Multidisciplinary Design Optimization of a Transonic Truss-Braced Wing using Physics-Based Models

Timothy R. Brooks*, Kenneth T. Moore †, and Eliot D. Aretskin-Hariton‡
NASA Glenn Research Center, Cleveland, OH, 44135, USA

Irian Ordaz§ and Jason Kirk¶
NASA Langley Research Center, Hampton, VA, 23681, USA

Shifting economic forces have placed pressure on the aviation industry to produce future aircraft designs with substantial performance improvements over those flying today. To meet these ambitious goals, new concepts are being considered that diverge from the traditional tube-and-wing design, such as the Transonic Truss-Braced Wing (TTBW). While this concept offers potential performance benefits over its traditional counterpart, it will also feature more coupling between various physics disciplines. These couplings must be considered during the design of the concept to have an accurate understanding of its performance and feasibility. This unique challenge provides a ripe opportunity for the application of physics-based multidisciplinary design optimization tool sets. The work presented here couples the mission analysis capability of Aviary with OpenAeroStruct for aerodynamic analysis and TACS for structural analysis. Using these tools, a system level design optimization on a TTBW concept minimizing design fuel burn is performed. This optimization includes design variables for mission trajectory, wing geometry and structural sizing variables, allowing for a more holistic design consideration.

I. Nomenclature

C_{D_0}	=	aircraft skin friction drag coefficient
CG	=	aircraft center of gravity
C_{M_y}	=	aircraft pitching moment coefficient
C_{f_e}	=	skin friction regression constant
b_{ob}	=	outboard span
d_j	=	distance state
D	=	aircraft total drag
g_{boundary}	=	phase linkage constraint
h_j	=	altitude state
KS_{fail}	=	aggregated structural failure constraint
L	=	aircraft total lift
R	=	residual constraint
S_{ref}	=	aircraft reference area
S_{wet}	=	aircraft wetted area
t	=	phase time
t_0	=	phase initial time
W_j	=	aircraft weight state
W_{fuel}	=	aircraft fuel weight
W_{gross}	=	aircraft gross weight
x^{struct}	=	Wing thickness design variables
α_j	=	angle of attack state

*Systems Engineer, Banner Quality Management Inc.
†Senior Analyst, Banner Quality Management Inc.
‡Research Aerospace Engineer, Propulsion Systems Analysis Branch
§Aerospace Engineer, Aeronautics Systems Analysis Branch
¶Aerospace Engineer, Aeronautics Systems Analysis Branch

η_j = tail incidence state
Δt = phase duration
θ_i^{wing} = wing span-wise twist vector

II. Introduction

In order to keep pace with ever-changing economic incentives, the U.S. aviation industry is working to significantly improve commercial aircraft efficiency by 2050. To accomplish this goal, NASA has invested in various initiatives such as the Sustainable Flight National Partnership (SFNP) [1] and the Advanced Aircraft Concepts for Environmental Sustainability (AACES) 2050 [2] efforts to mature and integrate key technologies for next-generation subsonic transports, as well as explore innovative aircraft concepts that can continue to improve the competitive performance within the aviation industry.

In support of the SFNP, NASA's Model-Based Systems Analysis and Engineering (MBSA&E) effort is working to develop a systems-level, digital integration across SFNP projects. This architecture will support the assessment, advancement, and adoption of ultra-efficient technologies for subsonic transport aircraft concepts with 2030 entry into service and beyond. The main objectives of this effort include the development of an open, cross-project, cross-program, external-capable MBSA&E framework building off NASA's investments and capabilities across various NASA projects. The effort will also support coordinated, integrated systems analysis studies including the development of common, open, reference, and vision vehicle models and technology benefit assessments and sensitivity studies informed by the SFNP demos.

The push for increased efficiency has sparked interest in novel and advanced concepts, which often introduce new challenges due to tighter coupling between design disciplines — challenges not encountered with their traditional counterparts. The Transonic Truss-Braced Wing (TTBW) is one such concept, offering greater aerodynamic performance due to its large span, but at the cost of more structural complexity due to the design of the wing and truss members [3]. This makes the concept an ideal candidate for Multidisciplinary Design Analysis and Optimization (MDAO).

To date, there has already been great interest in applying MDAO to the TTBW concept. In work by Aretskin-Hariton et al. [4] a fuel burn optimization for the concept was presented in which the optimizer was allowed to adjust: mission trajectory, electrified turbine size, electric motors size for climb assist, and battery capacity. This work coupled a number of physics-based disciplinary analyses together including an engine analysis using pyCycle [5] and a drag table built using aerodynamic analyses from VSPAERO [6]. While this study did include physics-based analyses, the consideration for aircraft design variables during the optimization were limited and did not include aerodynamic design optimization. This limited the benefits that could be captured through optimization. Leader et al. [7] expanded on this work by increasing the fidelity of the electrical analysis used to size the battery model for a hybrid-electric version of the TTBW model. Work by Fazal et al. [8] showed the capability for integrating uncertainty quantification into the TTBW design framework and demonstrated the importance of these considerations within the design process.

Past works on other concepts have further demonstrated the benefit that comes with considering aircraft and mission design under the same optimization. Hwang et al. [9], using the NASA Common Research Model [10], conducted a coupled aircraft design, airline flight route allocation, and mission trajectory optimization problem to maximize airline profit. The study found an additional 2.3% improvement in airline profit relative to a design in which the aircraft design and mission trajectory were optimized separately. Kaneko and Martins [11] performed a simultaneous design and trajectory optimization of two unmanned aerial vehicle concepts based on a series of aerostructural analyses and considering several optimization coupling architectures. In their work, they found an additional reduction of 4.2% in energy required to climb when compared to a design in which vehicle design and trajectory were optimized sequentially.

The present work will expand upon the work completed by Aretskin-Hariton et al. [4] in several ways. First, physics-based aerodynamic and structural analyses are included in the optimization problem. This allows for system-level metrics and trends to be more accurately resolved within the optimization, enabling a more thorough exploration of the concept's design space. Secondly, in addition to the original mission optimization problem, aircraft geometric and structural design variables are added to parameterize the wing design. This allows for both the aircraft design and mission to be simultaneously optimized for greater synergistic benefits.

III. Aircraft Model

The aircraft model used in this work is a non-proprietary concept referred to as the Transonic Truss-Braced Wing Tech Collector. The concept was developed by NASA to allow future vehicle technology studies, open publication of

results, and easier collaboration with parties outside of NASA. The model is based off of the Boeing SUGAR aircraft concept [3], a next-generation aircraft concept with an target for entry into service of 2035. The vehicle concept is shown in Fig. 1 and top-level design parameters for the aircraft can be found in Table 1.

Table 1 TTBW specifications.

Parameter	Value	Units
Takeoff gross weight	145,000	lb
Operational empty weight	88,000	lb
Passengers	154	
Design range	3,200	nm
Cruise Mach	0.8	
Cruise altitude	38,000	ft
Aspect ratio	19.7	
Reference area	1,475	ft^2
Span	170	ft
Sweep	21.8	deg

IV. Tool Background

The tools used in the current work are: Aviary [12], OpenAeroStruct [13], TACS [14], FunToFEM [15], pyGeo [16], and MPhys [17]. Aviary serves as the top-level backbone aircraft design framework, built on OpenMDAO [18], into which all other analysis tools are wrapped. OpenAeroStruct serves as a medium-fidelity aerodynamic model to be used during mission analysis. TACS is used to perform structural analysis of the wing and truss structure. The load and displacement interpolation between OpenAeroStruct and TACS is handled using the FunToFEM library, allowing aeroelastic analysis to be performed within the optimization. pyGeo serves as a geometry engine that ties top-level geometric design parameters to the computational mesh used by OpenAeroStruct and TACS for the aeroelastic analysis. Finally, the four libraries used for aeroelastic analysis are tied together into OpenMDAO using the MPhys framework. The details of each tool will be elaborated below.

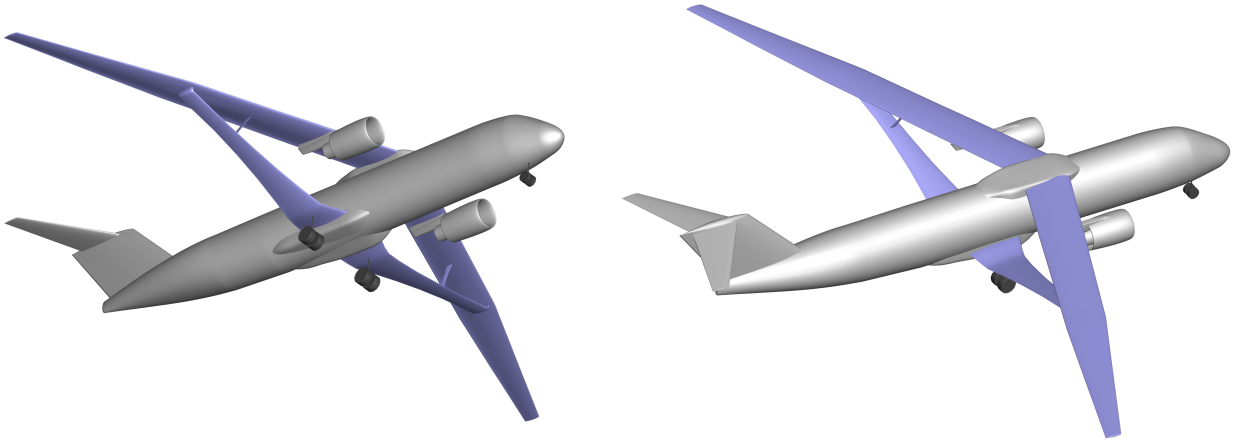

Fig. 1 TTBW Tech Collector geometry model [8].

