TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Woldhuis, T., Asaro, S., & Wang, X. (2025). Online Data-Driven Optimization of Aerodynamic Performance for an Unconventional Morphing Aircraft. *Journal of Guidance, Control, and Dynamics*, *49*(2), 492-506. https://doi.org/10.2514/1.G008985

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 49, No. 2, February 2026

Check for updates

Online Data-Driven Optimization of Aerodynamic Performance for an Unconventional Morphing Aircraft

Thymen Woldhuis,* Salvatore Asaro,† and Xuerui Wang‡
Delft University of Technology, 2629 HS Delft, The Netherlands

https://doi.org/10.2514/1.G008985

In nature, birds can intelligently adapt their wing shapes to their environment. This paper aims to replicate this capability by designing an online data-driven aerodynamic performance optimization framework for an unconventional morphing aircraft. Compared to state-of-the-art methods, the proposed framework can more efficiently search for optima with reduced computational load when addressing time-varying and nonlinear problems. It also demonstrates enhanced adaptability to unforeseen scenarios. In the event of a sudden actuator fault, the algorithm can automatically detect the fault, adapt the onboard data-driven model, and continue performing optimization and trimming tasks using the remaining healthy actuators. Additionally, the paper addresses the optimal number of actuators within a morphing surface, considering the tradeoff between aerodynamic optimization performance and weight penalty. High-fidelity simulations on a flying-wing aircraft platform demonstrate that through active morphing, the proposed framework achieves drag reductions of 1.9–4.9% during cruise and up to 12.6% at higher operational lift coefficients (due to heavier weight and lower speed), resulting in an overall drag reduction of 2.97% over a typical flight cycle, which corresponds to fuel savings of approximately 188.97 kg/h. This research represents a significant advancement in sustainable aviation, contributing to reduced fuel consumption, lower emissions, and improved fault tolerance for next-generation aircraft.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Nomenclature

BOBYQA	=	bound optimization by quadratic approximation
CMA-ES	=	covariance matrix adaptation evolution strategy
C_D	=	coefficient of drag
C_M	=	pitching moment coefficient
C_L	=	coefficient of lift
$C_{L_{\text{target}}}$	=	target lift coefficient
DIRECT	=	dividing rectangles
FTO	=	fault-tolerant optimization
F_{weight}	=	actuator weight
h	=	altitude
i	=	iteration index
J	=	cost
k_1	=	coefficient of lift deviation penalty
k_2	=	moment coefficient deviation penalty
MTOW	=	maximum takeoff weight
Mach	=	Mach number
m	=	number of actuators
opt	=	optimal
PYSWARM	=	particle swarm pattern search method
RANS	=	Reynolds-averaged Navier–Stokes
Re	=	Reynolds number
$\boldsymbol{s}$	=	spline control parameters
T	=	thrust
TRIC	=	translation-induced camber
U	=	airspeed
VCCTEF	=	variable-camber continuous trailing edge flap
VLM	=	vortex lattice method
α	=	angle of attack
β	=	low-pass filter coefficient
γ	=	flight path angle
$\boldsymbol{\delta}$	=	morphing angles
ε	=	fault threshold
ρ	=	density
σ	=	standard deviation
ψ	=	engine installation angle

Presented as Paper 2025-2804 at the AIAA SciTech Guidance, Navigation, and Control Conference, Orlando, FL, January 6–10, 2025; received 29 November 2024; accepted for publication 14 July 2025; published online 24 September 2025. Copyright © 2025 by Xuerui Wang, Delft University of Technology. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-3884 to initiate your request. See also AIAA Rights and Permissions https://aiaa.org/publications/publish-with-aiaa/rights-and-permissions/.

*Ph.D. Candidate, Faculty of Aerospace Engineering, Kluyverweg 1; t.woldhuis-1@tudelft.nl.

†Postdoctoral Researcher, Faculty of Aerospace Engineering, Kluyverweg 1; s.asaro@tudelft.nl.

‡Tenured Assistant Professor, Faculty of Aerospace Engineering, Kluyverweg 1; x.wang-6@tudelft.nl. Senior Member AIAA (Corresponding Author).

I. Introduction

IN MODERN aircraft design, offline optimization, where a fixed dataset or model is used to precompute an optimal solution, is typically performed to achieve optimal aerodynamic performance at a specific point within the flight envelope, known as the "design point." Extensive offline optimization using analytical models is then employed to determine the optimal aircraft configurations at various points in the flight envelope. These offline solutions are stored in predefined lookup tables [1], enabling real-time performance without repeated computational effort. However, when the aircraft operates under different flight conditions, the aerodynamic efficiency tends to degrade. Moreover, deviations from the nominal configuration, such as those arising from manufacturing variability, icing, and structural damage, can alter the actual aerodynamic characteristics from the ideal digital models, leading to suboptimal performance, increased fuel consumption, and emissions.

Online optimization, where optimization variables are dynamically updated in real-time using incoming sensor data, enables in-flight adjustments based on actual aerodynamic conditions, rather than relying on static offline models. This concept is inspired by nature, where birds continuously adjust their wing shapes to optimize performance in response to changing conditions. Applying this adaptive approach to aviation allows for significant performance improvements during flight, enhancing aerodynamic efficiency across a wider range of conditions. By dynamically responding to environmental changes, online optimization offers potential benefits such as improved fuel efficiency, reduced emissions, and overall enhanced aircraft performance, providing a more effective alternative to traditional optimization methods based on idealized, predefined models.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

With this motivation, a series of studies conducted by NASA Ames Research Center explored the online optimization of a variable camber continuous trailing edge flap (VCCTEF) system [2]. A real-time least-squares drag minimization method was developed, which proposed to use a recursive least-squares algorithm to estimate the aircraft's aerodynamic coefficients and fit them to a polynomial model [3]. The optimal wing shape was then determined using a gradient-based optimization method derived from the classical Newton–Raphson algorithm. This approach resulted in 8.4% drag reduction as compared to a clean wing for cruise [4]. Later wind tunnel experiments resulted in drag reductions of up to 9.4% for off-design lift coefficients [1].

While promising, these studies have several limitations. First, the use of a polynomial-based aircraft model imposes a fixed structure that is valid only near the trimming points, limiting its adaptability. Achieving detailed accuracy in local regions while preserving global validity would necessitate a substantial increase in the polynomial's degree. Second, the optimization problem for aircraft performance can be nonlinear and nonconvex. Gradient-based optimization strategies may converge to local minima or saddle points, exhibit sensitivity to initialization, and progress slowly in flat regions or near ill-conditioned points. Moreover, these methods struggle with nonsmooth functions and can involve costly gradient or Hessian calculations, particularly in large-scale problems.

To address these limitations, an alternative approach was proposed for the Smart-X morphing wing [5] and tested in a wind tunnel experiment [6], achieving up to 19.8% drag reduction. This study introduced the use of radial basis function neural networks to model the aircraft's aerodynamics in real time, followed by optimization using the covariance matrix adaptation evolution strategy (CMA-ES) [7]. While this approach improved the transition from local to global optimization, further enhancements in computational efficiency are necessary. Genetic optimization methods typically incur high computational costs, particularly as the number of variables increases. Nevertheless, they can be beneficial in problems with complex or poorly defined search spaces, which often arise in high-dimensional settings. Furthermore, although neural networks have proven effective in capturing aircraft dynamics, they operate as black-box models and impose significant computational demands.

Up to this point, we have identified several research gaps from an algorithm development perspective. It has become evident that an online and computationally efficient aerodynamic performance optimization framework is required. In addition to these algorithmic challenges, challenges also arise from a practical point of view.

The first challenge pertains to the hardware configuration on which the optimization is applied. The research conducted in [5,6] focused solely on wing optimization, neglecting challenges posed by free-flight conditions, such as maintaining trim and reserving control authority for stability augmentation and maneuverability. In contrast, the studies in [8,9] applied optimization to free-flying aircraft, but the configurations were conventional, i.e., tube-and-wing designs. Consequently, optimizing the wing's control surface settings in these studies had limited impact on mitigating the drag penalties caused by the large fuselage.

This research aims to develop an optimization framework for the Flying-V [10], an innovative wing–body aircraft concept developed by TU Delft. The design features a blended-wing–body (BWB) configuration in which the passenger cabin, fuel tanks, and cargo hold are integrated into a V-shaped flying wing. The primary goal of this configuration is to improve both aerodynamic and structural efficiency by reducing wetted surface area and induced drag. Comparative studies indicate that the Flying-V can achieve approximately 20% lower fuel consumption than the state-of-the-art twin-aisle aircraft (e.g., Airbus A350) for comparable mission profiles [10–12]. The feasibility and performance of the concept have been supported through multiple wind tunnel experiments and scaled free-flight tests [12–14]. However, this concept also faces several challenges. Due to the absence of conventional elevators and rudders, the control surfaces on the wing must perform multiple functions simultaneously, including three-axis trim, maneuverability, and performance optimization. Furthermore, Flying-V presents strong nonlinearity and nonconvexity at high angles of attack, requiring an optimization strategy capable of addressing such complexity.

The second practical challenge lies in the control surfaces used to implement the optimization algorithm. In recent years, there has been an increasing trend toward expanding the number of trailing-edge control surfaces. From a theoretical perspective, as the number of control surfaces increases, the degrees of freedom for precisely manipulating aerodynamic behavior also increase. This is why a distributed mini-flap configuration was proposed in [15]. However, this configuration does not ensure smooth transitions between adjacent flaps. In contrast, a seamless active morphing wing concept was introduced in [5]. Nevertheless, this research does not address the optimal number of actuators within the morphing surface area, considering the tradeoff between performance and the weight penalty of additional actuators.

Although increasing the number of actuators provides more control degrees of freedom, it also raises the likelihood of actuator faults in practice, as verified in the wind tunnel experiment in [1]. However, when an actuator fault occurred in that study, it was manually excluded from the control space. The algorithm lacked capabilities for automatic fault detection, fault tolerance, and onboard model adaptation, while still achieving optimal performance in postfault scenarios. These aspects are particularly crucial during real-world operations and will be addressed in this paper.

In addition, the performance of the optimization algorithm is highly dependent on the operational point and the aircraft's mass. While performance improvements of up to 9.4% and 19.8% were achieved in [1] and [6], respectively, these numbers may be overly optimistic when considering a typical flight cycle. This is because commercial aircraft spend the majority of their time in cruise conditions, where their shape is likely to be already near optimal. A more realistic assessment of optimization performance over a typical flight cycle will be conducted in this paper.

In light of the gaps in the literature, this paper proposes a fault-tolerant, online, and computationally efficient aerodynamic performance optimization framework and applies it to a nonlinear, free-flying, flying-wing morphing aircraft. The framework can automatically identify faulty actuators, adapt the onboard model, and achieve both trimming and performance optimization simultaneously using the remaining healthy actuators. It offers improved computational efficiency, enhanced performance, and better adaptability in handling time-varying, nonlinear, and nonconvex problems compared to state-of-the-art methods, as demonstrated by a quantitative comparative study in this paper. Moreover, the paper addresses the optimal number of actuators for a morphing control surface, considering the tradeoff between performance and weight penalty. Finally, high-fidelity simulations demonstrate the effectiveness of the proposed framework throughout the entire flight envelope and provide a more realistic estimation of drag reduction during a typical commercial aircraft flight cycle.

The rest of the paper is structured as follows: The detailed algorithm design, onboard model, and optimizer are explained in Sec. II. The simulation results are then presented in Sec. III, followed by conclusions in Sec. IV.

II. Methodology

This section presents the methodology used to model, simulate, and optimize the system. First, Sec. II.A outlines the overall optimization architecture. Section II.B delves into the development of the aerodynamic data sets, detailing how they are constructed and what flight conditions are considered. A data-driven onboard surrogate model is created in Sec. II.C. The onboard model will then be used to optimize the angle of attack and actuator inputs. Section II.D presents the optimizer design. Finally, Sec. II.E explores strategies for managing actuator faults, ensuring system robustness and continued operation under faulty conditions.

A. Optimization Architecture

The objective of the optimization is to find the best combination of the angle of attack (α) and actuation inputs ($\boldsymbol{\delta}$) that maximize

aerodynamic efficiency while satisfying the longitudinal trimming requirement. This should be achieved throughout the entire flight envelope, bridging the simulation-to-reality gap, adapting to new conditions, and tolerating unforeseen scenarios, including icing, actuator faults, and structural damage.

