

This is a preprint of the following article, which is available from mdolab.engin.umich.edu

Gray, A. C., Riso, C., Jonsson, E., Martins, J. R. R. A., and Cesnik, C. E. S., High-fidelity Aerostructural Optimization with a Geometrically Nonlinear Flutter Constraint, *AIAA Journal*, April 2023.

The original article may differ from this preprint and is available at

<https://doi.org/10.2514/1.J062127>.

High-fidelity Aerostructural Optimization with a Geometrically Nonlinear Flutter Constraint

Alasdair C. Gray^{1,*}, Cristina Riso², Eiríkur Jonsson¹,
Joaquim R. R. A. Martins¹, and Carlos E. S. Cesnik¹

¹*University of Michigan, Ann Arbor, MI, 48109*

²*Georgia Institute of Technology, Atlanta, GA, 30332*

Abstract

When designing aircraft, avoiding dynamic aeroelastic instabilities such as flutter is a key requirement. One way to meet this requirement is to use a multidisciplinary design optimization subject to a flutter constraint. Flutter-constrained design optimizations have used geometrically linear detailed models, which do not accurately predict flutter for very flexible aircraft, or geometrically nonlinear low-order models, which do not accurately trade off cruise range and structural mass. This paper presents a framework for integrating a geometrically nonlinear, subsonic flutter constraint that captures the large in-flight deflections of very flexible aircraft into high-fidelity gradient-based aerostructural optimization. The cruise range and stress constraints are computed accurately with detailed aerostructural analyses, which use a built-up finite element model coupled to RANS CFD. The detailed model is condensed to a low-order aeroelastic model to compute the geometrically nonlinear flutter constraint and its adjoint derivatives with computational cost and robustness suitable for optimization. The framework is demonstrated by maximizing the cruise range of a subsonic high-aspect-ratio wing with respect to panel thicknesses, sweep, and span. The impact of the geometrically nonlinear flutter constraint highly depends on the in-flight deformation level. At low deflection levels, sweeping the wing backward is the most effective flutter prevention method, and the flutter constraint has little impact on the achievable cruise range. At high deflection levels, shortening the wing span is necessary to suppress flutter, reducing the achievable cruise range by up to 5.9%. This work is a step toward making multidisciplinary design optimization a practical tool for designing the energy-efficient, very flexible aircraft of the future, which require geometrically nonlinear flutter analyses early in the design cycle to prevent flutter.

1 Introduction

Aircraft designs are evolving toward increasingly lightweight structures and higher-aspect-ratio wings for higher energy efficiency [1]. For these flexible wings, flutter is a critical design concern that must be considered early in the design process to avoid late redesign. One approach to addressing this problem is to design the wing using multidisciplinary design optimization (MDO)

¹Corresponding author: alachris@umich.edu

with a flutter constraint [2]. Many previous efforts have included flutter constraints in aircraft design optimization [2], with a recent thrust in gradient-based methods to tackle problems with computationally expensive function evaluations and many design variables. Despite the progress in this area, there are still significant gaps.

Flutter-constrained optimization only produces a truly flutter-free design if the underlying flutter analysis captures the relevant physics that dictate the flutter boundary and key instability mechanisms. In the aircraft industry, flutter analyses are typically conducted about a single reference shape that is independent of the flight condition. The structural and aerodynamic characteristics of the vehicle come from a linear built-up finite element model (FEM) of the airframe coupled to potential flow doublet-lattice method (DLM) lifting surfaces. For transonic aircraft, the DLM model is corrected with data from high-fidelity computational fluid dynamics (CFD), experiments, or both, to account for nonlinear effects due to shock waves and flow separation. However, wing deflections are assumed to be small enough that the aircraft’s structural characteristics can be described using the natural frequencies and mode shapes of the undeformed structure.

This geometrically linear approach is inadequate for flutter analysis of aircraft with very flexible high-aspect-ratio wings, which exhibit large in-flight deflections under operating loads within the flight envelope. These potentially large deflections cause changes in the aircraft’s structural and aerodynamic characteristics, which impact the flutter boundary and can lead to new instability mechanisms [3–6]. Thus, aircraft with very flexible wings require a geometrically nonlinear flutter analysis process that accounts for the impact of wing in-flight deflections and their dependency on the flight condition [7]. Such a geometrically nonlinear process consists of (1) computing the aircraft’s nonlinear static equilibrium at each flight condition of interest, (2) linearizing the equations of motion about each equilibrium state, (3) extracting the aeroelastic eigenvalues of the local linearized systems, and (4) using the obtained aeroelastic eigenvalue to compute the flutter boundary. Flutter-constrained optimization of very flexible wings must use this geometrically nonlinear analysis approach to accurately capture critical constraint values and produce a flutter-free design [2]. A constraint that captures such geometrically nonlinear effects is hereafter referred to as *geometrically nonlinear flutter constraint*, that is, a constraint where the underlying flutter analysis is about the statically deformed configuration for each flight condition.

To date, optimizations that considered geometrically nonlinear flutter constraints used low-order models to evaluate flutter and all other functions of interest. Various researchers performed aerostructural optimizations using geometrically nonlinear beam models coupled to potential flow thin airfoil or panel-based aerodynamics [8–12]. While beam models are suitable for modeling the global structural dynamics of slender wings, they cannot accurately predict localized stress distributions in the wingbox. Similarly, while potential flow aerodynamic models are advantageous for flutter analysis of subsonic wings, they miss viscous effects required to accurately predict cruise drag.

Most aerostructural optimization problems involve trading off structural weight, peak stress levels, and cruise drag, thus requiring models that can accurately predict these quantities. These accurate predictions can only be achieved using detailed structural FEMs coupled to aerodynamic models that capture viscous and compressible flow effects, such as RANS CFD [13, 14]. Additionally, RANS CFD can remove spurious multimodality in the design space that can appear when using lower fidelity inviscid methods [15].

To address this issue, other efforts have included flutter constraints in optimizations where cruise and maneuver performance, or flutter margin, are computed with high-fidelity models. He et al. [16, 17] demonstrated aerodynamic shape optimization using a time-spectral Reynolds-averaged Navier–Stokes (RANS) CFD flutter constraint. Jacobson and Stanford [18] performed aerostructural optimization with a flutter constraint computed using linearized frequency-domain CFD. Jonsson et

al. [19, 20] developed a robust flutter solver based on the DLM with efficient adjoint derivatives. They used it to compute a flutter constraint in large-scale optimizations of a transport aircraft model where static cruise and maneuver flight conditions were analyzed using RANS CFD [20]. However, these efforts considered geometrically linear structures and analyzed flutter considering the undeformed aircraft shape, which does not accurately capture the flutter boundary of aircraft with very flexible wings [7]. Lupp and Cesnik [21, 22] prototyped a framework to integrate geometrically nonlinear flutter constraints into a high-fidelity aerostructural optimization and used surrogate models in place of high-fidelity analyses to demonstrate the process.

Although rare, examples of geometrically nonlinear flutter analyses based on built-up FEMs exist in the literature. Cestino et al. [23] and Ritter et al. [24] analyzed the flutter of very flexible wings by solving for the geometrically nonlinear static aeroelastic equilibrium at multiple flight conditions, extracting the mode shapes and frequencies of the pre-loaded built-up (detailed) FEM, and using them to conduct a linearized flutter analysis based on nonplanar DLM. Other researchers have developed methods for producing geometrically nonlinear reduced order models of large built-up FEMs and coupling them to nonplanar DLM to compute flutter boundaries of very flexible wings [25, 26]. However, these efforts did not compute flutter derivatives nor perform optimization.

This paper presents a framework for integrating a geometrically nonlinear, subsonic flutter constraint that accounts for wing in-flight deflections into high-fidelity gradient-based aerostructural optimization. Like the DLM-based work of Jonsson et al. [19, 20], the framework centers around a detailed aerostructural model composed of a built-up FEM coupled to RANS CFD. This detailed model is used to perform high-fidelity static aerostructural analyses to compute cruise and maneuver performance metrics, such as range and peak stress levels, which require a detailed representation of the wing structure and aerodynamics. Like the approach prototyped by Lupp and Cesnik [22], the geometrically nonlinear flutter constraint is evaluated by condensing the built-up FEM at the current iteration to a low-order geometrically exact beam model, coupled to potential flow, unsteady thin airfoil theory. The resulting low-order aeroelastic model captures the impact of geometrically nonlinear effects due to wing in-flight deflections on the flutter boundary with computational effort and robustness adequate for optimization. By integrating models of different complexity within the optimization, the present framework can evaluate a geometrically nonlinear flutter constraint while also accurately predicting other quantities of interest, such as cruise range and peak stress levels, that are best evaluated by a detailed model.

The geometrically nonlinear flutter constraint is the aggregate of damping values over aeroelastic modes and flight conditions. These damping values are computed by geometrically nonlinear aeroelastic eigenvalue analyses conducted using the low-order model about the wing’s statically deformed shape at each flight condition. This formulation is suited to gradient-based optimization because it does not require the flutter boundary and is robust to mode switches and hump modes expected in practical design optimizations [2]. To compute derivatives, the flutter constraint is differentiated using the adjoint method to enable large-scale optimization with large numbers of structural sizing and geometric design variables.

This framework was previously demonstrated for pure structural optimization under fixed loads, minimizing structural mass subject to stress and flutter constraints [27]. The new contribution of this work consists of including fully coupled static aerostructural analyses for the cruise range and stress calculations, using a consistent geometric parameterization across the high- and low-detail representations of the wing, and implementing the adjoint derivatives of the flutter constraint with respect to geometric design variables.

The capabilities introduced in this work are demonstrated by conducting the first high-fidelity, gradient-based aerostructural optimizations of a subsonic wing with respect to structural sizing (panel thicknesses) and geometric (sweep and span) variables, subject to the geometrically nonlinear

flutter constraint evaluated at various wing root angles of attack. These optimizations demonstrate how in-flight deflections affect the optimized, very flexible wing design due to their impact on the flutter onset point and mechanism.

The remainder of the paper is organized as follows: Section 2 summarizes the geometrically nonlinear flutter constraint and its derivatives; Section 3 describes the computational framework that implements the methodology; Section 4 presents the test case and the optimization problem; Section 5 discusses the results; and a section of concluding remarks ends the paper.

2 Geometrically Nonlinear Flutter Constraint and Derivatives

The methodology to include a geometrically nonlinear flutter constraint into a high-fidelity gradient-based aerostructural optimization is presented here for an isolated wing with no loss in generality. The optimization integrates two wing models: (1) a detailed model composed of built-up FEM coupled to RANS CFD providing the cruise range and the stress constraints; and (2) a low-order model composed of a beam representation of the built-up FEM coupled to potential flow unsteady thin airfoil theory providing the geometrically nonlinear flutter constraint. Integrating such models enables accurate prediction of drag and peak stress levels in realistic wings while computing the geometrically nonlinear flutter constraint with accuracy, computational effort, and robustness suitable for optimization.

The detailed and low-order models are linked by a consistent geometric parameterization and by a computationally inexpensive structural model order reduction that extracts equivalent beam properties from the built-up FEM at each optimization step. The geometric parameterization and the model order reduction are differentiated to efficiently compute the total derivatives of the flutter constraint using the adjoint method.

Because this paper focuses on subsonic flutter boundaries, the flutter analysis neglects transonic effects. The potential flow model used in the flutter analysis could be replaced with a low-order transonic aerodynamic model [28] if appropriately parameterized to handle geometric updates and the related derivatives.

2.1 Model Condensation and Flutter Analysis

Figure 1 shows the process for evaluating the geometrically nonlinear flutter constraint, starting from a built-up FEM of the wing. The blue modules were added or extended in this work to optimize with geometric design variables. This section contains a high-level description of the analysis process. More detailed descriptions can be found in previous work [22, 27, 29, 30]

The process begins with the design variables, listed in the vector $\mathbf{x} \in \mathbb{R}^{N_x}$, which can be separated into structural sizing variables, $\mathbf{x}_{\text{struct}}$, and the newly added geometric variables, \mathbf{x}_{geo} . These design variables define the high-fidelity built-up FEM of the wingbox, whose nodal coordinates are referred to as \mathbf{p} .

The flutter constraint is computed on a low-order beam representation of the wing that follows the geometrically exact, strain-based formulation of Su and Cesnik [31]. In this formulation, the beam model is defined by its inertia and stiffness distributions associated with a user-specified reference axis close to the shear axis. The inertia distributions consist of constitutive properties that relate the beam generalized forces to the corresponding generalized velocities; the equivalent stiffness distributions consist of constitutive properties that relate the beam generalized forces to the generalized strains. The inertia and stiffness distributions are derived from the built-up FEM at each optimization using a previously developed model order reduction technique [29, 30]. They are referred to as *equivalent beam distributions* in the following.