A. Aviary

Aviary [*] is an open-source framework for performing aircraft conceptual design in an MDAO context [12]. It leverages the fundamental methods from two NASA tools: FLOPS (Flight Optimization System) [19] and GASP (General Aviation Synthesis Program) [20], both of which use low-fidelity correlations and empirical relationships that are relevant for existing traditional aircraft designs. The TTBW aircraft described here represents a deviation from the existing data, and Aviary provides a way to replace any existing calculation or subsystem with a user-provided tool that is more accurate or even higher fidelity. For certain key disciplines, the internal calculations are replaced with the tools described in the subsequent sections; for the remainder, which is primarily the subsystem mass buildup apart from the wing, the FLOPS correlations in Aviary are used.

Aviary is written entirely in Python with analytic derivatives and is built on top of the OpenMDAO framework [18]. This gives it access to the optimization capabilities of OpenMDAO, in particular, the support for efficient total derivative calculation across a large heterogeneous model for use in gradient-based optimization. Some of the efficiency comes from the use of graph-coloring algorithms to exploit model sparsity, so that independent linear systems can be simultaneously solved.[21]

To simulate the trajectory, Aviary uses Dymos, which is another tool built in OpenMDAO that can integrate Ordinary Differential Equations (ODE) using a variety of implicit and explicit methods [22]. While some methods (i.e, shooting methods) start at an initial point in time and integrate to a final time, other methods (collocation methods) solve for the entire time history of the integration variables, utilizing the optimizer to drive all of the defect residuals to zero. Aviary provides the most support for the latter, and it is used in this study. OpenMDAO's gradient support and graph-based coloring enable the collocation problem to be solved efficiently.

Aviary includes two levels of fidelity for the aircraft equations of motion – height-energy and the two-degree of freedom equations. For this problem the slightly-lower fidelity height-energy method was chosen due to its robustness in past works on the model [4]. The height-energy method uses a single differential equation that describes the aircraft motion using energy (potential and kinetic) as the time integration variable [23]. The trajectory to be simulated consists of three simple flight phases: climb, cruise, and descent, with boundary conditions that come from the TTBW's design mission. Each is implemented as a Phase in Dymos, and all three phases use the energy equation in the ODE, which contains the differential equation that Dymos solves. Other disciplines such as aerodynamics and propulsion are included in the phase to compute other quantities such as thrust or drag that contribute to the equations at the integration points. None of the disciplines used in this study provide additional differential equations at present, though the capability is supported.

B. OpenAeroStruct

OpenAeroStruct [†] is an open-source library for aerostructural analysis and optimization of flexible-winged aircraft [13]. The library is written using the OpenMDAO framework [18]. OpenAeroStruct features a Vortex Lattice Method (VLM) aerodynamic representation of the wing that can be optionally coupled to a 6 degree of freedom finite-element beam representation of the wingbox structure. OpenAeroStruct features adjoints for both its aerodynamic and structural solvers which allows the models to be integrated in larger gradient-based MDO problems using OpenMDAO. For this work, only OpenAeroStruct's aerodynamic model was utilized. This is due to the fact that it is not currently possible to model multiple connected lifting surfaces (i.e. the wing and strut) through the library's structural model.

Minor modifications were made to the library to improve memory performance when calling multiple OpenAeroStruct instances on the same machine. These modifications removed the nominal Aerodynamic Influence Coefficient (AIC) matrix assembly routine required to solve the linear aerodynamic potential equation. This routine was instead replaced with a memory-efficient matrix-free linear solver that utilizes Scipy's [24] Generalized Minimum Residual (GMRES) solver [25]. The core of OpenAeroStruct's derivatives are analytic and additional derivatives necessary for this work were computed using the JAX Automatic Differentiation library [26].

The VLM mesh used to model the TTBW aircraft in this work is shown in Fig 2. The model includes the aerodynamic influence of the wing, strut, and horizontal tail. The aerodynamic effect of the jury strut is neglected. Due to the symmetrical nature of the analyses considered in this work, only half the aircraft is modeled for computational efficiency. The mesh features 259 vortex lattice panels in total. Since VLM only captures inviscid drag, in addition to the drag generated by OpenAeroStruct, a viscous skin friction drag correction is computed using an empirical model based on

[*] `https://www.github.com/openmdao/aviary`

[†] `https://www.github.com/mdolab/openaerostruct`

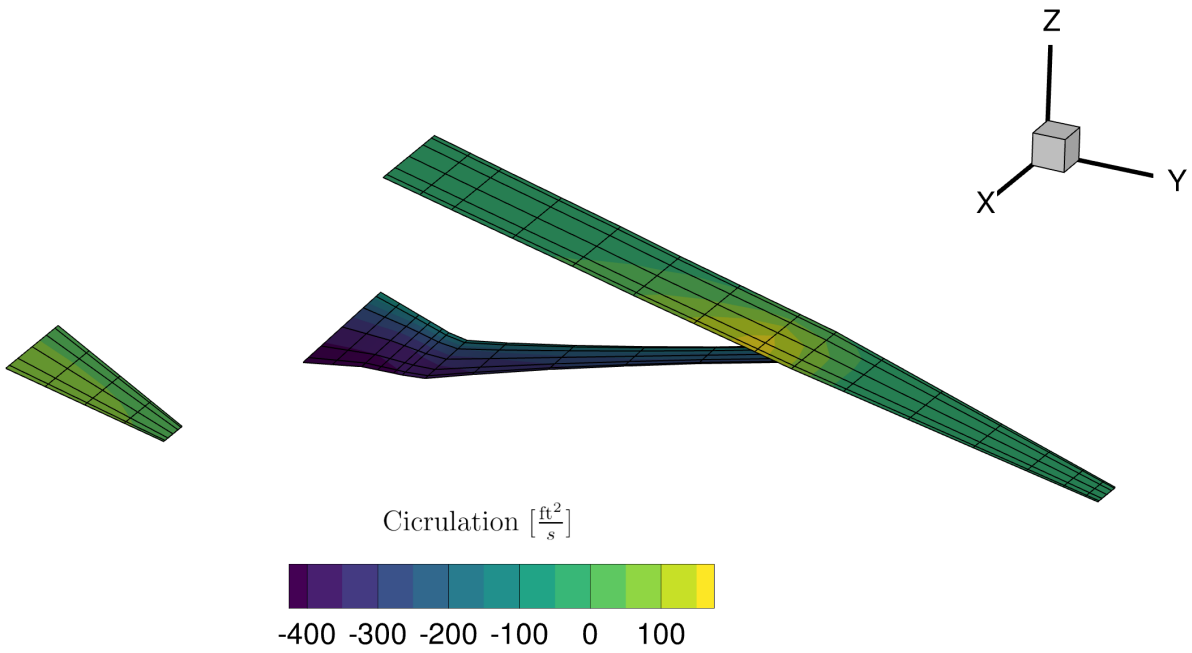

Fig. 2 TTBW half-mesh used in OpenAeroStruct aerodynamic VLM model

aircraft wetted area [27],

$$C_{D_0} = C_{f_e} \frac{S_{\text{wet}}}{S_{\text{ref}}}, \tag{1}$$

where C_{f_e} is chosen to be 0.0030, a value historically representative of civil transport type aircraft.

C. TACS

TACS [‡] is an open-source parallel finite-element analysis library for structural design and optimization [14]. The library is capable of solving multiple structural analysis problems, including static, dynamic, and modal and buckling eigenvalue problems. Each of these capabilities includes a relevant adjoint implementation allowing for design sensitivities to be readily computed. The core of the code is written in C++ and analytic derivatives for speed and features a Python user interface for ease of use.

The structural model that is used for this work is shown in Fig 3. Similar to the VLM model, the FEM model is a half model of the vehicle. The model includes the main wingbox structure and main and jury strut. The main wingbox structure (skins, ribs, and spars) is modeled using shell elements, the main strut using a series of beam elements, and the jury strut a rod element. The wingbox is modeled based on aluminum 2024-T3 series material shown in Tab 2. The joints at the intersection of each member are modeled as pinned using rigid body elements. The model also includes point mass elements for the engine and fuel which are attached to the main structure using RBE3 elements [28]. Unlike the aerodynamic model, the horizontal tail is not included in the structural model. The model contains 9,196 elements and 53,766 structural degrees of freedom in total. Structural thickness design variables are added for each colored panel on the wingbox as well as span-wise segments of the main strut, allowing each to be sized by the optimizer.