The optimization problem is formulated as follows:

$$\begin{aligned}
&\underset{\alpha,\boldsymbol{\delta}}{\text{minimize}}\ J(\alpha,\boldsymbol{\delta}) = C_D(\alpha,\boldsymbol{\delta}) + k_1\Big(C_{L,\text{target}}(\alpha,\boldsymbol{\delta}) - C_L(\alpha,\boldsymbol{\delta})\Big)^2 \\
&\qquad\qquad + k_2(C_M(\alpha,\boldsymbol{\delta}))^2 \\
&\text{subject to } \alpha \in [\alpha_{\min},\alpha_{\max}], \\
&\qquad \boldsymbol{\delta} \in [\boldsymbol{\delta}_{\min},\boldsymbol{\delta}_{\max}] \\
&\qquad \dot{\boldsymbol{\delta}} \in [\dot{\boldsymbol{\delta}}_{\min},\dot{\boldsymbol{\delta}}_{\max}] \\
&\qquad \boldsymbol{\delta} \in \left\{\boldsymbol{\delta} : [\boldsymbol{H},-\boldsymbol{H}]^\top\boldsymbol{\delta} \le \left[\bar{\boldsymbol{\delta}}_{\text{adj}}^\top,\bar{\boldsymbol{\delta}}_{\text{adj}}^\top\right]^\top\right\}
\end{aligned} \tag{1}$$

where C_L is the lift coefficient; C_M is the pitching moment coefficient that should be zero at trim; C_D is the drag coefficient to be minimized for reducing fuel consumption and emissions; and k_1 and k_2 are user-defined weights. This optimization problem is nonlinear and nonconvex; as will be shown in Sec. II.C, the aerodynamic coefficients are nonlinear and nonconvex functions of the optimization variables.

It is also possible to formulate the optimization problem by treating the conditions $C_{L,\text{target}}(\alpha,\boldsymbol{\delta}) - C_L(\alpha,\boldsymbol{\delta}) = 0$ and $C_M(\alpha,\boldsymbol{\delta}) = 0$ directly as equality constraints, in which case only $C_D(\alpha,\boldsymbol{\delta})$ remains in the cost function. Although the cost function $C_D(\alpha,\boldsymbol{\delta})$ becomes nearly convex under this formulation, the resulting optimization problem remains nonconvex because the two equality constraints are nonlinear and nonaffine functions of α and $\boldsymbol{\delta}$. Moreover, when the gains k_1 and k_2 are tuned sufficiently large, the proposed problem formulation can recover the one with explicit equality constraints.

The morphing surface is subject to position and velocity constraints of ± 20 deg and 45 deg/s, respectively [16]. In this context, "position" refers to the deflection angle of the aircraft's morphing surface, and "velocity" refers to the rate of change of this deflection (i.e., the morphing rate). Specifically, the deflection angle is defined by the angular displacement of a line connecting the hinge location to the trailing edge, capturing how many degrees this line rotates locally as a result of morphing. Both values are constrained due to physical actuator limits, with the angle measured in degrees and the rate in degrees per second.

The position constraint is further tightened to ± 16 deg in optimization to reserve margins for maneuvers. The tightening margin can also be designed to be flight-condition dependent (e.g., a larger margin for low speed and a smaller margin for high speed), and we propose that developing a dynamic, condition-aware margin allocation strategy could be a valuable direction for future work.

The morphing input $\boldsymbol{\delta}$ is a vector, indicated by the bold notation. The dimension of $\boldsymbol{\delta}$ depends on the number of actuators available to realize the morphing wing shape. We treat this dimension as a design variable, intending to answer the research question of what is the best number of actuators within the morphing surface. The tradeoffs involved in choosing it will be investigated in Sec. III.A. The morphing surface is composed of composite and polymeric elastic materials, which impose physical constraints on the relative positional differences between adjacent actuators [17]. Without loss of generality, let the number of actuators within the morphing surface be denoted as m, resulting in relative position constraints given by $|\delta_{i+1} - \delta_i| \le \bar{\delta}_{\text{adj},i}$, for $i = 1,2,\ldots,m-1$. This can be expressed in vector form as $[\boldsymbol{H},-\boldsymbol{H}]^\top\boldsymbol{\delta} \le [\bar{\boldsymbol{\delta}}_{\text{adj}}^\top,\bar{\boldsymbol{\delta}}_{\text{adj}}^\top]^\top$, where $\boldsymbol{H} \in \mathbb{R}^{(m-1)\times m}$ with $H_{i,i} = 1$ and $H_{i,i+1} = -1$ for $i = 1,2,\ldots,m-1$, and all other elements of $\boldsymbol{H}$ equal to zero [17]. Finally, the angle of attack α is constrained to $\alpha \in [0,10]$ deg for stall and pitch break protection.

The optimization focuses on steady, trimmed flight conditions over a longer time scale, as this has a more substantial impact on overall aerodynamic performance. While high-frequency dynamics can also affect drag and other aerodynamic properties, they generally occur over shorter durations and represent a relatively small portion of the overall flight envelope. The dynamic effects, including gusts and maneuvers, are thus considered shorter-term and can be decoupled from this optimization problem using low-pass filtering on the sensor measurements, based on the time-scale separation principle. In addition, outliers in the sensor measurements should be detected and removed from the optimization loop.

Perpendicular to the flight path, the trimming condition can be expressed as follows:

$$-T\sin(\alpha+\psi) - L + W\cos\gamma = 0 \tag{2}$$

where T, L, and W represent thrust, lift, and weight, respectively; ψ denotes the engine installation angle; and γ is the flight path angle. Assume that $T\sin(\alpha+\psi)$ is small compared to L, which is generally valid for commercial aircraft. For other aircraft with a higher thrust-to-weight ratio, this assumption can be removed by incorporating T into the optimization problem. Based on this assumption, the target lift coefficient, $C_{L_{\text{target}}}$, required to satisfy the trim condition, is calculated as follows:

$$C_{L_{\text{target}}} = \frac{2W\cos\gamma}{\rho U^2 S} \tag{3}$$

where $\rho U^2/2$ is the dynamic pressure and S is the wing surface area.

Figure 1 presents the optimization architecture in a schematic view. The proposed framework is designed to optimize aerodynamic performance across the entire flight envelope for any given combination of weight (W), Mach number (Mach), altitude (h), and flight path angle (γ). It is not constrained to specific values. That is why, in Fig. 1, these parameters are illustrated as arbitrary inputs. For validation purposes, the specific values simulated will be listed in Table 1. From any given W, Mach, h, and γ, the corresponding $C_{L_{\text{target}}}$ is calculated using Eq. (3). The computed $C_{L_{\text{target}}}$ acts as an

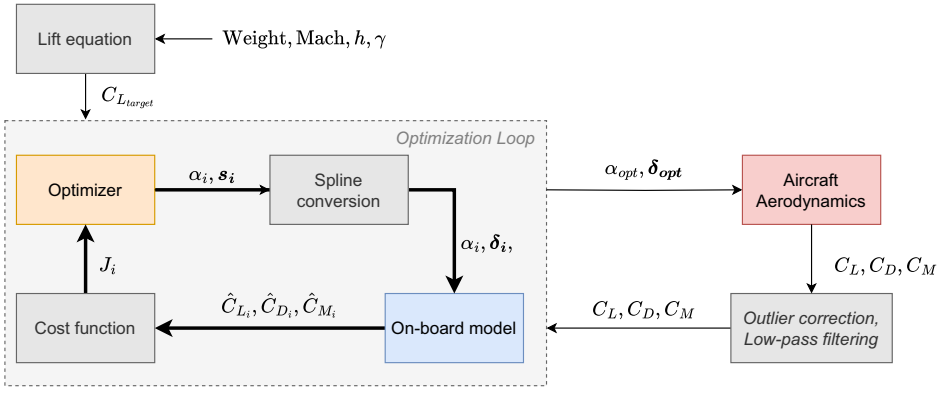


Fig. 1 Online shape optimization architecture.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

input of the optimization loop, in which an optimizer minimizes the cost function based on a surrogate onboard model. The choice of the optimizer will be detailed in Sec. II.D. To reduce computational loads of the optimizer and to ensure spanwise smoothness of the morphing surface, a spline function is created to map the real actuator motions to a virtual spline curve. Once an optimal combination of α_i and spline coefficient vector $\boldsymbol{s}_i$ are found, they are then converted back to the physical actuation angles $\boldsymbol{\delta}_i$ to be used by the onboard surrogate model. In this paper, the onboard surrogate model is trained by datasets generated by real-world wind tunnel experiments and simulations based on Reynolds-averaged Navier–Stokes (RANS) equations and the vortex lattice method (VLM). The aerodynamic coefficients estimated by the onboard model are then used by the cost function.

The optimization loop is run until the optimizer converges to an optimal α_{opt} and $\boldsymbol{\delta}_{\text{opt}}$, and then the suggested morphing configuration is tested on the real aircraft and the onboard model is updated online.

The proposed aerodynamic shape optimization framework can be readily integrated with higher-frequency control functions such as primary flight control, aeroelastic load control, and flutter control. From a physical perspective, the wing can morph around the optimized shape to perform maneuvers and other high-frequency control tasks. From a mathematical standpoint, the optimized wing shape and the corresponding morphing actuator configuration, denoted by $\boldsymbol{\delta}_{\text{opt}}$, serve as the new "zero" or baseline around which higher-frequency control functions operate.

For instance, in a 6-DoF control framework, dynamic stability and trajectory tracking are typically addressed using (nonlinear) optimal control methods based on deviations in states (or outputs) and control inputs, expressed as $\boldsymbol{x} - \boldsymbol{x}_{\text{ref}}$ and $\boldsymbol{u} - \boldsymbol{u}_{\text{ref}}$, where $\boldsymbol{u}_{\text{ref}}$ is usually zero. In our integrated approach, we propose setting $\boldsymbol{u}_{\text{ref}} = \boldsymbol{\delta}_{\text{opt}}$, effectively anchoring higher-frequency control around a performance-optimized morphing configuration.

Furthermore, it is also possible to design a unified cost function that simultaneously addresses aerodynamic performance, flight control, loads control, and flutter suppression. While such integration is feasible, we argue that the current time-scale separation approach is more computationally efficient. It enables slower update rates for morphing optimization, thereby reducing computational and communication demands.

In practical applications, the variables h, U, Mach number, ρ, and γ can be determined from measurements provided by the Pitot tube, air data computer, inertial navigation system, pressure altimeter, and Global Positioning System. The wing surface area S is typically a known constant. The gross aircraft weight W can be estimated onboard based on fuel consumption and the known gross takeoff weight, commonly recorded on commercial airliners. Additionally, W can be identified online by incrementally adjusting thrust and measuring the resultant accelerations. The aerodynamic coefficients C_L, C_D, and C_M can be estimated onboard using the three-axis force and moment dynamic equations, with linear and angular accelerations and rates obtained from the inertial measurement unit (IMU); gross weight is estimated or identified as previously discussed; thrust can be inferred from engine parameters, including measurements of fuel flow rate, rotational speeds, and inlet and exhaust pressures and temperatures. The variables α and $\boldsymbol{\delta}$ are directly measurable.

B. Aerodynamic Data Sets of the Flying-V

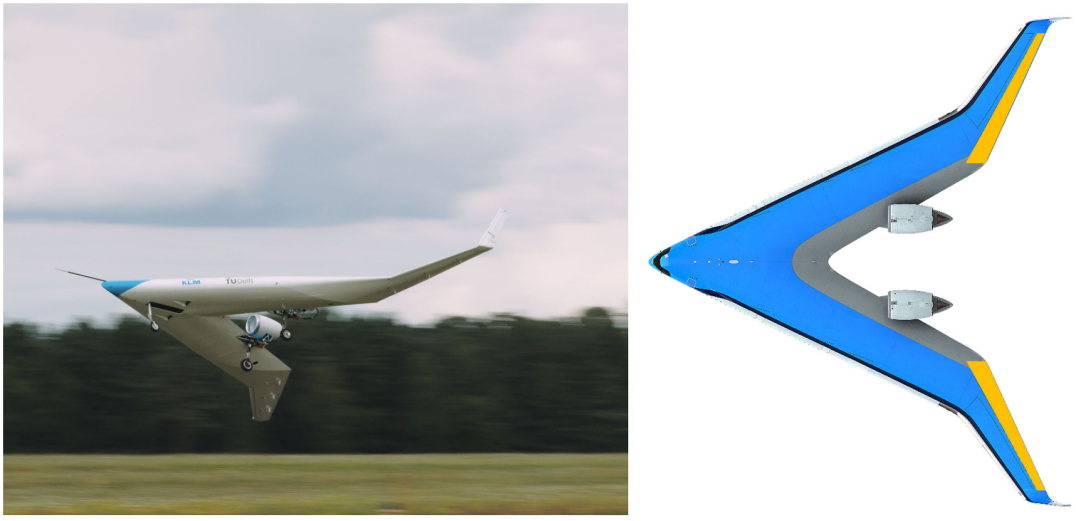
1. *The Flying-V Aircraft*

The Flying-V aircraft features an unconventional configuration in which the fuselage, wings, and passenger cabin are seamlessly integrated into a V-shaped BWB [12]. In the current design, the Flying-V is equipped with three discrete flaperons on each of the outboard wings. In this paper, we explore the aerodynamic benefits of replacing these discrete control surfaces with a continuous morphing surface. As shown in Fig. 2b, the area highlighted in yellow is designated as the morphing surface region. The morphing concept employed in this study is based on the seamless translation-induced camber (TRIC) mechanism, which has been experimentally validated in our prior work [6]. Because of its tailless configuration, the morphing surfaces are responsible for multiple objectives, including trimming, maneuver control, and online aerodynamic shape optimization.