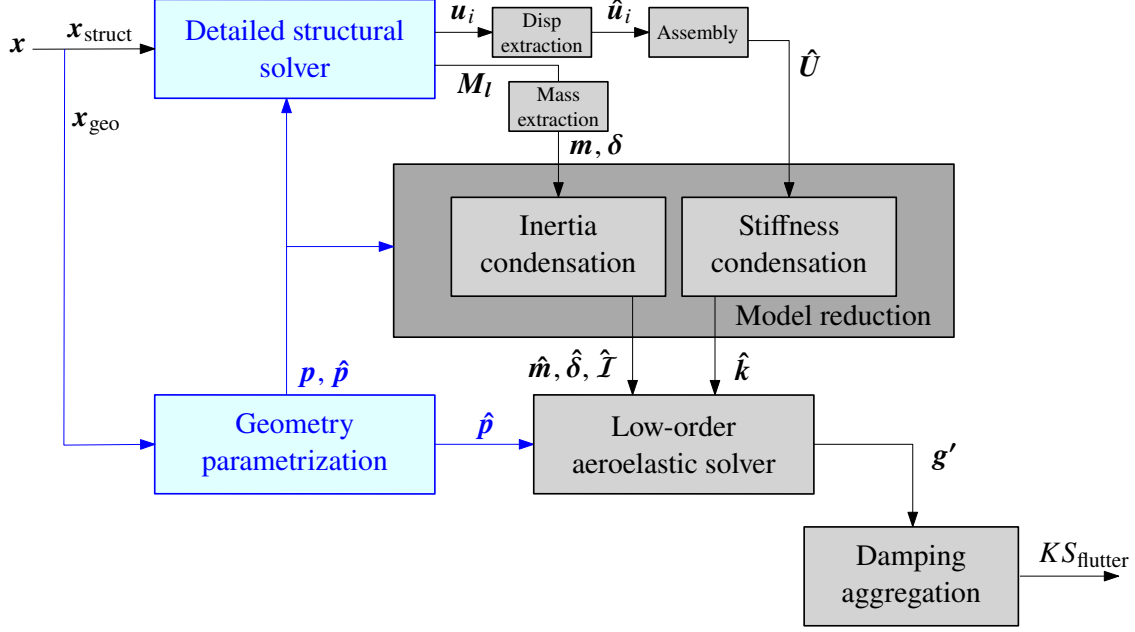


Figure 1: Geometrically nonlinear flutter constraint evaluation process.

The equivalent inertia distributions are given by a set of rigid-body elements with masses $\hat{\mathbf{m}}$, offsets $\hat{\boldsymbol{\delta}}$, and inertia tensors $\hat{\mathcal{I}}$ at the user-specified beam reference axis nodes, which have with coordinates $\hat{\mathbf{p}}$. These rigid-body elements are computed starting from a set of nodal masses \mathbf{m} associated with the built-up FEM nodes along with their offsets $\boldsymbol{\delta}$ and inertia tensors \mathcal{I} about the points $\boldsymbol{\delta} + \mathbf{p}$.

The masses of the built-up FEM nodes are extracted from its lumped mass matrix, \mathbf{M}_l , computed using the Hinton, Rock and Zienkiewicz (HRZ) method [32]. By definition, the mass lumping procedure computes concentrated masses at the built-up FEM nodes that approximate the consistent mass matrix, the offsets of these masses from the FEM nodes, $\boldsymbol{\delta}$, are therefore zero. The mass lumping process also computes lumped rotational inertias, $\mathcal{I}_{xx}, \mathcal{I}_{yy}, \mathcal{I}_{zz}$, for each node, but in practice, these have negligible effects on the condensation accuracy and are thus ignored. Non-structural point masses are added as additional masses with non-zero offsets and inertia tensors from the built-up FEM nodes, which are held constant during optimization.

The equivalent stiffness distributions of the beam model are computed through an identification process based on six linear static solutions of the built-up FEM under six independent tip loads (three independent forces and moments). Displacements and rotations at the beam reference axis nodes for each static solution, denoted by $\hat{\mathbf{u}}_i$ ($i = 1, \dots, 6$), are extracted using interpolation elements (akin to Nastran RBE3 elements) whose dependent nodes coincide with the beam reference axis nodes.

The stiffness condensation process uses these displacements, the beam node locations, and the tip load values to compute the equivalent stiffness properties for each beam element, which are assembled into the vector $\hat{\mathbf{k}}$. The beam formulation used in this work uses strains associated with axial, torsional, and bending deformations as the independent degrees of freedom. This requires identifying a 4×4 stiffness matrix per beam element, where the elements are the stiffness constants associated with axial, torsion, and bending strains along with the corresponding cross-coupling terms. These quantities are defined in the beam local coordinate system and do not need to be

updated with structural deflections [31]. This is because, in a strain-based beam formulation, the stiffness matrix comes from the beam stress-strain constitutive relation, which is materially linear. The geometric stiffening in typical displacement-based formulation is embedded in the geometrically nonlinear strain-based formulation with no requirement for beam stiffness matrix updates. Thus, the equivalent stiffness distributions are computed on the built-up FEM in its unstressed configuration and accurately capture its behavior under large deflections, provided the structure does not experience significant cross-sectional warping or localized buckling, which cannot be captured by a beam model identified in the undeformed shape [33]. Riso and Cesnik [34] compared results from a wing built-up FEM and its beam representation based on this approach, showing excellent agreement for vertical tip deflections up to 40% semispan.

After computing the inertial properties, $\hat{\mathbf{m}}$, $\hat{\delta}$, $\hat{\mathcal{I}}$, stiffness properties, $\hat{\mathbf{k}}$, and node coordinates, $\hat{\mathbf{p}}$, the beam model associated with the built-up FEM at a given optimization step is fully defined. The beam model is coupled to a two-dimensional aerodynamic model based on the potential flow unsteady thin airfoil theory of Peters et al. [35] to obtain the corresponding low-order aeroelastic model for the geometrically nonlinear flutter constraint. The methodology can be applied to aeroelastic models based on other types of beam formulations or unsteady aerodynamic models.

The flutter constraint is computed by a geometrically nonlinear process to account for the impact of wing in-flight deflections. Consider a nonlinear aeroelastic system governed by

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}(\mathbf{c}, \mathbf{x}), \mathbf{c}, \mathbf{x}), \quad (1)$$

where \mathbf{y} is the $N_y \times 1$ state vector and \mathbf{c} is the $N_c \times 1$ vector of parameters that identify a point in the flight envelope, such as load factor, root angle of attack, Mach number, or dynamic pressure. The flutter constraint for a given design \mathbf{x} is assessed considering N_s flight envelope points \mathbf{c}_i of interest for the stability (flutter) analysis ($i = 1, \dots, N_s$). The choice of the flight conditions is problem dependent and must ensure the range of operating conditions of interest is sampled with sufficient refinement to capture critical instability mechanisms, including hump modes.

The first step in the flutter analysis process is to compute the equilibrium state $\mathbf{y}_{e_i}(\mathbf{x}) := \mathbf{y}_e(\mathbf{c}_i, \mathbf{x})$ associated with the flight envelope point \mathbf{c}_i by solving

$$\mathbf{f}(\mathbf{y}_{e_i}(\mathbf{x}), \mathbf{c}_i, \mathbf{x}) = \mathbf{0} \quad (2)$$

The small-amplitude dynamics about the equilibrium state $\mathbf{y}_{e_i}(\mathbf{x})$ is governed by

$$\Delta \dot{\mathbf{y}} = \mathbf{A}_i \Delta \mathbf{y} \quad (3)$$

where $\Delta \mathbf{y} := \mathbf{y} - \mathbf{y}_{e_i}$ is the $N \times 1$ state perturbation vector and

$$\mathbf{A}_i = \mathbf{A}_i(\mathbf{x}) := \left. \frac{\partial \mathbf{f}(\mathbf{y}(\mathbf{c}, \mathbf{x}), \mathbf{c}, \mathbf{x})}{\partial \mathbf{y}} \right|_{\mathbf{y}(\mathbf{c}, \mathbf{x}) = \mathbf{y}_{e_i}(\mathbf{x}), \mathbf{c} = \mathbf{c}_i} \quad (4)$$

is the $N_y \times N_y$ Jacobian matrix of the system (1) with respect to the state vector \mathbf{y} at the equilibrium state \mathbf{y}_{e_i} .

The eigenvalues of $\mathbf{A}_i(\mathbf{x})$ are denoted by $\sigma_{ik}(\mathbf{x}) := g_{ik}(\mathbf{x}) + j\omega_{ik}(\mathbf{x})$ ($k = 1, \dots, N_y$), where $g_{ik}(\mathbf{x})$ is the damping associated with the k th mode at the i th equilibrium state, $\omega_{ik}(\mathbf{x})$ the corresponding angular frequency, and j the imaginary unit. The equilibrium state associated with the i th flight envelope point is stable if the damping values g_{ik} are all negative. This requirement translates to the constraints:

$$g'_{ik}(\mathbf{x}) \leq 0, \quad g'_{ik}(\mathbf{x}) := g_{ik}(\mathbf{x}) - G_i \quad \begin{array}{l} \forall k = 1, \dots, N_y \\ \forall i = 1, \dots, N_s \end{array} \quad (5)$$

The quantity $G_i = G(\mathbf{c}_i)$ in Eq. (5) is the value of a damping bounding curve at the i th flight envelope point. A negative value of the bounding curve helps the optimizer produce a more robust design by requiring a residual damping margin when the flutter constraint is active [19]. This approach can prevent instabilities (such as hump-mode instabilities) that the constraint may miss due to insufficient sampling of the operating envelope (e.g., speed range). A positive value of the bounding curve avoids constraint violations due to marginally stable modes that are not of practical concern because they are stabilized by unmodeled damping sources.

The constraints (5) are reduced to a scalar value using two stages of Kreisselmeier–Steinhauser (KS) aggregation [36–38] to enable efficient derivative computation using the adjoint method, which is necessary to tackle large-scale optimization problems with many design variables. The damping values are first aggregated over all modes at each flight point, giving the values \mathbf{g}' , which are then aggregated over all flight points to give the final flutter constraint value:

$$\text{KS}_{\text{flutter}}(\mathbf{x}) := g'_{\text{max}}(\mathbf{x}) + \frac{1}{\rho_{\text{KS}}} \ln \left[\sum_{i=1}^{N_s} e^{\rho_{\text{KS}}(g'_i(\mathbf{x}) - g'_{\text{max}}(\mathbf{x}))} \right] \leq 0 \quad (6)$$

The KS aggregation parameter, ρ_{KS} , must be carefully selected. Smaller values of ρ_{KS} yield more conservative designs. This may help ensure the optimized design is flutter-free even if the operating envelope is not sufficiently sampled to capture instability mechanisms associated with hump modes. Larger values represent the actual constraints more accurately but make optimization more difficult. The choice of the KS aggregation parameter is problem dependent.

The geometrically nonlinear flutter constraint, based on a set of aeroelastic analyses about the local equilibrium states at different flight conditions, captures the impact of in-flight deflections on the flutter boundary and critical mechanisms. However, this constraint is insufficient to capture other aeroelastic phenomena, such as limit-cycle oscillations, which require fully nonlinear dynamic calculations that are beyond the scope of this paper.

2.2 Geometric Parameterization

The geometry parameterization approach added to the framework in this work uses the free-form deformation (FFD) approach. This approach embeds the wingbox built-up and beam model nodes in a volume defined by a set of control points. Deformations at the control points are then smoothly mapped to the embedded points using spline-based interpolation. In this way, both the detailed wingbox and low-order beam models are deformed consistently and the dependent RBE3 nodes in the wingbox FEM remain coincident with the beam axis nodes (see Figure 2), which is critical for the condensation process. Additionally, the derivatives of the embedded points' coordinates with respect to the control point deformations can be computed quickly and accurately because the mapping is analytic [39].

While the geometry of the wingbox built-up FEM is defined by node coordinates, the low-order aeroelastic model requires the coordinates of the beam reference axis nodes $\hat{\mathbf{p}}$. It also requires the following derived information [4]: aerodynamic chord lengths, jig twist angles, and beam reference axis offsets from the wing leading edge. For a proof-of-concept of the methodology in this work, the chord, twist, and reference axis offsets are fixed, focusing on geometric changes that only alter the reference axis of the wing.

