On-Line Black-Box Aerodynamic Performance Optimization for a Morphing Wing With Distributed Sensing and Control

Mkhoyan, Tigran; Ruland, Oscar; Breuker, Roeland De; Wang, Xuerui

DOI
10.1109/TCST.2022.3210164

Publication date
2022

Document Version
Final published version

Published in
IEEE Transactions on Control Systems Technology

On-Line Black-Box Aerodynamic Performance Optimization for a Morphing Wing With Distributed Sensing and Control

Tigran Mkhoyan, *Member, IEEE*, Oscar Ruland, Roeland De Breuker, and Xuerui Wang, *Member, IEEE*

***Abstract*— Inspired by nature, smart morphing technologies enable the aircraft of tomorrow to sense their environment and adapt the shape of their wings in flight to minimize fuel consumption and emissions. A primary challenge on the road to this feature is how to use the knowledge gathered from sensory data to establish an optimal shape adaptively and continuously in flight. To address this challenge, this article proposes an online black-box aerodynamic performance optimization architecture for active morphing wings. The proposed method integrates a global online-learned radial basis function neural network (RBFNN) model with an evolutionary optimization strategy, which can find global optima without requiring in-flight local model excitation maneuvers. The actual wing shape is sensed via a computer vision system, while the optimized wing shape is realized via nonlinear adaptive control. The effectiveness of the optimization architecture was experimentally validated on an active trailing-edge (TE) camber morphing wing demonstrator with distributed sensing and control in an open jet wind tunnel. Compared with the unmorphed shape, a 7.8% drag reduction was realized, while achieving the required amount of lift. Further data-driven predictions have indicated that up to 19.8% of drag reduction is achievable and have provided insight into the trends in optimal wing shapes for a wide range of lift targets.**

***Index Terms*— Black-box optimization, evolutionary optimization, morphing, neural networks, vision-based control, wind tunnel experiment.**

I. Introduction

RECENT trends in aviation highlight the ever-increasing need for fuel economy and sustainability. Active morphing technology can offer significant benefits over conventional wing designs. Due to conflicting requirements [1], conventional wings are only optimized for a single flight condition (such as cruise). By contrast, the ability to morph wings into a desirable shape can allow aircraft to actively improve flight performance across the full flight envelope. While many challenges exist in morphing design, the key challenge to efficiently benefit from active morphing during in-flight operation is how to establish a shape optimization strategy that is adaptive and can find the global optimum.

The currently practiced method of "determining" the optimal wing shape is through the selection from an offline-determined and fixed lookup table. However, this method is only optimal for one particular reference aircraft design. Aircraft production variances and operating condition variations (including gross weight, airspeed, and altitude) can contribute to a wide range of uncertainty and thus suboptimality in the lookup table method [2]. By contrast, a promising method is online data-driven nonlinear optimization, which can tailor the wing shape adaptively and optimally to any specific flight condition, like birds do [3].

Online shape optimization strategies for active morphing wings do exist in the literature. In [4], a generative set search method was used to optimize the deflections of eight leading- and trailing-edge (TE) control surfaces at one fixed angle of attack (AOA) to reduce the drag on a wind tunnel model. This local black-box optimization strategy uses a linear lift coefficient model, the parameters of which have to be identified before optimization through sweeps of the control surfaces for the preselected AOA. The local scope of the linear lift model and the local search character of this black-box direct search method make this method prone to converge on local optima.

A real-time adaptive least-squares drag minimization approach has been proposed for the variable camber continuous TE flap (VCCTEF) concept [5]. It uses a recursive least-squares algorithm to estimate the derivatives of the aerodynamic coefficients with respect to the inputs. The optimal wing shape and elevator deflection are then calculated using the Newton-Raphson method from a constrained optimization problem. Improvements to the model excitation method, onboard model, and optimization methods were demonstrated in wind tunnel experiments to achieve up to 9.4% drag reduction on the common research model (CRM) with VCCTEF at low subsonic speeds [2]. Simulations have also indicated that a 3.37% drag reduction is achievable on the CRM with a distributed mini-plain flap system at Mach 0.85 [6].

While the linear-in-the-parameters multivariate polynomial model adopted in [2], [5], and [6] has a low computational cost, a significant shortcoming is that its model coefficients are only valid around a trimmed equilibrium. The implications

Manuscript received 5 August 2022; accepted 16 September 2022. Date of publication 10 October 2022; date of current version 25 April 2023. This work was supported by the Delft University of Technology. Recommended by Associate Editor Y. Pan. *(Corresponding author: Xuerui Wang.)*

Tigran Mkhoyan and Roeland De Breuker are with the Department of Aerospace Structures and Materials, Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands (e-mail: t.mkhoyan@tudelft.nl; r.debreuker@tudelft.nl).

Oscar Ruland is with the Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands (e-mail: o.l.ruland@student.tudelft.nl).

Xuerui Wang is with the Department of Aerospace Structures and Materials and the Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, 2629 HS Delft, The Netherlands (e-mail: x.wang-6@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2022.3210164.

Digital Object Identifier 10.1109/TCST.2022.3210164

of this approach are that the model parameters have to be reidentified at every operational point to perform real-time drag minimization throughout the flight envelope. The model excitation maneuvers required for reidentification need sweeps of the AOA and control surfaces which cause increased fuel consumption and reduced ride comfort. Although the inherent model structure assumptions reduce the model identification cost, they also make the model structure less adaptable. In fact, for preserving the same model accuracy, the polynomial model order and the number of coupling terms have to be varied in different flight regimes. Furthermore, gradient-based optimization methods combined with local models are prone to converge to local optima [7].

The issues of local gray-box model strategies can be overcome by an online black-box global model identification and optimization approach, which has two potentials: first, global wing shape solutions can be found, leading to more effective drag reductions; second, no additional model excitation maneuvers or changes to the model structure will be needed for operation at various flight conditions. The global optimization approach, however, does not come without challenges.

Global optimization methods generally require more objective function evaluations than local gradient-based methods, which makes them impractical for direct application to complex aerodynamic shape optimization. Low evaluation cost global surrogate models may provide a solution by sample-efficiently generalizing the information gathered by onboard sensors [8], [9]. Surrogate modeling methods in the literature include polynomial regression (PR) [10], artificial neural networks (ANNs) [11], radial basis function (RBF) models [12], and Gaussian process (GP), also referred to as kriging [13], [14]. In an online data-driven framework, ANNs are promising candidates due to their ability to approximate complex nonlinear functions and adaptability. A type of feed-forward ANN is the RBF neural network (RBFNN), which is featured by its ability in handling noisy, multiparameter, and scattered data and its sensitivity and adaptability to fresh data [15]. These characteristics make RBFNNs promising for in-flight aerodynamic model identification.

Evolutionary algorithms solve optimization problems by mimicking natural evolution. Analogous to the survival of the fittest principle in evolutionary biology, only the highest quality solutions and their offspring are selected for further consideration. Evolutionary algorithms can deal with discontinuities in the objective function and are suitable for multimodal and high-dimensional problems [16]. They have been demonstrated to be effective global black-box optimization tools to be combined with surrogate models for aerodynamic shape optimization problems [17], [18]. However, to the best of the authors' knowledge, the existing applications of evolutionary algorithms to the aerospace field are only limited to offline design optimization problems. Exploiting the merits of evolutionary algorithms in online and real-world scenarios remains an open challenge.

After an optimal wing shape is determined in-flight, a challenge still exists in realizing this optimized shape in real time. Intuitively, a feedforward mapping between the servo angle and the TE displacement can be utilized. However, owing to the nonlinear couplings between aerodynamics and structural dynamics, this mapping is uncertain and is perturbed by external disturbances. Moreover, in an earlier validation experiment of a morphing wing prototype [19], a nonlinear phenomenon named backlash was observed [19], [20]. Owing to the backlash, the output of the morphing mechanism not only depends on the actuator inputs at the current time instant but is also determined by the actuation history, leading to an undesirable hysteresis phenomenon [19], [21]. One promising approach to robustly observe and regulate the motions of a physical system is vision-based control, which has shown its effectiveness in aircraft position tracking [22] and inverted pendulum stabilization [23]. However, the effectiveness of vision-based control in dealing with nonlinear hysteresis and wind disturbances remains unknown.

This article proposes and experimentally validates an effective and adaptable online performance optimization architecture for active morphing wings. The main contribution is threefold.

1) Eliminates local model excitation maneuvers during operations and can find global optima by integrating an evolutionary optimization strategy with a global online-learned RBFNN model.
2) Realizes the optimal shape in the presence of backlash hysteresis, model uncertainties, and external gust disturbances using a nonlinear adaptive control algorithm supported by real-time computer vision sensing.
3) Validates the proposed architecture on a seamless active morphing wing demonstrator with distributed sensing and control in an open jet wind tunnel.

This article is organized as follows. Section II formulates the problem. The optimization architecture is then designed in Section III. The optimized shape realization using a distributed vision-based control is proposed in Section IV. The experimental setup and results are presented in Section V and Section VI, respectively. Finally, conclusion is drawn in Section VII.

II. Preliminaries and Problem Formulation

A. Morphing Wing

The active morphing wing considered in this research is illustrated in Fig. 1.[1] The wing has six distributed translation induced camber modules, allowing independent camber and span-wise twist morphing [19]. There are 12 servos (two per module) embedded in the wing box, allowing the TE bottom skin to slide in chord-wise and span-wise directions along a guided sliding interface. To reduce noise and drag, elastomeric skin segments are integrated between the modules. As shown in Fig. 1, the rotational motion of the servo is converted to the sliding motion of the skin by a ball joint linkage system, which results in active morphing of the wing from its nominal NACA6510 shape [19].

B. Online Optimization Problem

The goal of online shape optimization is to find the most aerodynamically efficient wing shape and AOA combination

[1]The project video can be found via https://www.youtube.com/watch?v=SdagIiYRWyA&t=319s.

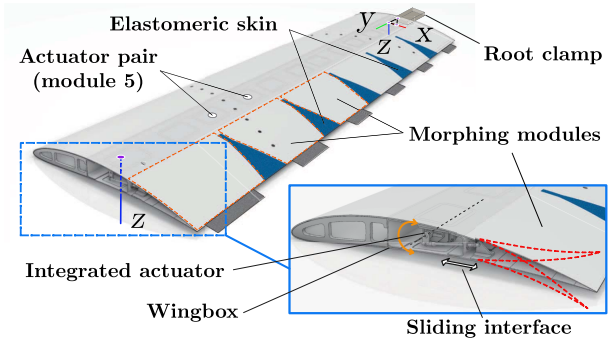


Fig. 1. Morphing wing actuation mechanism and coordinate system.

without altering the intended flight path of the aircraft. This task is complicated by the uncertainties, unsteadiness, and nonlinearities in aerodynamics. The wing shape is governed by a virtual shape input vector $\mathbf{u} \in \mathbb{R}^{q\times 1}$, which will be elaborated in Section III-A. The wing AOA is denoted as α. The mappings from $\alpha, \mathbf{u}$ to the lift and drag coefficients C_L and C_D are highly nonlinear and depend on many uncertain parameters. The maximization of lift-to-drag ratio (C_L/C_D) results in reduced fuel consumption of the aircraft. Moreover, the right amount of lift force must be generated to maintain level flight, which is determined by the target lift coefficient C_{L_t}. Therefore, the objective of the optimizer is to find the set of inputs $\alpha, \mathbf{u}$ that maximizes (C_L/C_D) while meeting the target lift coefficient C_{L_t} without violating the AOA and actuation limits, which is formulated as follows:

$$\begin{aligned} &\text{maximize}_{\alpha,\mathbf{u}} \ \frac{C_L(\alpha,\mathbf{u})}{C_D(\alpha,\mathbf{u})} \\ &\text{subject to } C_L(\alpha,\mathbf{u}) = C_{L_t} \\ &\qquad \alpha \in [\alpha_{\min}, \alpha_{\max}] \\ &\qquad \mathbf{u} \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}]. \end{aligned} \tag{1}$$

This optimization problem is nonlinear and non-convex, because C_L and C_D are nonlinear and non-convex functions of α and $\mathbf{u}$.

