l, c_d) for two consecutive intermediate shapes, C_{U_k} and C_{L_k} are defined as follows,

$$B_U[k] = y[T_{\text{ini}}] + \begin{bmatrix} k\lambda_\epsilon^{c_l} & 0 \\ 0 & k\lambda_\epsilon^{c_d} \end{bmatrix} \begin{bmatrix} \Delta_{c_l} \\ \Delta_{c_d} \end{bmatrix},$$

$$B_L[k] = y[T_{\text{ini}}] - \begin{bmatrix} k\lambda_\epsilon^{c_l} & 0 \\ 0 & k\lambda_\epsilon^{c_d} \end{bmatrix} \begin{bmatrix} \Delta_{c_l} \\ \Delta_{c_d} \end{bmatrix}$$

where $y[T_{\text{ini}}]$ is the most recent measurement of (c_l, c_d) and $\lambda_\epsilon^{c_l}, \lambda_\epsilon^{c_d} \in (0, 1]$ are correction factors based on error of prediction made by the surrogate model for the online gathered data.

As the data matrix is corrupted by the error inherent to surrogate model, the regularization parameters λ_g and λ_y are necessary.

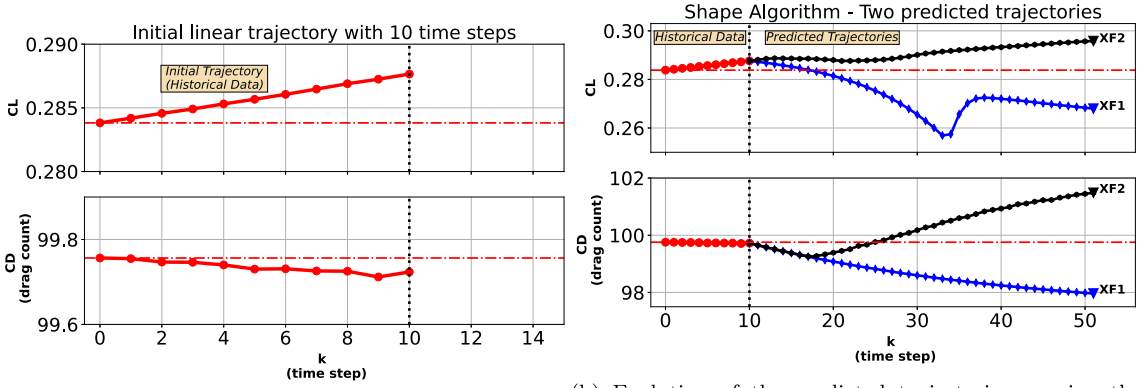
Remark 2. The data-enabled predictive shape algorithm uses a matrix of trajectories for prediction which might be computed for every new flight condition. The size of this matrix depends on the horizon of prediction. □

Numerical example

The data-enabled predictive shape algorithm uses the Hankel matrix and online measurements of c_l and c_d (historical data) to predict the values of these parameters for intermediate shapes during the morphing process. In this work we replaced the Hankel matrix by the data matrix $[U_L^{\mathrm{T}}, Y_L^{\mathrm{T}}]^{\mathrm{T}}$, a matrix compounded by trajectories that are sequence of input/output (u_k, y_k) as described in previous section. These trajectories are generated by using the surrogate model developed in Section 3.5.

The use of data-enabled predictive shape algorithm aims to mitigate the error on prediction inherent to the surrogate model developed in section 3.5 and shown in Fig. 9b. For a numerical example using, we implement the algorithm to predict online two trajectories of the system (1) when morphing from the X_0:Baseline to X_{F1} and to X_{F2}, X_{F1} is the final shape $x[L]$ of trajectory 1, while X_{F2} is the final shape $x[L]$ of trajectory 2. First, we morph the airfoil over a random linear trajectory and then collect the data for the first 10 time steps, the data gathered online is used to predict the c_l and c_d trajectories toward X_{F1} and toward X_{F2}.

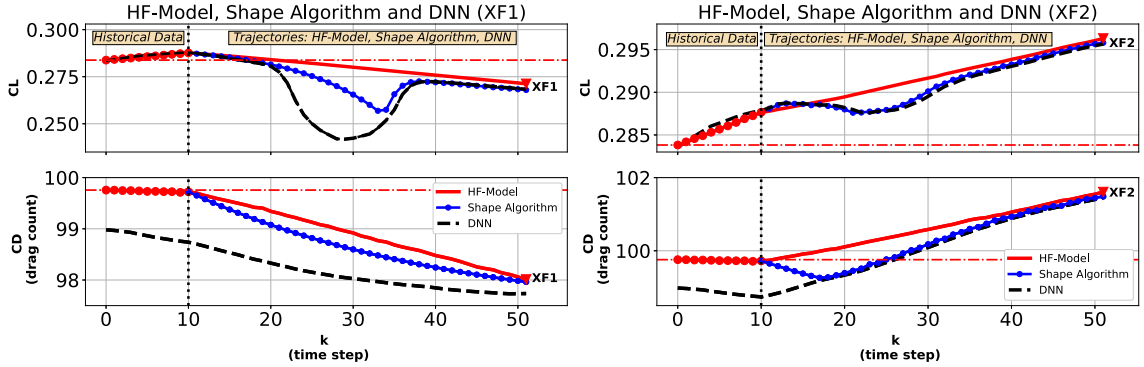
Fig. 11a presents the random initial trajectory used to make predictions of c_l and c_d. The $c_l[k]$ and $c_d[k]$ represent respectively the value of c_l and c_d for the shape x at time k, for this initial trajectory we have $c_l[0] = 0.2838$ and $c_l[10] = 0.2876$, a 1.25% c_l increase; and $c_d[0] = 99.7564$ and $c_d[10] = 99.7235$, a 0.06% c_d reduction. Ideally, as we are interested in collecting online data to be able to run the algorithm, this initial trajectory must produce even smaller variation on c_l and c_d to reduce the effects of morphing on the system under analysis. Fig. 11b presents predictions of c_l and c_d over the trajectories from X_0 toward X_{F1} and toward X_{F2} using the aforementioned initial trajectory. We are also interested to evaluate how accurate are these predicted values in comparison to the values given by high-order flow solver (CFD), the High-Fidelity Model (HF-Model). Fig. 12 shows the values of c_l and c_d predicted by the data-enabled predictive shape algorithm and DNN in comparison with the values given by HF-Model. Table 3 presents the comparison between DNN and the data-enabled predictive shape algorithm, showing that the data-enabled predictive shape algorithm has lower error of prediction on c_d than one made by the surrogate model for both trajectories.