2.3 Flutter Constraint Derivatives

Because the flutter constraint involves solving a large system of implicit equations (the 6 built-up FEM load cases required for the stiffness condensation), we use the adjoint method to efficiently

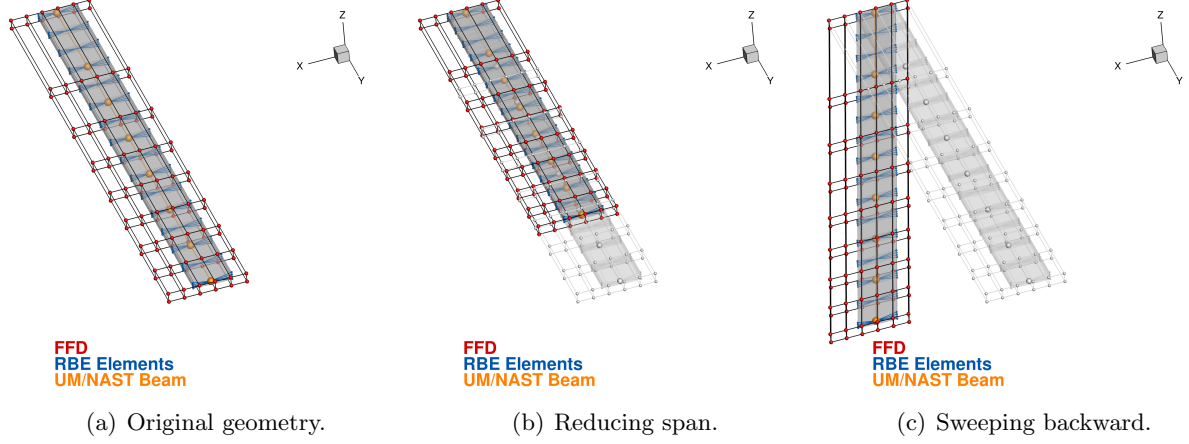


Figure 2: The FFD approach consistently maps deformations from the control points (red nodes) to both the built-up wingbox and low-order beam models.

compute the derivative of the flutter constraint with respect to an arbitrary number of design variables [40, Sec. 6.7]. Similarly to Ref. [27], where derivatives were taken with respect to structural variables, the total derivative with respect to geometric design variables is

$$\frac{d\text{KS}_{\text{flutter}}}{d\mathbf{x}_{\text{geo}}} = \frac{\partial\text{KS}_{\text{flutter}}}{\partial\mathbf{x}_{\text{geo}}} - \sum_{i=1}^6 \psi_i^T \left[\frac{\partial\mathbf{r}_i}{\partial\mathbf{x}_{\text{geo}}} \right], \quad (7)$$

where \mathbf{r}_i and \mathbf{u}_i are the residuals and states of the i th built-up FEM structural analysis, and the adjoint variables ψ_i are computed by solving

$$\left[\frac{\partial\mathbf{r}_i}{\partial\mathbf{u}_i} \right]^T \psi_i = \left[\frac{\partial\text{KS}_{\text{flutter}}}{\partial\mathbf{u}_i} \right]^T \quad (8)$$

Equation (8) is the same adjoint system solved in Jonsson et al. [27] because the only partial derivatives involved are with respect to the structural state variables \mathbf{u}_i . The $\partial\mathbf{r}_i/\partial\mathbf{x}_{\text{geo}}$ term in Eq. (7) is provided by the structural solver used in this work.

The new derivatives needed in Eq. (7) are contained in the explicit partial derivative term:

$$\begin{aligned} \frac{\partial\text{KS}_{\text{flutter}}}{\partial\mathbf{x}_{\text{geo}}} &= \frac{\partial\text{KS}_{\text{flutter}}}{\partial\mathbf{g}'} \frac{\partial\mathbf{g}'}{\partial\mathbf{x}_{\text{geo}}} \\ &= \frac{\partial\text{KS}_{\text{flutter}}}{\partial\mathbf{g}'} \left[\frac{\partial\mathbf{g}'}{\partial\hat{\mathbf{p}}} \frac{\partial\hat{\mathbf{p}}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\mathbf{g}'}{\partial\hat{\mathbf{k}}} \frac{\partial\hat{\mathbf{k}}}{\partial\hat{\mathbf{p}}} \frac{\partial\hat{\mathbf{p}}}{\partial\mathbf{x}_{\text{geo}}} \right. \\ &\quad + \frac{\partial\mathbf{g}'}{\partial\hat{\mathbf{m}}} \left(\frac{\partial\hat{\mathbf{m}}}{\partial\mathbf{m}} \frac{\partial\mathbf{m}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\mathbf{m}}}{\partial\hat{\mathbf{p}}} \frac{\partial\hat{\mathbf{p}}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\mathbf{m}}}{\partial\mathbf{p}} \frac{\partial\mathbf{p}}{\partial\mathbf{x}_{\text{geo}}} \right) \\ &\quad + \frac{\partial\mathbf{g}'}{\partial\hat{\delta}} \left(\frac{\partial\hat{\delta}}{\partial\mathbf{m}} \frac{\partial\mathbf{m}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\delta}}{\partial\hat{\mathbf{p}}} \frac{\partial\hat{\mathbf{p}}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\delta}}{\partial\mathbf{p}} \frac{\partial\mathbf{p}}{\partial\mathbf{x}_{\text{geo}}} \right) \\ &\quad \left. + \frac{\partial\mathbf{g}'}{\partial\hat{\mathcal{I}}} \left(\frac{\partial\hat{\mathcal{I}}}{\partial\mathbf{m}} \frac{\partial\mathbf{m}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\mathcal{I}}}{\partial\hat{\mathbf{p}}} \frac{\partial\hat{\mathbf{p}}}{\partial\mathbf{x}_{\text{geo}}} + \frac{\partial\hat{\mathcal{I}}}{\partial\mathbf{p}} \frac{\partial\mathbf{p}}{\partial\mathbf{x}_{\text{geo}}} \right) \right] \end{aligned} \quad (9)$$

This can be simplified by noting that $\partial \hat{\mathbf{m}} / \partial \hat{\mathbf{p}} = \partial \hat{\mathbf{m}} / \partial \mathbf{p} = 0$ as the connectivity between the built-up FEM and the beam nodes is fixed during optimization. Furthermore, $\partial \hat{\mathbf{I}} / \partial \hat{\mathbf{p}} = 0$ since $\hat{\mathbf{I}}$ represents the inertia tensors about the local center of mass, not the reference axis nodes.

Removing these terms and grouping by remaining terms with a dependence on the reference-axis coordinates, $\hat{\mathbf{p}}$, and the built-up FEM node coordinates, \mathbf{p} , yields

$$\begin{aligned} \frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{x}_{\text{geo}}} = \frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{g}'} & \left\{ \left(\frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{p}}} + \frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{k}}} \frac{\partial \hat{\mathbf{k}}}{\partial \hat{\mathbf{p}}} + \frac{\partial \mathbf{g}'}{\partial \hat{\delta}} \frac{\partial \hat{\delta}}{\partial \hat{\mathbf{p}}} \right) \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}_{\text{geo}}} \right. \\ & \left. + \left[\frac{\partial \mathbf{g}'}{\partial \hat{\delta}} \frac{\partial \hat{\delta}}{\partial \mathbf{p}} + \frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{I}}} \frac{\partial \hat{\mathbf{I}}}{\partial \mathbf{p}} + \left(\frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{m}}} \frac{\partial \hat{\mathbf{m}}}{\partial \mathbf{m}} + \frac{\partial \mathbf{g}'}{\partial \hat{\delta}} \frac{\partial \hat{\delta}}{\partial \mathbf{m}} + \frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{I}}} \frac{\partial \hat{\mathbf{I}}}{\partial \mathbf{m}} \right) \frac{\partial \mathbf{m}}{\partial \mathbf{p}} \right] \frac{\partial \mathbf{p}}{\partial \mathbf{x}_{\text{geo}}} \right\} \quad (10) \end{aligned}$$

The final term in Eq. (10) involves $\partial \mathbf{m} / \partial \mathbf{p}$, which requires the derivative of the built-up FEM mass matrix with respect to the built-up FEM nodal coordinates. If computed fully, the term would be an excessively large 3D tensor, even if stored in a sparse format. However, these terms can be rearranged for a significantly more efficient evaluation. The vector of lumped nodal masses, \mathbf{m} , can be written as

$$\mathbf{m} = \mathbf{Q} \mathbf{M}_D = \mathbf{Q} \mathbf{M}_l \mathbf{e}, \quad (11)$$

where \mathbf{M}_l is the full $6N \times 6N$ lumped mass matrix, \mathbf{e} is a $6N \times 1$ vector of ones, that extracts the diagonal of \mathbf{M}_l into the $6N \times 1$ vector \mathbf{M}_D , and \mathbf{Q} is a $N \times 6N$ matrix that extracts every 6th entry of \mathbf{M}_D to form the $N \times 1$ vector \mathbf{m} . Using this definition gives:

$$\begin{aligned} \frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{g}'} \left(\frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{m}}} \frac{\partial \hat{\mathbf{m}}}{\partial \mathbf{m}} + \frac{\partial \mathbf{g}'}{\partial \hat{\delta}} \frac{\partial \hat{\delta}}{\partial \mathbf{m}} + \frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{I}}} \frac{\partial \hat{\mathbf{I}}}{\partial \mathbf{m}} \right) \frac{\partial \mathbf{m}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{x}_{\text{geo}}} &= \frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{m}} \frac{\partial \mathbf{m}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{x}_{\text{geo}}} \\ &= \frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{m}} \mathbf{Q} \frac{\partial \mathbf{M}_l}{\partial \mathbf{p}} \mathbf{e} \frac{\partial \mathbf{p}}{\partial \mathbf{x}_{\text{geo}}} \quad (12) \end{aligned}$$

The derivative involving the lumped mass matrix is now part of a vector-matrix-vector product, $\phi^T (\partial \mathbf{M}_l / \partial \mathbf{p}) \theta$, where $\phi^T = \partial \text{KS}_{\text{flutter}} / \partial \mathbf{m} \mathbf{Q}^1$ and $\theta = \mathbf{e}$. This product can then be assembled efficiently on an element-by-element basis by the finite element solver without forming the full lumped mass matrix derivative.

For clarity, the individual terms from Equation 10 are shown in Table 1, grouped by the module used to compute them.

3 Computational Framework

This section describes the computational tools used to implement the methodology described in Section 2.

3.1 High-Fidelity Structural and Aerostructural Analysis

The presented flutter analysis framework requires only a detailed structural solver. However, this work considers optimizations that rely on both the flutter analysis and high-fidelity static aerostructural analyses in cruise and maneuver conditions. These analyses are performed using the MDO for Aircraft Configurations with High fidelity (MACH) framework [41].

¹Note that the matrix \mathbf{Q} is never explicitly formed. It's action is simply to expand the entries of $\partial \text{KS}_{\text{flutter}} / \partial \mathbf{m}$ into every 6th entry of a new larger vector.

Table 1: Partial derivative terms from Equation (10), grouped by the module that computes them.

Module	Terms computed
Damping aggregation	$\frac{\partial \text{KS}_{\text{flutter}}}{\partial \mathbf{g}'}$
Aeroelastic solver	$\frac{\partial \mathbf{g}'}{\partial \mathbf{p}}, \frac{\partial \mathbf{g}'}{\partial \mathbf{k}}, \frac{\partial \mathbf{g}'}{\partial \hat{\mathbf{m}}}, \frac{\partial \mathbf{g}'}{\partial \hat{\delta}}, \frac{\partial \mathbf{g}'}{\partial \hat{\mathcal{I}}}$
Model reduction	$\frac{\partial \hat{\mathbf{k}}}{\partial \hat{\mathbf{p}}}, \frac{\partial \hat{\delta}}{\partial \hat{\mathbf{p}}}, \frac{\partial \hat{\delta}}{\partial \hat{\mathbf{p}}}, \frac{\partial \hat{\mathcal{I}}}{\partial \hat{\mathbf{p}}}, \frac{\partial \hat{\mathbf{m}}}{\partial \hat{\mathbf{m}}}, \frac{\partial \hat{\delta}}{\partial \hat{\mathbf{m}}}, \frac{\partial \hat{\mathcal{I}}}{\partial \hat{\mathbf{m}}}$
Structural solver	$\phi^T \frac{\partial \mathbf{M}_l}{\partial \mathbf{p}} \theta$
Geometry parameterization	$\frac{\partial \mathbf{p}}{\partial \mathbf{x}_{\text{geo}}}, \frac{\partial \hat{\mathbf{p}}}{\partial \mathbf{x}_{\text{geo}}}$

The structural solver used in MACH is the Toolkit for the Analysis of Composite Structures (TACS) [42]², an open-source parallel finite element solver that handles poorly conditioned problems common in thin-walled aircraft structures. For such cases, the stiffness matrix condition numbers may exceed $\mathcal{O}(10^9)$, but TACS can solve these poorly conditioned problems using a Schur-complement-based parallel direct solver. Derivatives of structural functions of interest with respect to structural and geometric design variables are computed efficiently using the adjoint method [42].