The Flying-V presents nonlinear behavior at high angles of attack and has transonic cruise speeds, leading to nonlinear, nonconvex, and viscous behaviors. Therefore, high-fidelity simulations based on RANS and VLM are performed.

2. *High-Fidelity Dataset Based on RANS*

The RANS simulations are conducted with Ansys Fluent, and the grid is created with Integrated Computer Engineering and Manufacturing for Computational Fluid Dynamics (ICEM CFD). After a mesh independence study, the number of elements used to discretize the domain is 12.1×10^6, which leads to maximum differences with respect to the Richardson extrapolated value of less than 10% for C_L, C_D, and C_M. The mesh independence study is conducted at different α and Mach numbers. Further details on the meshing and simulation strategies are described in [18]. The simulations are performed in the range of Mach $= 0.2$ to Mach $= 0.7$ and $\alpha = 0°$ to $\alpha = 20°$. The RANS simulation was executed on the DelftBlue cluster, utilizing 24 cores per run. The DelftBlue cluster is TU Delft's high-performance computing infrastructure, designed to support large-scale scientific and engineering simulations through



a) Photo from one scaled flight testing [12] **b) Morphing area indicated in yellow**

Fig. 2 The Flying-V aircraft, with morphing surface area indicated in yellow (modified from [12]).

a tightly integrated architecture of over 17,000 CPU cores and GPU nodes, offering a reliable and scalable platform for advanced computational research across disciplines. In total, 63 data points were computed at 7 flight conditions with 9 different angles of attack. The simulations took 504 hours of runtime. This data is used as a basis for the zero-degree angle of attack coefficients and the coefficients with respect to α.

3. *VLM for Control Effectiveness*

To capture the aerodynamic mapping from actuation inputs to the coefficients, RANS simulations can also be performed. Nevertheless, this would be computationally inefficient because we treat the number of actuators as a variable in this paper, leading to a large number of possible compositions. To enhance computational efficiency, the contribution of the morphing surface is captured by VLM simulations. When properly calibrated, VLM can predict the change in forces and moment induced by the morphing surface for longitudinal trimming purposes for limited deflections as considered in this study. The VLM simulations are conducted with OpenVSP combined with VSPAERO [19]. A total of 24 elements are used to discretize the aircraft along the chord and 72 elements along the span.

For the VLM-based analysis, the number of data points depended on the number of morphing actuator sections. For each morphing configuration, we used four angles of attack and applied actuator deflections of $[-16°, -8°, 0°, 8°, 16°]$ per actuator. For example, with 18 actuators on the morphing surface, the total number of VLM simulation points was 94. These simulations were performed using OpenVSP combined with VSPAERO on a laptop equipped with an Intel Core i7-8750H processor (6 cores, 12 threads, base clock 2.20 GHz), 16 GB DDR4 RAM, and a 512 GB NVMe SSD, running Windows 11 with Python 3.9.18. Each VLM simulation took approximately 10 s to complete.

4. *Wind Tunnel Data Sets*

To increase fidelity, the VLM simulations are calibrated with wind tunnel experiments on a half-wing model of the Flying V at a freestream velocity of 25 m/s. Further details on the experiments can be found in [18]. The wind tunnel model is equipped with three control surfaces that span the same area as the morphing surface. Each control surface is tested individually, allowing to determine the impact of each of them on C_L, C_D, and C_M. The simulation data is then calibrated by multiplying the coefficients C_{L_δ}, $C_{D_{\delta^2}}$, and C_{M_δ} by the correction factors a, b, and c, as shown in the Appendix. A total of 75 wind tunnel data points were collected, each measured quasi-steadily over approximately 20 s to enable effective noise averaging. The construction of the aerodynamic data sets is summarized in Fig. 3.

5. *Spanwise Morphing Smoothness*

As mentioned above, the ideal number of actuators in the wing is set as a variable to be optimized and will be explored in Sec. III.A. To ensure that morphing along the span is smooth and to reduce the number of input variables for the optimizer, the virtual shape function can be used to map the real actuator deflections to the coefficients of virtual shape functions. The state-of-the-art practice is to adopt Chebyshev polynomials [20] as virtual shape functions [3,21]. However, the parameters used to tune the Chebyshev polynomial are coupled, and changing one parameter has a global influence on the shape of the polynomial. This hinders the state-of-the-art approach from precisely manipulating the aerodynamic properties at specific local locations. In view of this gap, this research adopts the univariate spline function as the base of virtual shape functions, which allows precise local manipulation and also ensures smooth transitions in between adjacent actuators. Figure 4 shows an example of actuator deflections governed by the univariate spline curve. A tradeoff was made regarding the number of spline control coordinates by increasing the number of control points and calculating the mean squared error (MSE) between 1000 random functions and their spline approximations. It was observed that after five control points, the improvement in approximation accuracy became negligible. Therefore, the number of spline parameters was set to five.

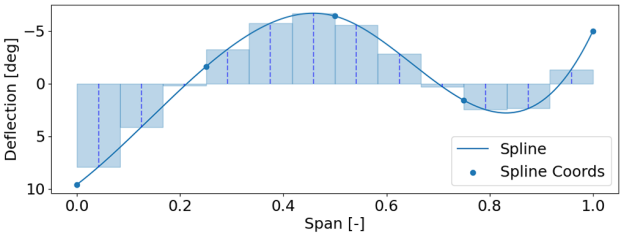


Fig. 4 Example of an univariate spline curve smoothly connecting 12 morphing actuators.

6. *Flight Conditions*

The flight conditions considered are given in Table 1 and are based on the available data from Ref. [18]. The flight path angle is estimated using an earlier study on the Flying-V [22]. The $C_{L_{\text{target}}}$ is calculated using Eq. (3) and the Flying-V parameters in Table 2. Essentially, Eq. (3) converts the flight condition to the trimming lift coefficient to be satisfied by the optimizer. More parameters of the Flying-V can be found in Ref. [12].

C. Onboard Data-Driven Model

1. *Model Structure*

To accelerate online computation, a data-driven global onboard surrogate model is learned from the aerodynamic datasets. Various approaches are available, including polynomial-based, spline-based, and neural network models [23], each offering distinct advantages. To determine the most suitable system identification model for this study, a thorough analysis of the requirements is essential. First, the model must be global in scope, capable of capturing nonlinear behavior. Second, it must be adaptive, allowing for updates as new data becomes available. Finally, it must be computationally efficient, with minimal evaluation costs.

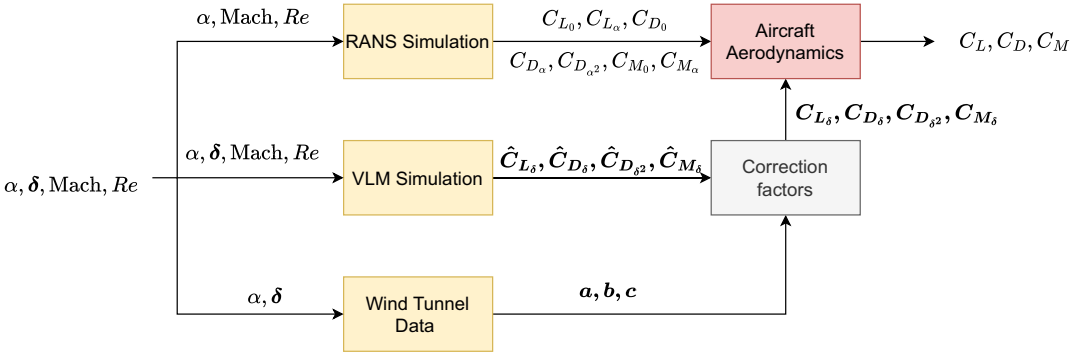


Fig. 3 The construction of the aerodynamic data sets of the Flying-V.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Table 1 Flight conditions considered for a typical Flying-V flight profile

Parameter	h, m	ρ, kg/m	U, m/s	Mach	Re (10e6)	γ, deg	Mass, kg	$C_{L_{\text{target}}}$
Takeoff	0	1.225	68.06	0.2	83.9	15.00	278000	**1.03**
Initial climb	3250	0.886	131.03	0.4	124	4.50	277539	**0.40**
Climb to cruise	6500	0.624	172.90	0.55	123	2.50	275816	**0.32**
Cruise[a]	9750	0.426	210.39	0.7	110	0.00	223500	**0.26**
Descent	6500	0.624	172.90	0.55	123	−2.50	172508	**0.20**
Approach	3250	0.886	131.03	0.4	124	−4.50	171357	**0.25**
Landing	0	1.225	68.06	0.2	83.9	−3.00	169000	**0.65**

[a]This Flying-V can cruise at higher Mach numbers. In this research, a Mach number of 0.7 is used because this is the highest Mach number at which accurate data is still available.
Note that $C_{L_{\text{target}}}$ is calculated using other values in the table based on Eq. (3).

Table 2 Key parameters of the Flying-V aircraft [12]

Variable	Value	Unit
Wing surface	898	m^2
Wing span	65	m
Planform length	55	m
Mean aerodynamic chord	18	m
Center of gravity	28.1	m
Maximum takeoff weight (MTOW)	278,000	kg
Fuel weight	109,000	kg

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Polynomial models are frequently employed for system identification due to their simplicity. However, their accuracy declines when applied to nonlinear systems. Additionally, polynomial models require data collected across a wide range of global variations to accurately represent the system. In contrast, neural networks are capable of modeling both global and complex behavior. Several studies have utilized neural networks to develop onboard models [24,25]. Despite this, neural networks are computationally demanding and often rely on global basis functions. Moreover, they lack transparency and are prone to numerical instabilities [26].

Multivariate splines, by contrast, have been shown to effectively generate a global aircraft model [23,27], and in the case of the Flying-V, under specific test conditions, they have even outperformed other identification methods [28]. Spline-based models are composed of piecewise functions, allowing different sections of the model to be fitted independently without interference. This characteristic is particularly advantageous for global models that require high accuracy in local regions, especially around trimming points.

The independent input parameters for the onboard model are angle of attack (α), control surface deflections ($\boldsymbol{\delta}$), and Mach number (Mach). Altitude is coupled with the Mach number, such that the appropriate altitude is selected based on Table 1 for a given Mach number. If all $\boldsymbol{\delta}$ values were used as variables in the multivariate spline, the model's complexity would increase rapidly. Therefore, the onboard model employs two separate methods. First, for each Mach number (Mach = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7), a cubic univariate spline is fitted for $C_L - \alpha$, $C_D - \alpha$, and $C_M - \alpha$. Linear interpolation is then used to estimate the splines for intermediate Mach numbers. To account for the influence of the actuators, estimates for C_{L_δ}, C_{D_δ}, $C_{D_{\delta^2}}$, and C_{M_δ} are generated based on the aerodynamic data. This strategy mirrors the approach used to generate the aerodynamic data of the onboard model; nevertheless, in a real-world scenario, the aerodynamic data can be directly obtained during flight.

A cubic spline is defined by a cubic polynomial on each interval $[x_i, x_{i+1}]$, such that

$$S(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i, \quad \text{for } x_i \le x < x_{i+1} \tag{4}$$

where a_i, b_i, c_i, d_i are the coefficients of the cubic polynomial on the ith interval. To ensure that the spline is smooth, it must be continuous under three conditions. Firstly, the spline function must be continuous at each knot to ensure that the function remains gap-free: $S(x_i^-) = S(x_i^+), \forall i$. Secondly, the first derivative of the spline must be continuous at each knot to ensure that there are no abrupt changes in slope: $S'(x_i^-) = S'(x_i^+), \forall i$. Lastly, the second derivative must be continuous at each knot to ensure smooth curve transitions: $S''(x_i^-) = S''(x_i^+), \forall i$.

2. *Excitation Approach*

To fit the onboard model, the system must be excited, for which several methods are discussed in the literature. In Ref. [3], the authors proposed applying bounded randomized inputs to the model, after which the onboard model is trained. In Ref. [1], the authors introduced the sweep excitation method, which deflects all control surfaces in an ordered sweeping motion, offering greater computational efficiency and time savings compared to the random excitation method. Additionally, in Ref. [21], the authors proposed a wandering phase method, which employs pseudo-random sample points using a Sobol sequence [29]. The principle behind all excitation methods is that they are performed once during a test flight, after which the model is fine-tuned and adapted during real flights. If the model remains accurate following the previous flight, further excitations may not be necessary. This study adopts the sweep method due to its computational efficiency, while acknowledging that any of the aforementioned methods could be suitable for the proposed optimization framework. Assuming a maximum actuator velocity of 45 deg/s [16], with the actuators starting and ending at the zero position, it would take 1.42 s per actuator to perform the sweep excitation method.