III. Optimization Architecture

To solve the online optimization problem formulated in Section II-B, the input $\mathbf{u}$ is first elaborated in Section III-A, followed by the cost function design in Section III-B. The evolutionary optimization strategy and neural network onboard model are designed in Section III-C and Section III-D, respectively. Finally, the overall optimization architecture is presented in Section III-E.

A. Virtual Shape

To ensure spanwise smoothness, within the optimization algorithm, the wing shape is represented by the virtual shape input vector $\mathbf{u} \in \mathbb{R}^{q\times 1}$, rather than the TE displacements at the 12 actuator locations $\mathbf{z} = [z_1, z_2, \ldots, z_{12}]^\mathsf{T}$. This choice has another promising benefit, i.e., the dimensionality of the optimization problem is decoupled from the input dimensionality of the morphing wing. Consequently, if q is chosen to be less than 12, the computational load is significantly reduced.

Essentially, the virtual inputs are the parameters of an approximation function that describes the wing shape.

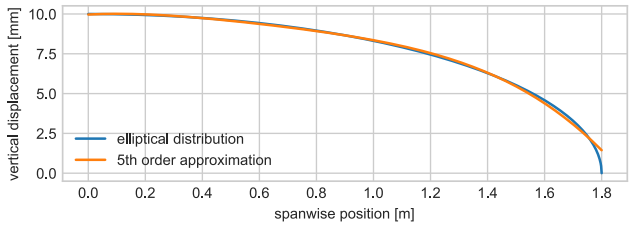


Fig. 2. Comparison of an elliptical distribution function and its 5th order Chebyshev polynomial approximation.

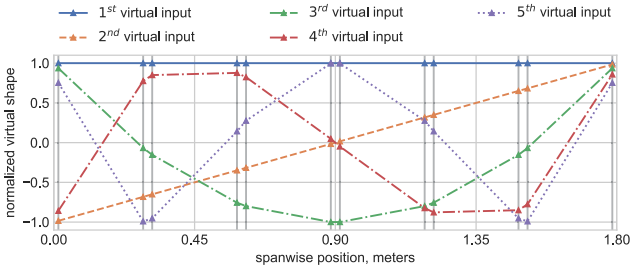


Fig. 3. Virtual input basis functions.

Chebyshev polynomials were chosen as the parametric wing shape approximation basis functions because of their nearly optimal property and orthogonality [24]. Consequently, the virtual inputs scale q basis shapes, which are described by the first q Chebyshev polynomials of the first kind: $T_q(x) = \cos(q \cdot \arccos(x))$, where q is a non-negative integer. These polynomials are orthogonal in the interval $[-1, 1]$ and are rescaled onto the $[0, 1.80]$ m domain, where 1.8 m is the half-wing span. The choice of q is determined by the tradeoff between reducing the number of measurements required to identify an onboard model and computational loads and reducing the shape approximation error. The order of the virtual shape function was chosen to be $q = 5$, because it resulted in the greatest reduction in computational loads without compromising the approximation power below acceptable levels for the expected wing shapes. Fig. 2 demonstrates that the 5th order Chebyshev polynomial has sufficient approximation power to make the normalized root mean square error (NRMSE) below 1.15% and approximate an elliptical distribution well.

As a consequence, the five virtual inputs $u_1, u_2 \ldots, u_5$ are the coefficients of a fifth-order Chebyshev approximation of the spanwise camber distribution function that describes the morphed wing shape. The distribution function is given as follows:

$$z(y) = \sum_{i=1}^{5} u_i T_i(y) \tag{2}$$

in which z is the TE vertical displacement as a function of the spanwise location y (Fig. 1). The local TE displacement z_i at the ith actuator is $z_i = z(y_i)$, where y_i is the spanwise location of the actuator. The shapes described by these basis polynomials and their contributions to the amount of camber at the actuator locations are shown in Fig. 3.

B. Cost Function

The cost function is used to score the desirability of the system outputs C_L and C_D. It is designed to maximize the

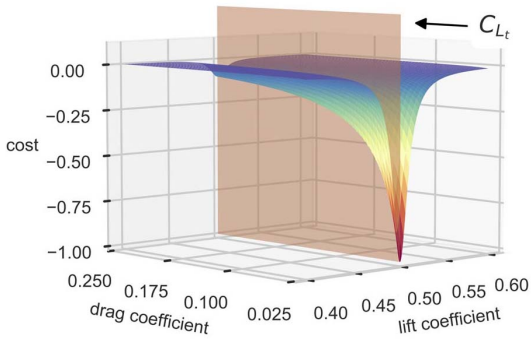


Fig. 4. Cost function for $C_{L_t} = 0.50$.

lift-to-drag ratio while regulating the error between the actual and the target lift coefficients, as presented in the following equation:

$$J\left(C_L, C_D, C_{L_t}\right) = \underbrace{-\frac{C_L}{C_D}}_{\text{efficiency}} \cdot \underbrace{\frac{k_2}{k_1 + \left(C_L - C_{L_t}\right)^2}}_{\text{deviation from lift target}}. \tag{3}$$

The efficiency and lift target terms are multiplied such that a low cost can only be reached when high efficiency and the correct amount of lift are achieved simultaneously. A constant $k_1 = 1 \times 10^{-4}$ is added to prevent singularities for small error values. The constant $k_2 = 2 \times 10^{-5}$ serves to scale the cost function output to $[-1, 0]$. As an example, the cost function for $C_{L_t} = 0.50$ is shown in Fig. 4.

It can be observed from Fig. 4 that a solution deviates from the target lift coefficient is undesirable even if it provides low drag. In this research, the virtual inputs and AOA bounds pose constraints on the inputs and are implemented directly as constraints on the input space. By contrast, the TE displacement bounds and target lift coefficient pose constraints on intermediate and output variables and are implemented indirectly through the cost function scoring. If the local TE displacement at any of the actuator locations is outside of the bounds, the cost value is determined by a penalty function. The cost is then equal to the square of the maximum TE displacement at any of the actuator locations plus a large constant, such that the cost penalty will always be higher than the cost of inputs that do not violate the constraints. The square of the maximum absolute local TE deflection provides a cost gradient to aid the optimizer in steering the solution back to the feasible space.

C. Covariance Matrix Adaptation—Evolutionary Strategy

To solve the nonlinear and non-convex morphing wing shape optimization problem, while considering online calculation efficiency, trivial approaches using either exhaustive search or naive random search is either infeasible or too slow considering the number of samples required. Instead, the covariance matrix adaptation evolutionary strategy (CMA-ES) [25], [26] is an evolutionary optimization strategy for black-box optimization of nonlinear non-convex continuous problems. With sufficiently large population sizes, CMA-ES has desirable global search performance as demonstrated by various black-box optimization benchmarking studies [25], [27]. Furthermore, as suggested by a recent comparative study, CMA-ES and its variants [28] have the ability to find the optima of a broader class of functions, with promising efficiency and speed [29], [30].

The applicability of CMA-ES to multitopology optimization of aircraft design combined with the nonlinear vortex lattice method aerodynamic analysis is demonstrated in [31]. It has also been applied for online optimization of feedback controllers of aeroacoustic instabilities of gas turbine combustors [28]. The adaptability and effectiveness of CMA-ES are further demonstrated in comparative studies with particle swarm optimization, genetic algorithm, and multipopulation genetic algorithm for damage detection and quantification of structural systems [32]. Various applications of CMA-ES for real-world online optimization are discussed in [30].

The principle of CMA-ES is to iteratively sample populations of candidate solutions from a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{C})$, which is uniquely identified by its mean $\mathbf{m} \in \mathbb{R}^n$ and covariance matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ [33]. Based on the returned costs of these candidate solutions, the mean and covariance matrix of the next generation's population are deterministically adapted. This process is repeated until the variation of the cost function converges to below a set threshold.

With g the generational counter, the k-th offspring from the subsequent generation $g + 1$ is sampled from a multivariate normal distribution $\mathcal{N}$, which is dependent on the current generation's mean search distribution value $\mathbf{m}^{(g)}$, overall standard deviation or step size $\sigma^{(g)}$, and covariance matrix $\mathbf{C}^{(g)}$ as shown in the following equation:

$$\boldsymbol{x}_k^{(g+1)} \sim \mathcal{N}\left(\mathbf{m}^{(g)}, \left(\sigma^{(g)}\right)^2, \mathbf{C}^{(g)}\right), \quad \text{for} \quad k = 1, \ldots, \lambda. \tag{4}$$

The distribution of $\mathcal{N}(\mathbf{m}^{(g)}, (\sigma^{(g)})^2, \mathbf{C}^{(g)})$ is equal to $\mathbf{m}^{(g)} + \sigma^{(g)}\mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)})$. In (4), $\mathbf{m}^{(g)}$ shifts the center of the multivariate normal distribution in a n-dimensional space; $\sigma^{(g)}$ scales the size of the distribution; $\mathbf{C}^{(g)}$ adapts the shape of the distribution. During each iteration of the algorithm, $\mathbf{m}^{(g)}$, $\sigma^{(g)}$, and $\mathbf{C}^{(g)}$ are updated based on the object parameter variations. As shown in (5), the mean of the next generation is a weighted average of the μ best scoring search points from the sample $\mathbf{x}_1^{(g+1)}, \ldots, \mathbf{x}_\lambda^{(g+1)}$. In other words, the center of the next generation's distribution is shifted in the average direction of the best performing candidates

$$\mathbf{m}^{(g+1)} = \sum_{i=1}^{\mu} w_i \, \mathbf{x}_{i:\lambda}^{(g+1)}. \tag{5}$$