The aerodynamic loading is provided by the open-source CFD solver ADflow, which is a parallel, finite-volume, cell-centered, multiblock solver with discrete adjoint derivative computation [43].³ ADflow solves the steady Reynolds averaged Navier–Stokes (RANS) equations with the one-equation Spalart–Allmaras turbulence model, using an approximate Newton–Krylov method [44].

The nonlinear aerostructural equations are solved using a block Gauss–Seidel approach, including an Aitken acceleration, and the linear adjoint equations are solved using a coupled Newton–Krylov approach [41].

The FFD geometry parameterization is provided by the open-source package pyGeo⁴ [39]. Once the CFD aerodynamic surface is deformed, either due to design changes or structural deformation, the volume mesh is updated using a robust inverse distance warping scheme [45], using the open-source package IDWarp.⁵

3.2 Low-Order Aeroelastic Framework

The geometrically nonlinear flutter analyses are conducted in the University of Michigan’s Nonlinear Aeroelastic Simulation Toolbox (UM/NAST) [4], a low-order multidisciplinary framework for simulating very flexible wings and complete aircraft. UM/NAST uses a strain-based geometrically exact beam formulation to represent aircraft components (such as wing and fuselage) undergoing structural deflections and rigid-body motions [31]. Aerodynamic loads can be computed using multiple options, such as the potential flow, unsteady thin airfoil theory of Peters et al. [35] used in this work. Compressibility effects, sweep, and tip losses are introduced using analytical relations.

The geometrically nonlinear flutter analysis at a given design iteration starts by computing the nonlinear equilibrium states of the system at the flight conditions of interest. Next, the equations of motion are linearized about such equilibrium states, which gives a state-space linearized model for each flight condition. The eigenvalues and eigenvectors (aeroelastic modes) of each full-order state-

²<https://github.com/smdogroup/tacs>

³<https://github.com/mdolab/adflow>

⁴<https://github.com/mdolab/pygeo>

⁵<https://github.com/mdolab/idwarp>

space matrix are obtained using a standard eigenvalue analysis. Finally, the eigenvalues are used for evaluating the flutter constraint or post-processed for analysis purposes. The flutter constraint and its partial derivatives with respect to the low-order model inputs $(\hat{\mathbf{k}}, \hat{\delta}, \hat{\mathbf{m}}, \hat{\mathbf{p}})$ are evaluated by an in-house developed OpenMDAO [46]⁶ wrapper that couples the UM/NAST component with the rest of the framework.

3.3 Model Order Reduction

Evaluating the geometrically nonlinear flutter constraint requires reducing the detailed model at a given optimization iteration to a low-order aeroelastic model in UM/NAST. The equivalent beam distributions of the TACS built-up FEM are computed using the University of Michigan’s Enhanced FEM2Stick (UM/EF2S) code [29, 30]. The outputs from UM/EF2S are used to create the beam structural model for the UM/NAST geometrically nonlinear flutter constraint and to compute its derivatives with respect to the equivalent beam distributions.

The geometry of the UM/NAST beam reference axis is obtained from the FFD deformation at each optimization step, while chord, twist, reference axis offset from the leading edge, and airfoil properties are fixed. The airfoil aerodynamics is described using a zero-thickness, flat-plate model in unsteady subsonic potential flow, with compressibility effects accounted for using the Prandtl–Glauert correction.

3.4 Optimization Algorithm

The optimization algorithm is SNOPT (Sparse Nonlinear OPTimizer) [47], a gradient-based optimizer that implements a sequential quadratic programming (SQP) algorithm. SNOPT uses an augmented Lagrangian merit function; the Hessian of the Lagrangian is approximated using a quasi-Newton approach suitable for optimization problems with many sparse nonlinear constraints. The interface with SNOPT is handled by pyOptSparse⁷ [48], an implementation of pyOpt [49] that eases defining large sparse Jacobians crucial to the performance of large scale optimizers like SNOPT.

4 Aerostructural Optimization

The methodology is now demonstrated by performing a series of two-point aerostructural optimizations both with and without the presented geometrically nonlinear flutter constraint. Section 4.1 describes the wing model being optimized, and Section 4.2 describes the formulation of the optimization problem.

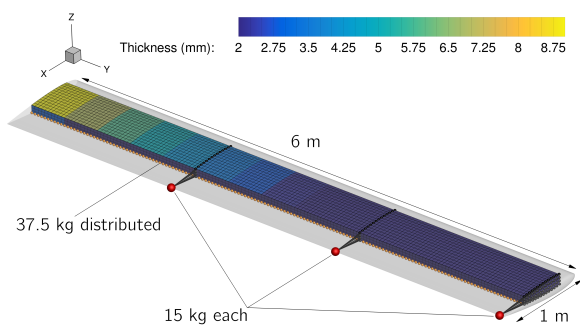
4.1 Wing Model

Figure 3 shows the baseline wing model that acts as the starting point for all optimizations. The wing consists of a rectangular, untwisted wing with a unit chord, an aspect ratio of 12, and a NACA 0012 aerodynamic cross-section. The wingbox spans from 15% to 65% of the aerodynamic chord and uses Aluminium 2024-T3 properties⁸ ($E = 73.1$ GPa, $\nu = 0.33$, $\sigma_Y = 345$ MPa, $\rho = 2780$ kg/m³). The wingbox mesh consists of 21,000 second-order MITC4 shell elements, totaling 123,120 degrees of freedom (DOF). The baseline thickness distribution (Figure 3(a)) comes from a structural pre-optimization under fixed 2.5-g aerodynamic loads, without the flutter constraint or

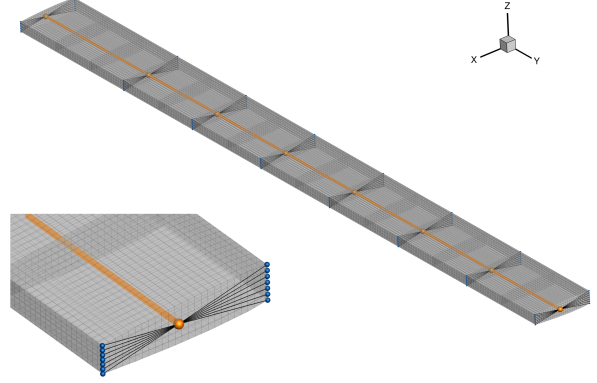
⁶<https://github.com/OpenMDAO/OpenMDAO>

⁷<https://github.com/mdolab/pyoptsparse>

⁸asm.matweb.com/search/SpecificMaterial.asp?bassnum=ma2024t3



(a) Baseline built-up FEM showing panel thicknesses, nonstructural masses, and wing outer mold line.



(b) Baseline wing built-up FEM and beam structural models.

Figure 3: Baseline wing model.

geometric variables, resulting in a mass of 68.5 kg. Non-structural masses are added to the wing to ensure it flutters within a dynamic pressure range that results in large but not excessive static deflections and where the subsonic flow assumption holds. Three 15 kg point masses are added along the trailing edge at each third of the semispan, and a mass of 37.5 kg is distributed on the lower skin to model fuel. These non-structural masses are also included as a static inertial load in the aerostructural analyses.

To evaluate the geometrically nonlinear flutter constraint, the TACS wingbox built-up FEM is reduced to an equivalent beam model in UM/NAST at each optimization step. The beam reference axis is at 41% of the aerodynamic chord and is discretized in eight reference nodes (seven three-node beam elements). Each reference node is connected to the leading and trailing edges of its corresponding cross-section by a TACS RBE3 element. Figure 3(b) shows the TACS and UM/NAST models overlapped.

The equivalent beam model is coupled to a potential flow, unsteady thin airfoil model [35] with zero-thickness, flat-plate cross-sectional properties. The resulting low-order aeroelastic model provides the geometrically nonlinear flutter constraint, while the other functions of interest come from the detailed aerostructural model. The geometrically nonlinear flutter analysis accounts for subsonic compressibility effects using the Prandtl–Glauert correction. Three-dimensional aerodynamic effects are not considered, but they can be added using tip loss factors or variable airfoil properties along the span derived from the detailed model. Transonic effects are beyond the scope of this paper. They could be included by replacing the potential flow model with a low-order transonic aerodynamic model [28], if parametrized and differentiated to enable geometric updates.

The aerostructural analyses use a structured multiblock CFD mesh (Figure 4). The wing surface contains 44 cells along the chord, 66 along the span, and 8 across the trailing edge. This surface mesh is extruded to 100 chord lengths from the wing using the hyperbolic mesh generation tool pyHyp⁹. The first layer height is 2.7734×10^{-6} m, resulting in a y^+ value of approximately 1 at the maximum studied Reynolds number of 12.7×10^6 .

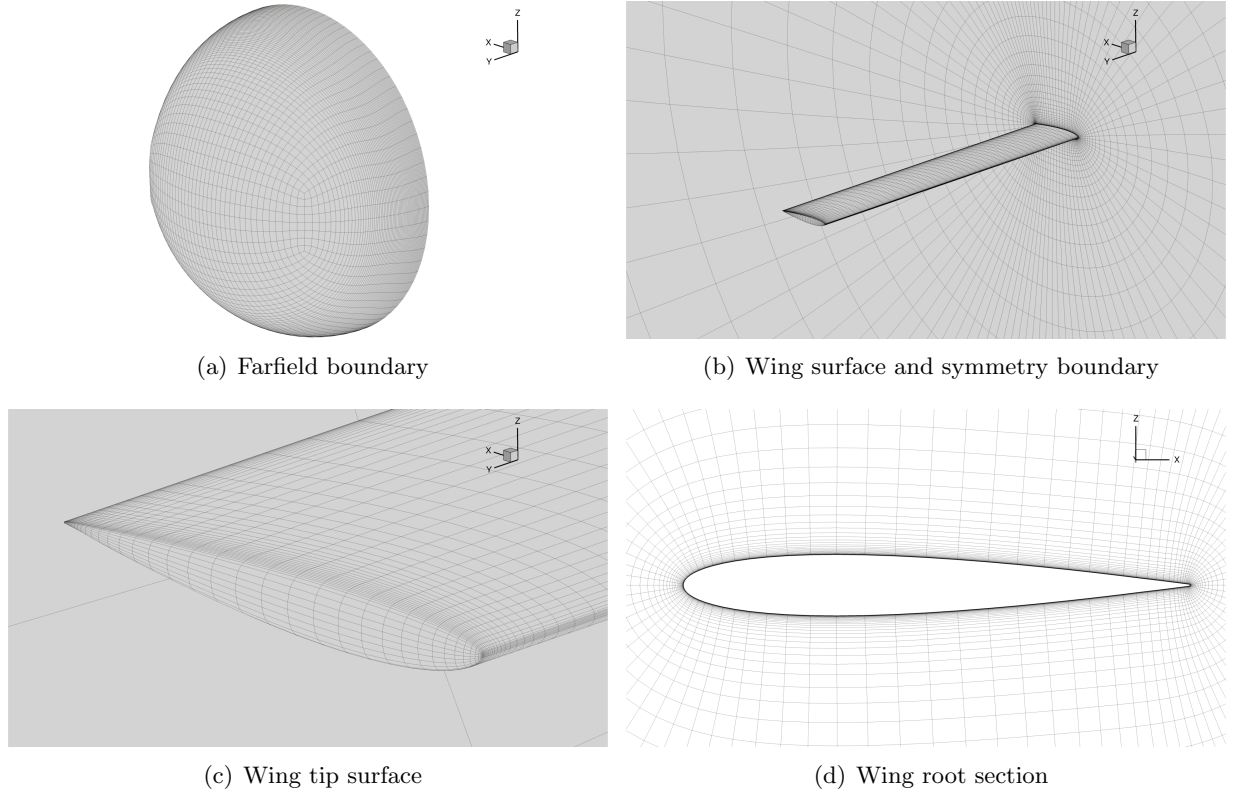


Figure 4: Baseline wing CFD aerodynamic model.

Table 2: Optimization formulation.

	Function/variable	Description	Quantity
maximize with respect to	R	Cruise range	
	t	Skin/spar/rib thicknesses	76
	x_{Span}	Wing semispan scaling	1
	x_{Sweep}	Wing sweep	1
	α_i	Root angle of attack at i th flight condition	2
	Total		80
subject to	$KS_{\text{flutter}} \leq 0$	KS aggregate of damping values	4
	$1.5KS_{\text{stress}} \leq 1$	KS aggregates of 2.5 g stresses	4
	$ t_{\text{skin}, i} - t_{\text{skin}, i+1} \leq 1 \text{ mm}$	Skin adjacency constraints	28
	$ t_{\text{spar}, i} - t_{\text{spar}, i+1} \leq 1.5 \text{ mm}$	Spar adjacency constraints	28
	$L_i = n_i \text{MTOW}$	Lift constraint at i th flight condition	2
	Total		66

4.2 Optimization Problem

The optimization problem is to maximize the wing's cruise range, subject to maneuver stress and flutter constraints. The formal problem definition is detailed in Table 2. The cruise and maneuver conditions in Table 3 for the detailed aerostructural analyses have a Mach number of 0.5. The

⁹github.com/mdolab/pyhyp

Table 3: Flight conditions for optimization.