(a) Evolution of the initial trajectory.

(b) Evolution of the predicted trajectories running the data-enabled predictive shape algorithm.

Fig. 11. Data-enabled predictive shape algorithm.



(a) Evolution of the HF-Model trajectory and predicted trajectories.

(b) Evolution of the HF-Model trajectory and predicted trajectories.

Fig. 12. Comparison of the three trajectories.

Table 3
Prediction error on c_d.

	XF1: $X[11]$ to $X[51]$		XF2: $X[11]$ to $X[51]$	
	Max Abs error	Mean Abs error	Max Abs error	Mean Abs error
DNN	1.04%	0.79%	0.96%	0.48%
Proposed Algorithm	0.33%	0.22%	0.76%	0.34%

Fig. 11b shows the prediction of c_l and c_d for two different trajectories of system (1). The trajectory is defined as a linear sequence of states starting at $X_0 = [x_1[0], \ldots, x_{20}[0]]$ and ending at $X_F = [x_1[L], \ldots, x_{20}[L]]$ with $L = 51$ for all trials running the data-enabled predictive shape algorithm. Every state at time k defines a shape which is associated with a value of c_l and a value of c_d as described by system (1). For convenience, the shapes between X_0 and X_F, i.e., $x[k]$ with $k = [1, 2, \ldots, 50]$ are called intermediate shapes. Fig. 13a shows the evolution of state X for the first 10 time steps of the initial linear trajectory while Fig. 13b shows the main shapes reached by the system for the two trajectories described in this numerical example.

5. Online learning algorithm

Now having a way to obtain candidates for final shape that provide better values for c_l and c_d, we proceed to design an online data-based algorithm to efficiently morph the airfoil from initial shape X_0 to a final shape X_F preventing the system from reaching the "unsafe" values of c_l and c_d.

The proposed algorithm is compound by two main parts well defined: The offline procedure, before start to morph the shape, we run the search algorithm (SA) to determine the final shape to be reached by the system; The online procedure, we start to morph the airfoil using small steps; for every m steps, the online algorithm evaluates the benefits to keep morphing. The learning algorithm can be described by the following procedure.

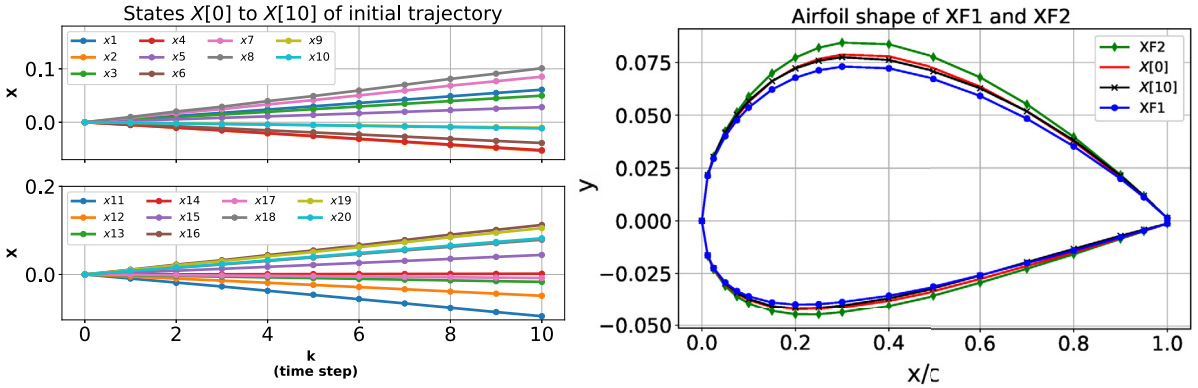
5.1. Numerical results

To show the efficiency of the proposed framework, we use the algorithm in three different flight conditions. We consider $m = 10$ which means that for every 10 steps of morphing toward the final shape, we run the data-driven controller to predict the c_l and c_d values for the remaining steps and plan to keeping morphing.

Every prediction made in this procedure is called cycle. For $m = 10$ we run four cycles before reaching the final shape X_F starting from the initial shape X_0 (baseline). For all scenarios, the data-driven controller is able to reduce the prediction error inherent to the surrogate model described in Section 3.5. The following subsections describe the results of the proposed procedure for different scenarios.