The adaptation law for the covariance matrix is shown as follows:

$$\begin{aligned}\mathbf{C}^{(g+1)} = (1 - c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \underbrace{\mathbf{p}_c^{(g+1)}\mathbf{p}_c^{(g+1)^{\mathrm{T}}}}_{\text{rank-one update}} \\ + c_{\text{cov}}\left(1 - \frac{1}{\mu_{\text{cov}}}\right) \\ \times \underbrace{\sum_{i=1}^{\mu} w_i \left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)\left(\frac{\mathbf{x}_{i:\lambda}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}}\right)^{\mathrm{T}}}_{\text{rank}-\mu\text{update}}\end{aligned} \tag{6}$$

in which c_{cov} and μ_{cov} are the learning rate for updating the covariance matrix and weighting parameter between rank-one and rank-μ updates, respectively. The rank-μ update uses information from previous generations to improve the reliability of the covariance matrix estimator for small population sizes. The rank-one update exploits the directional information from past generations using the evolution path $\mathbf{p}_c^{(g+1)}$, which is a sum of successive steps defined as follows:

$$\mathbf{p}_c^{(g+1)} = (1-c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2-c_c)\mu_{\text{eff}}}\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \quad (7)$$

where c_c is the learning rate for cumulation for the rank-one update; μ_{eff} is the variance effective selection mass defined as follows:

$$\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1}. \quad (8)$$

The overall standard deviation $\sigma^{(g)}$ scales the size of the search distribution based on the length of the evolution path compared with its expected length under random selection as follows:

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathrm{E}\|\,\mathcal{N}(\mathbf{0},\mathbf{I})\|} - 1\right)\right) \quad (9)$$

in which c_σ and d_σ are the learning rate for the cumulation for the step size control and a damping parameter, respectively. The scaling of the distribution with $\sigma^{(g)}$ can be used to either broaden the distribution's search space or to focus it. When an evolution path is relatively long, the successive steps are roughly in the same direction and the step size can be automatically increased so that fewer iterations are needed to cover the distance. Conversely, when the evolution path is short, the successive steps, at least partially, cancel out each other and the step size can be automatically decreased [33].

D. Onboard Model—RBFNNs

To reduce the online objective function evaluation cost, a global surrogate model (also referred to as the onboard model) is established to approximate the mappings from the AOA and wing shape inputs to the lift and drag coefficient outputs using two RBFNNs. These types of ANNs use RBFs as activation functions and are widely used as function approximators, particularly suitable for establishing multivariate and nonlinear mappings from multiparameter, noisy, and scattered data sets [15].

The architectures of the two single-hidden-layer RBFNNs used in this research are shown in Fig. 5. Both of them use six inputs: the AOA α and the five virtual inputs that describe the wing shape $u_1, \ldots, u_5$. Their respective outputs are the lift and drag coefficients C_L and C_D.

Equation (10) represents the hidden unit activations given by the basis functions ϕ_j (e.g., Gaussian basis functions). These depend on the input activations from the previous layer $\boldsymbol{x}$, and on the parameters $\boldsymbol{\mu}$ and σ_j [15], where $\boldsymbol{\mu}$ represents the RBF location in $\mathbb{R}^{q+1}$ and σ_j denotes the RBF radius

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right). \quad (10)$$

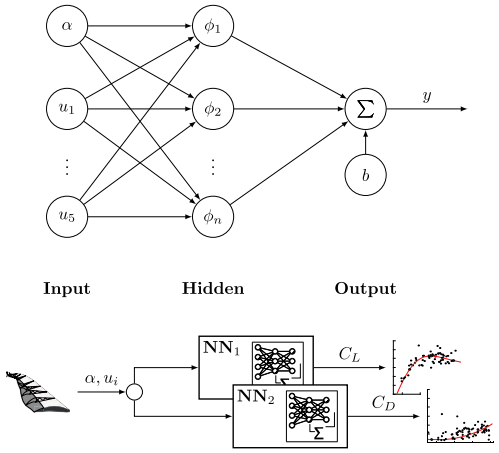


Fig. 5. Single-hidden-layer RBFNN architecture.

Both the center locations and the radii of the RBF basis functions are determined by the network training process. To train the onboard model, the adaptive gradient algorithm (Adagrad) is used [34], which maintains and adapts one learning rate for each dimension using historical data as follows:

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \quad (11)$$

in which $g_{t,i}$ is the gradient of the loss function with respect to parameter i at time t, and $G_{t,ii} = \sum_{\tau=1}^{t} g_{\tau,i}^2$ is the cumulative sum of the squares of the past gradients. The effect of $G_{t,ii}$ in (11) is that the effective learning rate is diminished over time. Since this diminishing depends on the parameter gradient histories, higher learning rates are automatically used for parameters relating to infrequent features, whereas lower learning rates are automatically used for parameters relating to frequent features. This makes Adagrad well-suited for dealing with sparse data.

E. Optimization Architecture and Algorithms

An overview of the optimization architecture is shown in Fig. 6. The left-hand side of Fig. 6 presents a fast model optimization loop, which comprises three main parts: 1) the cost function (Section III-B); 2) the optimizer (Section III-C); and 3) the onboard model (Section III-D). A pseudocode description of the online optimization algorithm is given by Algorithm 1.

During each iteration of the optimization procedure, the evolutionary optimizer generates a population of λ candidate solutions and queries the onboard model (RBFNNs) with their AOA and wing shape combinations $\alpha_k, \mathbf{u}_k$, where $k = 1, \ldots, \lambda$. In turn, the onboard model predicts the steady-state lift and drag coefficients $\hat{C}_{L_k}, \hat{C}_{D_k}$ resulting from each of these inputs, and the cost function scores the desirability $\gamma_k = J(\hat{C}_{L_k}, \hat{C}_{D_k}, C_{L_t})$ of these predicted outputs. Subsequently, the scores of the evaluated input combinations are used by the optimizer to generate a more promising population of input combinations. This cycle is repeated with a frequency of approximately 15 Hz, depending on the population size and available computing power. This process continues until the optimizer converges onto a single most optimal $\alpha, \mathbf{u}$ combination, whose index is defined as β. This input combination

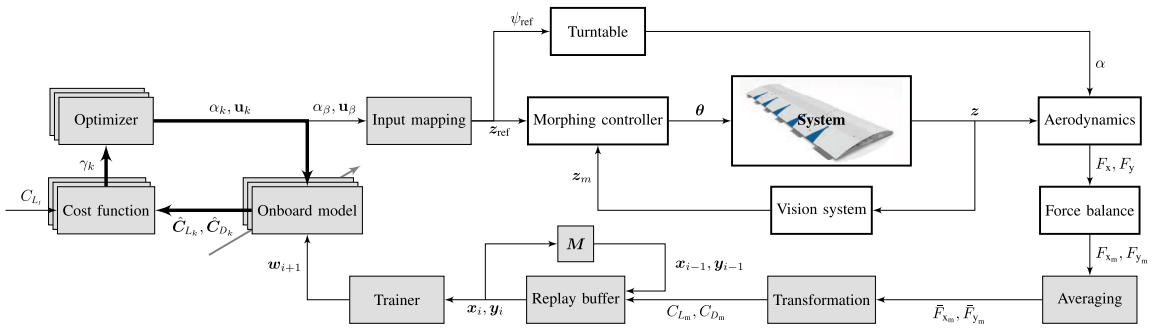


Fig. 6. Online shape optimization architecture diagram with components of the optimization algorithm marked in gray.

Algorithm 1 Online Morphing Shape Optimization

Input: $C_{\mathrm{L_t}}$
Output: $C_{\mathrm{L_m}}, C_{\mathrm{D_m}}$
Initilialize: $\epsilon = 1.0$
while running **do**
  while calculating the running averages $\bar{F}_{\mathrm{x_m}}, \bar{F}_{\mathrm{y_m}}$ **do**
    while $\epsilon > 1 \times 10^{-6}$ **do**
      Generate a population of λ candidates;
      for $k = 1, \ldots, \lambda$ **do**
        Compute candidate $\hat{C}_{\mathrm{L}_k}, \hat{C}_{\mathrm{D}_k}$ using RBFNNs;
        Evaluate cost function $\gamma_k = J(\hat{C}_{\mathrm{L}_k}, \hat{C}_{\mathrm{D}_k}, C_{\mathrm{L_t}})$;
      end
      $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_\lambda]$;
      $\epsilon = \max(\boldsymbol{\gamma}) - \min(\boldsymbol{\gamma})$;
    end
    Define the most promising candidate as the β-th that leads to $\min(\boldsymbol{\gamma})$;
    Transform $\mathbf{u}_\beta \rightarrow \boldsymbol{z}_{\mathrm{ref}}$;
    $\psi_{\mathrm{ref}} = \alpha_\beta - \varepsilon$;
    Command ψ_{ref} and $\boldsymbol{z}_{\mathrm{ref}}$;
  end
  Transform $\bar{F}_{\mathrm{x_m}}, \bar{F}_{\mathrm{y_m}}$ to $C_{\mathrm{L_m}}, C_{\mathrm{D_m}}$;
  Add $C_{\mathrm{L_m}}, C_{\mathrm{D_m}}$ to training buffer;
  Retrain RBFNNs;
end

is then actuated on the real system and the resulting measurements are subsequently used to improve the onboard model.

Since the zero position of the turntable did not coincide with the zero AOA, the turntable angle to be commanded to realize any true aerodynamic AOA was unknown beforehand. To address this challenge, the angular difference ε between the commanded turntable angle ψ_{ref} and the aerodynamic AOA α, was estimated using measurements of a sweep in AOA with the wing jig (baseline) shape and the iterative approach is outlined in Algorithm 2. During this procedure, the turntable misalignment constant ε was iteratively estimated by matching the measurement-based zero-lift AOA $\hat{\alpha}_{C_L=0}$ estimation to the theoretical zero-lift AOA $\alpha^*_{C_L=0}$. This theoretical zero-lift AOA was determined to be $-6.5570°$ using the 3-D panel viscous solver *XFLR5 v 6.48*. This estimation procedure was performed once, at the beginning of the experiment.

Algorithm 2 Turntable Misalignment Estimation

Input: $\psi, F_{\mathrm{x_m}}, F_{\mathrm{y_m}}, \alpha^*_{C_{\mathrm{L}}=0}$
Output: ε
Initilialize: $\epsilon = 1.0,\ \varepsilon = 1.0$
while $\epsilon > 1 \times 10^{-8}$ **do**
  Adjust ε based on error ϵ: $\varepsilon = \varepsilon + 0.5 \cdot \epsilon$;
  Calculate angles of attack $\alpha = \psi + \varepsilon$;
  Transform $F_{\mathrm{x_m}}, F_{\mathrm{y_m}}$ to $C_{\mathrm{L}}, C_{\mathrm{D}}$ using α;
  Estimate $\mathrm{d}C_{\mathrm{L}}/\mathrm{d}\alpha$;
  Estimate zero lift angle of attack $\hat{\alpha}_{C_L=0}$;
  Update error: $\epsilon = \hat{\alpha}_{C_L=0} - \alpha^*_{C_L=0}$;
end

Using the ε estimation, the most promising aerodynamic AOA α_β is converted to an equivalent turntable angle reference $\psi_{\mathrm{ref}} = \alpha_\beta - \varepsilon$. In addition, the virtual shape inputs $\mathbf{u}_\beta$ are transformed to the reference TE displacements at the 12 actuator locations $\boldsymbol{z}_{\mathrm{ref}} \in \mathbb{R}^{12}$. Then, the reference table angle and TE displacements are sent to the turntable and morphing controllers, respectively. The morphing controller uses the TE displacement feedback $\boldsymbol{z}_m$ from a vision system to steer the TE displacements to their reference values by controlling the morphing actuators $\boldsymbol{\theta}$. The vision system and the morphing controller will be elaborated in Section IV.