D. FunToFem

FunToFEM [§] is an open-source aeroelastic coupling library. The library provides common load and displacement transfer schemes that can be used to couple various aerodynamic and structural solvers. Similar to TACS the core of the code is written in C++ for speed, features analytic derivatives, and features a Python user interface for easier integration

[‡] `https://www.github.com/smdogroup/tacs`
[§] `https://www.github.com/smdogroup/funtofem`

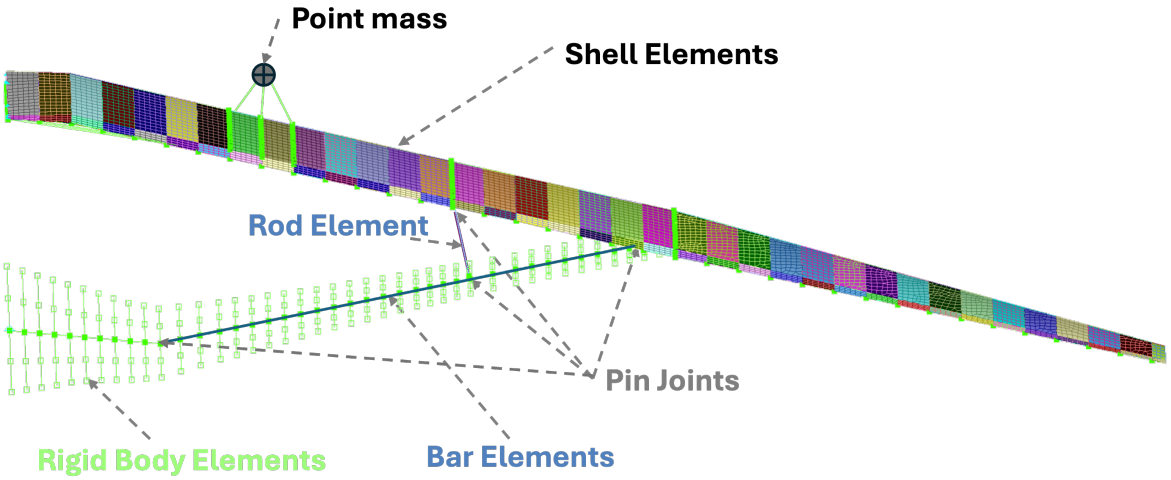

Fig. 3 TTBW half-mesh used in TACS structural FEM model

Table 2 Wing box structural specifications

Parameter	Value	Units
Youngs' modulus (E)	10,600	kpsi
Poisson ratio (ν)	0.33	
Yield strength (σ_y)	47.0	kpsi
Density (ρ)	0.1004	lb/in^3

with other codes. For this work the library's MELD formulation [15] is utilized, a matching-based extrapolation scheme that uses weighted least-squares fitting and singular value decomposition to transfer loads and displacements between aerodynamic and structural meshes. This allows for accurate, smooth, and adjoint-consistent coupling independent of aerodynamic and structural mesh connectivity.

We use MELD to couple together the structural representation of the wing and strut with their aerodynamic counterparts. MELD's load transfer scheme is limited to nodal forces; nodal moments are not directly supported. Despite this, the method still requires the resultant forces and moments arising from the distribution of aerodynamic loads to be consistent between the aerodynamic and structural models. This can lead to issues in models featuring co-linear elements like those used on the strut, since not all moments can be reacted through coupling forces alone [15]. To avoid this issue, these regions are expanded out using RBE2 elements [28] (shown in green in Fig 3) to help with transferring moments back to the strut elements. Since the horizontal tail is neglected in the structural model, it is treated as rigid in the aerostructural model.

E. pyGeo

pyGeo [¶] is an open-source Python library for handling geometry parameterization in MDAO [16]. The library has multiple interfaces for geometry parameterization, including: free-form deformation volume [29], OpenVSP [6], and Engineering Sketch Pad [30]. In this work an OpenVSP model (seen in Fig. 1) is utilized to define the geometry of the aircraft. pyGeo is responsible for updating the spatial locations of the nodes on the VLM and FEA meshes as the OpenVSP geometry design variables are perturbed during optimization. Since OpenVSP does not currently support geometric design sensitivities, which are necessary for gradient-based optimization, pyGeo must finite-difference across the geometry engine to estimate the sensitivities of the VLM nodal locations with respect to OpenVSP geometry parameters. Figure 4 illustrates the design variables used to parameterize the geometry in this work. These include

[¶] `https://www.github.com/mdolab/pygeo`

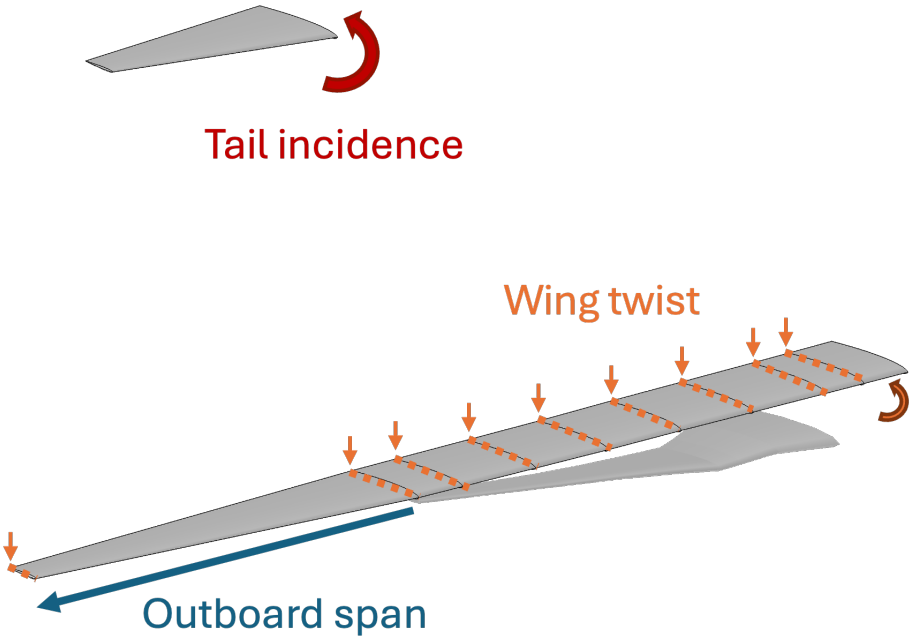

Fig. 4 TTBW geometric parameterization used for study

wing twist, outboard span, and tail incidence. The wing twist is defined at 9 stations along the span of the wing. We don't include the twist at the root of the wing to avoid redesigning the wing-body fairing. The outboard span is defined as the portion of the wing beyond the strut junction. This means that the absolute location of the wing-strut junction remains fixed for this study. Finally, the horizontal tail incidence can be controlled by rotating the surface about its local span axis. For this work the geometry of the strut is left untouched.

F. MPhys

OpenAeroStruct, TACS, FunToFem, and pyGeo are all connected using MPhys [‖] [17], an open-source modular multi-physics library built on top of OpenMDAO. This library standardizes interfaces for coupling multidisciplinary simulation models, making it easier to assemble complex systems for design optimization. It provides a flexible, hierarchical model structure along with automated derivative computation using the adjoint method, enabling efficient gradient-based optimization even for large-scale, high-fidelity aerospace applications. MPhys reduces development effort by allowing interchangeable solvers and standardized data flow between disciplines, facilitating rapid prototyping and reliable integration for aerostructural, aeropropulsive, and other multi-physics design problems. By wrapping these tools together and exposing them to OpenMDAO, each solver's single-disciplinary adjoint formulations can be taken advantage of, which allows OpenMDAO to form the coupled adjoint automatically. This allows optimizations with large numbers of design variables ($O(100)$) to be performed without sacrificing on computational performance.