3. *Surrogate Model: Offline Fitting*

Using the sweep method, a dataset was generated from the aerodynamic model, to which the onboard model was fitted. Figure 5 illustrates the influence of Mach number and angle of attack on the aerodynamic coefficients. Markers in the figure indicate points from the dataset to which a spline is fitted. The moment coefficient is calculated using the center of gravity from Table 2. The nonlinear and nonconvex behavior is clearly visible in the $C_{M-\alpha}$ graph, where the unstable pitch break phenomenon appears at a high angle of attack. The pitch break phenomenon refers to a sudden and nonlinear change in an aircraft's pitching moment behavior, typically occurring at moderate to high angles of attack, where C_{M_α} transits from negative (stable) to positive (unstable). Furthermore, Fig. 5 also shows that the pitch break margin decreases as the Mach number increases. Nonlinearities also emerge in the lift and drag coefficients. Figure 6 depicts the influence of actuator deflections on the aircraft when the Mach number is 0.7. Upward deflection of the actuators decreases the lift coefficient but increases the moment coefficient, while downward deflection has the opposite effect. The drag coefficient increases symmetrically with both upward and downward deflection, as $C_{D_\delta} \approx 0$. It is noteworthy that, for clarity of presentation, only a subset of the data is shown in Figs. 5 and 6. The actual dataset used is considerably more extensive (referring to Sec. II.B) than what is displayed.

It is noteworthy that the surrogate model was not constructed through traditional "training" in the regression or machine learning sense. Instead, spline interpolation was applied to all reliable data

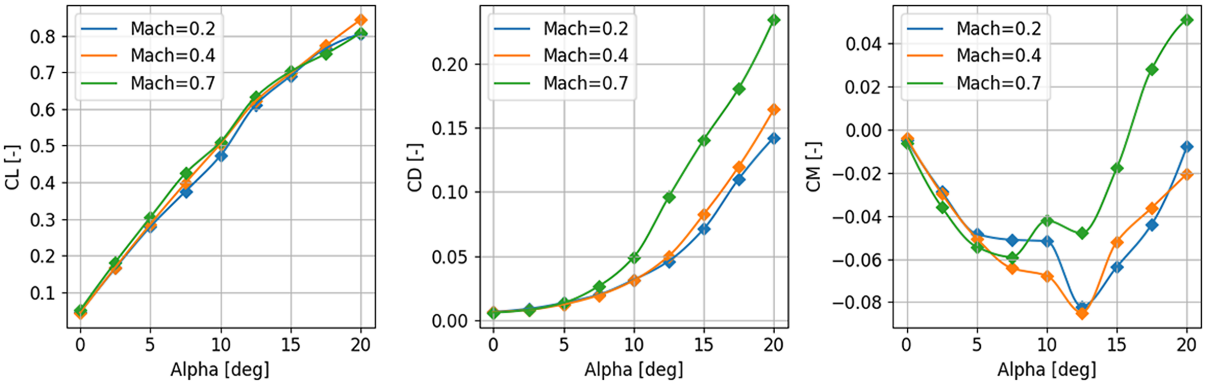


Fig. 5 Impacts of Mach number on lift, drag, and pitching moment coefficients.

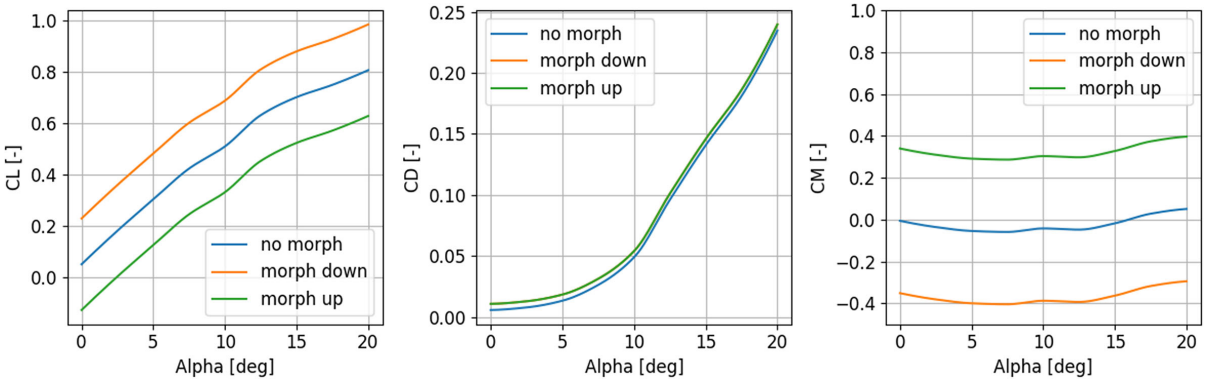


Fig. 6 Influence of actuator deflections on aerodynamic coefficients at $\delta = -16$ deg, $\delta = 0$ deg, and $\delta = 16$ deg.

points, with outliers removed during preprocessing. As a result, the model passes exactly through the available data, with zero fitting error at those locations, as can be visually confirmed in Fig. 5.

While concerns regarding the model's generalization capability are acknowledged, it should be noted that spline evaluation was strictly confined to the original parameter domain. No extrapolation was performed in the constrained optimization process, ensuring the surrogate's reliability within its defined bounds.

An 80/20 data split for validation can also be considered, wherein the spline would be fitted on the training subset and evaluated on the remainder using standard metrics such as mean-squared error (MSE), maximum absolute error (MaxAE), and the R^2 score. However, given the precious data points, the full dataset was retained for interpolation to avoid information loss. Nonetheless, the importance of quantitative validation is fully recognized, and more rigorous training-validation procedures are planned in future work.

In addition, since the surrogate model employs cubic univariate interpolating splines, which pass exactly through all trustworthy sampled data points (outlines have been removed in the preprocessing phase), the interpolation error at these points is identically zero, and no training or approximation is involved. The knot positions are fixed by the input data locations, and no adaptive placement or tuning is applied. Consequently, classical error bounds or convergence analysis, as well as sensitivity to knot placement, are not applicable in this context.

4. *Surrogate Model: Online Adaptation*

While Figs. 5 and 6 illustrate how the surrogate model trained on offline data behaves (where offline data refers to precollected, static datasets from simulations or experiments), it is also crucial that the onboard surrogate model can adapt to real-world scenarios online (during operations). This implies that trustworthy fresh data points should update the model, even if they sometimes contradict the existing model.

To facilitate this, a forgetting mechanism is incorporated into the onboard model. If a new data point is within a small distance of an existing one, the older point is discarded, and the spline is refitted. For the angle of attack, a small disturbance is defined as 0.1°, which corresponds to 0.5% of the considered domain (0°–20°). For the Mach number, a small disturbance is defined as 0.02 Mach, representing 4% of the considered domain (Mach 0.2–0.7).

This adaptation is instantaneous for each available, reliable data point. It does not need to wait for the collection of a data batch, as is typically required by neural networks or polynomial models. Figure 7 illustrates the effectiveness of updating the model from an initial configuration. Specifically, 40 new randomly generated data points are introduced to the surrogate model to simulate real-time measurement data that were not part of the original offline dataset. The aerodynamic coefficients, C_L, C_D, and C_M, are plotted against the angle of attack. As shown in Fig. 7, the proposed algorithm is capable of effectively adapting to these new data points online.

D. Optimizer

1. *Comparison of Optimization Methods*

To solve the problem formulated in Sec. II.A, an optimizer is designed. Optimizers can generally be classified as either gradient-based or derivative-free. In our application, the objective function is defined over simulated and experimental data, where analytical gradients are not available. Finite difference approximations are either computationally costly or sensitive to noise and discretization artifacts, especially in high-fidelity simulations. These limitations make gradient-based optimizers less suitable, motivating the choice of a derivative-free approach.

Another motivation for choosing a derivative-free optimizer is its potential to search for global optima. From a theoretical point of view, the Flying-V aerodynamic model is nonlinear and nonconvex. To confirm this, we first applied a local gradient-based optimization algorithm with random initial conditions, which indeed reveals various local optima. The nonlinear and nonconvex property of the optimization problem is further revealed in Fig. 8a. When there is only one optimization variable, namely the angle of attack, the

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

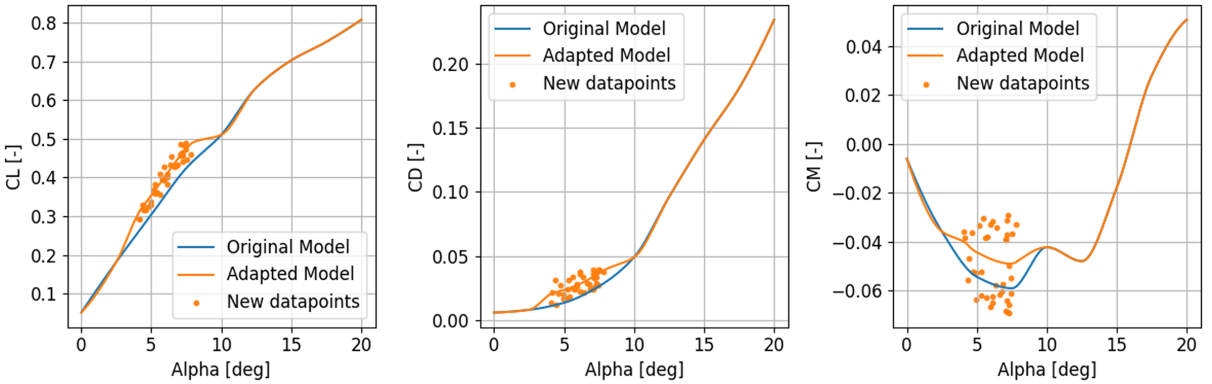
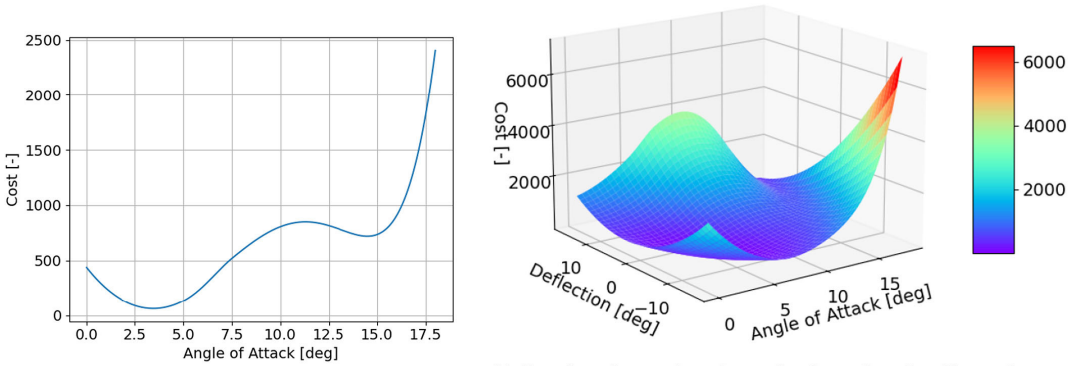


Fig. 7 Onboard model adaptation to 40 new data points at Mach = 0.7.



a) Cost function against the angle of attack

b) Cost function against the angle of attack and uniformed morphing

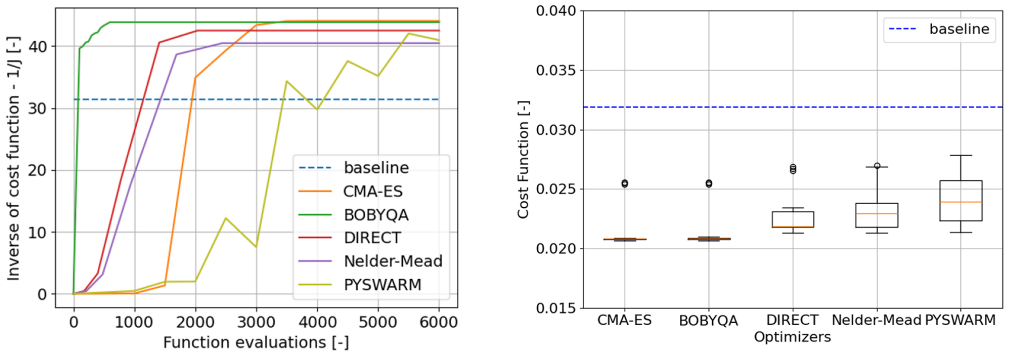
Fig. 8 Cost function behavior when one or two independent optimization variables present.

cost function already shows two local minima. Moreover, the cost function becomes more complicated when the uniformed morphing along the span is added as an additional optimization variable (Fig. 8b). The cost function is a nonlinear function of both α and the morphing input vector $\boldsymbol{\delta}$ and therefore forms a multidimensional surface. What is illustrated in Figs. 8a and 8b are merely 1D and 2D slices of this multivariable surface. When distributed morphing is present, the cost function is expected to become more complicated, preserving the nonlinear and nonconvex properties, motivating the choice of derivative-free optimization.