After the controllers have converged and the intended wing shape and AOA are actuated on the system, the resulting aerodynamic forces F_{x_m}, F_{y_m} are measured using a force balance which is mounted to the turntable. The 40-s running averages of the resulting measurements denoted as $\bar{F}_{x_m}, \bar{F}_{y_m}$ are then transformed to the lift and drag coefficients C_{L_m}, C_{D_m} using an α-dependent nonlinear mapping. Both the inputs used, and the resulting aerodynamic coefficients, are added to the training set which is kept in memory. Finally, the loop is closed by training the onboard model, which results in updated parameters $\boldsymbol{w}_{i+1}$. This completes the architecture.

IV. Optimized Shape Realization—A Vision-Based Control Approach

A vision-based control approach is adopted to realize the optimized shape on the real physical system while resisting external disturbances, model uncertainties, and mechanical imperfections. The distributed vision system is described in Section IV-A, while an adaptive nonlinear shape control algorithm is presented in Section IV-B.

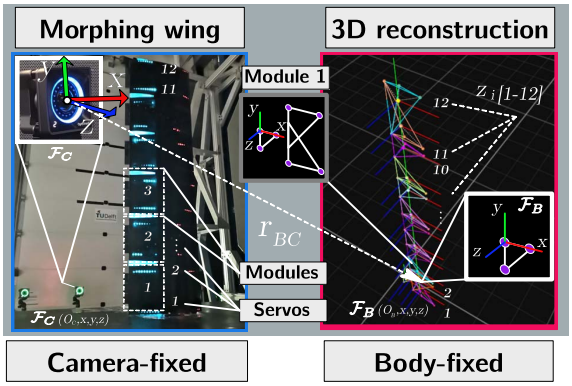


Fig. 7. Coordinate systems and vision-based wing shape reconstruction.

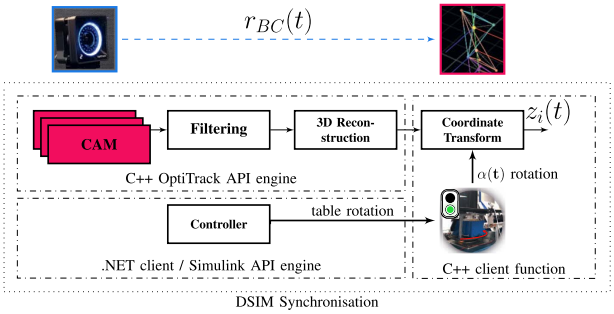


Fig. 8. Real-time vision-based tracking pipeline.

A. Vision System

1) Vision-Based Shape Reconstruction: A distributed vision system is developed to provide the knowledge of the morphing wing' shape to the controller in real time. The variable of interest to the controller is the local vertical displacement of the wing trailing edge with respect to a body-fixed coordinate system. The body-fixed coordinate system $\mathcal{F}_B$ is chosen to be near the root of the wing with an origin O_B (Fig. 7). The displacement of the trailing edge is reconstructed in the $\mathcal{F}_B$ frame in real time by means of a 5-view vision-based tracking system. Each morphing module is fit with a pair of active infrared (IR) light-emitting diodes (LEDs), with 3 additional markers for the definition of $\mathcal{F}_B$. An overview of the vision-based tracking pipeline is shown in Fig. 8.

2) 3-D Reconstruction: The n-view 3-D reconstruction problem is concerned with finding the optimal estimation of an object $\hat{X}$ in a 3-D global coordinate frame (i.e., locations in the x–z-axes), which is observable in $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$ noisy points correspondence in n camera views. The point correspondences $\bar{x}_i$ are generally defined by markers in u, v coordinates of a 2-D image plane and transformed to a camera fixed reference frame $\mathcal{F}_C$ via triangulation. Prior to this process, several successive image processing steps are implemented to refine the observation of point correspondences. The most common setup for triangulation is a calibrated 2-view stereo camera setup, which was demonstrated in a previous study for the reconstruction of flexible wing motion [35]. However, the tracking accuracy and redundancy can be improved with the addition of more distributed camera observations. As highlighted in [35], this is particularly beneficial for objects subject to adverse environmental conditions (such as disturbing flow conditions and mechanical vibrations), where calibration drift can be accumulated over time. Therefore, a five-camera setup was adopted in this research.

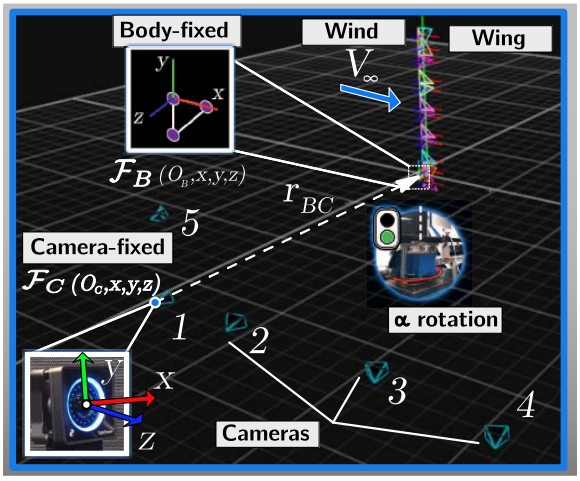


Fig. 9. 3-D orientation of the cameras with respect to the morphing wing positioned in the wind tunnel.

The principle of the n-view reconstruction relies on back-projecting the 3-D point onto the respective camera views, allowing to define a minimization problem for the reprojection error, $E = \sum_{k=1}^{n} ||x_k - \bar{x}_k||^2$. The n-view minimization problem is commonly solved by an expanded linear system of equations similar to singular value decomposition (SVD) in a direct linear transform (DLT) procedure [36]. Global optimization methods can be applied, such as algebraic, matrix inequality, and the $\mathcal{L}_\infty$ approach [37]. More computationally intensive methods also exist, such as bundle adjustment, where the camera calibration parameters are not known prior and are included in the minimization problem [36]. This research adopted a proprietary 3-D point cloud reconstruction engine in the OptiTrack application programming interface (API) system [38]. Multicamera calibration was performed by a wanding process, resulting in an average calibration error of 0.25 mm for all cameras.

3) Coordinate Transformation: As illustrated in the 3-D view of Fig. 9, a coordinate system transformation from $\mathcal{F}_C$ to $\mathcal{F}_B$ is required for shape reconstruction. This transformation is performed by a translation, followed by 3-axis rotations in pitch, roll, and yaw axes (θ, ϕ, ψ). It is noteworthy that $\mathcal{F}_B$ is attached to the wing and thus is rotating along with the turntable. Therefore, the transformations have to be performed continuously in real time. The applications for processing, reconstructing, and accessing the data are written in low-level C++ programming language for performance enhancement. The average total processing latency of the complete processing pipeline (Fig. 8) was in the range of 5–7 ms, which is smaller than the control sampling interval (16.67 ms).

B. Nonlinear Adaptive Vision-Based Control

The objective of the morphing controller is to steer the distributed actuators to morph the wing to the optimal shape commanded by the optimizer, using the real shape reconstructed by the vision system as a feedback signal (Fig. 6). The morphing wing can be represented by the following nonlinear

multiinput multioutput system:

$$\dot{x}(t) = f(x(t), u(t)) + d(t), \;\; y(t) = h(x(t)) \tag{12}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^p$ are the system state, input, and measurable output, respectively. $d(t) \in \mathbb{R}^n$ is the external disturbance vector. For this morphing control problem, the output signal y is provided by computer vision. $f : (\mathbb{R}^n, \mathbb{R}^m) \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are assumed to be Lipschitz continuous on their domains.

To make the controller constantly adapt to changes in the system and environment, online system identification is necessary. A precise online identification method for dynamic systems that only requires output feedback and less computation power than neural networks has been proposed in [39]. This method identifies the incremental [40] model using sampling interval-based linearization and discretization.

Theorem 1 [39]: If the nonlinear system given by (12) is observable, then under sufficiently high sampling frequency, the output tracking error increment $\Delta e_{t+1} = \Delta y_{t+1} - \Delta y_{\text{ref},t+1}$ can be determined uniquely from the observations and control inputs over a sufficiently long time horizon, $[t - M + 1, t]$, $M \geq (n + p)/p$

$$\Delta e_{t+1} \approx F_t \overline{\Delta e}_{t,M} + G_t \overline{\Delta u}_{t,M} \tag{13}$$

where $F_t \in \mathbb{R}^{p \times Mp}$ is the augmented transition matrix, and $G_t \in \mathbb{R}^{p \times Mm}$ is the augmented input distribution matrix. $\overline{\Delta u}_{t,M} = [\Delta u_t^{\mathsf{T}}, \Delta u_{t-1}^{\mathsf{T}}, \ldots, \Delta u_{t-N+1}^{\mathsf{T}}] \in \mathbb{R}^{Nm}$ and $\overline{\Delta e}_{t,M} = [\Delta e_t^{\mathsf{T}}, \Delta e_{t-1}^{\mathsf{T}}, \ldots, \Delta e_{t-N+1}^{\mathsf{T}}] \in \mathbb{R}^{Np}$ are the input and tracking error data of N previous samples, respectively. For the selection of N, readers are recommended to [39]. Using (13), the nonlinear dynamic inversion control in its discrete form can be applied, which regulates the tracking error to zero asymptotically.

It is noteworthy that this nonlinear adaptive control is superior to the incremental nonlinear dynamic inversion control in the literature [40] mainly because of its abilities of online adaptation and dealing with hysteresis effects [41]. The Lyapunov-based stability and robustness proofs of the proposed control method are presented in our recent publication [41]. Real-world experiment results show that this nonlinear adaptive controller is able to decrease the tracking error by more than 62% despite model uncertainties, external disturbances, frictions, and nonlinear backlash hysteresis [41].

V. Experimental Design and Setup

To assess the performance of the proposed online shape optimization architecture, an experiment was conducted with the SmartX-Alpha morphing wing at the open jet facility (OJF) wind tunnel [42].

A. Apparatus

The experimental apparatus is shown in Fig. 10. The system consists of the SmartX-Alpha morphing wing, mounted vertically on an actively controlled turntable system, and placed in the wind tunnel test section. The operational point was selected as $V = 15$ m/s. The wing is clamped on a three-axis external balance measurements system, allowing to measure root reaction forces and moments at 1000 Hz.