[‖] `https://www.github.com/openmdao/mphys`

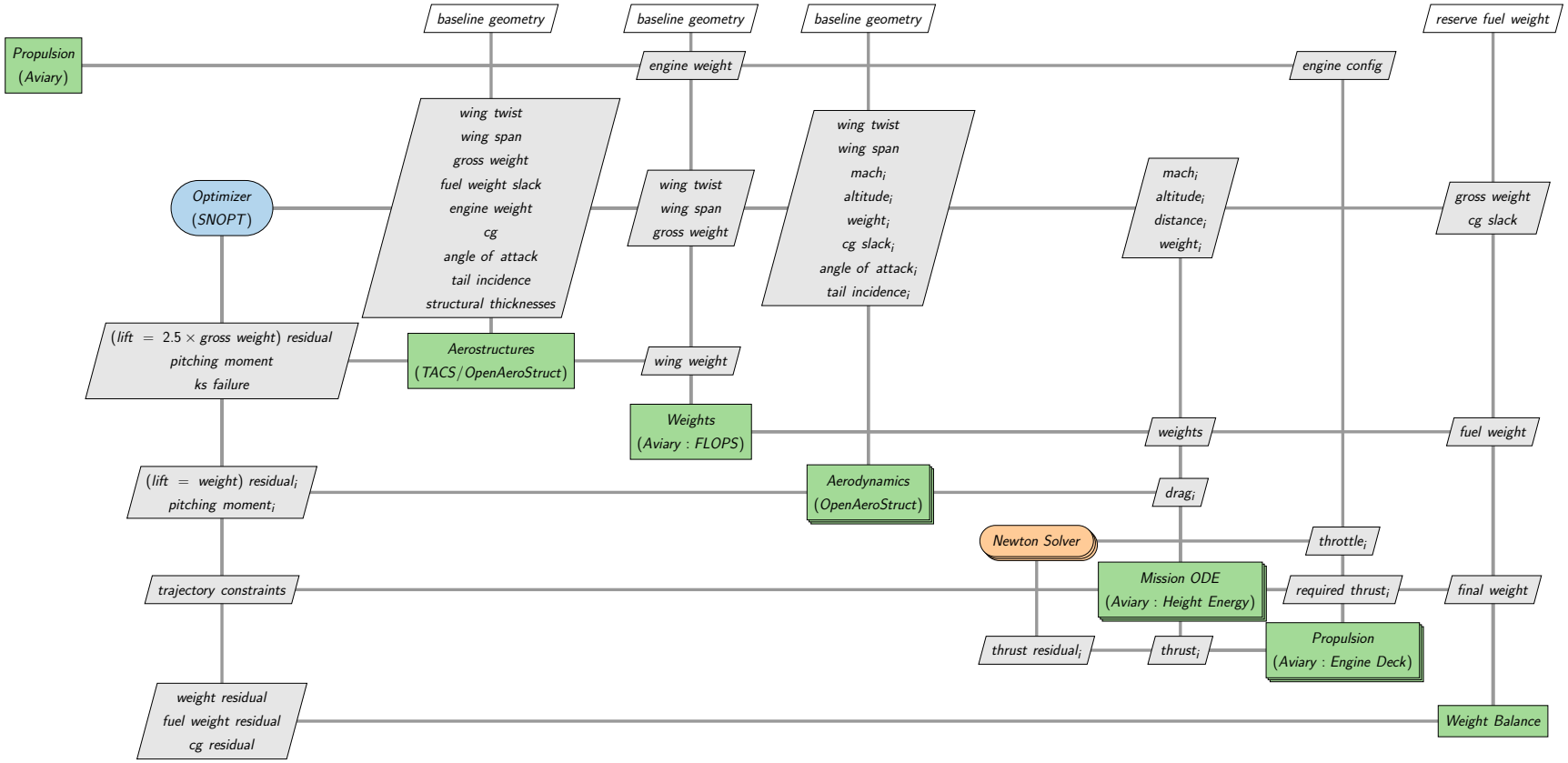

Fig. 5 Top-level XDSM for integrated TTBW model

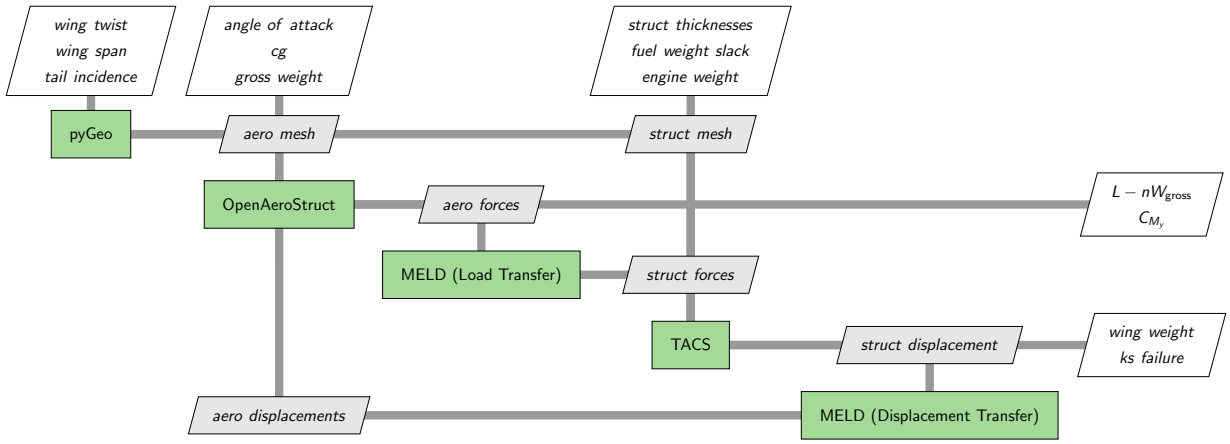

Fig. 6 XDSM breakdown for aerostructural analysis

V. Tool Integration

An eXtended Design Structure Matrix (XDSM) [31] is shown in Fig. 5 which illustrates how the optimization problem is connected across the various disciplines. SNOPT [32] is utilized as the top-level optimizer for this problem. SNOPT is an optimizer that has been designed to efficiently handle large sparse optimizations problems, such as those considered here. In the aircraft sizing problem, the design gross weight and center of gravity of the aircraft are declared as design variables, and constraints are added to balance the fuel computed in the pre-mission detailed mass subsystem against the fuel consumed in the mission and the computed or calculated reserve fuel. This avoids introducing a costly coupling loop over the entire model, which would need to be converged every design iteration. The only subsystem that is outside of the optimization loop is the pre-mission propulsion analysis, which computes the engine mass, and sets a few configuration parameters. Since there are no engine design variables in this problem, it only needs to be run once before the optimizer iteration begins.

Figure 6 shows a more detailed breakdown of the aerostructures discipline block. This discipline performs an aerostructural sizing of the wing based on a 2.5 g pull up maneuver case. The flight condition is defined at a flight Mach of 0.6415 at standard sea level. The analysis begins with a pyGeo component that takes in geometry design variables—wing twist, span, and horizontal tail incidence—as inputs and outputs the updated node locations for the wing jig mesh used in the OpenAeroStruct VLM and TACS FEA solvers. OpenAerostruct then computes the aerodynamic forces for the undeformed wing shape. These aerodynamic forces are processed by the MELD load transfer scheme and interpolated to the nodes of the FEA mesh. The forces are then taken as inputs and used to solve the elastic displacements of the structure. These displacements from the structure are passed through MELD's displacement scheme and interpolated back to the VLM mesh. Finally the VLM is re-solved with the newly deformed wing and the process is iterated until convergence using a Gauss-Seidel solver. Once converged, the lift and pitching moment from the aerodynamic model are computed and provided as constraints for the optimizer. The structural wing weight is then passed to the weights module where the full wing weight, along with other aircraft component weights, are calculated.

The structural wing weight computed by TACS in the aerostructures subsystem is fed into the Aviary weight subsystem, which in turn computes the total weight of the wing, W_{wing}. This is computed using the following equation adapted from FLOPS [33],

$$W_{\text{wing}} = W_{\text{bend}} + W_{\text{shear/cs}} + W_{\text{misc}}, \tag{2}$$

where W_{bending} is the structural wing weight required to sustain bending loads, $W_{\text{shear/cs}}$ is the wing weight associated with shear loading and control surfaces, and W_{misc} is the non-structural weight of the wing (i.e. leading/trailing edges, wing tips, etc.). TACS is responsible for providing the bending weight, W_{bend}, while Aviary computes the remaining terms using historical regressions borrowed from FLOPS. Aviary then uses empirical-based relations to predict each of the remaining weights of the aircraft, culminating in the fuel weight as the final output.

In order to couple OpenAeroStruct's aerodynamic solver to the mission analysis, a direct coupling approach [11] is utilized, in which multiple OpenAeroStruct solver instances are connected to each mission phase, one instance for each discrete time integration point. Each solver instance is passed a weight state from its corresponding mission integration

point that dictates the amount of lift the model must generate. OpenAeroStruct then computes the total drag at each point, which is then used to complete equations of motion for the mission analysis. In this case, these OpenAeroStruct instances are treated as rigid (i.e. no aerostructural coupling). This simplification was made in order to save on computational expense due to the large number flight cases required by the time integration analysis. Furthermore, it is assumed that the time-scales of the ODE integrations are such that the aerodynamics can be approximated using a series of steady analyses. Like the aerostructures model, the current geometric design variables are passed to an internal pyGeo component that then generates an updated mesh for the analyses. Due to the fact that each OpenAeroStruct instance can be solved independently of one another, the process is parallelized by running the cases on multiple processors using an OpenMDAO *ParallelGroup* [18]. This approach differs from the coupling strategy used by Aretskin-Hariton et al. [4], in which VSPAERO was coupled to the mission analysis using a precomputed drag table outside of the optimization loop. This allows the effects aerodynamic/geometric design within the optimization to be considered and gives access to a larger design space.

The mission ODE analysis consists of three Dymos phases: climb, cruise, and descent. The climb and cruise phases are discretized using 3 third-order segments, yielding 12 integration points, while the descent phase has 1 third-order segment, giving 4 points, for a total of 28 integration points. The distance and mass are states in the equations of motion, so there are constraints on the defects between the computed and approximated time derivatives, which the optimizer drives to zero. There are additional boundary and compatibility constraints between each pair of phases, and at the start and the end of the trajectory.