Condition	Speed (m/s)	Mach	Load factor (g)	Altitude (m)	Reynolds
Cruise	149.7	0.5	1	10000	4.38×10^6
Maneuver	170.1	0.5	2.5	0	12.0×10^6
Flutter	10–180	0.029–0.529	–	0	12.7×10^6

cruise altitude is 10 000 m while the maneuver is at sea level, with air properties based on standard atmospheric values. The geometrically nonlinear flutter constraint is also evaluated for sea-level conditions at 20 flight speeds between 10–180 m/s, within the applicability of subsonic potential flow assumptions.

This speed sampling is appropriate for this problem, which aims to demonstrate the methodology rather than conducting a practical design optimization. In a production environment, the speed range (or other parameter range) should be sampled finely enough that the geometrically nonlinear flutter constraint captures all critical instability mechanisms, including hump modes. However, too fine a sampling increases the computational burden of evaluating the constraint. This is critical for the proposed geometrically nonlinear formulation, which solves a nonlinear static aeroelastic equilibrium problem followed by linearization at each flight condition. The appropriate sampling is problem dependent and based on the trade-off of constraint accuracy and computational burden. Constraint parameters such as the bounding curve and the KS aggregation parameter help compensate for instability mechanisms missed with a coarser sampling of the operating envelope, such as hump modes. For instance, specifying a negative bounding curve, decreasing the KS aggregation parameter, or both, make the flutter constraint more conservative, increasing the optimized design’s robustness. These approaches can address potentially missed instability mechanisms and other unmodeled effects.

Demonstrating the methodology requires a multidisciplinary objective that combines aerodynamic and structural performance. The optimization aims to maximize cruise range, computed using the Breguet range equation:

$$R = \frac{V}{c_T} \frac{C_L}{C_D} \ln \left(\frac{M_{\text{init}}}{M_{\text{init}} - M_{\text{fuel}}} \right) \quad (13)$$

In this equation, R is the flight range, V the cruise speed, c_T the thrust-specific fuel consumption ratio (0.5 kg/h) and C_L , C_D the cruise lift and drag coefficients. The initial cruise mass M_{init} consists of a fixed 400 kg plus the wingbox structural mass and the fuel mass of 37.5 kg. These three masses are all half of the full-vehicle values.

The design variables are 76 wingbox panel thicknesses, span, and sweep. The angles of attack for the cruise and maneuver flight conditions are also design variables to ensure the lift matches the load factor. The primary constraints are the stress and flutter constraints. The von Mises stresses in the wingbox under the maneuver loads are KS-aggregated into four constraint values, one each for the ribs, spars, upper skin, and lower skin panels. These values are constrained to lie below the material yield strength by a safety factor of 1.5.

The flutter damping values are first aggregated across all modes at each flight condition and then aggregated across all flight conditions. This gives a scalar constraint value that closely but smoothly approximates the maximum damping value across all flutter flight conditions, which is constrained to lie below a threshold of 0.12 rad/s.

To produce a realistic structural design, the optimization includes linear adjacency constraints. For the upper and lower skins, the thickness of adjacent panels must be within 1 mm of one another,

while for the spar panels, the limit is 1.5 mm. Finally, the lift produced in the cruise and maneuver flight conditions must equal half the maximum take-off weight (MTOW) (1280 kg) multiplied by the desired load factor.

To investigate how geometric variables suppress flutter, optimizations are solved with only span, only sweep, or both span and sweep variables, including the structural variables in all cases. To highlight the importance on geometrically nonlinear effects on the flutter constraint and optimal design, the wing is optimized multiple times with the flutter constraint computed at a different fixed wing root angle of attack between 0.5 and 6°. Finally, an optimization without the flutter constraint for each parameterization serves as a reference to judge the impact of the flutter constraint.

Although the discussion of the results herein focuses on the impact of the flutter constraint root angle of attack on the optimal design, this is merely a proxy for the in-flight deformation level of the wing, which is the true driver of the geometric nonlinearity. In a more realistic design problem, the deformations would be dictated by the load factor enforced for each flutter flight condition and by the resulting trim parameters. However, to reduce the complexity of this example problem, we use the root angle of attack to control the deformation level.

Each optimization runs on 72 Intel Xeon Gold 6154 processor cores. The cruise and maneuver aerostructural analyses and adjoint derivative evaluations use 26 and 42 cores, respectively. The flutter flight points are split across separate instances of MACH-UM/NAST on the remaining four processors, each performing its aggregation, resulting in the four distinct flutter constraint values (Table 2). This splitting is done evenly such that one instance computes the flutter damping aggregate over the first five points in the sweep, another in the next five, and so on. The optimizations are terminated when SNOPT reaches a feasibility (maximum constraint violation) below 10^{-5} and an optimality (Lagrangian gradient norm) below 10^{-4} , a reduction of three orders of magnitude from the initial design. This optimization requires about 50 major iterations and 4 hours of wall time.

5 Results

5.1 Derivative Verification

The analytic geometric derivatives in Section 2.3 are verified against finite-difference approximations. Table 4 compares the analytic derivatives with central-difference approximations computed by perturbing the span and sweep design variables by $\pm 10^{-4}$. The values are computed with the baseline wing, using 10 flight speeds over the same range used in the optimization problem, at an root angle of attack of 2°. The most significant difference between analytic and finite difference values is 0.00108%.

Table 4: Flutter constraint derivatives with respect to the geometric variables $d\mathbf{KS}_{\text{flutter}}/d\mathbf{x}_{\text{geo}}$.

	$d\mathbf{KS}_{\text{flutter}}/d\mathbf{x}_{\text{Span}}$	$d\mathbf{KS}_{\text{flutter}}/d\mathbf{x}_{\text{Sweep}}$
Analytic	33.09999319	−5.25644508
Finite-difference	33.09997429	−5.25650208
Δ (%)	6×10^{-5}	-1.08×10^{-3}

5.2 Optimization Results

Before discussing the optimized designs, we discuss the flutter mechanisms of the wings. Figure 5 shows the root loci of the wings optimized for each case and root angle of attack, α . In all cases, the wings optimized without the flutter constraint are flutter critical. In almost all cases, the flutter mode has the second-lowest frequency (referred to as mode 2 herein). For the undeformed wing in a vacuum, mode 2 is the first in-plane bending mode. As the wing deflects, this mode couples with torsion due to geometrically nonlinear bending kinematics¹⁰. With the addition of aerodynamics, torsion couples with out-of-plane bending, leading to a 3-DOF flutter mechanism that involves in-plane bending, torsion, and out-of-plane bending motions, as previously discussed by Hodges and Dowell [50], Patil et al. [3], and Cestino et al. [23], among others.

Because the in-plane bending and torsion motions only couple when the wing deflects, the resulting flutter mechanism is not captured by a geometrically linear flutter constraint that only considers the undeformed wing. Increasing the root angle of attack leads to larger in-flight deformations, strengthening the coupling, causing mode 2 to flutter at a lower speed and stray further into the right half-plane of the root locus.

The optimizer has two options for suppressing this flutter mode. The first is to increase the wing's torsional and in-plane bending stiffness by reinforcing the wingbox spars. The other is to increase the out-of-plane bending stiffness, either by thickening the upper and lower skins or reducing the span, in order to decrease in-flight deformations and weaken the geometrically nonlinear coupling.

The only cases where mode 2 is not the critical flutter mode are in the wings optimized with either only sweep or with span and sweep at the lowest root angle of attack of 0.5° . At this low root angle of attack, with the wash-out introduced by backward sweep, in-flight deflections are low enough that the geometrically nonlinear coupling that causes mode 2 to flutter is not significant, and mode 3 is the critical flutter mode. In the undeformed wing, mode 3 is a coupled out-of-plane bending and torsion mode. This bending-torsion coupling is exacerbated by the backward sweep, explaining why this mode is flutter critical in the wings optimized with a sweep variable. The wing optimized with only a span variable remains straight; thus, mode 2 is always the critical flutter mode.

Figure 6 presents the strain mode shapes computed by UM/NAST for the wing optimized with span and sweep variables and without a flutter constraint. The plots show the normal vibration modes for the undeformed wing and the aeroelastic modes at the flutter speed for each root angle of attack. Note that the aeroelastic eigenvectors are complex, and thus the magnitudes of the complex strain values are plotted. The central column of plots demonstrates that mode 2 originates as a purely in-plane bending mode in the undeformed structure but gains significant torsion and out-of-plane bending components as in-flight deflections increase, leading to a flutter mode that involves in-plane, torsion, and out-of-plane bending. Notice that mode 1 and mode 3 do not develop a significant in-plane bending component, and remain primarily a mix of torsion and out-of-plane bending.

Figure 7 further demonstrates the role of in-plane bending in the 3-DOF flutter mechanism. The figure shows results of the flutter analyses from Figure 5(e) repeated with in-plane bending removed. To compute these, the aeroelastic system is linearized around the same static deformed state (including in-plane bending deformations) at each flight point, but the rows and columns corresponding to in-plane bending strain and strain rate DOFs are removed from the aeroelastic state-space matrix \mathbf{A} (Eq. (3)) before its eigenvalues are computed. The two plots show that the flutter mechanism associated with mode 2 disappears when in-plane bending is restrained, proving

¹⁰Figure 9(c) by Cestino et al. [23] illustrates this effect.

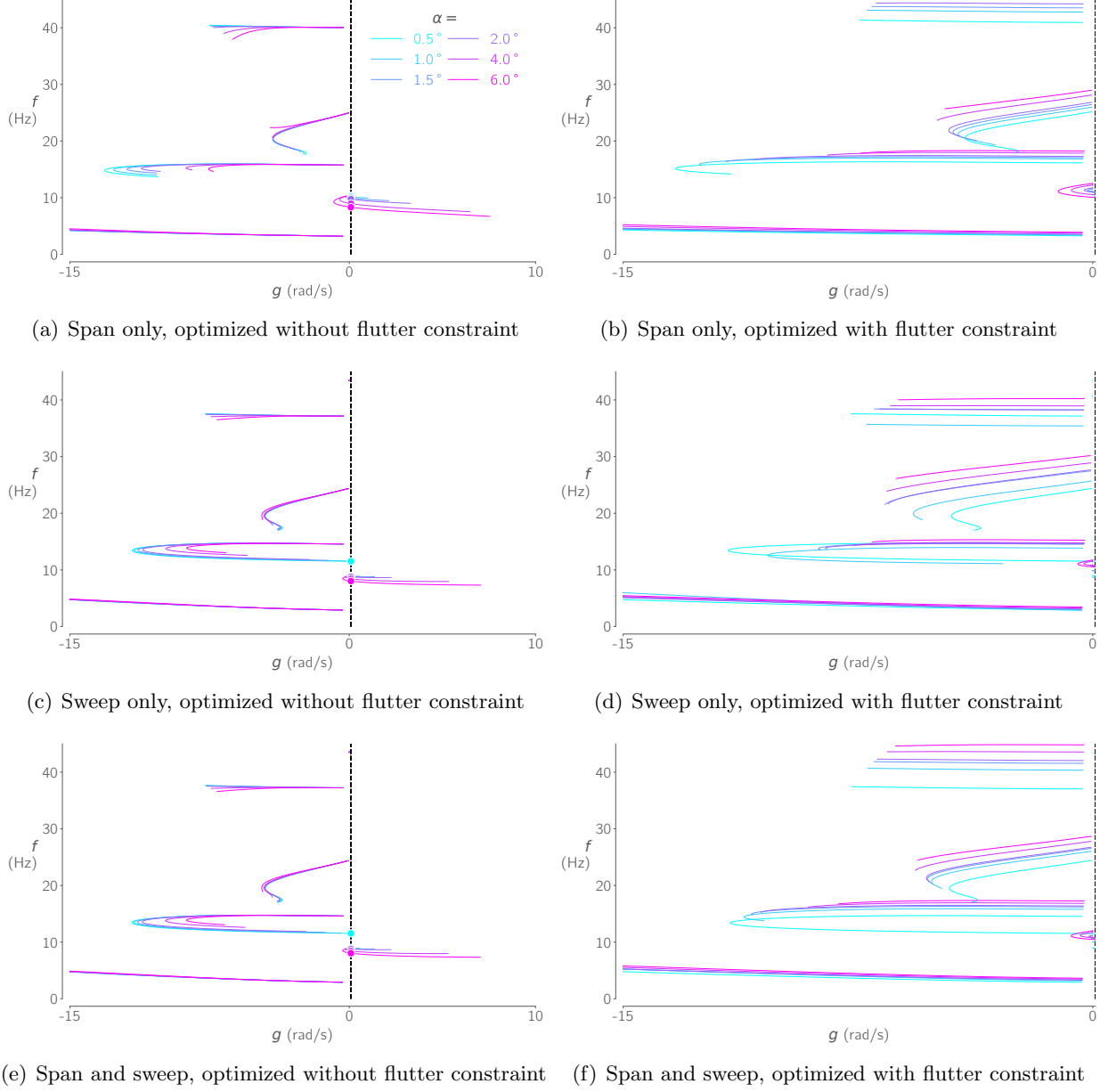


Figure 5: Root loci for all optimized wings.

that in-plane bending is critical to the 3-DOF flutter mechanism.