5.1.1. Flight condition 1

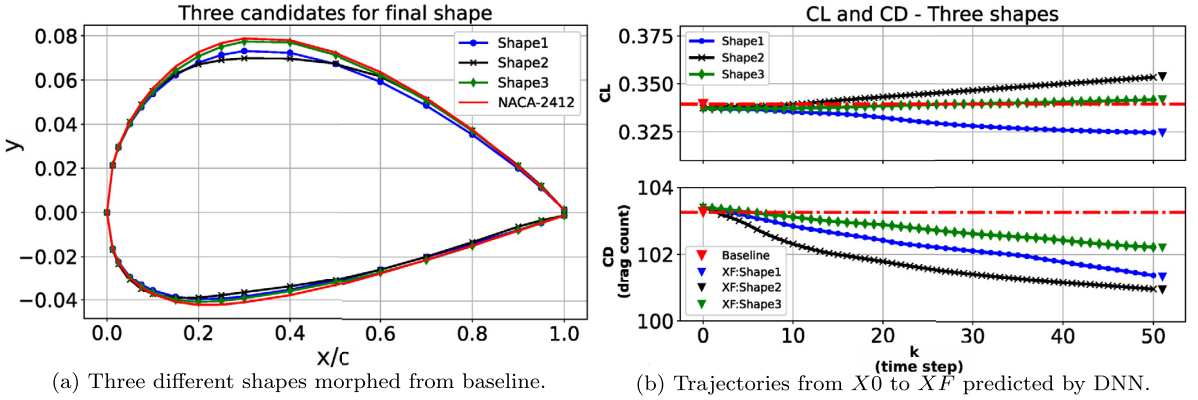
For this flight condition, we consider Re=2.0e6, Mach=0.2, and $\alpha = 1.0°$. The first steps of the algorithm are to identify candidates for final shape using the S.A algorithm. Fig. 14 presents three candidate shapes and their respectively trajectories from $X0$ to XF provided by DNN. A trajectory S is defined by 50 intermediate



(a) Evolution of state X over 10 time steps.

(b) Airfoil shapes.

Fig. 13. Airfoil shapes and evolution of the trajectories.



(a) Three different shapes morphed from baseline.

(b) Trajectories from $X0$ to XF predicted by DNN.

Fig. 14. Three candidates for the final shape.

Procedure 1 Online learning algorithm.

Input: Flight condition: (Re, Mach, α).
Output: $x_c^\star$, shape with improved c_d and c_l values.

procedure
  Run SA for the given flight condition using the developed surrogate model.
  Select the best N candidates for final shape.
  Choose one shape from the candidates and generate the reference trajectory (u_r^1, y_r^1).
  Apply u_r to morph the airfoil by T_{ini} steps toward the chosen shape.
  Update u_{ini} and y_{ini} to the T_{ini} most recent input/output measurements.
  Based on data gathered online, adjust B_L and B_U in (8).
  Considering (u_r^1, y_r^1), solve (8) for $g^\star$. Compute $y_N^1 = Y_N g^\star$ and $u_N^1 = U_N g^\star$
  At state $x[T_{\text{ini}}]$, generate the reference trajectories (u_r^i, y_r^i), $i = \{2, \ldots, N\}$, toward candidate shape i defined at step 3.
  Considering (u_r^i, y_r^i), solve (8) for $g^\star$,
  Compute $y_N^i = Y_N g^\star$ and $u_N^i = U_N g^\star$, $\forall i = \{2, \ldots, N\}$.
  Evaluate the predicted values for c_l and c_d given by y_N^i, $i = \{1, \ldots, N\}$, and choose the optimal trajectory $(u_N^\star, y_N^\star)$.
  Morph m steps following the optimal trajectory $(u_N^\star, y_N^\star)$ defined at step 12.
  while *Not_at_final_shape* and *keep_morphing* **do**
    Repeat step 6 and 7.
    Update (u_r, y_r) based on $(u_N^\star, y_N^\star)$.
    Considering (u_r, y_r), solve (8) for $g^\star$. Compute $y_N = Y_N g^\star$ and $u_N = U_N g^\star$
    Evaluate the predicted values for c_l and c_d given by y_N. Make a decision about keep morphing.
  end while
end procedure

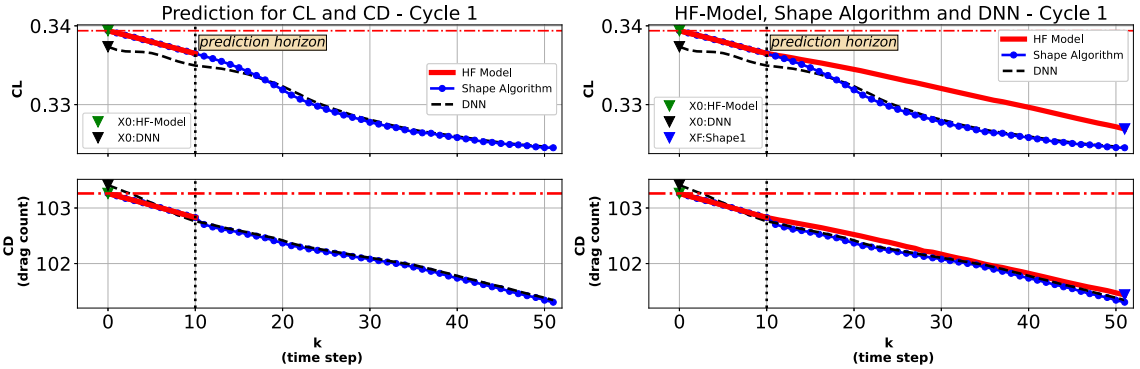
shapes from $X0$ to XF, i.e., $S = [X_0, X_1, X_2, ..., X_{49}, X_{50}, X_F]$ where $X_k = [x_1[k], x_2[k], \ldots, x_{20}[k]]$.