Numerous derivative-free optimization strategies exist, each with distinct strengths and weaknesses. Selecting the most appropriate depends on the specific characteristics and behavior of the cost function. As previously mentioned, the onboard model is nonconvex and, being spline-based, is a smooth function. In Ref. [30], the authors studied various optimization algorithms for nonconvex and smooth cost functions. Based on this study, the tradeoff will consider covariance matrix adaptation evolution strategy (CMA-ES) [7], bound optimization by quadratic approximation (BOBYQA) [31], dividing rectangles (DIRECT) [32], the particle swarm pattern search method (PYSWARM) [33], and the Nelder–Mead method as viable optimization methods.

The optimizers are compared based on performance, the number of required evaluations, and sensitivity. The results of this comparison are shown in Fig. 9a. CMA-ES and BOBYQA converge to the same optimum; however, BOBYQA achieves this with fewer required function evaluations. In contrast, DIRECT and the Nelder–Mead method converge to two different local optimal values. In addition, PYSWARM shows highly viable results and also presents a tendency of converging to a lower optimal value.

When updating the onboard model, more complex behavior emerges due to the potential for nonlinear and nonconvex behavior



a) Optimizer performance against the number of function evaluations

b) Variability in optimal cost for different optimizers when adapting the on-board model with new datapoints

Fig. 9 Comparison of various global optimizers in terms of performance, efficiency, and sensitivity.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

in local regions. To prevent significant deviations from the optimum, the optimizers must return results with low variability. By using an initial coarse onboard model and introducing random data points, the variability of the solutions can be analyzed. As illustrated in Fig. 9b, CMA-ES and BOBYQA performed similarly. Despite that, BOBYQA converges with much fewer function evaluations. Therefore, this study considers BOBYQA the most suitable optimizer for this application.

2. *BOBYQA Optimizer*

BOBYQA is a derivative-free optimization algorithm tailored for solving smooth, bound-constrained, nonlinear optimization problems. It belongs to the class of trust-region methods, which iteratively build local quadratic models of the objective function within a dynamically adjusted "trust region" around the current iterate. Theoretically, it has local convergence guarantees under the assumptions of 1) smooth objective function, 2) well-posed interpolation set, and 3) exact function evaluations [31]. Fundamentally, although BOBYQA is derivative-free, it does not possess inherent global optimization capabilities.

In this work, we adopt an enhanced version of BOBYQA, incorporating heuristic strategies inspired by Refs. [34,35]. These include adaptive stagnation detection, soft and hard restarts, and global-seeking mechanisms designed to improve the algorithm's robustness in nonconvex landscapes. While these modifications do not transform BOBYQA into a global optimizer in the theoretical sense, they significantly improve its practical ability to escape local minima, making it more effective in challenging nonconvex scenarios. This has been verified by results in Fig. 9b that the enhanced version of BOBYQA even outperforms global algorithms, including CMA-ES, DIRECT, and PYSWARM, for the problem considered in this paper.

The BOBYQA algorithm initializes using samples of the cost function, usually $n+2$ (where n is the number of function variables), and then approximates the objective function ($f(x)$) using a quadratic model ($Q(x)$), which can be written as

$$Q(x) = c + g^{\top}(x - x_0) + \frac{1}{2}(x - x_0)^{\top} H (x - x_0) \tag{5}$$

where x_0 is the current point (the center of the trust region); g is the gradient approximation; H is the Hessian approximation; and c is a constant term. The algorithm minimizes the quadratic model within a trust region defined by the radius Δ_k. The subproblem to solve is

$$\min Q(x), \quad \text{subject to} \quad \|x - x_k\| \leq \Delta_k \tag{6}$$

where Δ_k is the radius of the trust region, and x_k is the current best estimate. If the solution to the subproblem improves the objective function sufficiently, the trust region is expanded; otherwise, it is shrunk. After each iteration, the algorithm uses new function evaluations to update the quadratic model $Q(x)$ around the new current best estimate.

BOBYQA is particularly well-suited for problems involving noisy data, nonlinear smooth functions, bound constraints, and low to medium dimensionality. Using the numerical results from [31], it can be estimated that BOBYQA requires $O(n \cdot \log(n))$ function evaluations, where n is the number of optimization variables. Regarding computational efficiency, the paper reported computation times scaling as $O(n^2)$. Since the optimization problem in this study involves less than 10 independent variables, the time constraints imposed by this scaling are not expected to significantly impact performance.

E. Fault Tolerance of Actuator Faults

The aim of fault tolerance is to detect any possible actuator failure based on fresh data and adjust the remaining actuators so that the system remains trimmed and optimized for minimal drag.

Figure 10 shows how the system identifies and optimizes in the presence of an actuator failure. An evaluation function is designed to judge whether the aircraft is actually trimmed. If an unforeseen actuator failure presents, a mismatch between the onboard model and reality would appear, leading to a deviation from the real optimum. If the aircraft is not trimmed correctly, despite the optimizer still thinking the aircraft is in an optimum based on existing knowledge, the proposed algorithm would first identify the faulty actuator through a new model excitation sequence. If the actuator is stuck, then the identified C_{L_δ} would be approximately zero.

Once the system has identified the faulty actuator, the splines will be adjusted to incorporate this fixed point, ensuring that the faulty actuator remains in the same position in the model while ensuring spanwise smoothness. The algorithm will then determine the optimal positions for the remaining actuators, taking into account the position of the faulty actuator. Figure 11 illustrates how the splines adapt to accommodate the faulty actuator. The dashed line represents the faulty actuator location, and it is evident how the splines consistently pass through this fixed point while allowing the other actuators to remain variable.

It is worth noting that fault detection and diagnosis (FDD) is not the primary focus of this research. Therefore, a simple zero-gain check-based FDD approach is adopted. Nevertheless, more advanced FDD techniques, such as model-based residual analysis or machine learning-based classifiers, can be readily integrated into the proposed framework. Since the zero-gain check method can be affected by sensor noise, two measures are incorporated to tackle this issue: 1) temporal filtering (zero-gain conditions are only confirmed if the lack of response persists across a defined time window); 2) thresholding (a gain threshold above noise level is

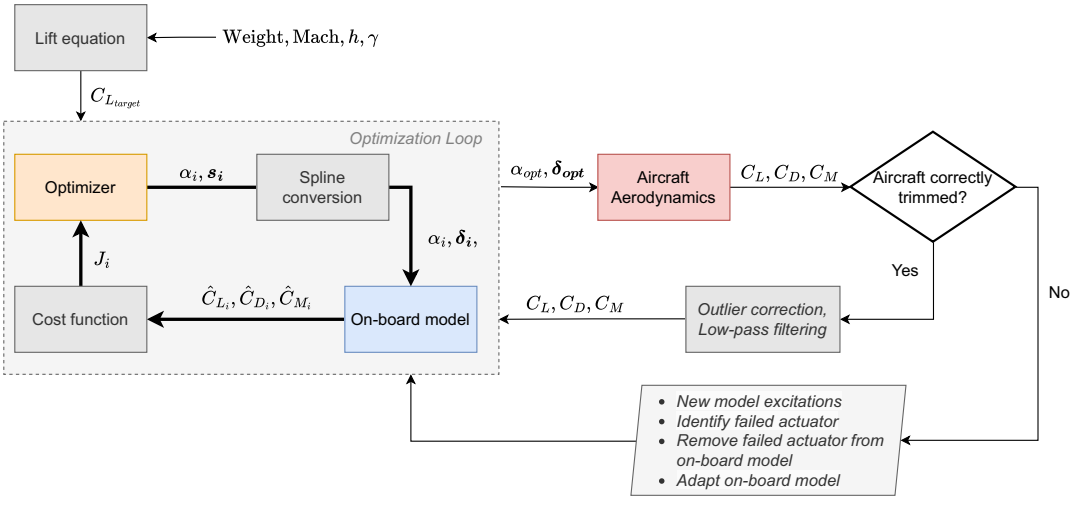


Fig. 10 The optimization architecture that can identify and tolerate actuator faults.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

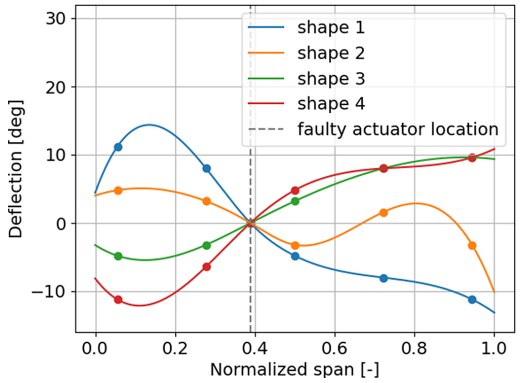


Fig. 11 Illustration of actuator deflections based on the spline curve accounting for a faulty actuator.

established, below which actuator behavior is flagged as suspect but not immediately declared faulty).

In this paper, the time window for temporal filtering is set to one and a half seconds. The threshold is defined as $\varepsilon = \lambda_f \cdot \sigma$, where σ is the standard deviation of the filtered signal (C_L, C_D, or C_M) in the nominal (healthy) case, and λ_f is a tunable constant. The standard deviation of the filtered (noisy) signal depends on sensor characteristics. For the purposes of this high-level fault-tolerant analysis, the noise is assumed to be normally distributed with a standard deviation of 1.0% of the aerodynamic coefficient (either C_L, C_D, or C_M) with a frequency of 1000 Hz. For context, a typical sweep procedure of one actuator takes 1.42 s. The following low-pass filter is then applied to the signal $y_t = (1-\beta)y_{t-1} + \beta x_t$, where y_t is the filtered output at time t, x_t is the raw input signal at time t, and $\beta \in (0,1)$ is the smoothing factor (filter coefficient). For a noise strength of 1.0% of the aerodynamic coefficient and a smoothing factor of 0.1, the standard deviation on the final signal of a sweeping maneuver of one actuator is 0.179%. The variable λ_f imposes a tradeoff between low false alarm rate and low detection latency. The choice of $\lambda_f = 4$ results in a false alarm rate of 5.80% and a detection latency of 112 ms.

III. Results

In this section, the effectiveness of the proposed optimization architecture is presented. Section III.A determines the optimal number of actuators considering the tradeoff between performance and weight penalty. Section III.B presents the optimization effectiveness on various flight conditions. Section III.C presents the fault tolerance results. Section III.D evaluates the overall performance improvement in one typical flight profile.

A. Ideal Number of Morphing Actuators

In conventional aircraft design, each wing typically has only one control surface, namely the aileron, responsible for roll control. However, as aircraft design evolves, we are seeing an increasing number of control surfaces on aircraft wings. The more control surfaces available, the greater the degrees of freedom for control, which allows for more precise manipulation of local aerodynamic properties and potential improvements in aerodynamic performance. Recently, Boeing has expressed interest in distributed trailing-edge control surfaces [15]. However, adding more control surfaces also results in a weight penalty. This subsection aims to quantify this tradeoff.

Typically, commercial aircraft such as the Airbus A350 use electro-hydraulic servo actuators (EHSA, 15–25 kg) for primary aileron actuation, while electro-hydrostatic actuators (EHA, 25–35 kg) or electric backup hydraulic actuators (EBHA, 20–30 kg) are employed for secondary flight controls or as redundant systems [36,37]. Therefore, a sensitivity study is conducted, considering an additional weight range of 10–40 kg per actuator, which includes both the actuator mass and the associated structural weight required to integrate the actuators.

The tradeoff associated with selecting the number of actuators is illustrated in Fig. 12 for three different flight phases and their corresponding target lift coefficients: climbing ($C_{L_{\text{target}}} = 0.4$), cruise ($C_{L_{\text{target}}} = 0.26$), and landing ($C_{L_{\text{target}}} = 0.65$). It can be observed that, regardless of the flight condition or phase, a consistent trend emerges: as the number of actuators increases, the drag coefficient initially decreases significantly, followed by a moderate increase.

Under cruise conditions, the optimal number of actuators is 9 when the individual actuator weight is 10 kg, and it decreases to 6 when the individual actuator weight is 20 or 40 kg. Based on the slope observed in Fig. 12, this optimal number can be relaxed to 5 without incurring a notable performance penalty. The climbing and landing phases exhibit similar behavior to that of the cruise phase. Generally, the required trimming lift coefficients for climbing and landing are higher, leading to correspondingly higher drag coefficients. In the climbing phase, the optimal number of actuators is 9 for all considered actuator weights. In the landing phase, the optimal number is 6 when the individual actuator weight is 40 kg and becomes 9 for 10 and 20 kg.

Overall, as the weight of each individual actuator increases, the optimal number of actuators tends to decrease or remain constant, while the associated drag coefficient increases. This behavior is attributed to the dependence of $C_{L_{\text{target}}}$ on aircraft weight [Eq. (3)]. Specifically, a heavier aircraft requires a higher angle of attack to achieve trim, which in turn results in increased lift-induced drag.