For the mission propulsion, an engine deck provides thrust and fuel burn that are needed during integration of the mission equations of motion. Aviary includes the capability to read a static engine deck and create a meta-model that computes thrust and fuel burn for the altitudes, Mach numbers, and throttle settings that are needed at the evaluation points in the mission. Note that the throttle setting is computed by balancing thrust versus the required thrust computed from the height-energy method. A Newton solver is used to drive this balance to zero, rather than letting the optimizer control the throttle.

The center of gravity (CG) of the aircraft is calculated at every point along the mission using the fuel weight states and aircraft subsystem weights calculated by Aviary. Coarse geometry representations of major subsystems such as cabin, cargo compartment, control surfaces, auxiliary power unit (APU), fuel tanks, etc., are modeled using conformal components in OpenVSP. These subsystem geometry components are mapped to subsystem weights calculated by Aviary and the CG is calculated with a custom CG and inertia analysis component that leverages the OpenVSP Python API to determine the centroid of each subsystem geometry.

VI. Problem Statement

The optimization problem considered here is summarized in Table 3. The objective of the optimization is to minimize the aircraft fuel burn by simultaneously sizing the aircraft gross weight while optimizing the trajectory, wing structure and geometry. As mentioned in the previous section, the trajectory is split into 3 phases: climb, cruise, and descent. In each of these phases, the altitude, distance, and aircraft mass are design variables that are chosen by the optimizer in a way that satisfies the collocation defects for the mass and distance states, the intra-phase linkage constraints, and any initial and final constraints. In addition, the climb and descent phases include monotonic path constraints that prevent the trajectory from descending during climb or climbing during descent; this is done to prevent any oscillations that might occur in the trajectory profile. To close the aircraft, a final weight residual assures that the fuel weight consumed in the mission plus the reserve fuel weight balances the total take-off weight. To prevent introducing an additional analysis coupling loop, a slack variable is added for the fuel weight. This decouples the dependency of the aerostructural solution on the weight of the fuel, which is an output of the weights model. We add a corresponding slack constraint on the fuel weight slack variable to ensure that it is consistent with the calculated amount of fuel. The aerodynamics subsystem introduces an angle of attack and tail incidence design variable for each OpenAeroStruct model instance, because these trimming variables are not computed in the ODE. The optimizer chooses an angle of attack that constrains the lift output from each OpenAeroStruct model to match the target lift for its associated mission point. Likewise, the tail incidence is chosen to constrain the aerodynamic pitching moment about the aircraft center of gravity. Similarly, the aerostructures 2.5 g maneuver case introduces one more angle of attack/tail incidence design variable pair. Structural design variables are also added for sizing the wing and strut, one for each panel shown in 3, 192 in total. In addition to the lift and moment constraints, the aerostructural case also introduces a structural failure constraint to ensure that the structure is adequately sized. The constraint is computed for every element in the FEM based on the von Mises yield criteria with a 1.5 factor of safety. It is normalized to a value of unity such that values

greater than 1.0 signify a violation. All of these element-wise constraints are aggregated into a single scalar value the Kreisselmeier–Steinhauser (KS) aggregation [34, 35]. The span-wise twist of the wing is included as a design variable, with 9 in total. Lastly, a design variable is assigned for the outboard span, which controls the span of the wing from the strut intersection to the wing tip.

VII. Results

In order to provide a baseline for comparison, three modified versions of the optimization problem described in Table 3 are run. The first case freezes the geometry of the design by removing the twist and span design variables. This case is referred to as the *baseline optimized* design and it demonstrates the benefits of optimizing for structural sizing and trajectory only. The second case adds twist design variables in, but still keeps the span fixed. We designate this case the *twist optimized* design and it demonstrates the benefits of optimizing wing geometry without modifying planform. Finally, in the last case span is added back in, meaning all optimization design variables described in Table 3 are utilized. This last case is referred to as the *fully optimized* design and demonstrates the benefits of considering planform within the optimization. Each optimization case takes roughly 48 hours of wall time for optimization. All cases were run on a MacBook Pro M3 Max 2023 using a total of 12 processors. A sizeable portion of the run time is due to the finite-differencing over the OpenVSP model required for the geometric design sensitivities. This is due to the fact that updating the model in between finite difference perturbations is significantly expensive.

The top-level performance metrics for each optimized case can be seen in Table 4. It should come as no surprise that in terms of fuel burn performance, the *fully optimized* design performs the best, followed by the *twist optimized* design, and lastly the *baseline optimized* design. This is due to the larger design space that each case is able to explore relative to the prior. Comparing the *baseline* and *twist optimized* designs first a noticeable improvement in fuel burn can be seen (20,390 lbs vs 19,080 lbs) corresponding to a decrease of roughly 6%. Considering the improvement in L/D of the *twist optimized* designs leads to the conclusion that the fuel burn improvement is due to an increase in aerodynamic efficiency of the design. This improvement in L/D comes at a cost of a heavier wing weight leading to an increase in gross weight of the aircraft. Next, comparing the *fully* and *twist optimized* designs, a negligible benefit in fuel burn is found (19,080 lbs vs 19,070 lbs) corresponding to a decrease of less than 0.1%. The gross weight, wing weight, and L/D are similarly very close between these two designs. Looking at the final optimal span for the *fully optimized* design, only a slight increase in the outboard span is seen, about a foot, which explains why the improvements in the other performance metrics are so modest. Figure 8 shows the normalized lift distribution across the aerodynamic model (wing, strut, and tail) at the start of cruise for both designs. The *twist* and *fully optimized* designs achieve a more elliptical lift distribution to reduce induced drag and minimize fuel burn. In both of these designs this is accomplished by adding washout to the wing through increasing the twist along the span. We see that the twist distribution is fairly smooth with exception of small kink at wing root due to twist not being fixed there by design. Since the *baseline* geometry features an untwisted wing, it is likely that the 6% improvement in fuel burn found in Table 4 represents an overly-optimistic improvement relative to a more realistic baseline design featuring a pre-twisted wing. All three designs also take advantage of passive load alleviation due to the natural bend-twist coupling of swept-back wings. As the loading on the wing increases the wing tips aeroelastically twist down. This leads to a more "bell-shaped" lift distribution during maneuver which reduced bending loads and allows for a lighter wing structure without comprising aerodynamic performance at cruise. The altitude and Mach flight profile over time is shown in Fig. 7. We see only minor differences in the optimal mission for all three designs, mainly in the duration of the climb and descent phases. This result is likely a sign that wing twist and span do not play a large role in the optimal aircraft trajectory for this design. Figure 9 shows the final structural sizing and failure stress contours for each design. From these results it can be seen that the heavier wing weights seen in the *twist* and *fully optimized* designs in Table 4 comes from thicker panels needed to reinforce the structure, particularly near the wing root, for the higher bending loads seen during maneuver (relative to the *baseline* design) in Fig. 8. The failure contours show that, as should be expected, the highest sizing stresses for the wing occur near the root and strut junction due to the reaction of the bending loads.

The results found so far suggest that the design has a low sensitivity with respect to the outboard span. To confirm this, a design sweep over span to investigate the design space around the optimal span was performed. This was completed by fixing the span design variable at different values and re-running the *twist* optimization and plotting the resulting fuel burn against outboard span. The resulting plot, including the position of the nominal twist and fully optimized designs, is shown in Fig. 10. The position of the nominal *twist* and *fully optimized* designs are also annotated on this plot. Examining this plot closely reveals that, like previously suspected, the design is indeed relatively flat around the optimal span design. This plot shows that, varying the span by up to 20 ft only results in a difference of