Fig. 14b presents the prediction made by the surrogate model for all three shapes and shows that Shape1 is not a good candidate for final as it reaches an "unsafe" value of c_l. To evaluate the efficiency of the online algorithm, we choose Shape1 to start the search for improved shape. We want to verify that after m steps of morphing, the algorithm can find a new trajectory.

In Cycle1, we morph m steps toward chosen shape and predict the values for c_l and c_d for next steps, the decision on keeping morphing is based on predicted values. Fig. 15a shows the prediction for c_l and c_d and based on this prediction, we can see that Shape1 is not a good choice. In order to evaluate the efficiency of the proposed framework in Cycle1, Fig. 15b shows the three trajectories toward the Shape1; the continuous line represents the high-fidelity system, the system that we intend to track; the dashed line represents the prediction made by the surrogate model based on DNN described in Section 3.5; and the dotted line represents the prediction made by the data-driven controller based on the data-enabled predictive shape algorithm.

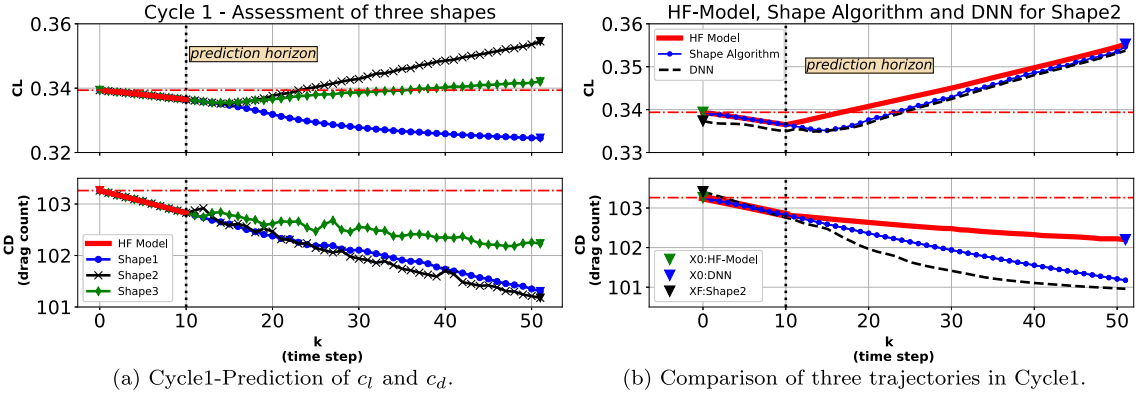
Fig. 15a shows the trajectory toward Shape1 is not appropriate and reaches "unsafe" values of c_l. For Cycle1, the online algorithm decides to stop morphing and evaluates other candidates for final shape by running a local search using the data gathered online. Fig. 16a shows the assessment of two other shapes where the Shape2 is chosen to be the new morphed shape.

During Cycle1, the Shape2 is chosen, and a new trajectory is predicted. Fig. 16b shows the predictions of c_l and c_d over the



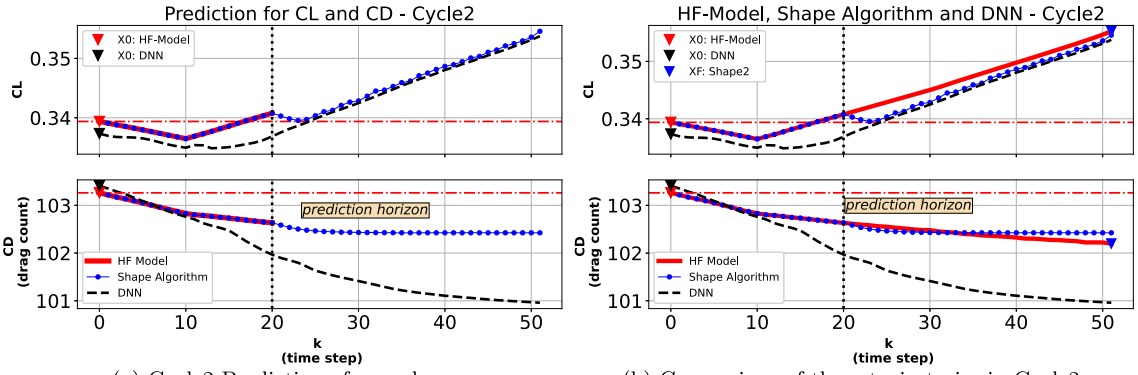
(a) Cycle1 - Prediction of c_l and c_d. (b) Comparison of three trajectories in Cycle1.

Fig. 15. Assessment of Cycle1.



(a) Cycle1-Prediction of c_l and c_d. (b) Comparison of three trajectories in Cycle1.

Fig. 16. Assessment of Cycle1.



(a) Cycle2-Prediction of c_l and c_d. (b) Comparison of three trajectories in Cycle2.