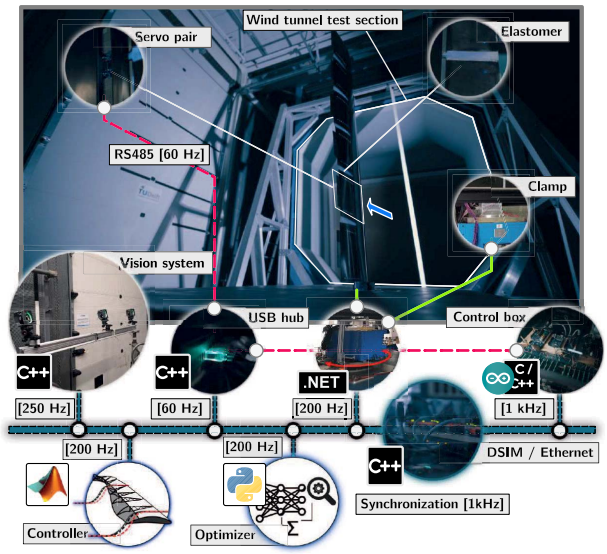


Fig. 10. Experimental apparatus with various hardware, software, and vision-based control components. The green, red, and blue lines represent mechanical, electrical, and synchronization paths, respectively.

The shape command and the resulting lift distribution are achieved by 12 high-performance servos, connected to an array of recommended standard (RS)-485 devices communicating serially via the RS-485 protocol. The update rate is constrained by the physical universal serial bus (USB) host interface with a fixed time delay of 15 ms. The actuation angles of the servo are constrained to $\pm 25°$ as to not exceed the physical limits of the morphing system. The online optimization was carried out on an Intel Xeon W-2223 3.60 GHz central processing unit (CPU) system with 16.00 GB random access memory (RAM).

An array of IR-LEDs of type 3528 850NM WLP PLCC2, characterized by 850 nm wavelength is installed on the wing bottom surface and powered by a 12 V direct current (DC) power supply. The brightness is actively controlled by the IRF520 Power metal–oxide–semiconductor field-effect transistor (MOSFET) dimmer circuit. Five Prime^{x}41 4.1 Mpixel IR cameras are responsible for marker tracking at a frame rate of 250 frames per second (FPS) [38]. The shape-reconstruction algorithm is written in C++ and deployed on a Dell Optiplex 7400 i5-8500 3.00 GHz CPU system with 8.00 GB RAM.

To continuously control the wing AOA, a real-time turntable control loop has been implemented. The Franke turntable of type LTB 400 is equipped with a brushless TC-60-1.3 motor with encoder and breaking systems. The table angle is measured by an MSR-40-MOR rotary encoder. The turntable is controlled by a proportional–integral–derivative algorithm whose parameters are tuned to provide smooth table angle command tracking while satisfying servo rate and position limits. The servo commands are communicated via an RS232 protocol over a USB controller. A.NET-based software control interface is developed to set control parameters and received the encoder feedback signal, which is interfaced to the synchronization framework in real time at 200 Hz.

B. Real-Time Synchronization

Various hardware and software components are needed to cooperate coherently and harmoniously in real time. To facili-

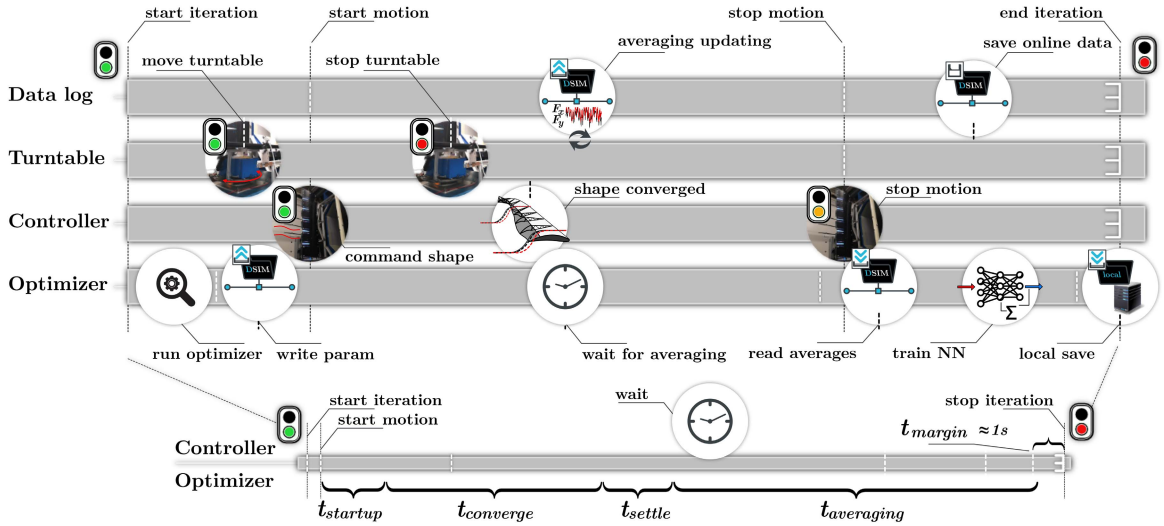


Fig. 11. Control flow and timeline of the experimental system.

tate this, a distributed data-sharing architecture was developed based on the decentralized communication principle, which allows parallel integration of hardware and software components in various programming languages (Python, MATLAB, Simulink, C++,.NET, etc.) and various communication protocols (RS485, Ethernet, ModBus, etc.). The architecture software is developed in C++ with the real-time D-SIM framework, connecting several personal computer (PC) nodes over a local Ethernet network [43], and enabling synchronization as depicted in the bottom part of Fig. 10. This approach provides several key benefits over conventional centralized systems: 1) running hardware and software processes in parallel at nonuniform sampling rates with a 1 kHz synchronization of shared variables between processes; 2) scalability and easy to be modified system structure; and 3) mixing various programming languages and protocols for various experimental components.

C. Optimization Experiment Configuration

The CMA-ES algorithm was used to solve the optimization problem. A relatively large population size of $\lambda = 150$ was used to improve the global search performance of the algorithm. The middle of the input domain was used as the initial solution point $\mathbf{x}_0$. To allow global convergence, the initial standard deviation $\boldsymbol{\sigma}_0$ and the scaling of the input variables were selected such that $\mathbf{x}_0 \pm 2\boldsymbol{\sigma}_0$ spanned the width of the domain in each of the input axes. A suitable cost function variation convergence threshold was found at 1×10^{-6}, such that the optimizer yielded adequate convergence considering the computational load. The computational time was further reduced by a parallel query of 150 candidate solutions.

The onboard model on which the candidate solutions from the optimizer are evaluated comprises two single-hidden-layer RBFNNs. These networks are continuously improved with training being performed each iteration using training data kept in memory in the replay buffer. After balancing approximation power and computational load, the lift and drag coefficient networks (Fig. 5) were configured with 500 and 940 neurons, respectively. The higher neuron count in the drag coefficient network is necessary for capturing the higher degree of nonlinearities and spanwise distribution dependencies in drag. During training, the RBF center locations, radii, network weights, and the bias parameter are updated using the Adagrad algorithm with a mean squared error loss function. The initial learning rates for both networks were configured as 0.01.

The training data, comprised of previously evaluated inputs and their lift and drag coefficients, is stored in a buffer. In this research, the replacement of old data points when the buffer is full is based on the nearest neighbor search on all points in the buffer, inspired by the coverage maximization strategy presented in [44]. The data point with the lowest mean Euclidean distance to its 10 closest neighbors is replaced with the latest available data point. This replacement strategy aims to maximize the coverage domain of the training set by replacing the data points in regions of high data density and holding onto samples in data scare regions of the domain.

Because of the backlash effects in the actuation mechanism, the required actuator angles for any given wing shape are not unique and are unknown beforehand. Therefore, limits were imposed on the commanded local z-displacements of the trailing edge at the actuator positions $z_1, z_2, \ldots, z_{12}$. The maximum absolute displacement achievable at any actuator position is dependent on the actuation of the neighboring actuators. Actuator pairs that deflect in unison are able to effectuate larger TE displacements than actuator pairs that deflect in opposite directions. The local vertical displacements allowed for the optimizer were selected as ± 10 mm.

D. Experiment Procedures

The control flow is shown in Fig. 11, responsible for the operation of various system components during the experiment. The order and measurement conditions of the performed runs are shown in Table I. The three types of runs performed are baseline, wandering, and optimization. The baseline runs are AOA sweep with a fixed (jig) wing shape, to establish a performance baseline. During the wandering phase runs, pseudo-random (PR) inputs were actuated on the system to explore the input space for onboard model identification. Throughout the optimization runs, the optimal angles of attack and wing shapes established by the optimizer were commanded to maximize the lift-to-drag ratio.

TABLE I
EXPERIMENT TEST MATRIX, MEASUREMENT WITH ANGLE-OF-ATTACK BIAS IN GRAY (PR = PSEUDORANDOM)

C_{L_t}	α	wing shapes	type	no. of samples
-	sweep	jig shape	baseline	18
-	PR	PR	wandering	150
0.65	optimal	optimal	optimization	15
-	sweep	jig shape	baseline	18
-	PR	PR, reduced bounds	wandering	57
0.40	optimal	optimal	optimization	30
0.75	optimal	optimal	optimization	30
0.90	optimal	optimal	optimization	40

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, the experimental results are presented and discussed. First, a performance baseline is established, and the wandering phase measurements are presented in Section VI-A. Then, the convergence property and computational loads are discussed in Section VI-B. In Section VI-C, the results from online optimization for $C_{L_t} = 0.65$ are elaborated upon. Experimental data-driven optimization predictions are shown in Section VI-D for a wider range of target lift coefficients. Finally, real-world operation considerations are discussed in Section VI-E.

A. Baseline and Wandering Phase

A performance baseline was developed by measuring the aerodynamic forces of the wing jig shape at various angles of attack. This wing jig shape was realized by performing a doublet maneuver without wind. Subsequently, 40-s averaged force measurements were taken at table angles from $-18°$ to $10°$ in increments of two degrees at a wind speed of 15 m/s.

The performance baseline was established by interpolation of the jig shape measurements with a 16-th degree polynomial. This relatively high polynomial order was selected considering the fitting performance. The validity of this interpolation in the region of interest was not compromised by Runge's phenomenon [45] as the bounds of the jig sweep measurements are much wider than the region of interest. Both the jig shape measurements and the fit model are shown in Fig. 12. During the wandering phase, the actuation space was explored with PR inputs. As shown in Fig. 12, naturally, the performances of the PR wandering inputs are distributed around the jig shape performance curve. While the jig shape curve serves as a baseline, the distribution of the wandering phase performances roughly indicates the physical bounds of the attainable performance with active morphing for SmartX-Alpha.

B. Convergence Property and Computational Loads

The computational load of the proposed method is mainly comprised of the computational loads of the CMA-ES optimizer and the RBFNN onboard model. These in turn are determined by the convergence rate of the optimizer, and the amount of training of the onboard model required.