Table 3 Optimization problem formulations

	Variable/function	Description	#	*Baseline Optimized*	*Twist Optimized*	*Fully Optimized*
Minimize	W_{fuel}	Mission fuel weight		✓	✓	✓
with respect to	W_{gross}	Aircraft gross weight	1	✓	✓	✓
	W^*_{fuel}	Mission fuel weight slack variable	1	✓	✓	✓
	W_{engine}	Engine weight	1	✓	✓	✓
	$t_0^{\text{climb}}, t_0^{\text{cruise}}, t_0^{\text{descent}}$	Climb, cruise, descent phase initial time	3	✓	✓	✓
	$\Delta t^{\text{climb}}, \Delta t^{\text{cruise}}, \Delta t^{\text{descent}}$	Climb, cruise, descent phase duration	3	✓	✓	✓
	$W_j^{\text{climb}}, W_j^{\text{cruise}}, W_j^{\text{descent}}$	Climb, cruise, descent weight state	23	✓	✓	✓
	$CG_j^{\text{climb}}, CG_j^{\text{cruise}}, CG_j^{\text{descent}}$	Climb, cruise, descent CG state	84	✓	✓	✓
	$d_j^{\text{climb}}, d_j^{\text{cruise}}, d_j^{\text{descent}}$	Climb, cruise, descent distance state	23	✓	✓	✓
	$h_j^{\text{climb}}, h_j^{\text{cruise}}, h_j^{\text{descent}}$	Climb, cruise, descent altitude polynomial controls	12	✓	✓	✓
	$\alpha_j^{\text{climb}}, \alpha_j^{\text{cruise}}, \alpha_j^{\text{descent}}$	Climb, cruise, descent angle of attack state	28	✓	✓	✓
	$\eta_j^{\text{climb}}, \eta_j^{\text{cruise}}, \eta_j^{\text{descent}}$	Climb, cruise, descent tail incidence state	28	✓	✓	✓
	α^{man}	Maneuver angle of attack state	1	✓	✓	✓
	η^{man}	Maneuver tail incidence state	1	✓	✓	✓
	x^{struct}	Wing/strut structural thicknesses	192	✓	✓	✓
	θ_i^{wing}	Wing twist	9	✗	✓	✓
	b_{ob}	Wing outboard span	1	✗	✗	✓
		Total design variables	**411**			
Subject to	$0.0 < \text{throttle}^{\text{traj}} < 1.0$	Throttle constraints	17	✓	✓	✓
	$\frac{dh_j}{dt}^{\text{climb}} > 0.0$	No descent during climb	12	✓	✓	✓
	$\frac{dh_j}{dt}^{\text{descent}} < 0.0$	No climb during descent	4	✓	✓	✓
	$\frac{dW_j}{dt}^{\text{traj}} - \frac{dW_j}{dt}^{\text{traj}}_{\text{approx}} = 0.0$	Weight collocation defects	21	✓	✓	✓
	$\frac{dd_j}{dt}^{\text{traj}} - \frac{dd_j}{dt}^{\text{traj}}_{\text{approx}} = 0.0$	Distance collocation defects	21	✓	✓	✓
	$g^{\text{traj}}_{\text{boundary}} = 0.0$	Phase linkage constraints	9	✓	✓	✓
	$R_{\text{weight}} = 0.0$	Weight residual	1	✓	✓	✓
	$R_{\text{range}} = 0.0$	Range residual	1	✓	✓	✓
	$W_{\text{fuel}} - W^*_{\text{fuel}} = 0.0$	Mission fuel weight slack constraint	1	✓	✓	✓
	$L_j^{\text{climb}} - W_j^{\text{climb}} = 0.0$	Climb lift constraint	12	✓	✓	✓
	$C_{M_y}^{\text{climb}} = 0.0$	Climb trim constraint	12	✓	✓	✓
	$L_j^{\text{cruise}} - W_j^{\text{cruise}} = 0.0$	Cruise lift constraint	12	✓	✓	✓
	$C_{M_y}^{\text{cruise}} = 0.0$	Cruise trim constraint	12	✓	✓	✓
	$L_j^{\text{descent}} - W_j^{\text{descent}} = 0.0$	Descent lift constraint	4	✓	✓	✓
	$C_{M_y}^{\text{descent}} = 0.0$	Descent trim constraint	4	✓	✓	✓
	$L^{\text{man}} - n_{man} \cdot W_{\text{gross}} = 0.0$	Maneuver lift constraint	1	✓	✓	✓
	$C_{M_y}^{\text{man}} = 0.0$	Maneuver trim constraint	1	✓	✓	✓
	$\text{KS}_{\text{fail}} \leq 1.0$	Maneuver failure constraint	1	✓	✓	✓
		Total constraints	**146**			

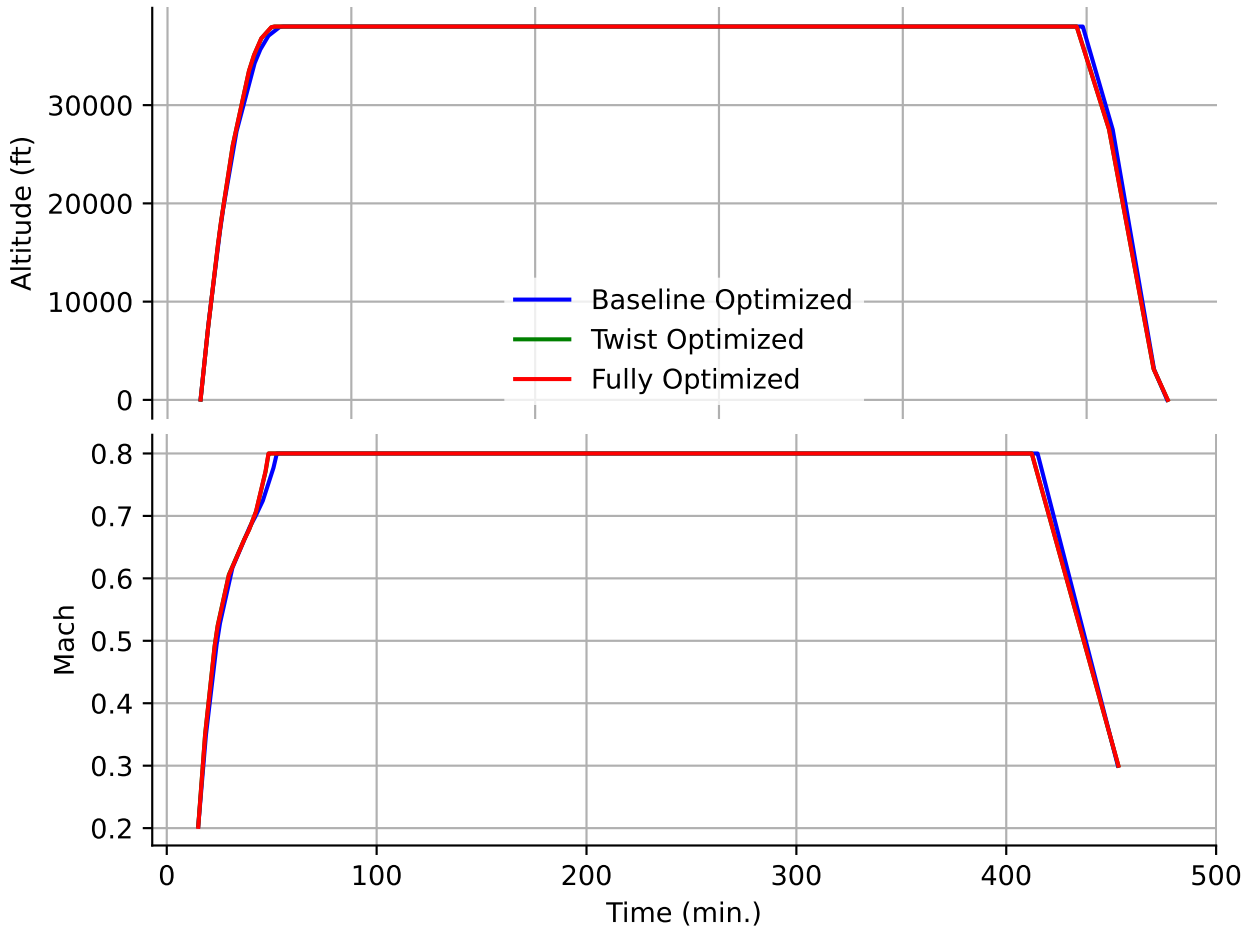

Fig. 7 Comparison of optimized altitude/Mach mission profiles for baseline, twist, and fully optimized designs