Fig. 17. Assessment of Cycle2.

trajectory toward Shape2 and the real values of c_l and c_d of the system represented by the HF-Model.

During Cycle2, we change the initial trajectory by morphing m steps towards Shape2 and by running the data-driven controller, we predict the values of c_l and c_d for next steps of morphing toward Shape2. Fig. 17 presents the result of the algorithm in Cycle2.

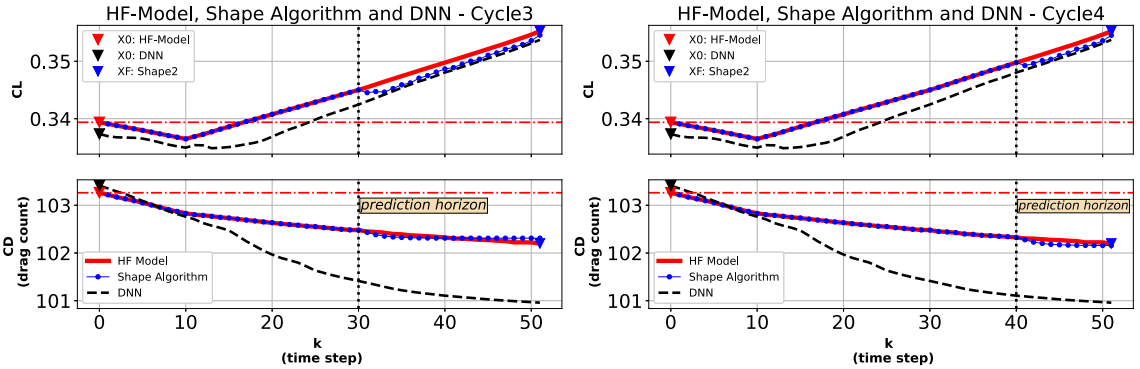
Figs. 15, 17, and 18 show that the data-driven controller, is more accurate when tracking the high-fidelity model c_d values than the one using the surrogate model. Table 4 presents the error on prediction of next m morphing steps toward Shape2 for the DNN and the data-enabled predictive shape algorithm. For all the cycles, the prediction made by the data-driven controller is more accurate than the one using the surrogate model.

In this example, we deal with a situation where the chosen trajectory does not provide the desired aerodynamics performance.

Table 4
MAE on prediction of m steps of morphing for flight condition 1.

	Prediction error on c_d			
	$X_{11} - X_{20}$	$X_{21} - X_{30}$	$X_{31} - X_{40}$	$X_{41} - X_{50}$
Cycle 1	0.117%	0.391%	0.6187%	0.857%
Cycle 2	–	0.047%	0.057%	0.169%
Cycle 3	–	–	0.05%	0.06%
Cycle 4	–	–	–	0.06%
DNN	0.343%	0.882%	1.137%	1.208%

The online algorithm was able to correct the trajectory and predict a new path to reduce c_d without compromising the value of c_l. The final shape reached by the solution provides a 1.04% reduction of c_d and a 4.67% increase of c_l, as shown in Table 5. In



(a) Cycle3-Prediction of c_l and c_d. (b) Cycle4-Prediction of c_l and c_d.

Fig. 18. Assessment of Cycle3 ans Cycle4.

Table 5
Final results after four cycles of morphing.

	c_l	c_d
Initial shape (X_0)	0.339	103.26
Final shape (X_F)	0.355	102.19
Variation	+4.67%	−1.04%

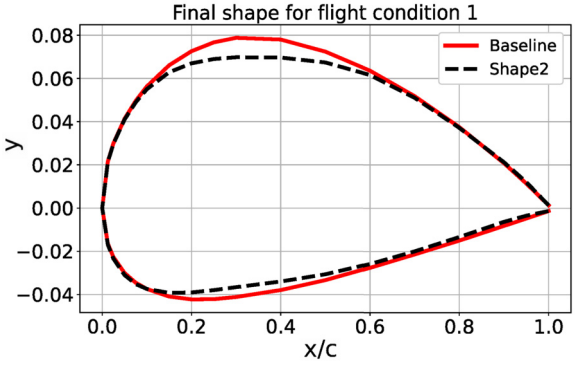


Fig. 19. Final Shape vs initial shape for flight condition 1.

order to achieve this c_d reduction, we relaxed the geometric constraints to allow shapes beyond the boundaries presented in Fig. 2. The final airfoil shape for flight condition 1 is shown in Fig. 19.

5.1.2. Flight condition 2

The prediction made by the data-driven controller is given by solving the equation (8). The constraints L_k and M_k in equation (8) are based on online gathering data and on performance of the surrogate model. These constraints guarantee that small variation on shape results on small variation on c_l and c_d.

In this second numerical example, we intend to evaluate the performance under an inaccurate surrogate model. For this example, we consider Re = 1.25×10^6, Mach = 0.19, and $\alpha = 0.5°$. Fig. 20 shows the trajectory predicted by the surrogate model for the chosen candidate for final shape, the Shape1. According to this prediction, the value of c_d reduces from 110.47 drag count to 108.01 drag count, a 2.23% reduction of c_d.

Fig. 20 shows that for the baseline, the prediction error for c_d using the surrogate model is greater than 1.0 drag count. We use the online algorithm to morph the airfoil toward Shape1 in an efficient way. The first step of the solution is to move $m = 10$ steps towards Shape1. Fig. 21 presents the prediction of c_l and c_d in Cycle1.