To the best of our knowledge, there are no theoretical proofs that guarantee convergence of CMA-ES in the literature. However, much evidence exists to show that the algorithm convergences on many functions [46]. This promising convergence property can be explained as follows: as shown

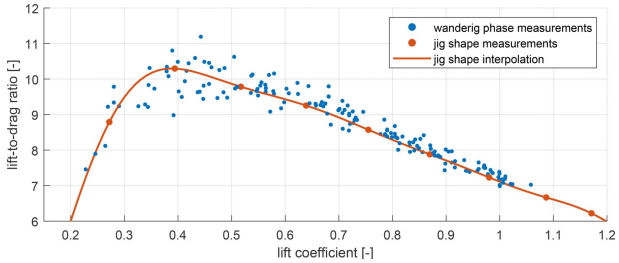


Fig. 12. Wing jig shape baseline performance measurements (orange dots), wing jig shape performance fit model (orange line), and performance measurements from PR exploration (blue dots).

in Section III-C, the scaling of the CMA-ES distribution depending on the length of the evolution path serves to either focus or broaden the search. When the successive search steps partially cancel out, the distribution is scaled down. This reduces the footprint of the generated population in the search space and consequently reduces the variation in their function evaluation values until a set threshold is reached upon which convergence is declared. Thus, in principle, convergence could only not occur if the optimizer were to step in roughly the same direction indefinitely.

This evolution-path-dependent scaling strategy is also the primary mechanism responsible for the acceleration of the convergence rate of this method. When the evolution path is relatively long, owing to a series of successive steps in a similar direction, the distribution is automatically scaled up to reduce the number of successive steps needed. Referring to [25], [33], the rate of convergence is dependent on the ratio of the population size to the number of dimensions of the input space. Therefore, in order to improve the convergence rate in our experiment, a large population size of $\lambda = 150$ was used, whilst the number of dimensions in the input space was reduced from 13 to 6 by the employment of the 5-th order virtual shape functions to describe the wing shape. In addition, the convergence threshold was selected as 1×10^{-6} which was as high as possible without causing variations in the lift-to-drag ratios of the solutions over 1×10^{-4}. This ensures that no further optimization steps are performed when the change in result would be negligible, avoiding unnecessary computations.

During both the wandering and the optimization phases of the experiment, the CMA-ES optimizer converged in every iteration. The mean time for convergence was 6.6 s on the hardware as described in Section V-A, and convergence happened within 9.8 s in 95% of cases. This time span is deemed acceptable, because it is small compared with the time required for reaching steady-state aerodynamics and noise averaging and filtering, which add up to 60 s.

The computational time required for the training of the onboard RBFNN model is mainly influenced by the number of times the model weights are updated during the training process. This in turn is largely dependent on the size of the training data set. While large training data sets improve the accuracy of the neural network model, the time required for training also increases significantly. Therefore, a limit on the training set size was introduced. When this limit is reached, new measurements replace old data in the training

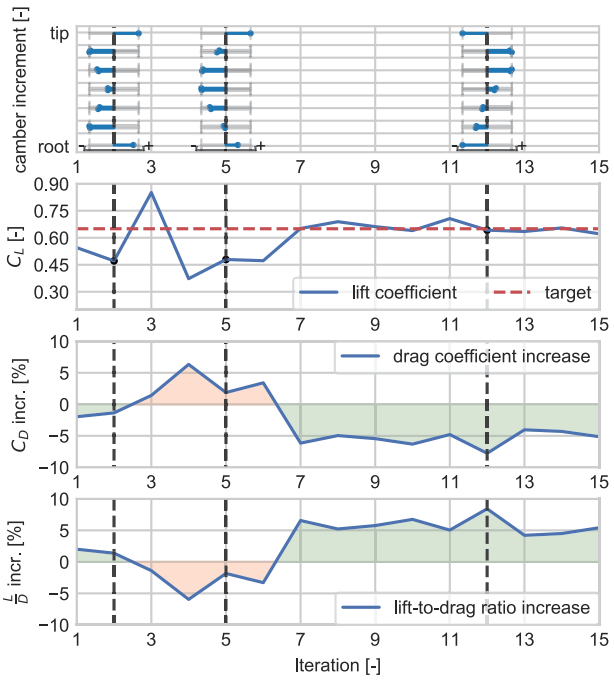


Fig. 13. Lift and drag coefficients, and selected wing shapes from online optimization with a target lift coefficient of 0.65 for 15 iterations.

set using a replacement strategy which focuses on retaining a sufficient coverage of the data throughout the optimization domain. Although the maximum amount of training data was not reached during our wind tunnel experiment, in real-world operations when large amounts of data from simulations and previous flights are available this sample replacement strategy will be a vital part of the framework.

During the experiment, the combined training of both neural networks required 3.0 s on average and remained below 3.2 s in 95% of cases. This time span is also deemed acceptable as it is even shorter than the time required for optimization.

C. Online Optimization

After the first 150 wandering phase measurements, 15 iterations of online wing shape optimization were performed with a target lift coefficient $C_{L_t} = 0.65$. The resulting lift and drag coefficient measurements are shown in Fig. 13. During the first six iterations, the discrepancies between the targeted and the measured lift coefficients are relatively large, i.e., ± 0.25. This is caused by the small number of measurements collected during the wandering phase relative to the dimensionality of the input space. The 150 samples used correspond to a grid with only $\sqrt[6]{150} \approx 2.305$ points in each of the six dimensions. Hence it is very challenging to capture the true aerodynamic mapping in such a small data set. At the same time, the measured drag coefficients also fluctuate considerably. This is also reasonable as the total drag is dominated by the lift-induced drag. Furthermore, these initially evaluated wing shapes are suboptimal, attributed to the observation of increased amounts of camber near the root and tip. While the true optimal shape is unknown, the amounts of camber at those locations are expected to decrease for constructing a nearly elliptical optimal lift distribution.

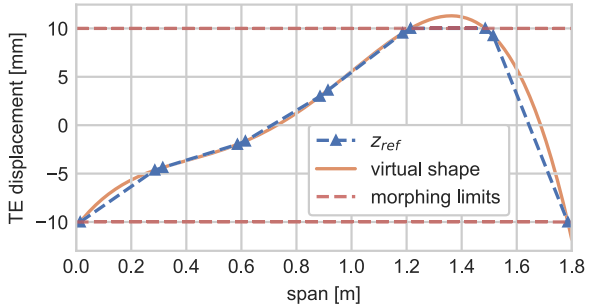


Fig. 14. Shape evaluated at iteration 12 during online optimization with a target lift coefficient of 0.65.

The variations during the first six iterations will not result in instability or severe ride discomfort on an aircraft, because the fluctuated C_L values shown in Fig. 13 are only the C_L commands. These commands and the corresponding wing shapes are realized by the inner-loop nonlinear vision-based adaptive controller which damps out the transit responses and high-frequency fluctuations in the real C_L responses. Moreover, the wing shape is adapted at a low frequency based on the steady-state aerodynamic forces and will not cause large accelerations. Furthermore, the initial fluctuations can be greatly reduced by more training on an expanded data set. This has also been proved by the experimental results that after iteration 6, the C_L values have much fewer variations, because the RBFNN onboard model has sufficiently learned the wing aerodynamic properties. Luckily, in real-world operations, the training data set generated by higher-fidelity aerodynamic simulations, test flights, and accumulated flight operations will be much more abundant than what we can afford in the wind tunnel test, thus reducing the initial model errors and C_L variations.

After iteration six, the measured lift coefficient converges to its target. Moreover, Fig. 13 shows that the wing shape and AOA combinations evaluated from iteration seven onward not only realized C_{L_t} but also achieved a lower C_D. In other words, the executed wing shapes are more efficient as indicated by the positive lift-to-drag ratio increase percentage. The most desirable performance was measured for the input combination evaluated during iteration 12 with $C_L = 0.642$. The measured lift-to-drag ratio at this iteration was 10.015, which corresponds to a 7.8% drag reduction compared with the wing jig shape at the same C_L.

The wing shape evaluated at iteration 12 is shown in Fig 14. This shape comprises maximum positive camber between 1.4 and 1.6 m from the wing root (corresponds to module 5), a steep decrease at the wing tip, and a gradual reduction of the local camber toward the root end of the wing. It is mainly the reduction of the local camber near the wing ends, which is supposed to reduce the strength of the wing tip vortices, that results in the observed improvement in the aerodynamic efficiency. These vortices are the result of the spanwise flow components caused by the "leaking" of high pressure air from the bottom side of the wing around the wing tips toward the lower pressure regions on the upper wing surface. Nevertheless, the optimal location of the maximum camber is expected to be in between the wing root and the

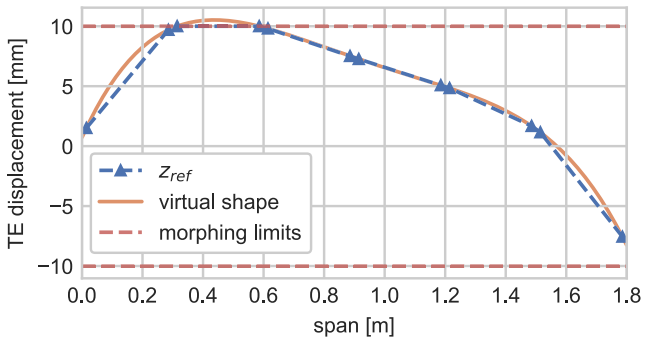


Fig. 15. Experimental data-driven optimal shape prediction for $C_{L_t} = 0.65$.

center of the wing, which corresponds to an interpolation between the elliptical distributions associated with wings with one and two free ends. There, it is suspected that even though the given shape already offers a 7.8% drag reduction over the wing jig shape for this target lift coefficient, even more efficient wing shapes do exist.

An advanced efficient wing shape is predicted using the measurements from the second wandering phase. After the first optimization run, the zero angle of the turntable was reset by a reconfiguration of the turntable hardware and software as part of the shut-down and startup procedure. The shift in the turntable zero position was approximated by matching the force balance readings to earlier established benchmarks through variation of the table angle. However, later analysis revealed this approximation method to be less accurate than was expected. Post-processing of the measurements using Algorithm 2 showed that the shift in the table angle zero position was overestimated by approximately 1.6°. As a result, the true AOA and the AOA-dependent lift and drag coefficients of the second wandering phase were unintentionally biased, which were found to negatively affect the performance of the later optimization runs. Nevertheless, more accurate optimal wing shape predictions are achieved by correcting these biases in post-processing. These results are presented in Section VI-D.

D. Experimental Data-Driven Optimization Predictions

To make an improved estimation of the optimal shape and its corresponding drag reduction, and to make a more general prediction about the optimal wing shapes at other target lift coefficients, the online training was simulated using experimental data collected during wandering phases. In post-processing, the samples measured after the AOA bias was introduced in the turntable were corrected. These samples were then fed to the optimization algorithm on a per-sample basis to simulate the wandering phase experiment. A validation subset, which contained 25% of the samples, was used to estimate the predictive accuracy of the trained model.