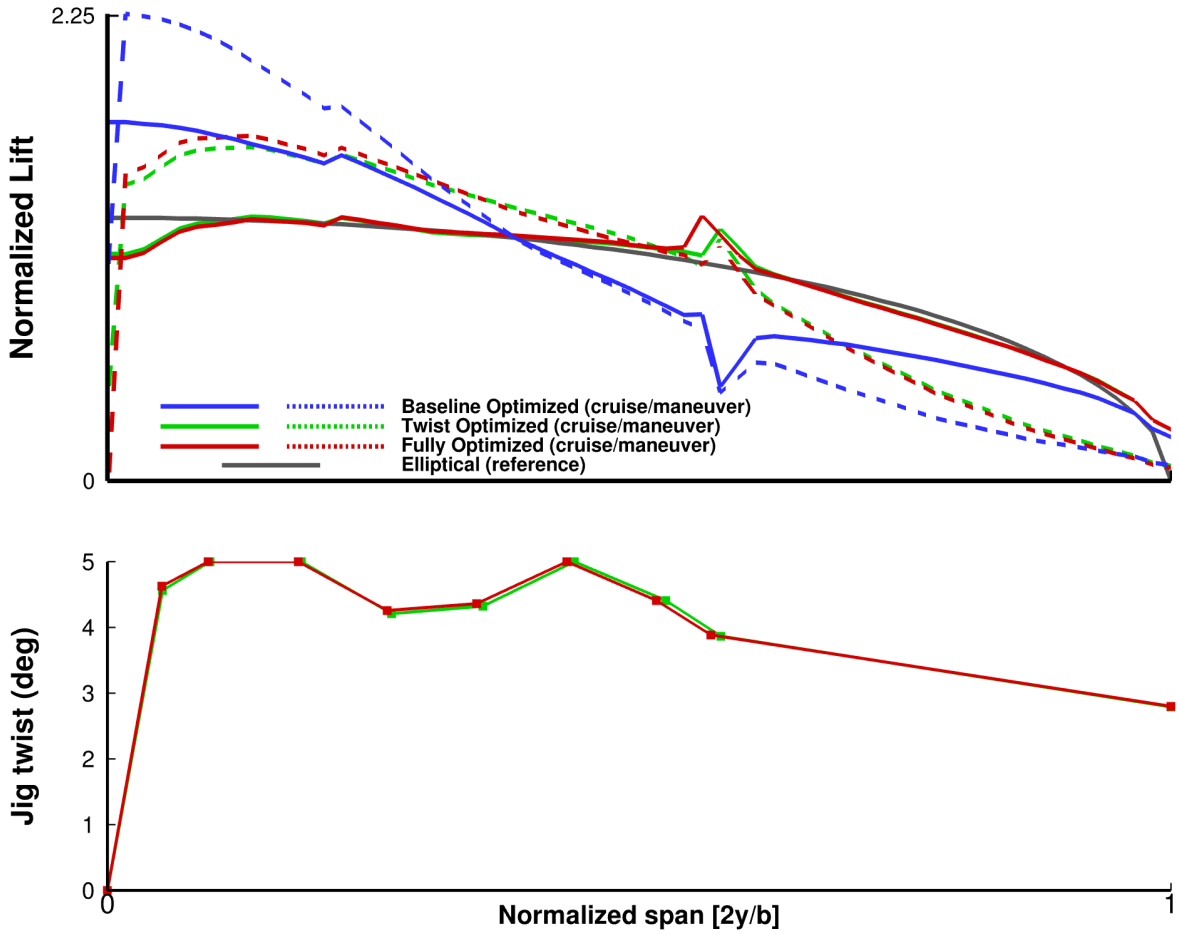

Fig. 8 Comparison of normalized lift distribution (wing, strut, and tail combined) between baseline, twist, and fully optimized designs

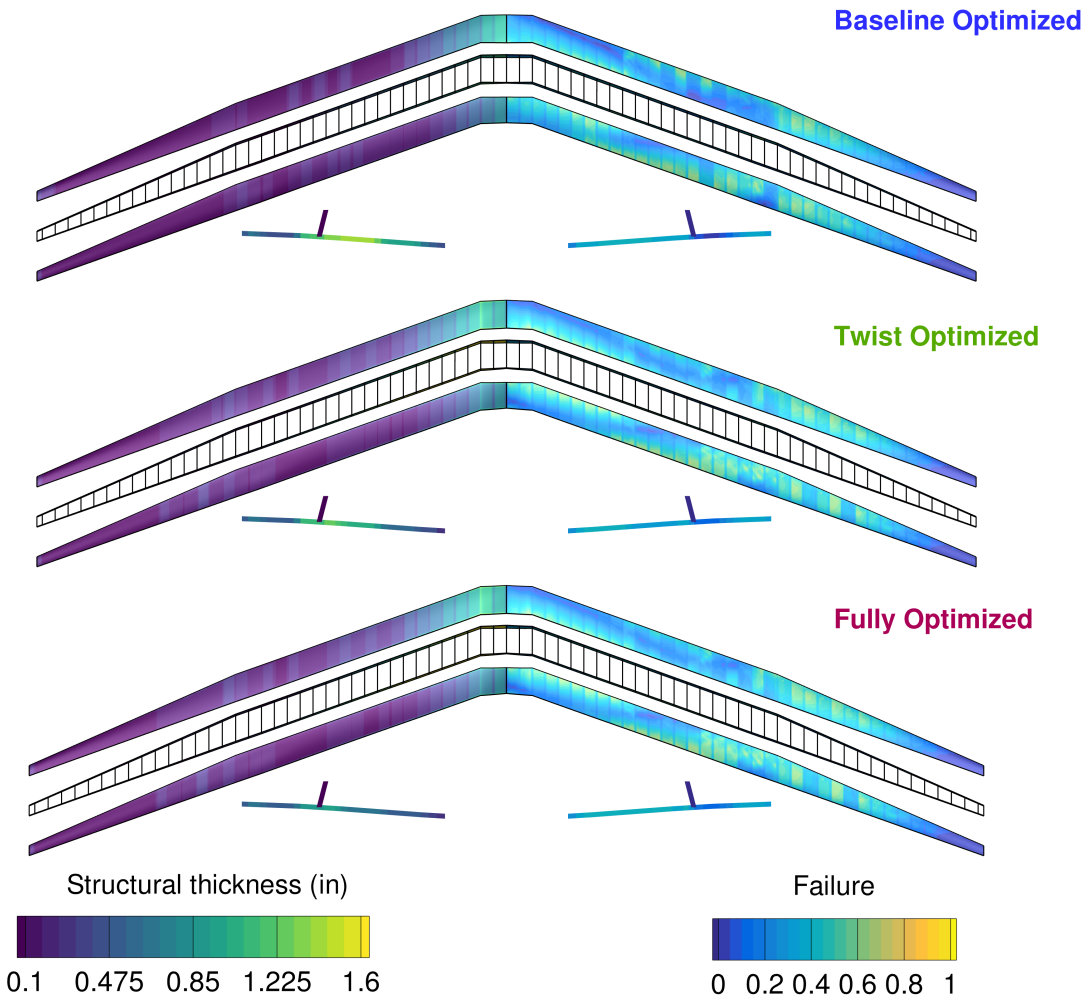

Fig. 9 Optimized panel thicknesses and failure contours for optimized wing and truss structures

Table 4 Optimization results

Design	Fuel burn (lb)	Gross weight (lb)	Wing weight (lb)	Span (ft)	L/D @ cruise start (·)
Baseline Optimized	20,390	147,980	18,410	35.97	25.31
Twist Optimized	19,080	148,420	20,140	35.97	27.68
Fully Optimized	19,070	148,450	20,160	37.36	27.72

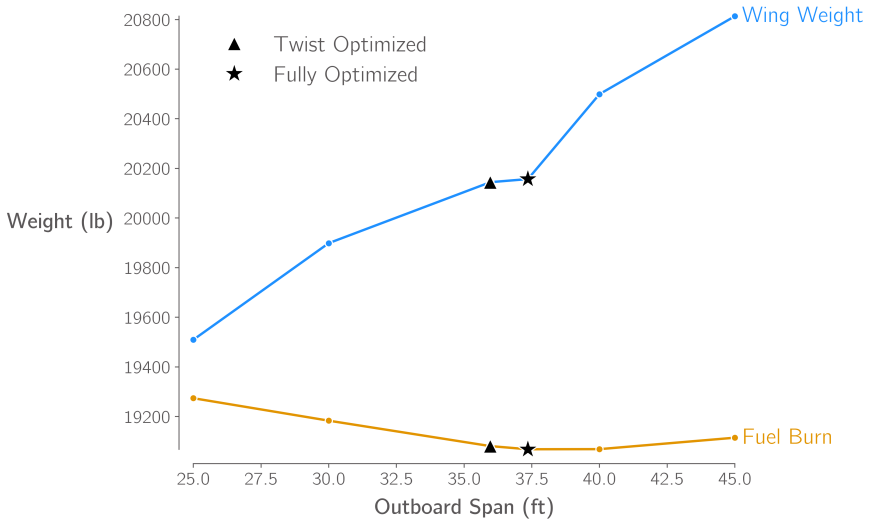

Fig. 10 Design fuel burn and wing weight versus outboard span

200 lb in mission fuel burn, roughly 1%. One explanation for this low variation in fuel burn performance with respect to span can be found in inviscid aerodynamic theory. Based on this theory, increasing the wing span primarily offers benefits in reduction of induced drag. However, these benefits offer diminishing returns: the larger the span, the less induced drag that can be reduced. It is likely that the Tech Collector TTBW design's already considerably large wing span doesn't stand to gain much from increasing the wing span further. Increasing the wing span beyond this point incurs steeper penalties in both viscous drag and structural weight.

VIII. Conclusion

We have demonstrated a system level design optimization of a next generation TTBW aircraft concept. The optimization demonstrated how the low-fidelity methods in Aviary can be augmented with medium-fidelity and high-fidelity analyses including: aerodynamic, geometric, and structural design variables. From this study it was found that wing twist had a more pronounced effect on optimal fuel burn performance relative to the outboard span, offering up to a 6% reduction in fuel burn. A design variable sweep was conducted to quantify the effect of outboard span on the optimization. The optimization problem was found to have a relatively flat design space around the nominal span, explaining the low sensitivity in span observed earlier.

There are several directions for future work to build upon what is presented here. The aerodynamic model used in this work was a low-fidelity simplification that could be improved in future work. In particular, while Prandtl–Glauert corrections were included in the VLM analysis to account for compressibility effects, in order to accurately capture transonic effects, a higher fidelity analysis, such as Computational Fluid Dynamics (CFD), must be considered. This becomes especially important when adding the strut intersection geometry into the optimization, as it will be crucial for ensuring that aerodynamic shocks are not introduced into the model, reducing performance. In the current work, the number of ribs in the structural model was held fixed. This means that as the span of the model is varied, the rib spacing is modified as well. To accurately capture the effect that this has on the structural sizing, buckling must be considered in the design optimization. Considering buckling on the strut member will similarly be an important sizing consideration that will require additional aerostructural sizing conditions, such as a -1 g maneuver or gust condition, to ensure that the

structure is robustly sized for buckling.