Fig. 21 presents the result of the solution in Cycle1 to predict the trajectory for c_l and c_d toward Shape1. Since the predicted trajectory shows that Shape1 does not have a better aerodynamics performance than the baseline airfoil, the airfoil stops morphing.

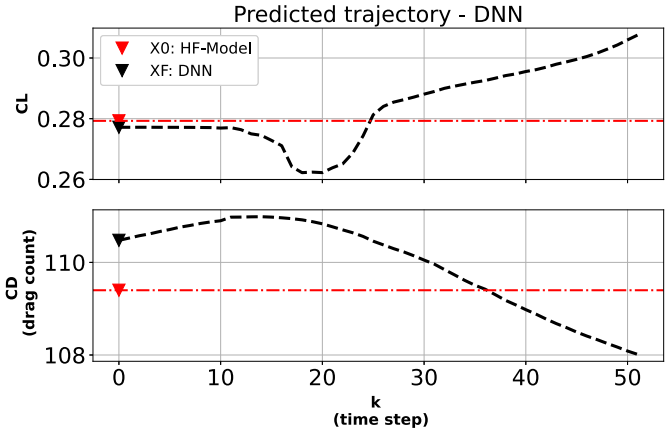


Fig. 20. Predicted Trajectory using DNN for Shape1.

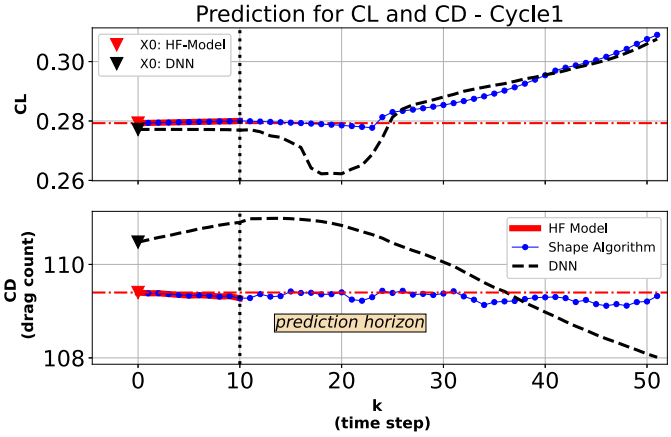


Fig. 21. Assessment of Shape1 in Cycle1 for flight condition 2.

Fig. 22 shows the evolution of the values of c_l and c_d towards Shape1 and provides a comparison to the predicted trajectories. The Shape1 indeed increases the value of c_d instead of reducing it. If we rely only on the prediction made by the surrogate model, the system may reach "unsafe" values of c_l or c_d. The data-driven controller prevents this situation.

To compute the accuracy, Table 6 shows the prediction error using the surrogate model and the framework that combines the surrogate model and the data-driven technique. For the first m

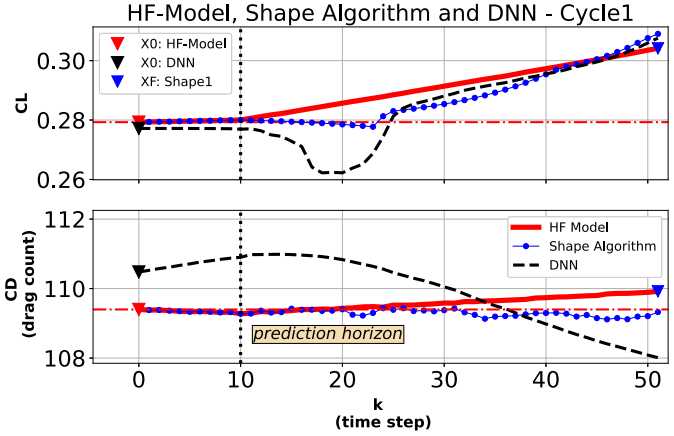


Fig. 22. Assessment of Shape1 for Cycle1.

Table 6
MAE on prediction of m morphing steps for flight condition 2.

	Prediction error on c_d			
	$X_{11} - X_{20}$	$X_{21} - X_{30}$	$X_{31} - X_{40}$	$X_{41} - X_{50}$
Cycle 1	0.024%	0.15%	0.368%	0.571%
DNN	1.44%	0.832%	0.311%	1.234%

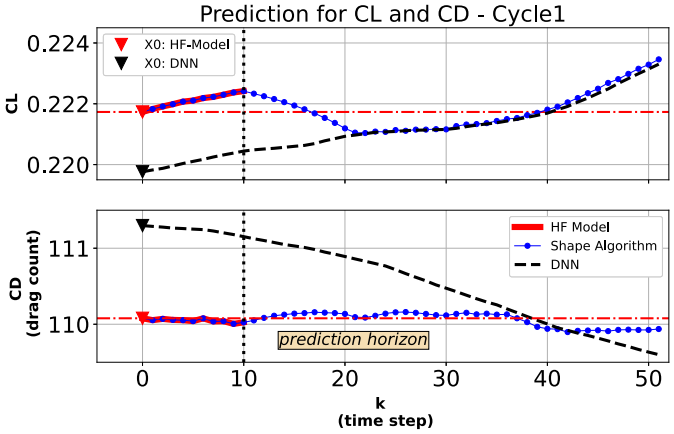


Fig. 23. Assessment of Shape1 for Cycle1.

morphing steps, the prediction error for the proposed framework is 60 times less than the one using the surrogate model.