Figure 8 shows the structural models of the optimized wings, displaying the maneuver stress and structural thickness distributions. The designs optimized with the flutter constraint have shorter wingspans and thicker spars. Figure 9 shows the optimized cruise range, lift-to-drag ratio, structural mass, semispan, and sweep angle of each optimized wing for the root angle of attack where the flutter constraint was computed. Also shown are reference values of these metrics for the wings optimized without a flutter constraint. Finally, to show the differences across the 3 geometric parameterizations more clearly, Figure 10 plots the same data with the values for all parameterizations on the same axes.

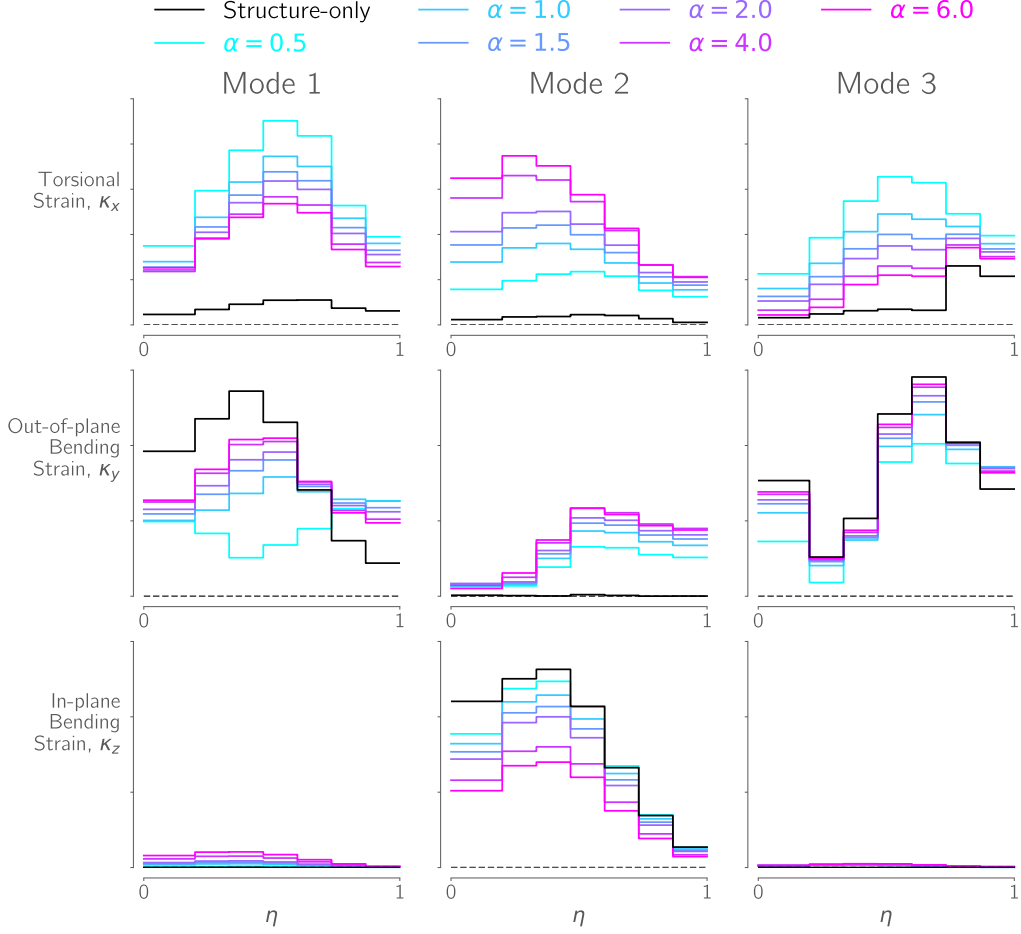


Figure 6: Breakdown of the first three structural and aeroelastic strain modes computed by UM/NAST for the wing optimized without a flutter constraint. Mode 2 transitions from a pure in-plane bending mode to a combination of in-plane bending, out-of-plane bending, and torsion DOF (to become the flutter mode) as the level of static deformation increases. Modes 1 and 3 are less affected by in-flight deformations and remain as a coupled out-of-plane bending and torsion DOF.

The primary trade-off in the optimization is between suppressing flutter by reducing the span or increasing structural thicknesses, both detrimental to the cruise range. In this case, the optimizer finds it more effective to reduce the span, sacrificing drag while maintaining a lower structural mass, rather than retaining a higher span resulting in lower drag but higher structural mass. This is because reducing span increases the in-plane and out-of-plane bending stiffness for a given panel thickness distribution.

In the span-only cases, at the lowest root angle of attack of 0.5° , the optimizer achieves the same range as without the flutter constraint but with a different design. The wingspan is reduced by 1.6%, sacrificing drag to satisfy the flutter constraint, but the structural mass is lower, which counteracts the loss in aerodynamic efficiency. This lower mass is not solely due to the reduced volume of the wingbox. In the wing optimized without flutter constraint, the front spar is greatly reinforced (see Figure 8(a)) to induce passive load alleviation; in the flutter-constrained design, this reinforcement is reduced because of the lower bending moment resulting from the lower span. As the root angle of attack increases, the optimized span decreases, but the wingbox mass does not. At

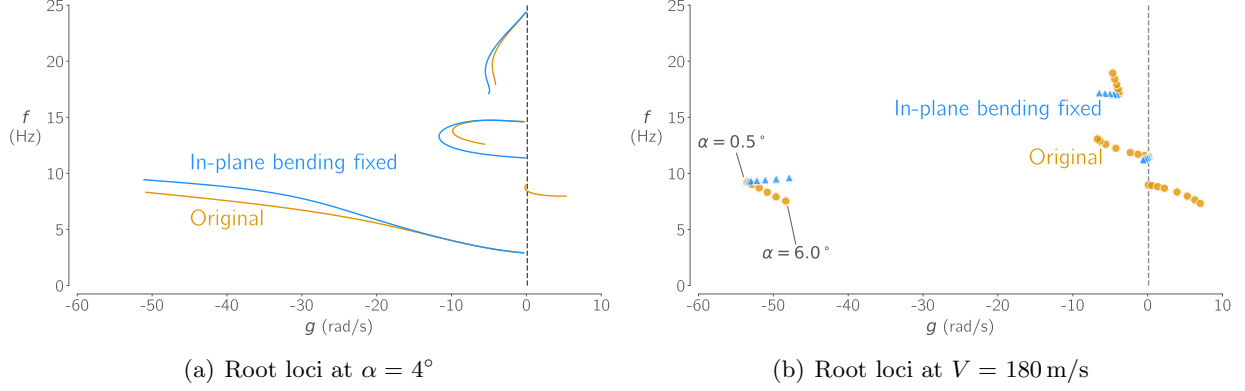


Figure 7: When in-plane bending DOF are removed from the aeroelastic stability analysis, the flutter mechanism changes, showing that in-plane bending plays a critical role in the geometrically nonlinear 3-DOF flutter mechanism.

the highest root angle of attack of 6° , the span is 7.7% lower than for the non-flutter-constrained design, but the structural mass is 2.1% higher. At this point, the range loss due to the flutter constraint is a modest 2.8%.

In the cases where the optimizer can only vary the sweep, the range loss due to the flutter constraint varies significantly more with the root angle of attack. At $\alpha = 0.5^\circ$, the non-flutter-constrained design is barely flutter-critical. Thus, unlike the span-only case, the optimizer satisfies the flutter constraint with almost no design change. Comparing Figures 8(a) and 8(b) shows that the optimizer achieves passive load alleviation through backward sweep rather than by reinforcing the front spar. This gives a lighter wing and a range of 5248.9 km, 1.2% higher than the span-only case. However, as the root angle of attack for the flutter constraint increases, the loss in range is more significant. The flutter mode at higher angles of attack is mode 2, which is suppressed by reducing in-flight deformation. Without reducing deformations by shortening the wing, the optimizer is forced to satisfy the flutter constraint by reinforcing the wingbox. The optimized wingbox mass increases by 45.4% at $\alpha = 6^\circ$ resulting in a range loss of 5.9% over the non-flutter-constrained design, and 2% over the span-only case.

When the optimizer can vary span and sweep, it reaps the benefits of both. Like the sweep-only cases, the non-flutter-constrained wing is just flutter-critical at $\alpha = 0.5^\circ$, and the flutter-constrained design is almost identical to the non-flutter-constrained one. As the root angle of attack increases, the optimizer reduces the wingspan to control in-flight deformation while keeping a lower structural mass, as in the span-only cases. The wings optimized with span and sweep variables achieve the same high range of 5248.9 km as the sweep-only case at $\alpha = 0.5^\circ$ but also maintain a lower structural mass and higher range as the root angle of attack increases, as the span-only cases did. This is because the optimizer can meet the stress constraints while minimizing structural thickness increases by varying sweep in addition to span.

Figures 10(e) and 10(f) compare the masses of the wingbox skins and spars, respectively. These trends demonstrate how the optimizer’s method for suppressing the mode 2 flutter mechanism changes at different levels of in-flight deformation in the span-only and span + sweep cases. For $\alpha = 0.5$ - 2° , the optimizer appears to favor increasing the torsional and in-plane bending stiffness, with the mass of the spars increasing, even as the skin and total wingbox masses decrease. Above $\alpha = 2^\circ$, the optimizer increases the mass of the wingbox skins and decreases the mass of the spars. This suggests a greater focus on limiting in-flight deformations to reduce the strength of the in-plane bending torsion coupling that drives the 3-DOF flutter mechanism.

Geometrically nonlinear effects play a pivotal role in the outcome of the presented optimizations. For low in-flight deformations, optimizing the wing with only a sweep geometric design variable results in a cruise range 1.2% higher than with only span and equal to the range achieved without considering flutter. For the highest in-flight deformations, reducing span is necessary to suppress the geometrically nonlinear coupling of in-plane bending, out-of-plane bending, and torsion involved in the flutter mode. This is effective at preventing flutter than maintaining a larger span while reinforcing the wingbox. At this point, this advantage has reversed, and the wings optimized with the span variable have a range advantage of at least 2%. The optimizer changes sweep to provide passive load alleviation, while it relies on span and (to a lesser extent) on thickness for meeting the flutter constraint.

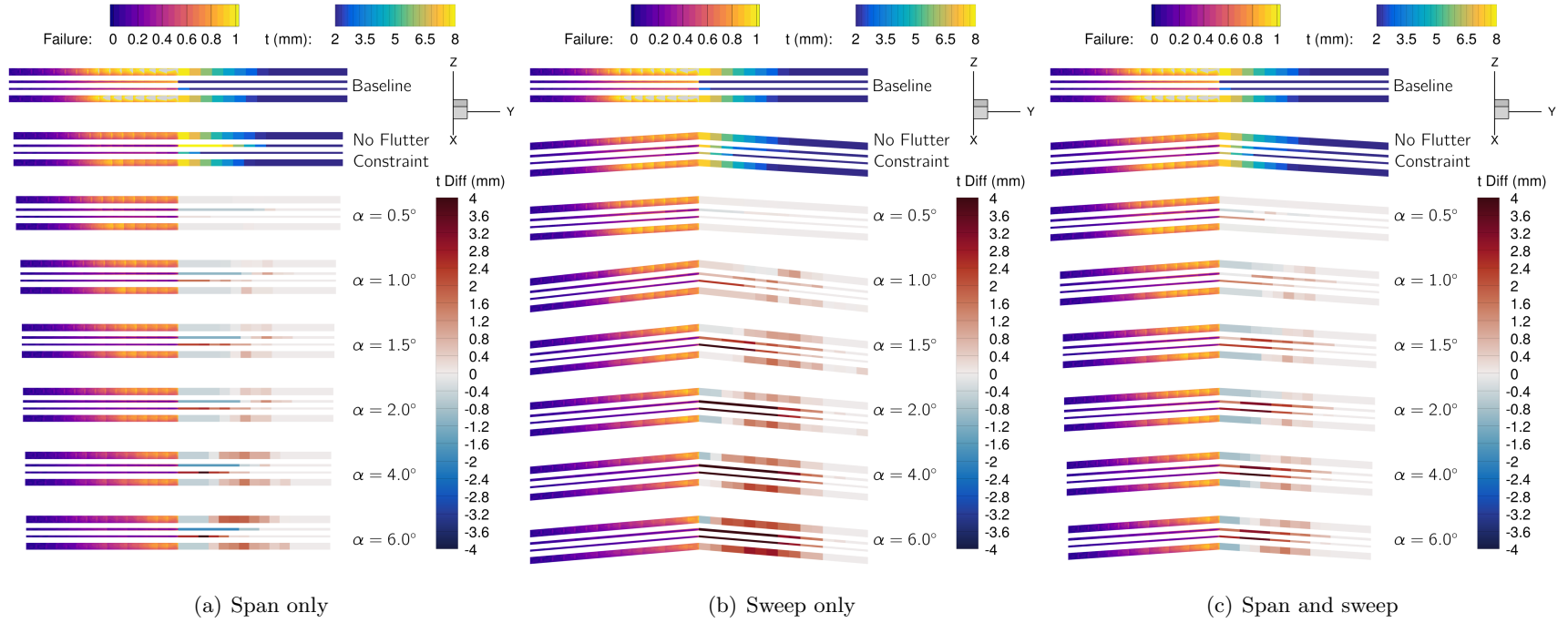


Figure 8: Selected optimized wings showing the normalized stress (left), and the structural sizing (right). The thickness difference plotted on the wings optimized with the flutter constraint is relative to the wing optimized without the flutter constraint.