Acknowledgments

This work was completed with support from the Systems Analysis and Integration sub-project under the Advanced Air Transport Technology (AATT) project in the Advanced Air Vehicles Program (AAVP). Funding for the Programs is provided by the Aeronautic Research Mission Directorate (ARMD).

References

[1] Wahls, R. A., "NASA Sustainable Flight National Partnership Overview," *8th International Workshop on Aviation and Climate Change*, 2023. URL https://ntrs.nasa.gov/api/citations/20230007095/downloads/UTIAS8th-ACC-SFNPoverview-2023may31_final-Wahls.pdf.

[2] Heidmann, J. D., "An Overview of NASA Sustainable Aviation," *8th International Aerospace Innovation Forum*, 2022. URL https://ntrs.nasa.gov/api/citations/20220011765/downloads/Presentation_An%20Overview%20of%20NASA%20Sustainable%20Aviation_Heidmann.pdf.

[3] Harrison, N. A., Hoffman, K., Lazzara, D. S., Reichenbach, E. Y., Sclafani, A. J., and Droney, C. K., "Subsonic Ultra Green Aircraft Research: Phase IV Final Report–Volume I Mach 0.80 Transonic Truss-Braced Wing High-Speed Design Report," 2023.

[4] Aretskin-Hariton, E., Gratz, J., Kirk, J., Lyons, K., Jasa, J., Moore, K., Falck, R., Caldwell, D., Kuhnle, C., Recine, C., Hendricks, E., and Olson, E., *Multidisciplinary Optimization of a Transonic Truss-Braced Wing Aircraft using the Aviary Framework*, 2024. https://doi.org/10.2514/6.2024-1084, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-1084.

[5] Hendricks, E. S., and Gray, J. S., "pyCycle: A Tool for Efficient Optimization of Gas Turbine Engine Cycles," *Aerospace*, Vol. 6, No. 87, 2019. https://doi.org/10.3390/aerospace6080087.

[6] McDonald, R. A., and Gloudemans, J. R., *Open Vehicle Sketch Pad: An Open Source Parametric Geometry and Analysis Tool for Conceptual Aircraft Design*, 2022. https://doi.org/10.2514/6.2022-0004, URL https://arc.aiaa.org/doi/abs/10.2514/6.2022-0004.

[7] Leader, M. K., Aretskin-Hariton, E., and Moore, K., *Multidisciplinary Optimization of a Transonic Truss Braced Wing Aircraft With Hybrid-Electric Propulsion*, 2024. https://doi.org/10.2514/6.2024-4294, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-4294.

[8] Fazal, B., Schmidt, J., Phillips, B. D., Ordaz, I., and Moore, K., *Integration of Uncertainty Quantification in a Model-Based Systems Analysis and Engineering Framework*, 2024. https://doi.org/10.2514/6.2024-4559, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-4559.

[9] Hwang, J. T., Jasa, J. P., and Martins, J. R. R. A., "High-Fidelity Design-Allocation Optimization of a Commercial Aircraft Maximizing Airline Profit," *Journal of Aircraft*, Vol. 56, No. 3, 2019, pp. 1164–1178. https://doi.org/10.2514/1.C035082, URL https://doi.org/10.2514/1.C035082.

[10] Vassberg, J., Dehaan, M., Rivers, M., and Wahls, R., *Development of a Common Research Model for Applied CFD Validation Studies*, 2008. https://doi.org/10.2514/6.2008-6919, URL https://arc.aiaa.org/doi/abs/10.2514/6.2008-6919.

[11] Kaneko, S., and Martins, J. R., *MDO Formulations for Simultaneous Design and Trajectory Optimization*, 2024. https://doi.org/10.2514/6.2024-2231, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-2231.

[12] Gratz, J., Kirk, J., Recine, C., Jasa, J., Aretskin-Hariton, E., Moore, K., and Marfatia, K., *Aviary: An Open-Source Multidisciplinary Design, Analysis, and Optimization Tool for Modeling Aircraft With Analytic Gradients*, 2024. https://doi.org/10.2514/6.2024-4219, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-4219.

[13] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A., "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, Vol. 57, No. 4, 2018, pp. 1815–1827. https://doi.org/10.1007/s00158-018-1912-8.

[14] Kennedy, G. J., and Martins, J. R., "A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures," *Finite Elements in Analysis and Design*, Vol. 87, 2014, pp. 56–73. https://doi.org/https://doi.org/10.1016/j.finel.2014.04.011, URL https://www.sciencedirect.com/science/article/pii/S0168874X14000730.

[15] Kiviaho, J. F., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2237–2243. https://doi.org/10.2514/1.J057318, URL https://doi.org/10.2514/1.J057318.

[16] Hajdik, H. M., Yildirim, A., Wu, E., Brelje, B. J., Seraj, S., Mangano, M., Anibal, J. L., Jonsson, E., Adler, E. J., Mader, C. A., Kenway, G. K. W., and Martins, J. R. R. A., "pyGeo: A geometry package for multidisciplinary design optimization," *Journal of Open Source Software*, Vol. 8, No. 87, 2023, p. 5319. https://doi.org/10.21105/joss.05319.

[17] Yildirim, A., Jacobson, K. E., Anibal, J. L., Stanford, B. K., Gray, J. S., Mader, C. A., Martins, J. R. R. A., and Kennedy, G. J., "MPhys: A Modular Multiphysics Library for Coupled Simulation and Adjoint Derivative Computation," *Structural and Multidisciplinary Optimization*, 2024. (In press).

[18] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, 2019, pp. 1075–1104. https://doi.org/10.1007/s00158-019-02211-z.

[19] Wells, D. P., Horvath, B. L., and Mccullers, L. A., "The Flight Optimization System Weights Estimation Method," 2017. URL https://ntrs.nasa.gov/api/citations/20170005851/downloads/20170005851.pdf.

[20] Hague, D., "GASP - General Aviation Synthesis Program," Vol. 5, 1978. URL https://ntrs.nasa.gov/api/citations/.

[21] Gray, J. S., Hearn, T. A., and Naylor, B. A., *Using Graph Coloring To Compute Total Derivatives More Efficiently In OpenMDAO*, 2019. https://doi.org/10.2514/6.2019-3108.

[22] Falck, R., Gray, J. S., Ponnapalli, K., and Wright, T., "dymos: A Python package for optimal control of multidisciplinary systems," *Journal of Open Source Software*, Vol. 6, No. 59, 2021, p. 2809. https://doi.org/10.21105/joss.02809, URL https://doi.org/10.21105/joss.02809.

[23] Capristan, F. M., and Welstead, J. R., "An Energy-Based Low-Order Approach for Mission Analysis of Air Vehicles in LEAPS," 2018. https://doi.org/10.2514/6.2018-1755.

[24] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, Vol. 17, 2020, pp. 261–272. https://doi.org/10.1038/s41592-019-0686-2.

[25] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. https://doi.org/10.1137/0907058, URL https://doi.org/10.1137/0907058.

[26] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q., "JAX: composable transformations of Python+NumPy programs," 2018. URL http://github.com/jax-ml/jax.

[27] Raymer, D. P., *Aircraft design: A conceptual approach*, 4$^{\text{th}}$ ed., AIAA education series, American Institute of Aeronautics and Astronautics, Reston, Va., 2006. URL http://www.loc.gov/catdir/toc/ecip068/2006004706.html.

[28] MSC Software Corporation, *MSC Nastran 2012 Quick Reference Guide*, MSC Software Corporation, Newport Beach, CA, 2012. Available from MSC Software Corporation.

[29] Sederberg, T. W., and Parry, S. R., "Free-form deformation of solid geometric models," *SIGGRAPH Comput. Graph.*, Vol. 20, No. 4, 1986, p. 151–160. https://doi.org/10.1145/15886.15903, URL https://doi.org/10.1145/15886.15903.

[30] Haimes, R., and Dannenhoffer, J., *The Engineering Sketch Pad: A Solid-Modeling, Feature-Based, Web-Enabled System for Building Parametric Geometry*, 2013. https://doi.org/10.2514/6.2013-3073, URL https://arc.aiaa.org/doi/abs/10.2514/6.2013-3073.

[31] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. https://doi.org/10.1007/s00158-012-0763-y.

[32] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Rev.*, Vol. 47, No. 1, 2005, p. 99–131. https://doi.org/10.1137/S0036144504446096, URL https://doi.org/10.1137/S0036144504446096.

[33] Wells, D. P., Horvath, B. L., and Mccullers, L. A., "The Flight Optimization System Weights Estimation Method," Tech. rep., 2017. URL https://ntrs.nasa.gov/citations/20170005851.

[34] Kreisselmeier, G., and Steinhauser, R., "Application of vector performance optimization to a robust control loop design for a fighter aircraft," *International Journal of Control*, Vol. 37, No. 2, 1983, pp. 251–284. https://doi.org/10.1080/00207179.1983.9753066, URL https://doi.org/10.1080/00207179.1983.9753066.

[35] 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. https://doi.org/10.1007/s00158-006-0061-7, URL https://doi.org/10.1007/s00158-006-0061-7.