5.1.3. Flight condition 3

In this third numerical example we evaluate the performance when the surrogate model is not accurate to provide the trajectory of c_l. We considered Re = 1.25×10^6, Mach = 0.12, and $\alpha = 0°$. Fig. 23 shows the trajectory towards the final shape predicted.

To compute the accuracy in this third numerical example, the table presents the error on prediction using only the surrogate model and using the proposed framework. For the first m morphing steps, the prediction error is 60 times less than one using only the surrogate model.

Fig. 23 presents the result in Cycle1 to predict the trajectory for c_l and c_d toward Shape1. Since the predicted trajectory shows that the final shape does not have a better aerodynamics performance than the baseline airfoil.

Fig. 24 presents the real values of c_l and c_d in the trajectory toward Shape1 and compares it to the predicted trajectories. This final shape decreases the value of c_d in 0.15% that represents a reduction of 0.17 drag count while increase the value of c_l from 0.221 to 0.224, an increase of 1.27% in c_l. To compute the accuracy of Table 7 presents the error on the prediction using the proposed framework and the surrogate model. For the first m morphing steps, the prediction error using the proposed framework is 8.8 times less.

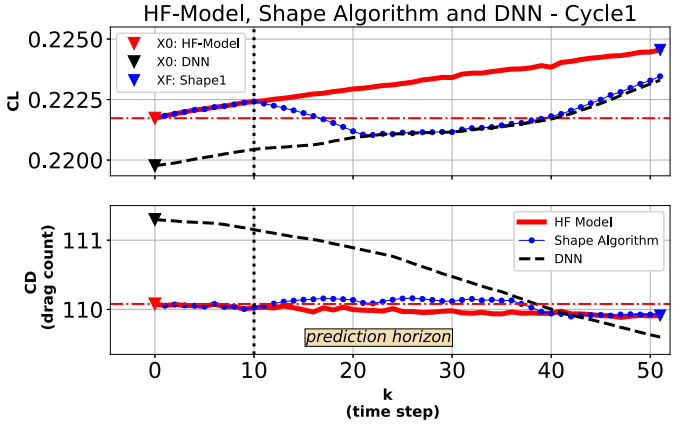


Fig. 24. Assessment of Shape1 for Cycle1.

Table 7
MAE on prediction of m morphing steps for flight Condition 3.

	Prediction error on c_d			
	$X_{11} - X_{20}$	$X_{21} - X_{30}$	$X_{31} - X_{40}$	$X_{41} - X_{50}$
Cycle 1	0.103%	0.137%	0.124%	0.02%
DNN	0.91%	0.64%	0.25%	0.11%

In this work, we developed an online solution that is more accurate than the surrogate model to predict the values of c_d in a trajectory with small deformation on airfoil shape. For all three presented scenarios, there is a significant reduction on the prediction error. Table 6 shows that the proposed framework for flight condition 2 is 60 times more accurate than DNN to predict c_d over the trajectory from X_{11} to X_{20} reducing the prediction error from 1.44% to 0.024%.

It is important to mention that the c_d reduction shown in the three numerical examples might be greater if we consider a higher deformation on airfoil shape and the use of a shape optimizer based on a gradient descent approach instead of the one based on SA. For all three scenarios, the trajectory from initial shape to final shape was defined as a linear sequence of states. For the present investigation, we focused on subtle airfoil shape deformations with the goal of ensuring that the trajectory between initial and final shapes is smooth.

6. Conclusion and future work

In this work, we developed an online, data-based framework for morphing airfoil. We use airfoil shape control points which can move vertically. The online solution is based on a data-driven controller combined with a surrogate model used to generate, in off-line manner, trajectories of the non-linear system representing the morphing airfoil. Without full knowledge of the aerodynamic parameters (lift and drag coefficients), the proposed learning framework searches for an airfoil shape that minimizes a metric of performance, associated to drag force and subject to geometric and lift constraints.

The solution uses data gathered online to improve the accuracy of c_d prediction provided by the surrogate model along the trajectory with small deformation on shape for every time step. The optimization framework focuses on subtle airfoil deformations to assure a smooth trajectory between initial and final shapes with no critical values of c_l in a morphing technology application.

The efficiency and robustness of the proposed solution was shown in three different scenarios, resulting in a significant reduction on prediction error. The proposed online learning algorithm successfully predicted, with high accuracy, the trajectory toward a shape that truly provided reduction on value of c_d without compromising c_l.

Future work includes the implementation of a shape optimizer based on the gradient descent approach which will obtain more expressive drag coefficient reductions; Finally, we plan to extend the current analysis to a 3D morphing wing and evaluate the effect 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 National Science Foundation S&AS-1849198 and by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, Brazil, under the Research Grant No. 99999.005340/2015-02.