When $C_{L_t} = 0.65$, the optimal wing shape as computed by the optimizer on the trained onboard model, was shown in Fig. 15. With an AOA of 0.8°, the predicted lift-to-drag ratio of this shape is 10.35. This corresponds to a 11.1% of drag reduction compared with the wing jig shape. Furthermore, the shape shown closely represents the expected optimal shape described in Section VI-C, with a gradual reduction of airfoil

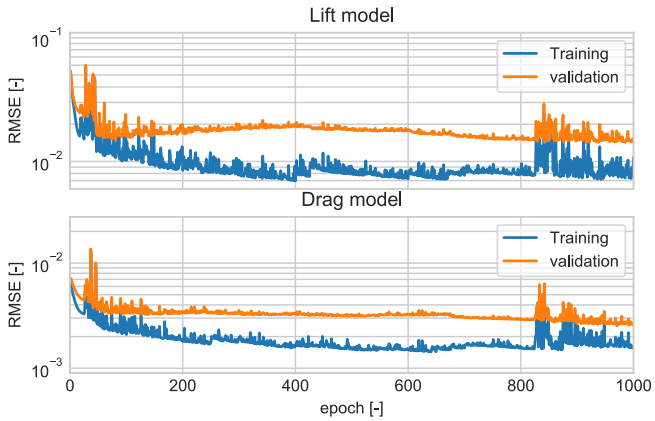


Fig. 16. Experimental data-driven model training evolution.

camber toward the free wing tip end, and a more moderate amount of camber reduction at the root end as a result of pressure leakage at the root intersection.

The predictive accuracy of the trained onboard model was evaluated using error measures computed on the validation data set. The evolution of the root mean square errors (RMSEs) of the lift and drag models on the training and validation data sets are shown in Fig. 16, in which 25 training epochs equal one iteration of the optimization algorithm. Naturally, the RMSE of the model is naturally higher on the validation data set than on the training set as the former represents unseen rather than familiar data. It can also be observed that after an initial steep decline, the validation RMSE values continue to decrease as more measurements become available.

The training of the RBFNNs is conducted using a mini-batch size of 32, meaning that the network weights are adapted based on batches of up to 32 samples. One complete pass through the entire training data set is referred to as an epoch. When the total number of samples in the training data set is not a multiple of the mini-batch size, as many mini-batches of the maximum size are drawn as possible. The last batch containing the remainder of the samples will naturally contain fewer samples than the others. Hence, if the number of the remainder of the samples is too small, the last model weight update of an epoch will be performed based on only a small number of samples. Such small-sample-size training passes generally result in increased loss and increased variance between the losses over different epochs. This phenomenon can be observed at iteration 65, which corresponds to epoch 825, containing 65 samples divided into batches of 32 samples. Consequently, the third batch contains only one sample, which causes a temporary increase in loss and noisiness. Nevertheless, with subsequent iterations, the size of the last mini-batch increases and the losses decrease again and smooth out.

The final RMSE values, and their normalized counterparts are shown in Table II. The normalization was performed using the domain width of the corresponding output variables, i.e., $\text{NRMSE} = \text{RMSE}/(y_{\max} - y_{\min})$. Table II shows that the NRMSE values of the lift and drag coefficient RBFNN models are close to each other. By contrast, the lift-to-drag ratio NRMSE is higher, because the lift-to-drag ratio is not approximated with a dedicated neural network, but by the ratio

TABLE II
MODEL ERROR MEASURES ON THE VALIDATION DATA SET

	RMSE [-]	NRMSE [%]
C_{L}	0.0147	2.34
C_{D}	0.0026	2.50
L/D	0.2818	7.75

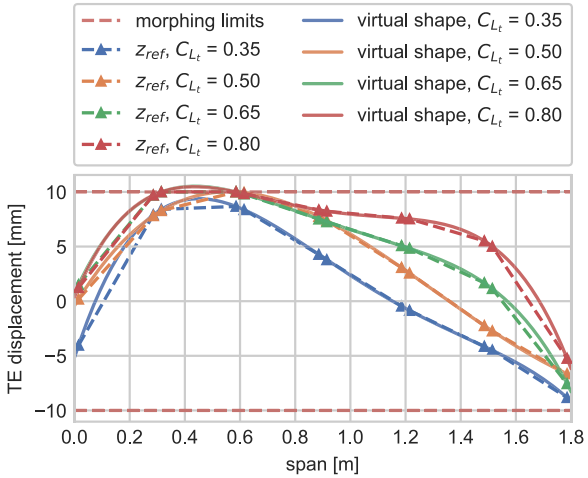


Fig. 17. Experimental data-driven optimal shape predictions for four target lift coefficients.

of two estimated outputs. Nevertheless, based on the relatively small training data set, the onboard model is able to predict the lift and drag coefficients of the validation data set samples with an average prediction error that is approximately 2.5% of their respective domain widths.

To reveal the trend in the estimated optimal wing shape, the shape optimization was also conducted for target lift coefficients of 0.35, 0.50, and 0.80. The predicted optimal shapes for these lift coefficients are shown in Fig. 17. In general, the maximum amount of camber morphing is commanded at approximately one-quarter span, with a gradual reduction approaching the minimum camber limit toward the wing tip. For higher target lift coefficients, the area under the virtual shape curves, which can be interpreted as the overall amount of camber, is increased.

For active camber morphing airfoils, the amount of lift generated can be changed with variations of both the AOA and the airfoil camber. Hence, multiple combinations of the AOA and camber morphing can be employed to realize any given C_{L_t}. However, due to the underlying aerodynamics, these different solutions will not necessarily come with the same drag penalties. Computational fluid dynamics (CFDs) simulations for active camber morphing airfoils have shown that the TE deflection for best lift-to-drag ratio increases with increasing lift coefficients [47]. Moreover, in the case of a 3-D distributed active camber morphing wing, C_{L_t} can be realized with different combinations of the AOA and the spanwise camber distribution. Earlier wind tunnel experiments on SmartX-Alpha have shown that for a uniform spanwise camber morphing, the optimal amount of TE displacement also increases with increasing lift coefficients. Therefore, the trend of increasing the overall camber with increasing C_{L_t} conforms to the expectations.

TABLE III
ANGLES OF ATTACK AND AERODYNAMIC COEFFICIENTS CORRESPONDING TO PREDICTED OPTIMAL WING SHAPES

$C_{\mathrm{L_t}}$ [-]	C_{D} [-]	$\frac{\mathrm{L}}{\mathrm{D}}$ [-]	α [deg]	C_{D} reduction [%]
0.35	0.02762	12.67	-2.50	19.8
0.50	0.04188	11.94	-1.05	17.4
0.65	0.06282	10.35	0.81	11.1
0.80	0.09023	8.87	2.97	6.5

As long as the morphing limits are not reached, both the best overall amount of camber and the ideal spanwise lift distribution, which is elliptical for induced drag reduction, can be achieved simultaneously. Comparing the camber distributions for $C_{L_t} = 0.35$ and $C_{L_t} = 0.50$, the largest difference between the two distributions is a flat increase of camber along the entire wing span. For the $C_{L_t} = 0.50$ distribution, the fourth and fifth servo units have already reached their upper limits. Consequently, for $C_{L_t} = 0.65$ and $C_{L_t} = 0.80$, no flat camber increases across the span are observed, but rather reshaping of the spanwise distribution occurs. This can be explained by the tradeoff between increasing the overall camber for achieving C_{L_t} and retaining an ideally shaped spanwise lift distribution when the morphing saturation limit is locally reached.

The lift and drag coefficients and the angles of attack of the predicted optimal shapes are given in Table III, which proves that over 6.5% of drag reduction has been successfully reached for all the tested target lift coefficients. It can be seen from Table III that higher C_D reductions are achieved for lower target lift coefficients. As mentioned before, the relatively highly cambered baseline wing shape (NACA6510) is naturally efficient at inducing higher lift coefficients and consequently, the wing can benefit more from active morphing for lower target lift coefficients. In addition, for a lower C_{L_t}, the associated drag is also lower, meaning that the relative error of the drag coefficient prediction becomes larger, and the maximum lift-to-drag ratio may become optimistic. Owing to the experimental time constraint, the measured data set is of a relatively small size but is adequate for demonstrating the effectiveness of the proposed optimization architecture. In the future, with more data, even higher-accuracy neural network models could be trained such that more accurate evaluations can be made on the highest achievable drag reductions.

E. Real-World Operational Considerations

For the application of the optimization framework to real-world commercial aircraft, no further changes to the optimization architecture are needed. The only required changes will be to the RBFNN training data and the sensors used.

Because gathering enough steady-state flight data to train a global model in every flight is impractical, some initial training data needs to be available. In the presented experiment, this data was gathered by measuring the forces of PR wing-shape and angle-of-attack combinations throughout the domain, in the wandering phase. In real-world operation, this data can be generated with high-fidelity simulations, and later on with measurements from previous flights. Note that this data is only used to maintain a reasonably accurate model in regions of the domain in which no measurements have yet

been gathered. The onboard RBFNN model is still actively updated as new measurements become available.

Secondly, the external force-balance as used in our experimental setup is essentially a group of strain gauges, which can also be realized on real-world aircraft structures. Fiber optics is also an alternative to strain gauges. Moreover, considering the whole aircraft free-flying operation, in the steady-state equilibrium conditions for which the optimization framework is proposed, the lift and drag forces balance the weight and thrust forces. These forces can be estimated using the gross aircraft weight, pitch angle, AOA, fuel consumption rate and the engine's thrust-specific fuel consumption (TSFC). The gross aircraft weight can be calculated from the aircraft's gross take-off weight and the fuel burned. These are known quantities on commercial airliners, as are fuel consumption rate and the angles of attack and pitch. The TSFC varies with flight speed and altitude, which are also known knowledge.

Finally, our experimental IR camera setup can be converted to a fuselage-mounted in-flight vision tracking system as has been experimentally validated in [48]. With these considerations and adaptations, the proposed optimization architecture can be realistically applied to real-world free-flying aircraft in the future.

VII. Conclusion

An online black-box shape optimization architecture for active distributed camber morphing wings has been proposed and experimentally validated. Compared with the unmorphed wing base shape, a drag reduction of 7.8% was achieved on the SmartX-Alpha demonstrator for a target lift coefficient of 0.65. For a wide range of target lift coefficients, the predicted drag reductions vary between 6.5% and 19.8%, with higher drag reductions being associated with lower lift coefficients.

The ability of the proposed architecture to realize the best wing shape for various lift coefficients online eliminates the need for model excitation maneuvers at every trim condition, as is characteristic for existing gray-box methods employing local models. By virtue of its black-box nature, this architecture can be applied to other morphing wing platforms with relative ease. For future applications to real-world commercial aircraft, the external force-balance and the vision system used in the experimental setup can all be made onboard. These features make the proposed optimization architecture both effective and practical for achieving sustainable aviation.

References

[1] T. A. Weisshaar, "Morphing aircraft systems: Historical perspectives and future challenges," *J. Aircr.*, vol. 50, no. 2, pp. 337–353, Mar. 2013.

[2] N. Nguyen et al., "Real-time adaptive drag minimization wind tunnel investigation of a flexible wing with variable camber continuous trailing edge flap system," in *AIAA Aviation Forum*. Dallas, TX, USA: AIAA, Jun. 2019, p. 3156.