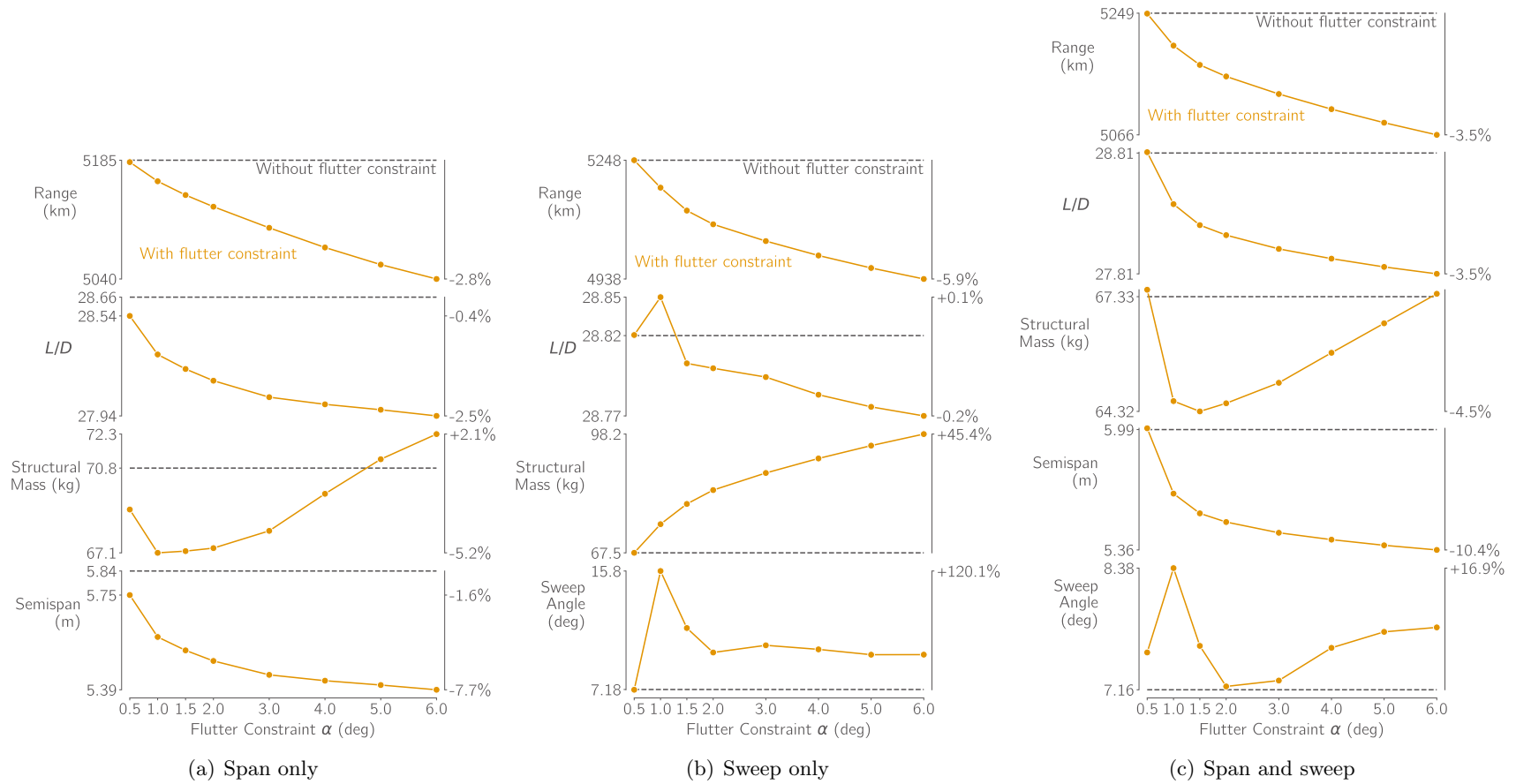
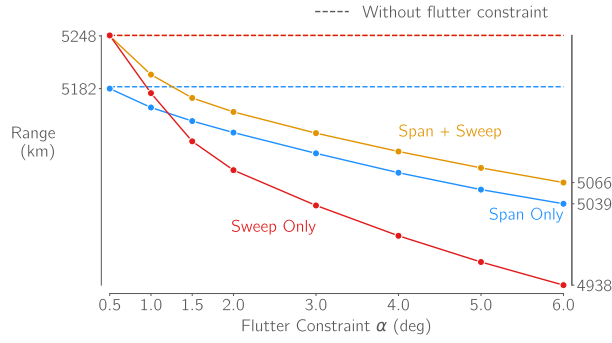
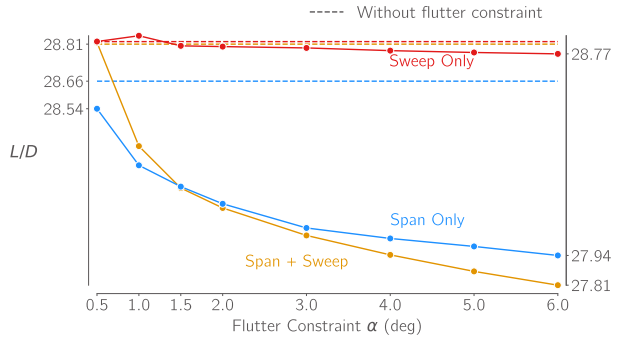


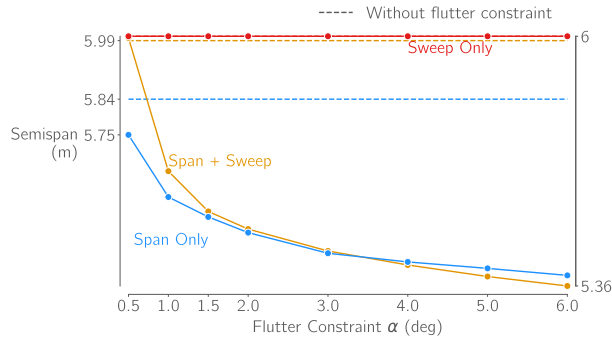
Figure 9: Trends in the functions of interest for each optimized wing design.



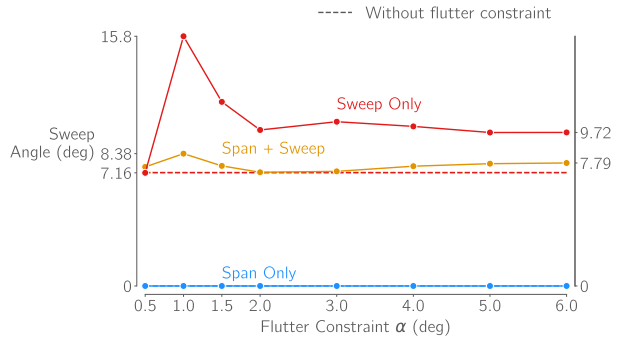
(a) Cruise range



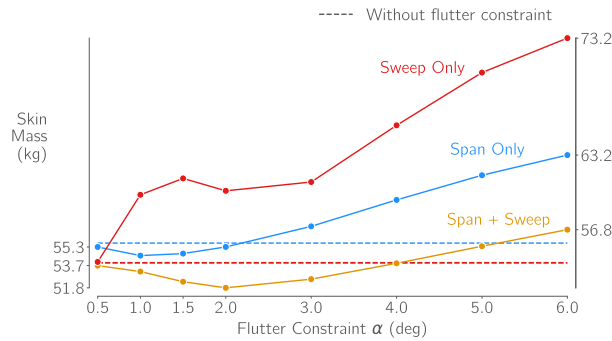
(b) Lift-to-drag ratio



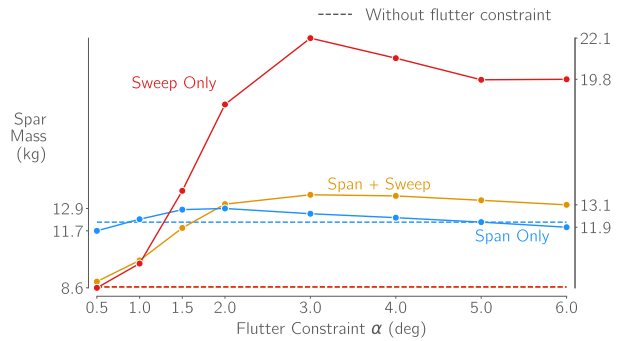
(c) Semispan



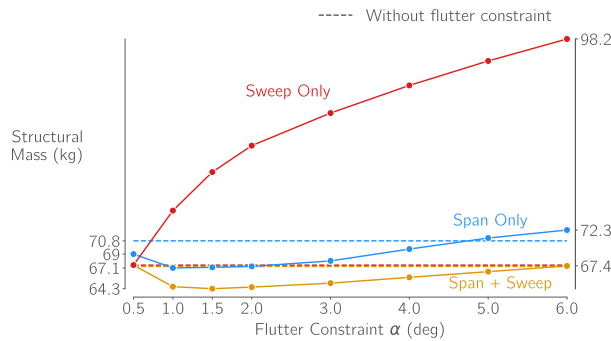
(d) Sweep angle



(e) Wingbox skin mass



(f) Wingbox spar mass



(g) Total wingbox mass

Figure 10: Trends in the functions of interest for each optimized wing design compared across the three geometric parameterizations.

6 Concluding Remarks

If MDO is to be a practical approach for designing the energy-efficient, very flexible aircraft of the future, modeling and constraining geometrically nonlinear flutter behaviors is critical. This paper presented a framework incorporating a low-order geometrically nonlinear flutter constraint into high-fidelity, gradient-based aerostructural optimization problems. The framework uses a built-up wingbox FEM coupled with RANS CFD to accurately predict quantities that require detailed models, such as cruise range and peak stress levels. The built-up FEM is then reduced to a beam representation coupled to potential-flow, thin airfoil theory to predict geometrically nonlinear flutter onset at tractable cost for large-scale optimization.

This work extended the framework’s capabilities to include a consistent geometric parameterization of both wing representations. The adjoint method was used to efficiently compute derivatives of the geometrically nonlinear flutter constraint with respect to large numbers of detailed design variables, which were verified with finite differences. The framework was demonstrated by optimizing the structural sizing and planform geometry of a high-aspect-ratio wing to maximize its cruise range, subject to maneuver stress and flutter constraints. The effect of the flutter constraint on optimum designs was shown to highly depend on the in-flight wing deflections. The flutter constraint was not limiting for low deflection levels, and the optimization results were close to those obtained without considering flutter. At increasing angles of attack, geometrically nonlinear effects resulted in a 3-DOF flutter mechanism due to the coupling of in-plane bending with torsion and out-of-plane bending. The optimizer prevented this mechanism by decreasing span and increasing wingbox thicknesses to limit deflections, leading to a loss in range of up to 5.9%. Comparing optimized designs achieved with different planform parameterizations showed that varying sweep and thickness results in more efficient flutter-safe designs at lower deflection levels. At the same time, span reduction is needed to prevent flutter for higher deflection levels. This work is a step toward bringing geometrically nonlinear flutter analyses into multidisciplinary design optimization of practical aircraft configurations, helping enable the next generation of higher-energy-efficiency aircraft.

Acknowledgments

The material in this paper is based upon work supported by Airbus under the frame of the Airbus-Michigan Center for Aero-Servo-Elasticity of Very Flexible Aircraft. The authors thank Patrick Teufel (Airbus Operations GmbH) for his technical monitoring of this effort and feedback on the manuscript. This research was supported in part through computational resources and services provided by Advanced Research Computing at the University of Michigan, Ann Arbor.