B. Optimization Performance and Computational Efficiency

This subsection presents the performance and computational efficiency of the proposed optimization architecture. First, at one specific cruise condition, three approaches are compared: 1) the baseline, which is the practical state-of-the-art that only uses one single flap/actuator on each wing; 2) the derivative-free local optimizer Nelder–Mead; 3) the derivative-free global optimizer CMA-ES; and 4) the proposed derivative-free trust-region optimization method BOBYQA. The last three approaches use a wing with

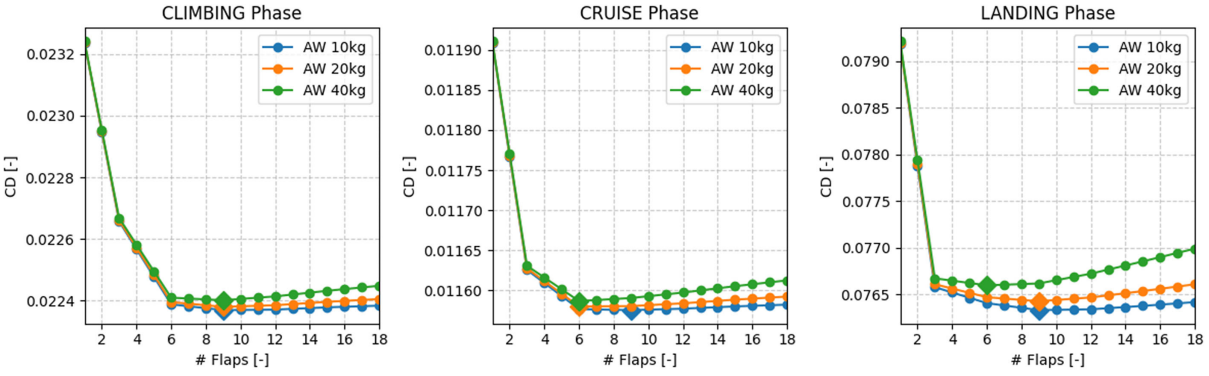


Fig. 12 Drag reduction effectiveness against the number of actuators (AW = actuator weight).

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

9 actuators, and an actuator weight of 10 kg. For a fair comparison, the flight condition and control surface area for the four methods are identical.

The onboard model is initially configured with a sweeping model excitation to get the aerodynamic coefficients at Mach = 0.7 and $h = 9750$ m. Then, the optimization is run on the onboard surrogate model. The number of iterations is limited to 1300 or until the minimal tolerance is met. Since the minimum tolerance is met in both cases, running the same experiment for more iterations will not change the final values. As seen from Fig. 13, Nelder–Mead, CMA-ES, and BOBYQA all present an improvement as compared to the baseline. BOBYQA demonstrates faster convergence and achieves a lower cost function value compared to Nelder–Mead. The Nelder–Mead method becomes trapped in a local minimum for a substantial number of iterations before escaping to another suboptimal solution. While CMA-ES ultimately converges to the same optimum as BOBYQA, it requires significantly more iterations and exhibits severe initial oscillations due to its stochastic sampling around the initial guess. In contrast, BOBYQA converges more efficiently, with only minor fluctuations during the early exploration of its trust-region approximations.

To assess the statistical robustness of the BOBYQA-based optimization and the reliability of the results, we conducted 100 independent optimization runs with randomized initial guesses x_0 and a fixed initial trust region size of 0.1, under constant operating conditions (Mach = 0.7, $C_{L_{\text{target}}} = 0.26$). As discussed in Sec. II.D, an enhanced version of BOBYQA was employed, incorporating heuristic strategies to improve the algorithm's robustness. Simulation results show that the final objective function values across the 100 runs had a mean of 0.0116 and a standard deviation of 8.84×10^{-8}, indicating low sensitivity to initialization. Furthermore, varying the initial trust region size from 10^{-5} to 1 does not affect the final optimization result, further verifying the algorithm's statistical robustness.

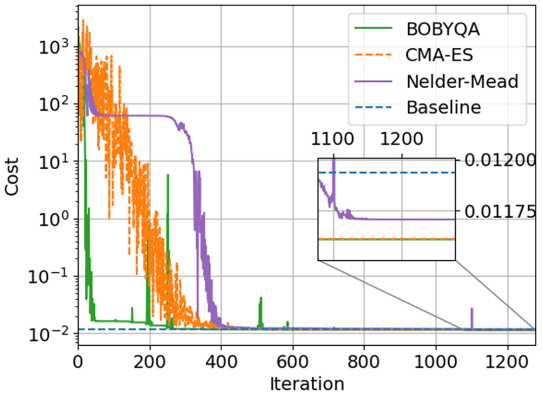
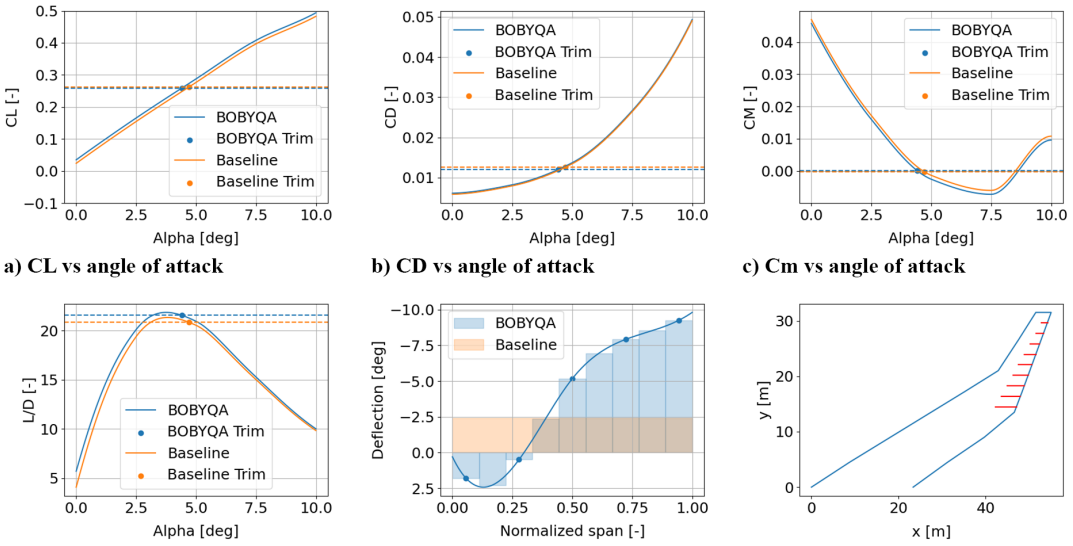


Fig. 13 Optimization performance comparisons at a cruise condition.

The comparison with the baseline is further detailed in Fig. 14. Since this baseline approach only has two inputs (α, and one single actuator) to satisfy both lift and moment trim, there is only one feasible solution. It can be seen that the optimized solution has a higher L/D than the simple trim solution while keeping both the moment and lift coefficients at the desired value. At the flight condition of $C_{L_{\text{target}}} = 0.26$ and Mach = 0.7, the proposed approach can increase the aerodynamic efficiency (L/D) by 2.98%. This is by virtue of the distributed actuators configuration that can modify the aerodynamic profile regionally. Specifically, the outboard actuators have a larger moment arm to the center of gravity while having a smaller influence on lift and drag. The optimizer can therefore use more of the outboard actuators to trim without overdeflecting the inner actuators, which minimizes the drag. In addition, having the inner actuators be able to deflect downward increases the lift and therefore lowers the angle of attack, which is also beneficial for reducing lift-induced drag.

After achieving a 2.98% of aerodynamic efficiency enhancement at $C_{L_{\text{target}}} = 0.26$ and Mach = 0.7, the effectiveness of the proposed approach on various flight conditions is then evaluated in Fig. 15. The target lift coefficient is an indication of flight conditions and is related to aircraft weight, fuel burn, and flight phases. It can be seen from Fig. 15a that the optimized solution outperforms the baseline at all the tested conditions. Moreover, the optimizer gains more as the target lift coefficient increases, reaching an increase 8.19% at $C_{L_{\text{target}}} = 0.4$. When going to even higher target lift coefficients, the L/D ratio increase even reaches 12.6%. The main reason for this is that at lower target lift coefficients the system is easier to trim with just one actuator because the ideal shape, as can be seen in Fig. 15b, is more similar to a horizontal straight line. Conversely, at higher target lift coefficients, a single-flap system loses excessive lift in its attempt to maintain a zero C_M, necessitating a higher angle of attack, which in turn increases drag.

It is noteworthy that morphing is a three-dimensional effect, whereas Fig. 15b presents a 2D visualization. Specifically, the curves in the figure represent the position of the trailing edge along the span, as if viewing the morphing wing from the rear. The x-axis denotes the normalized spanwise position, and the y-axis indicates the amount of trailing-edge morphing (with positive values meaning morphing downward).

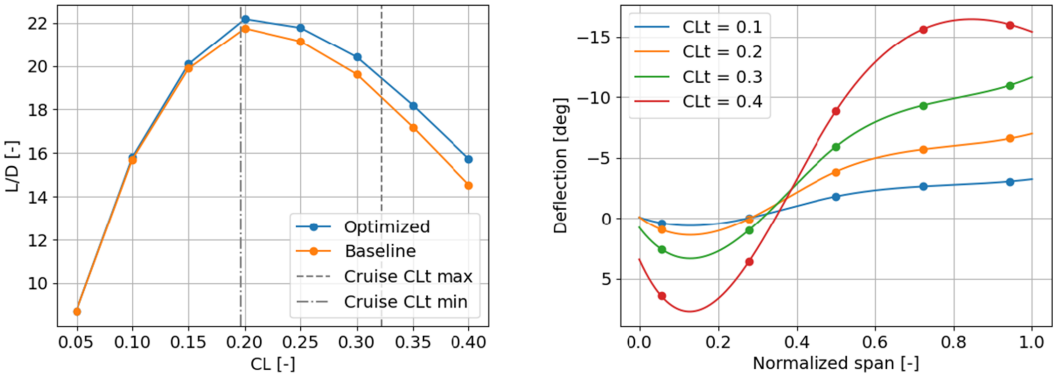


a) CL vs angle of attack **b) CD vs angle of attack** **c) Cm vs angle of attack**

d) L/D vs angle of attack **e) Deflections** **f) Morphing actuator locations**

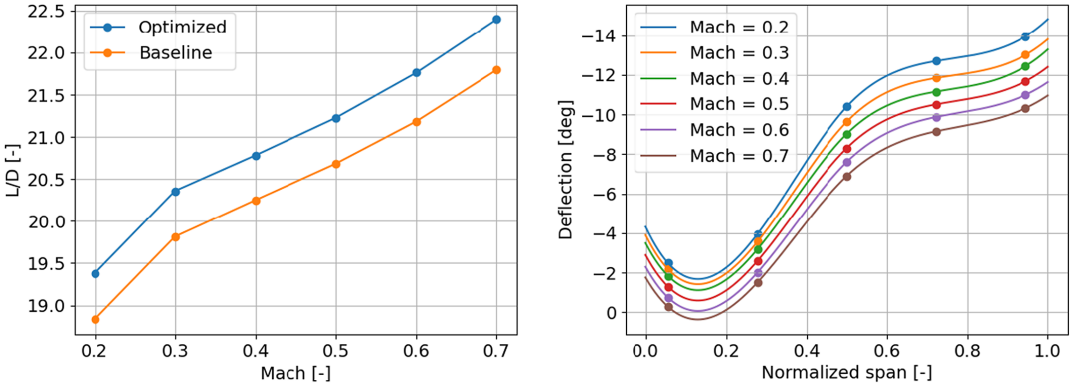
Fig. 14 Optimization results compared to the baseline for $C_{L_{\text{target}}} = 0.26$ and Mach = 0.7.

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985



a) Optimization for various $C_{L_{\text{target}}}$ during cruise b) Optimal shape for different $C_{L_{\text{target}}}$ during cruise

Fig. 15 Optimization performed for various target lift coefficients.



a) Optimization for different Mach numbers b) Optimal shape for different Mach numbers

Fig. 16 Optimization performed for various Mach numbers when $C_{L_{\text{target}}} = 0.26$.

Figure 16 illustrates the increase in L/D ratio and the corresponding actuator configurations for different Mach numbers. The L/D ratio increases at higher Mach numbers, even in the baseline case. The optimization benefits remain relatively consistent across the range of Mach numbers. It is important to note that under identical flight conditions, the $C_{L_{\text{target}}}$ increases at lower Mach numbers due to the reduction in speed. As shown in Fig. 15, a higher $C_{L_{\text{target}}}$ would lead to greater increases in the L/D ratio. However, to ensure a meaningful comparison across different Mach numbers, the $C_{L_{\text{target}}}$ is maintained at a constant value. Figure 16b indicates that the actuators require less upward morphing at higher Mach numbers because of the increase in control effectiveness.