[3] D. Lentink et al., "How swifts control their glide performance with morphing wings," *Nature*, vol. 446, no. 7139, pp. 1082–1085, Apr. 2007.

[4] M. Jacobsen, "Real time drag minimization using redundant control surfaces," *Aerosp. Sci. Technol.*, vol. 10, no. 7, pp. 574–580, Oct. 2006.

[5] Y. Ferrier, N. T. Nguyen, and E. Ting, "Real-time adaptive least-squares drag minimization for performance adaptive aeroelastic wing," in *Proc. 34th AIAA Appl. Aerodynamics Conf.*, Jun. 2016, p. 3567.

[6] N. T. Nguyen and J. Xiong, "Real-time drag optimization of aspect ratio 13.5 common research model with distributed flap system," in *AIAA Scitech Forum*. Virtual Event: AIAA, Jan. 2021, p. 69.

[7] J. Snyman, *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. New York, NY, USA: Springer, 2005, p. 24.

[8] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, Jun. 2011.

[9] W. Gong, A. Zhou, and Z. Cai, "A multioperator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 746–758, Oct. 2015.

[10] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 12, pp. 4371–4381, Dec. 2016.

[11] W. Zhang, F. Feng, J. Jin, and Q.-J. Zhang, "Parallel multiphysics optimization for microwave devices exploiting neural network surrogate," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 4, pp. 341–344, Apr. 2021.

[12] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 326–347, Jun. 2014.

[13] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 180–192, Apr. 2014.

[14] J. Luo, A. Gupta, Y.-S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates," *IEEE Trans. Cybern.*, vol. 49, no. 99, pp. 1708–1721, May 2019.

[15] C. S. K. Dash, A. K. Behera, S. Dehuri, and S.-B. Cho, "Radial basis function neural networks: A topical state-of-the-art survey," *Open Comput. Sci.*, vol. 6, no. 1, pp. 33–63, Jan. 2016.

[16] I. Bajaj, A. Arora, and M. M. F. Hasan, *Black-Box Optimization: Methods and Applications*. Cham, Switzerland: Springer, 2021, pp. 35–65.

[17] E. Iuliano and D. Quagliarella, "Efficient aerodynamic optimization of a very light jet aircraft using evolutionary algorithms and Reynolds-averaged Navier–Stokes flow models," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 225, no. 10, pp. 1109–1129, Oct. 2011.

[18] D. Keidel, G. Molinari, and P. Ermanni, "Aero-structural optimization and analysis of a camber-morphing flying wing: Structural and wind tunnel testing," *J. Intell. Mater. Syst. Struct.*, vol. 30, no. 6, pp. 908–923, Apr. 2019.

[19] X. Wang, T. Mkhoyan, I. Mkhoyan, and R. De Breuker, "Seamless active morphing wing simultaneous gust and maneuver load alleviation," *J. Guid., Control, Dyn.*, vol. 44, no. 9, pp. 1649–1662, Sep. 2021.

[20] M. Ruderman and L. Fridman, "Model-free sliding-mode-based detection and estimation of backlash in drives with single encoder," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 2, pp. 812–817, Mar. 2021.

[21] D. Papageorgiou, M. Blanke, H. H. Niemann, and J. H. Richter, "Robust backlash estimation for industrial drive-train systems—Theory and validation," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 5, pp. 1847–1861, Sep. 2019.

[22] P. Serra, R. Cunha, T. Hamel, C. Silvestre, and F. L. Bras, "Nonlinear image-based visual servo controller for the flare maneuver of fixed-wing aircraft using optical flow," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 2, pp. 570–583, Mar. 2015.

[23] Y. Xu, S. Yin, S. X. Ding, H. Luo, and Z. Zhao, "Performance degradation monitoring and recovery of vision-based control systems," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 6, pp. 2712–2719, Nov. 2021.

[24] M. M. Gomroki, F. Topputo, F. Bernelli-Zazzera, and O. Tekinalp, "Solving constrained optimal control problems using state-dependent factorization and Chebyshev polynomials," *J. Guid., Control, Dyn.*, vol. 41, no. 3, pp. 618–631, Mar. 2018.

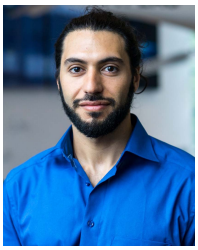
[25] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, Jun. 2001.

[26] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi, "Theoretical foundation for CMA-ES from information geometry perspective," *Algorithmica*, vol. 64, no. 4, pp. 698–716, Sep. 2012.

[27] Z. Li, Q. Zhang, X. Lin, and H.-L. Zhen, "Fast covariance matrix adaptation for large-scale black-box optimization," *IEEE Trans. Cybern.*, vol. 50, no. 5, pp. 2073–2083, May 2020.

[28] N. Hansen, A. S. P. Niederberger, L. Guzzella, and P. Koumoutsakos, "A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 180–197, Feb. 2009.

[29] P. Pošík, W. Huyer, and L. Pál, "A comparison of global search algorithms for continuous black box optimization," *Evol. Comput.*, vol. 20, no. 4, pp. 509–541, Dec. 2012.
[30] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, May 2015.
[31] M. Parenteau, É. Laurendeau, and G. Carrier, "Combined high-speed and high-lift wing aerodynamic optimization using a coupled VLM-2.5D RANS approach," *Aerosp. Sci. Technol.*, vol. 76, pp. 484–496, May 2018.
[32] B. Ghahremani, M. Bitaraf, and H. Rahami, "A fast-convergent approach for damage assessment using CMA-ES optimization algorithm and modal parameters," *J. Civil Struct. Health Monitor.*, vol. 10, no. 3, pp. 497–511, Apr. 2020.
[33] N. Hansen, *The CMA Evolution Strategy: A Comparing Review*. Berlin, Germany: Springer, 2006, pp. 75–102.
[34] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[35] T. Mkhoyan, C. C. De Visser, and R. De Breuker, "Adaptive real-time clustering method for dynamic visual tracking of very flexible wings," *J. Aerosp. Inf. Syst.*, vol. 18, no. 2, pp. 58–79, Feb. 2021.
[36] K. Kanatani, Y. Sugaya, and Y. Kanazawa, "Multiview triangulation," in *Guide to 3D Vision Computation*. Cham, Switzerland: Springer, 2016, pp. 133–147.
[37] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," in *Computer Vision—ACCV* (Lecture Notes in Computer Science), vol. 4843, Y. Yagi, S. B. Kang, I. S. Kweon, and H. Zha, Eds. Berlin, Germany: Springer, 2007, doi: 10.1007/978-3-540-76386-4_2.
[38] OptiTrack. *OptiTrack—Primex 41—in Depth*. Accessed: Aug. 15, 2021. [Online]. Available: https://optitrack.com/cameras/primex-41/
[39] B. Sun and E.-J. van Kampen, "Intelligent adaptive optimal control using incremental model-based global dual heuristic programming subject to partial observability," *Appl. Soft Comput.*, vol. 103, Feb. 2021, Art. no. 107153.
[40] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 1203–1218, May 2021.
[41] B. Sun, T. Mkhoyan, E.-J. Van Kampen, R. D. Breuker, and X. Wang, "Vision-based nonlinear incremental control for a morphing wing with mechanical imperfections," *IEEE Trans. Aerosp. Electron. Syst.*, early access, May 19, 2022, doi: 10.1109/TAES.2022.3175679.
[42] J.-W. van Wingerden et al., "Two-degree-of-freedom active vibration control of a prototyped 'Smart' rotor," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 2, pp. 284–296, Mar. 2011.
[43] multiSIM. (2021). *multiSIM: Distributed Simulation D-SIM*. [Online]. Available: https://multisim.nl/d-sim/
[44] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proc. 32nd AAAI Conf. Artif. Intell.* New Orleans, LA, USA: American Institute of Aeronautics and Astronautics, 2018, pp. 3302–3309.
[45] C. Runge, "Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten," *Zeitschrift für Mathematik und Physik*, vol. 46, pp. 224–243, Jan. 1901.
[46] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.
[47] S. J. Huntley, C. B. Allen, and B. K. Woods, "Computational analysis of the aerodynamics of camber morphing," in *AIAA Aviation Forum*. Dallas, TX, USA: AIAA, Jun. 2019, p. 2914.
[48] T. Mkhoyan, C. C. de Visser, and R. De Breuker, "Adaptive state estimation and real-time tracking of aeroelastic wings with augmented Kalman filter and kernelized correlation filter," in *AIAA Scitech Forum*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2021, p. 0666.



Tigran Mkhoyan (Member, IEEE) received the B.Sc. and M.Sc. degrees in advanced control and motion cueing for dynamic flight simulation from the Department of Control and Operations, Delft University of Technology, Delft, The Netherlands, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Aerospace Structures and Materials, Faculty of Aerospace Engineering, working on the smart-X project, an autonomous smart morphing wing.

He is determined to bridge the gap between aeroelasticity, design, and advanced control and push the advancements in aircraft technology toward a new generation of smart aircraft.

Oscar Ruland received the B.Sc. and M.Sc. degrees in aerospace engineering from the Delft University of Technology, Delft, The Netherlands, in 2018 and 2021, respectively. He has worked on online shape optimization for SmartX-Alpha as part of his M.Sc. thesis research.

From July 2019 to December 2019, he was an Intern at Lockheed Martin Aeronautics, Fort Worth, TX, USA. He is currently an Aerospace Engineer working at Netherlands Organisation for Applied Scientific Research (TNO), The Hague, The Netherlands. His research interests include simulation, optimization, and control.

Roeland De Breuker received the Ph.D. degree in aerospace engineering from the Delft University of Technology, Delft, The Netherlands, in 2011.

He joined the Faculty of Aerospace Engineering, Delft, as an Assistant Professor, in 2011, and became an Associate Professor in 2017. He has been a Visiting Researcher at Clarkson University, Potsdam, NY, USA, and a Visiting Professor at Airbus Innovations, Hamburg, Germany. He graduated 12 Ph.D. researchers and 69 M.Sc. researchers. He holds 32 journals and 58 conference papers and two patents.

Dr. De Breuker is the Associate Fellow of the American Institute of Aeronautics and Astronautics. He also serves as an Associate Editor for the *Journal of Fluids and Structures* and the *Journal of Intelligent Material Systems and Structures*.

Xuerui Wang (Member, IEEE) received the Ph.D. degree in aerospace engineering from the Delft University of Technology (TU Delft), Delft, The Netherlands, in 2019.

From May 2019 to May 2020, she was a Post-Doctoral Researcher with the Smart and Aeroelastic Structure Laboratory, TU Delft. Since May 2020, she has been an Assistant Professor with the Faculty of Aerospace Engineering, TU Delft. Her tenure track program is co-funded by the Department of Control and Operations and the Department of Aerospace Structures and Materials. Her research interests include nonlinear control, aeroelasticity, aerial robotics, and active morphing structures.