References

- [1] Afonso, F., Vale, J., Oliveira, É., Lau, F., and Suleman, A., “A review on Non-Linear Aeroelasticity of High Aspect-Ratio Wings,” *Progress in Aerospace Sciences*, Vol. 89, No. Supplement C, 2017, pp. 40–57. doi:10.1016/j.paerosci.2016.12.004.
- [2] Jonsson, E., Riso, C., Lupp, C. A., Cesnik, C. E. S., Martins, J. R. R. A., and Epureanu, B. I., “Flutter and Post-Flutter Constraints in Aircraft Design Optimization,” *Progress in Aerospace Sciences*, Vol. 109, 2019, p. 100537. doi:10.1016/j.paerosci.2019.04.001.
- [3] Patil, M. J., Hodges, D. H., and Cesnik, C. E. S., “Limit-Cycle Oscillations in High-Aspect-Ratio Wings,” *Journal of Fluids and Structures*, Vol. 15, No. 1, 2001, pp. 107–132.
- [4] Su, W., and Cesnik, C. E. S., “Nonlinear Aeroelasticity of a Very Flexible Blended-Wing-Body Aircraft,” *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1539–1553. doi:10.2514/1.47317.
- [5] Tang, D., and Dowell, E. H., “Experimental and Theoretical Study on Aeroelastic Response of High-Aspect-Ratio Wings,” *AIAA Journal*, Vol. 39, No. 8, 2001, pp. 1430–1441.
- [6] Drachinsky, A., Avin, O., Raveh, D. E., Ben-Shmuel, Y., and Tur, M., “Flutter Tests of the Pazy Wing,” *2022 AIAA SciTech Forum*, San Diego, CA and Online, 2022, pp. 1–19. doi:10.2514/6.2022-2186, aIAA-2022-2186.
- [7] Cesnik, C. E. S., Palacios, R., and Reichenbach, E. Y., “Reexamined Structural Design Procedures for Very Flexible Aircraft,” *Journal of Aircraft*, Vol. 51, No. 5, 2014, pp. 1580–1591. doi:10.2514/1.C032464.
- [8] Variyar, A., Economon, T. D., and Alonso, J. J., “Design and Optimization of Unconventional Aircraft Configurations with Aeroelastic Constraints,” Grapevine, TX, 2017. doi:10.2514/6.2017-0463, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2017-0463>.
- [9] Xie, C., Meng, Y., Wang, F., and Wan, Z., “Aeroelastic Optimization Design for High-Aspect-Ratio Wings with Large Deformation,” *Shock and Vibration*, 2017, pp. 1–16.
- [10] Werter, N. P., and De Breuker, R., “A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation,” *Composite Structures*, Vol. 158, 2016, pp. 369–386. doi:10.1016/j.compstruct.2016.09.044, URL <http://dx.doi.org/10.1016/j.compstruct.2016.09.044>.
- [11] Rajpal, D., Gillebaart, E., and De Breuker, R., “Preliminary aeroelastic design of composite wings subjected to critical gust loads,” *Aerospace Science and Technology*, Vol. 85, 2019, pp. 96–112. doi:10.1016/j.ast.2018.11.051, URL <https://doi.org/10.1016/j.ast.2018.11.051>.
- [12] Bordogna, M. T., Lancelot, P., Bettebghor, D., and De Breuker, R., “Aeroelastic tailoring for static and dynamic loads with blending constraints,” *International Forum on Aeroelasticity and Structural Dynamics*, IFASD, Como, Italy, 2017.
- [13] Kenway, G. K. W., and Martins, J. R. R. A., “Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration,” *Journal of Aircraft*, Vol. 51, No. 1, 2014, pp. 144–160. doi:10.2514/1.C032150.
- [14] Brooks, T. R., Martins, J. R. R. A., and Kennedy, G. J., “Aerostructural Trade-offs for Tow-steered Composite Wings,” *Journal of Aircraft*, Vol. 57, No. 5, 2020, pp. 787–799. doi:10.2514/1.C035699.

- [15] Bons, N. P., He, X., Mader, C. A., and Martins, J. R. R. A., “Multimodality in Aerodynamic Wing Design Optimization,” *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 1004–1018. doi:10.2514/1.J057294.
- [16] He, S., Jonsson, E., Mader, C. A., and Martins, J. R. R. A., “Aerodynamic Shape Optimization with Time Spectral Flutter Adjoint,” *2019 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019. doi:10.2514/6.2019-0697.
- [17] He, S., “Aerodynamic Shape Optimization using a Time-Spectral Approach for Limit Cycle Oscillation Prediction,” Ph.D. thesis, University of Michigan, 2020.
- [18] Jacobson, K. E., and Stanford, B. K., “Flutter-Constrained Optimization with the Linearized Frequency-Domain Approach,” *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*, 2022. doi:10.2514/6.2022-2242.
- [19] Jonsson, E., Mader, C. A., Kennedy, G. J., and Martins, J. R. R. A., “Computational Modeling of Flutter Constraint for High-Fidelity Aerostructural Optimization,” *2019 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019. doi:10.2514/6.2019-2354.
- [20] Jonsson, E., “High-fidelity Aerostructural Optimization of Flexible Wings with Flutter Constraints,” Ph.D. thesis, University of Michigan, 2020.
- [21] Lupp, C. A., and Cesnik, C. E. S., “A Gradient-Based Flutter Constraint Including Geometrically Nonlinear Deformations,” *2019 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, San Diego, California, 2019.
- [22] Lupp, C. A., Cesnik, C. E. S., Beran, P., Deaton, J., and Easterling, D., “Including Geometrically Nonlinear Flutter Constraints in High Fidelity Aircraft Optimization,” *International Forum on Aeroelasticity and Structural Dynamics*, IFASD, Savannah, Georgia, 2019.
- [23] Cestino, E., Frulla, G., Catelani, D., and Linari, M., “Numerical Simulation and Experimental Validation of Slender Wings Flutter Behaviour,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 16, 2019, pp. 5913–5928. doi:10.1177/0954410019879820.
- [24] Ritter, M., Hilger, J., and Zimmer, M., “Static and Dynamic Simulations of the Pazy Wing Aeroelastic Benchmark by Nonlinear Potential Aerodynamics and Detailed FE Model,” *2021 AIAA SciTech Forum*, 2021, pp. 1–16. doi:10.2514/6.2021-1713.
- [25] Drachinsky, A., and Raveh, D. E., “Nonlinear Aeroelastic Analysis of Highly Flexible Wings using the Modal Rotation Method,” *AIAA Journal*, 2022, pp. 1–13. doi:10.2514/1.j061065, article in advance.
- [26] Cea, A., and Palacios, R., “Geometrically Nonlinear Effects on the Aeroelastic Response of a Transport Aircraft Configuration,” *Journal of Aircraft*, 2022, pp. 1–16. doi:10.2514/1.C036740.
- [27] Jonsson, E., Riso, C., Monteiro, B. B., Gray, A. C., Martins, J. R. R. A., and Cesnik, C. E. S., “High-Fidelity Gradient-Based Wing Structural Optimization Including a Geometrically Nonlinear Flutter Constraint,” *AIAA Journal*, 2023. doi:10.2514/1.J061575, to appear.

- [28] Skujins, T., and Cesnik, C. E. S., “Reduced-Order Modeling of Unsteady Aerodynamics Across Multiple Mach Regimes,” *Journal of Aircraft*, Vol. 51, No. 6, 2014, pp. 1681–1704. doi:10.2514/1.C032222.
- [29] Riso, C., Sanghi, D., Cesnik, C. E. S., Vetrano, F., and Teufel, P., “Parametric Roll Maneuverability Analysis of a High-Aspect-Ratio-Wing Civil Transport Aircraft,” *AIAA SciTech Forum*, Orlando, FL, 2020, pp. 1–23. doi:10.2514/6.2020-1191, aIAA-2020-1191.
- [30] Riso, C., and Cesnik, C. E. S., “Correlations Between UM/NAST Nonlinear Aeroelastic Simulations and the Pre-Pazy Wing Experiment,” *AIAA SciTech Forum*, Online, 2021, pp. 1–24. doi:10.2514/6.2021-1712, aIAA-2021-1712.
- [31] Su, W., and Cesnik, C. E. S., “Strain-Based Geometrically Nonlinear Beam Formulation for Modeling Very Flexible Aircraft,” *International Journal of Solids and Structures*, Vol. 48, No. 16–17, 2011, pp. 2349–2360. doi:10.1016/j.ijsolstr.2011.04.012.
- [32] Hinton, E., Rock, T., and Zienkiewicz, O. C., “A note on mass lumping and related processes in the finite element method,” *Earthquake Engineering & Structural Dynamics*, Vol. 4, No. 3, 1976, pp. 245–249. doi:https://doi.org/10.1002/eqe.4290040305.
- [33] Medeiros, R. R., Cesnik, C. E., Stodieck, O., Calderon, D. E., Cooper, J. E., and Coetzee, E. B., “Comparison of structural model reduction methods applied to a large-deformation wing box,” *Aeronautical Journal*, Vol. 125, No. 1292, 2021, pp. 1687–1709. doi:10.1017/aer.2021.73.
- [34] Riso, C., and Cesnik, C. E. S., “Impact of Low-Order Modeling on Aeroelastic Prediction for Very Flexible Wings,” *Journal of Aircraft*, 2023. doi:10.2514/1.C036869, to appear.
- [35] Peters, D. A., Hsieh, M. C. A., and Torrero, A., “A State-Space Airloads Theory for Flexible Airfoils,” *Journal of the American Helicopter Society*, Vol. 52, No. 4, 2007, pp. 329–342. doi:10.4050/JAHS.52.329.
- [36] Kreisselmeier, G., and Steinhauser, R., “Systematic Control Design by Optimizing a Vector Performance Index,” *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems, Zurich, Switzerland*, Elsevier, Zurich, Switzerland, 1979. doi:10.1016/S1474-6670(17)65584-8.
- [37] Wrenn, G. A., “An Indirect Method for Numerical Optimization Using the Kreisselmeier–Steinhauser Function,” Tech. Rep. CR-4220, NASA Langley Research Center, Hampton, VA, 1989.
- [38] Poon, N. M. K., and Martins, J. R. R. A., “An Adaptive Approach to Constraint Aggregation Using Adjoint Sensitivity Analysis,” *Structural and Multidisciplinary Optimization*, Vol. 34, No. 1, 2007, pp. 61–73. doi:10.1007/s00158-006-0061-7.
- [39] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, AIAA, Fort Worth, TX, 2010. doi:10.2514/6.2010-9231.
- [40] Martins, J. R. R. A., and Ning, A., *Engineering Design Optimization*, Cambridge University Press, Cambridge, UK, 2021. doi:10.1017/9781108980647, URL <https://mdobook.github.io>.

- [41] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations,” *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935–951. doi:10.2514/1.J052255.
- [42] Kennedy, G. J., and Martins, J. R. R. A., “A parallel aerostructural optimization framework for aircraft design studies,” *Structural and Multidisciplinary Optimization*, Vol. 50, No. 6, 2014, pp. 1079–1101. doi:10.1007/s00158-014-1108-9.
- [43] Kenway, G. K. W., Mader, C. A., He, P., and Martins, J. R. R. A., “Effective Adjoint Approaches for Computational Fluid Dynamics,” *Progress in Aerospace Sciences*, Vol. 110, 2019, p. 100542. doi:10.1016/j.paerosci.2019.05.002.
- [44] Yildirim, A., Kenway, G. K. W., Mader, C. A., and Martins, J. R. R. A., “A Jacobian-free approximate Newton–Krylov startup strategy for RANS simulations,” *Journal of Computational Physics*, Vol. 397, 2019, p. 108741. doi:10.1016/j.jcp.2019.06.018.
- [45] Secco, N., Kenway, G. K. W., He, P., Mader, C. A., and Martins, J. R. R. A., “Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization,” *AIAA Journal*, Vol. 59, No. 4, 2021, pp. 1151–1168. doi:10.2514/1.J059491.
- [46] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “Open-MDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. doi:10.1007/s00158-019-02211-z.
- [47] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP algorithm for large-scale constrained optimization,” *SIAM Journal of Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:10.1137/S1052623499350013.
- [48] Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., “pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems,” *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. doi:10.21105/joss.02564.
- [49] Perez, R. E., Jansen, P. W., and Martins, J. R. R. A., “pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization,” *Structural and Multidisciplinary Optimization*, Vol. 45, No. 1, 2012, pp. 101–118. doi:10.1007/s00158-011-0666-3.
- [50] Hodges, D. H., and Dowell, E. H., “Nonlinear equations of motion for the elastic bending and torsion of twisted nonuniform rotor blades,” Tech. rep., 1974. URL <https://ntrs.nasa.gov/citations/19750005242>.