The optimization computations were performed on a laptop equipped with an Intel Core i7-8750H processor (6 cores, 12 logical processors, base clock 2.20 GHz), 16 GB DDR4 RAM, and a 512 GB NVMe SSD. The system ran Microsoft Windows 11 with Python 3.9.18. Using this setup, the BOBYQA optimization converged in an average CPU time of 11.63 s and an average wall clock time of 11.79 s, as measured across multiple runs for different Mach numbers and lift coefficients as found in the typical flight profile.

C. Fault-Tolerant Optimization

In the presence of an unforeseen actuator fault, the system should still be able to optimize and trim the aircraft. This subsection presents the results of a case study where the system is optimized on the original onboard model while there is actually an unforeseen actuator fault present. The faulty actuator is simulated by setting the corresponding control effectiveness to zero.

Figure 17 compares the results with and without fault-tolerant optimization when $C_{L_{\text{target}}} = 0.26$, Mach $= 0.7$. It shows that because of the lack of knowledge of the fault, the spline curve for the system without fault tolerance does not go through the failed actuator's actual position, leading to an untrimmed and nonoptimal solution as highlighted in Fig. 17c. By contrast, the algorithm proposed in Sec. II.E is able to automatically identify which actuator is stuck as well as the jammed position, thus still fulfilling the optimization and trim requirements effectively.

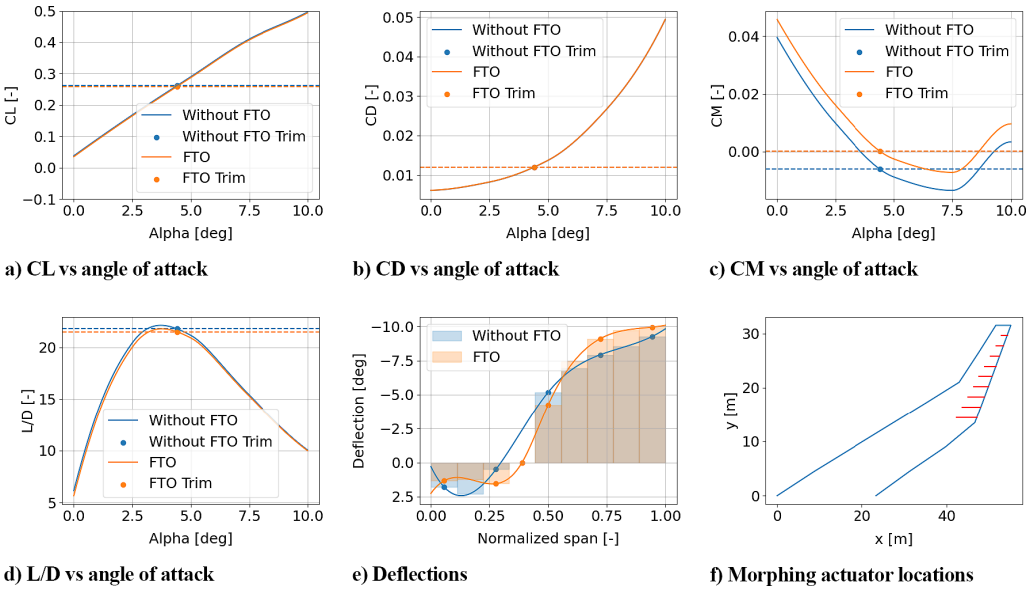
To further assess the robustness of the proposed fault-tolerant optimization approach, another case study is presented involving the simultaneous failure of two flaps when $C_{L_{\text{target}}} = 0.26$, Mach $= 0.7$. In this scenario, the fourth flap is jammed at $0°$, and the sixth flap is jammed at $-4°$. The results of this case are illustrated in Fig. 18. Similar to the single fault case, the faults are simulated by setting the corresponding control effectiveness values to zero while accounting for the additional aerodynamic effects of jamming at nonzero locations in the model.

Figure 18 demonstrates that the proposed algorithm is capable of automatically identifying both faulty actuators as well as accurately estimating their jammed positions. The optimization and trimming objectives are still successfully achieved by reconfiguring the remaining healthy actuators to compensate for the faults. This highlights the adaptability and resilience of the fault-tolerant system even under two simultaneous actuator failure conditions.

Up to this point, fault tolerance for one and two actuator failures has been demonstrated; these results can be generalized to scenarios involving a higher number of actuator failures. This is because, once an actuator failure is identified, the updated model excitation sequence includes all actuators, both functional and failed. The spline-based parameterization can adapt to account for the faulty actuators, while the optimizer incorporates multiple failure locations, respecting physical constraints such as adjacent actuator relative motion limits. This process leads to a new optimal solution that accommodates the fault scenarios while maintaining aircraft trim.

In the extreme case where all actuators fail, it becomes physically impossible to trim the BWB aircraft. However, as long as sufficient control degrees of freedom remain after trimming, the fault-tolerant

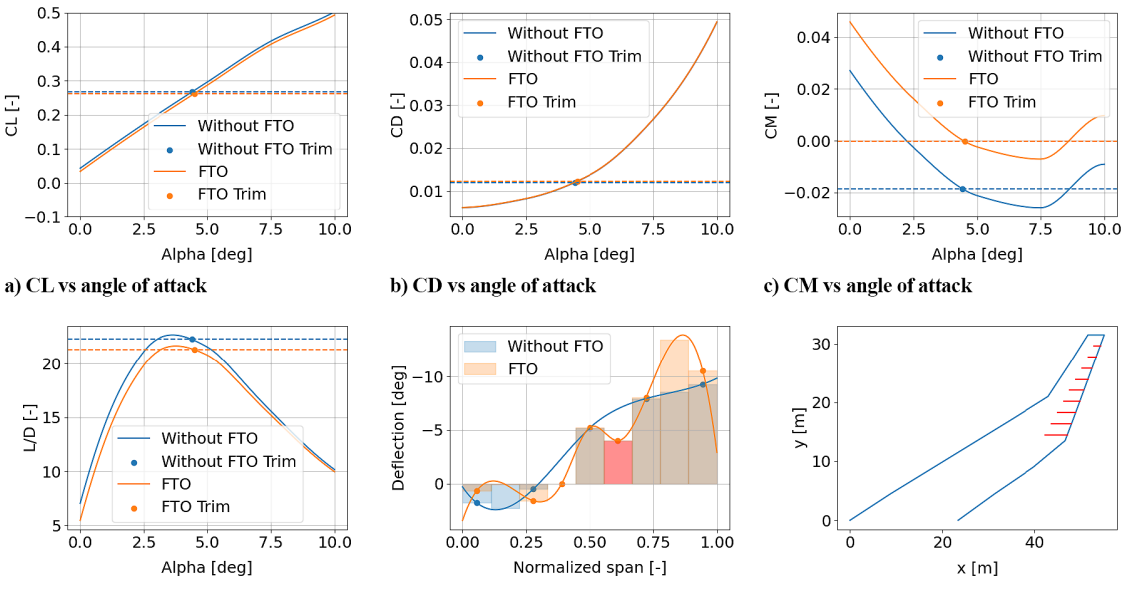
Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985



a) CL vs angle of attack b) CD vs angle of attack c) CM vs angle of attack

d) L/D vs angle of attack e) Deflections f) Morphing actuator locations

Fig. 17 Optimized results for fault-tolerant optimization (FTO) with a single actuator fault.



a) CL vs angle of attack b) CD vs angle of attack c) CM vs angle of attack

d) L/D vs angle of attack e) Deflections f) Morphing actuator locations

Fig. 18 Optimized results for fault-tolerant optimization (FTO) with two actuator faults.

optimization algorithm should, in principle, have the potential of identifying all failed actuators and providing an improved solution relative to the baseline configuration.

D. Effectiveness on a Typical Flight Cycle

In Sec. III.B, the effectiveness of the proposed method on various flight conditions has been demonstrated. Depending on the flight condition, the algorithm can achieve up to 12.6% of increase in aerodynamic efficiency (L/D). Nevertheless, during one typical flight cycle, an aircraft experiences different durations in different flight conditions. For example, for most of the time, the aircraft is cruising. This subsection aims to evaluate the effectiveness of the proposed method during one typical flight cycle.

Although a scaled model of the Flying-V has undergone flight testing, the full-scale version has not yet entered service, and therefore its real operational flight profile is not available. In practice, flight profiles are not only aircraft-dependent but also strongly influenced by air traffic control procedures and airline operations. Given that the Flying-V is designed as a BWB counterpart to the A350, we consider the A350's flight profile a reasonable reference. We do not expect significant deviations in the phase durations for similar long-range missions, especially for high-level conceptual analysis.

A typical flight profile is considered based on data provided in Ref. [38] for the Airbus A350-900. The results are presented in Table 3. The time per flight phase has been calculated using rate of climb/descent data from Ref. [38]. Without loss of generality, this research considers a 10-hour flight. The drag reduction is calculated for every flight phase, and a time-weighted average is calculated, resulting in a drag reduction of 2.97% in one typical flight cycle.

For jet aircraft, fuel consumption is proportional to the thrust required, so if drag decreases, thrust and fuel consumption decrease proportionally. While direct fuel consumption data is typically not disclosed by aircraft manufacturers, aircraft performance models (APMs) provide a reliable means to estimate fuel burn based on flight profiles. In this study, we employed the Poll–Schumann APM [39,40], as implemented in the `pycontrails` library [41], to compute fuel flow rates across all phases of a representative flight profile for the A350-900. Using data presented in Table 3, along with fuel flow rate estimates derived from the Poll–Schumann

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Table 3 Full flight savings for a typical flight profile with 10 hours of cruise (flight profile based on [38])

Parameter	Time, min	Est. flight fuel rate, kg/s	Mach	Est. $C_{L_{\text{target}}}$	C_D baseline	C_D opt	ΔC_D, %	Est. fuel savings, kg
Takeoff	1.00	3.875	0.2	1.03	0.1658	0.1654	0.24	0.6
Initial climb	1.67	3.101	0.4	0.40	0.0225	0.0217	3.61	11.2
Climb to cruise	9.97	2.687	0.55	0.32	0.0161	0.0155	3.35	53.8
Cruise	600.00	1.780	0.7	0.26	0.0124	0.0120	2.98	1908.7
Descent	10.57	0.216	0.55	0.20	0.0096	0.0094	2.13	2.9
Approach	6.67	0.686	0.4	0.25	0.0121	0.0118	2.49	6.8
Landing	1.00	0.683	0.2	0.65	0.0722	0.0675	6.44	2.6
Total	**630.88**	— —	— —	— —	— —	— —	**2.97**	**1987**

model, we estimate a total fuel savings of 1987 kg, of which 1909 kg is attributed to reductions during the cruise phase, and the remaining 79 kg to other phases of flight. On average, this corresponded to 188.97 kg/h of fuel saving over a typical flight cycle.

IV. Conclusions

This paper proposes a derivative-free online data-driven aerodynamic performance optimization framework, which can efficiently and smoothly account for nonlinearity and nonconvexity, adapt to changes onboard, and automatically detect and adapt to actuator faults.

Quantitative comparison results based on a free-flying Flying-V aircraft model confirm that the adopted data-driven BOBYQA algorithm outperforms several considered derivative-free optimization methods in this specific problem setting. Moreover, a spline-based onboard model effectively captures the nonlinearities of the Flying-V model and adapts to new data. Furthermore, the framework shows effectiveness in determining the optimal number of flaps considering the tradeoff between aerodynamic performance and weight penalty.

By optimizing the angle of attack and actuator positions, the proposed online optimization framework can achieve a drag reduction of 1.9% to 4.9% during cruise. At higher target lift coefficients (caused by heavier weight and lower speed), the drag reduction can reach up to 12.6%. This framework has shown effectiveness throughout the entire flight envelope. For a typical flight cycle of 10 hours, it can achieve a 2.97% drag reduction and it is approximately 188.97 kg/h of fuel. Last but not least, in the presence of unforeseen actuator faults, the proposed algorithm is able to automatically identify, adapt, trim, and optimize.

One limitation of this research is that high-lift devices used during takeoff and landing are still under development. In future work, these can be incorporated into the framework either as unknown optimization variables or as known inputs.

Another potential direction for future work is to validate the adaptability of the proposed framework to other unforeseen scenarios. The framework continuously learns the mapping from optimized variables to aerodynamic coefficients using fresh online data. In theory, other unmodeled variations, such as those caused by buckling or icing, can also be captured/identified during operation. As long as the resulting aerodynamic behavior remains within the controllable range of the morphing mechanism, the optimization algorithm can potentially adapt the aircraft shape accordingly, leveraging the updated mapping.

In conclusion, the proposed online aerodynamic performance optimization framework demonstrates significant drag reduction and fault-tolerance capabilities. This advancement represents a crucial step toward more sustainable aviation, contributing to reduced fuel consumption and lower emissions in future aircraft designs.