References

[1] J. Sobieszczanski-Sobieski, R.T. Haftka, Multidisciplinary aerospace design optimization: survey of recent developments, Struct. Optim. 14 (1997) 1–23.
[2] C. Kassapoglou, Design and Analysis of Composite Structures: with Applications to Aerospace Structures, John Wiley & Sons, 2013.
[3] J.A. Samareh, Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization, AIAA J. 39 (2001) 877–884.
[4] A. Jameson, Optimum aerodynamic design using cfd and control theory, in: 12th Computational Fluid Dynamics Conference, 1995, p. 1729.
[5] P.E. Rubbert, Cfd and the changing world of airplane design, in: ICAS Proceedings, vol. 19, American Inst of Aeronautics and Astronautics, 1994, pp. LVII–LVII.
[6] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, S. Kaushik, Prediction of aerodynamic flow fields using convolutional neural networks, Comput. Mech. 64 (2019) 525–545.
[7] X. Du, P. He, J.R.R.A. Martins, Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling, Aerosp. Sci. Technol. 113 (2021) 106701.
[8] M.A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction, Struct. Multidiscip. Optim. 53 (2016) 935–952.
[9] M.A. Bouhlel, N. Bartoli, J. Morlier, A. Otsmane, An improved approach for estimating the hyperparameters of the kriging model for high-dimensional problems through the partial least squares method, Math. Probl. Eng. (2016) 6723410.
[10] M.A. Bouhlel, N. Bartoli, R.G. Regis, A. Otsmane, J. Morlier, Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method, Eng. Optim. 50 (2018) 2038–2053.
[11] M.A. Bouhlel, J.R.R.A. Martins, Gradient-enhanced kriging for high-dimensional problems, Eng. Comput. 1 (2019) 157–173.
[12] M.A. Bouhlel, J.T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J.R.R.A. Martins, A Python surrogate modeling framework with derivatives, Adv. Eng. Softw. 135 (2019) 102662.
[13] J. Li, M.A. Bouhlel, J.R.R.A. Martins, Data-based approach for fast airfoil analysis and optimization, AIAA J. 57 (2019) 581–596.
[14] M.A. Bouhlel, S. He, J.R.R.A. Martins, Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes, Struct. Multidiscip. Optim. 61 (2020) 1363–1376.
[15] J. Valasek, M. Tandale, J. Rong, A reinforcement learning - adaptive control architecture for morphing, J. Aerosp. Comput. Inf. Commun. 2 (2005) 174–195.
[16] M. Tandale, J. Rong, J. Valasek, Preliminary results of adaptive-reinforcement learning control for morphing aircraft, in: Guidance, Navigation and Control Conference and Exhibit, AIAA, 2004, pp. 3215–3225.
[17] A. Lampton, A. Niksch, J. Valasek, Morphing airfoils with four parameters, in: Guidance, Navigation and Control Conference and Exhibit, AIAA, 2008.
[18] A. Niksch, J. Valasek, T.W. Strganac, L.A. Carlson, Morphing aircraft dynamical model: longitudinal shape changes, in: Guidance, Navigation and Control Conference and Exhibit, AIAA, 2008.
[19] 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.
[20] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models, in: Proc. SIGGRAPH 86, 1986, pp. 151–160.
[21] 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. (2020).
[22] 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, 1981, AIAA 1981-1259.
[23] E. Turkel, V.N. Vatsa, Effects of artificial viscosity on three-dimensional flow solutions, AIAA J. 32 (1994) 39–45.
[24] 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.
[25] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[26] 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.
[27] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, Rech. Aérosp. 1 (1994) 5–21.
[28] 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.
[29] 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. (2022), https://arc.aiaa.org/doi/10.2514/1.J060481, in press.
[30] J.C. Vassberg, M.A. DeHaan, S.M. Rivers, R.A. Wahls, Development of a Common Research Model for Applied CFD Validation Studies, 2008.
[31] N. Garg, B.W. Pearce, P.A. Brandner, A.W. Phillips, J.R.R.A. Martins, Y.L. Young, Experimental investigation of a hydrofoil designed via hydrostructural optimization, J. Fluids Struct. 84 (2019) 243–262.
[32] T.W. Sederberg, S.R. Parry, Free-form deformation of solid geometric models, SIGGRAPH Comput. Graph. 20 (1986) 151–160.
[33] 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.
[34] J. Prochazkova, Free form deformation methods - the theory and practice, in: 16th Conference on Applied Mathematics, APLIMAT 2017 - Proceedings, 2017, pp. 1276–1282.
[35] J.A. Samareh, Aerodynamic Shape Optimization Based on Free-Form Deformation, 2004.
[36] J.N. Kutz, Linear stability-based smooth Reynolds-averaged Navier–Stokes transition model for aerodynamic flows, J. Fluid Mech. 814 (2017) 1–4.
[37] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, J. Fluid Mech. 11 (4) (1998) 761–767.
[38] J.W. Poldeman, J.C. Willems, Introduction to Mathematical Systems Theory: A Behavioral Approach, vol. 3, Springer, 1998.
[39] 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.
[40] J.C. Willems, P. Rapisard, I. Markovsky, A note on persistency of excitation, Syst. Control Lett. 44 (2005) 561–580.
[41] J. Berberich, J. Köhler, M. Müller, F. Allgöwer, Data-driven model predictive control with stability and robustness guarantees, IEEE Trans. Autom. Control 44 (2021) 561–580.
[42] J.L.J. Coulson, F. Dörfler, Distributionally robust chance constrained data-enabled predictive control, IEEE Trans. Autom. Control (2021).
[43] A.A.H. Damen, P.M.J. Van den Hof, A.K. Hajdasinski, Approximate realization based upon an alternative to the Hankel matrix: the page matrix, Syst. Control Lett. 2 (1982) 202–208.
[44] F. Fiedler, S. Lucia, On the relationship between data-enabled predictive control and subspace predictive control, in: Proc. European Control Conference, 2021.