Appendix: Control Effectiveness Data Calibrations

Table A1 presents the coefficients in relation to the actuator deflection from the wind tunnel data and VLM data and presents the correction factors for each section. Wind tunnel data was gathered for a three-actuator layout [18], where the controllable surface is divided into three individual sections. Any actuator of the VLM simulations that falls within the geometric bounds of a wind tunnel actuator configuration is corrected in the same manner.

Table A1 Correction factors for control effectiveness

Coefficient	Wind tunnel	VLM	Correction factors
$C_{L_{\delta_1}}$	2.59e − 3	5.02e − 3	0.52
$C_{L_{\delta_2}}$	1.88e − 3	2.13e − 3	0.88
$C_{L_{\delta_3}}$	1.18e − 3	1.22e − 3	0.97
$C_{D_{\delta_1^2}}$	1.32e − 5	1.05e − 5	1.25
$C_{D_{\delta_2^2}}$	6.75e − 6	3.99e − 6	1.69
$C_{D_{\delta_3^2}}$	4.89e − 6	2.70e − 6	1.81
$C_{M_{\delta_1}}$	−2.80e − 3	−6.21e − 3	0.45
$C_{M_{\delta_2}}$	−2.41e − 3	−3.73e − 3	0.65
$C_{M_{\delta_3}}$	−1.63e − 3	−2.65e − 3	0.62

Acknowledgments

The authors would like to acknowledge Professor Roelof Vos for sharing the Flying-V aerodynamic model. The authors also would like to thank Professor Erik-Jan van Kampen and Professor Coen de Visser for their suggestions. This publication is part of the project Flying V Flight Control with project number 19511 of the research program Open Technology Programme, which is (partly) financed by the Dutch Research Council (NWO). The authors would like to acknowledge the in-kind contribution of KLM Royal Dutch Airlines and ADSE.

References

[1] Nguyen, N., Cramer, N., Hashemi, K., Ting, E., Drew, M., Wise, R., Boskovic, J., Precup, N., Mundt, T., and Livne, E., "Real-Time Adaptive Drag Minimization Wind Tunnel Investigation of a Flexible Wing with Variable Camber Continuous Trailing Edge Flap System," *AIAA Aviation 2019 Forum*, AIAA Paper 2019-3159, 2019. https://doi.org/10.2514/6.2019-3156

[2] Nguyen, N., "Elastically Shaped Future Air Vehicle Concept," NASA Innovative Partnerships Program Office TR ARC-E-DAA-TN3743, Oct. 2010, https://ntrs.nasa.gov/citations/20110023698.

[3] Ferrier, Y., Nguyen, N., and Ting, E., "Real-Time Adaptive Least-Squares Drag Minimization for Performance Adaptive Aeroelastic Wing," *34th AIAA Applied Aerodynamics Conference*, AIAA Paper 2016-3567, 2016. https://doi.org/10.2514/6.2016-3567

[4] Ting, E., Chaparro, D., Nguyen, N., and Fujiwara, G. E. C., "Optimization of Variable-Camber Continuous Trailing-Edge Flap Configuration for Drag Reduction," *Journal of Aircraft*, Vol. 55, No. 6, 2018, pp. 2217–2239. https://doi.org/10.2514/1.C034810

[5] Mkhoyan, T., Thakrar, N. R., Breuker, R. D., and Sodja, J., "Design and Development of a Seamless Smart Morphing Wing Using Distributed Trailing Edge Camber Morphing for Active Control," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0477, 2021. https://doi.org/10.2514/6.2021-0477

[6] Mkhoyan, T., Ruland, O., De Breuker, R., and Wang, X., "On-Line Black-Box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control," *IEEE Transactions on*

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985

Control Systems Technology, Vol. 31, No. 3, 2023, pp. 1063–1077. https://doi.org/10.1109/TCST.2022.3210164

[7] Hansen, N., and Ostermeier, A., "Completely Derandomized Self-Adaptation in Evolution Strategies," *Evolutionary Computation*, Vol. 9, No. 2, 2001, pp. 159–195. https://doi.org/10.1162/106365601750190398

[8] Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," *AIAA Applied Aerodynamics Conference*, AIAA Paper 2008-6919, 2008. https://doi.org/10.2514/6.2008-6919

[9] Nguyen, N., and Xiong, J., "Real-Time Drag Optimization of Aspect ratio 13.5 Common Research Model with Distributed Flap System," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0069, 2021. https://doi.org/10.2514/6.2021-0069

[10] Faggiano, F., Vos, R., Baan, M., and Van Dijk, R., "Aerodynamic Design of a Flying V Aircraft," *17th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-3589, 2017. https://doi.org/10.2514/6.2017-3589

[11] Benad, J., and Vos, R., "Design of a Flying v Subsonic Transport," *33rd Congress of the International Council of the Aeronautical Sciences*, AIAA Paper 2022-0358, 2022. https://doi.org/10.2514/6.2022-0358

[12] TU Delft, Faculty of Aerospace Engineering, "Flying-V Project," 2022, https://www.tudelft.nl/en/ae/flying-v [accessed 15 June 2025].

[13] Palermo, M., and Vos, R., "Experimental Aerodynamic Analysis of a 4.6%-Scale Flying-v Subsonic Transport," *AIAA Scitech 2020 Forum*, AIAA Paper 2020-2228, 2020. https://doi.org/10.2514/6.2020-2228

[14] Ruiz Garcia, A., Brown, M., Atherstone, D., Arnhem, N.v, and Vos, R., "Aerodynamic Model Identification of the Flying v from Sub-Scale Flight Test Data," *AIAA Scitech 2022 Forum*, AIAA Paper 2022-0713, 2022. https://doi.org/10.2514/6.2022-0713

[15] Xiong, J., Nguyen, N. T., and Bartels, R. E., "Numerical Simulation of An Aspect Ratio 13.5 Common Research Model with Trailing Edge Mini-Flaps," *AIAA Aviation 2023 Forum*, AIAA Paper 2023-4525, 2023. https://doi.org/10.2514/6.2023-4525

[16] Hanke, C. R., and Nordwall, D. R., "The Simulation of a Large Jet Transport Aircraft. Volume II: Modeling Data," NASA CR-114494, 1970, https://ntrs.nasa.gov/citations/19730001300.

[17] Wang, X., Mkhoyan, T., Mkhoyan, I., and De Breuker, R., "Seamless Active Morphing Wing Simultaneous Gust and Maneuver Load Alleviation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 9, 2021, pp. 1649–1662. https://doi.org/10.2514/1.G005870

[18] Asaro, S., and Vos, R., "Synthesis of the Aerodynamic Model of a Flying Wing Aircraft," *AIAA Scitech 2025 Forum*, AIAA Paper 2025-0852, 2025. https://doi.org/10.2514/6.2025-0852

[19] McDonald, R. A., and Gloudemans, J. R., "Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design," *AIAA Science and Technology Forum and Exposition, AIAA Scitech Forum*, AIAA Paper 2022-0004, 2022. https://doi.org/10.2514/6.2002-0004

[20] Gomroki, M. M., Topputo, F., Bernelli-Zazzera, F., and Tekinalp, O., "Solving Constrained Optimal Control Problems Using State-Dependent Factorization and Chebyshev Polynomials," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 3, 2018, pp. 618–631. https://doi.org/10.2514/1.G002392

[21] Ruland, O., Mkhoyan, T., De Breuker, R., and Wang, X., "Black-Box Online Aerodynamic Performance Optimization for a Seamless Wing with Distributed Morphing," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 3, 2023, pp. 560–570. https://doi.org/10.2514/1.G006573

[22] de Zoeten, G. J., Varriale, C., and Vos, R., "Flight Performance Evaluation of the Flying-V," *AIAA Aviation 2023 Forum*, AIAA Paper 2023-3484, 2023. https://doi.org/10.2514/6.2023-3484

[23] Tol, H. J., De Visser, C. C., Van Kampen, E., and Chu, Q. P., "Nonlinear Multivariate Spline-Based Control Allocation for High-Performance Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1840–1862. https://doi.org/10.2514/1.G000065

[24] Calise, A. J., and Rysdyk, R. T., "Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Control Systems*, Vol. 18, No. 6, 1998, pp. 14–25. https://doi.org/10.1109/37.736008

[25] Calise, A. J., Lee, S., and Sharma, M., "Development of a Reconfigurable Flight Control Law for Tailless Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 5, 2001, pp. 896–902. https://doi.org/10.2514/2.4825

[26] Babuška, R., and Verbruggen, H., "Neuro-Fuzzy Methods for Nonlinear System Identification," *Annual Reviews in Control*, Vol. 27, No. 1, 2003, pp. 73–85. https://doi.org/10.1016/S1367-5788(03)00009-9

[27] Sun, L. G., de Visser, C. C., Chu, Q. P., and Mulder, J. A., "Online Aerodynamic Model Identification Using a Recursive Sequential Method for Multivariate Splines," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1278–1288. https://doi.org/10.2514/1.60375

[28] Ruiz-García, A., Vos, R., and de Visser, C. C., "Aerodynamic Model Identification of the Flying V from Wind Tunnel Data," *AIAA Aviation 2020 Forum*, AIAA Paper 2020-2739, 2020. https://doi.org/10.2514/6.2020-2739

[29] Sobol, I., "On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals," *USSR Computational Mathematics and Mathematical Physics*, Vol. 7, No. 4, 1967, pp. 86–112. https://doi.org/10.1016/0041-5553(67)90144-9

[30] Rios, L. M., and Sahinidis, N. V., "Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations," *Journal of Global Optimization*, Vol. 56, No. 3, 2013, pp. 1247–1293. https://doi.org/10.1007/s10898-012-9951-y

[31] Powell, M. J. D., "The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives," Univ. of Cambridge, Dept. of Applied Mathematics and Theoretical Physics TR DAMTP 2009/NA06, Jan. 2009, https://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

[32] Jones, D. R., Perttunen, C. D., and Stuckman, B. E., "Lipschitzian Optimization Without the Lipschitz Constant," *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, 1993, pp. 157–181. https://doi.org/10.1007/BF00941892

[33] Vaz, A. I. F., and Vicente, L. N., "A Particle Swarm Pattern Search Method for Bound Constrained Global Optimization," *Journal of Global Optimization*, Vol. 39, No. 2, 2007, pp. 197–219. https://doi.org/10.1007/s10898-007-9133-5

[34] Cartis, C., Roberts, L., and Sheridan-Methven, O., "Escaping Local Minima with Local Derivative-Free Methods: A Numerical Investigation," *Optimization*, Vol. 71, No. 8, 2022, pp. 2343–2373. https://doi.org/10.1080/02331934.2021.1883015

[35] Cartis, C., Fiala, J., Marteau, B., and Roberts, L., "Improving the Flexibility and Robustness of Model-Based Derivative-Free Optimization Solvers," *ACM Transactions on Mathematical Software*, Vol. 45, No. 3, 2019, pp. 32–41. https://doi.org/10.1145/3338517

[36] Xiong, J., Bartels, R. E., and Nguyen, N., "Aerodynamic Optimization of Mach 0.745 Transonic Truss-Braced Wing Aircraft with Variable-Camber Continuous Trailing-Edge Flap," *AIAA Scitech 2021 Forum*, AIAA Paper 2021-0337, 2021. https://doi.org/10.2514/6.2021-0337

[37] Haitao, Q., Yongling, F., Xiaoye, Q., and Yan, L., "Architecture Optimization of More Electric Aircraft Actuation System," *Chinese Journal of Aeronautics*, Vol. 24, No. 4, 2011, pp. 506–513. https://doi.org/10.1016/S1000-9361(11)60058-7

[38] EUROCONTROL, "Aircraft Performance Database: Airbus A350-900 (ICAO: A359)," 2018, https://contentzone.eurocontrol.int/aircraftperformance/details.aspx?ICAO=A359 [accessed18 Sept. 2024].

[39] Poll, D. I. A., and Schumann, U., "An Estimation Method for the Fuel Burn and Other Performance Characteristics of Civil Transport Aircraft During Cruise: Part 2, Determining the Aircraft's Characteristic Parameters," *Aeronautical Journal*, Vol. 125, No. 1284, 2021, pp. 296–340. https://doi.org/10.1017/aer.2020.124

[40] Poll, D. I. A., and Schumann, U., "An Estimation Method for the Fuel Burn and Other Performance Characteristics of Civil Transport Aircraft in the Cruise. Part 1 fundamental Quantities and Governing Relations for a General Atmosphere," *Aeronautical Journal*, Vol. 125, No. 1284, 2021, pp. 257–295. https://doi.org/10.1017/aer.2020.62

[41] Shapiro, M., Engberg, Z., Teoh, R., Stettler, M., Dean, T., and Abbott, T., "Pycontrails: Python Library for Modeling Aviation Climate Impacts (v0.54.10)," 2025. https://doi.org/10.5281/zenodo.15426480

V. Mukhopadhyay
Associate Editor

Downloaded by Technische Universiteit Delft on March 3, 2026 | http://arc.aiaa.org | DOI: 10.2514/1.G008985