Multidisciplinary Optimization of a Transonic Truss-Braced Wing Aircraft using Aviary

Eliot Aretskin-Hariton, Jennifer Gratz, John Jasa*, Kenneth Moore*
Robert Falck, Caroline Kuhnle, Mark Leader, Jeff Chapman, Eric Hendricks
NASA Glenn Research Center, Cleveland, OH, USA

Jason Kirk, Darrell Caldwell**, Christopher Eggert, Irian Ordaz, Erik Olson
NASA Langley Research Center, Hampton, VA USA

Carl Recine, Kenneth Lyons
NASA Ames Research Center, Mountain View, CA USA

The continuous push to decrease fuel burn of single-aisle commercial aircraft has led to interest in a Transonic Truss-Braced Wing (TTBW) concept vehicle. Sporting high-slung wings that are long and slender to increase aerodynamic efficiency, a TTBW can also accommodate higher-bypass turbine engines. The combination of these two changes potentially leads to an overall decrease in fuel consumption. In this paper, a TTBW concept vehicle is assembled in the Aviary open-source tool for conceptual aircraft design. The conceptual-level aerodynamics and propulsion systems that come prepackaged with Aviary are replaced with higher-fidelity vortex lattice method for aerodynamics (VSPAERO) and a one-dimensional cycle analysis tool for propulsion (pyCycle). The vehicle is then optimized to minimize fuel burn for a representative commercial mission. Design parameters for the vehicle include the electrified turbine size, the size of the electric motors which are used for climb assist, and battery capacity. The TTBW concept vehicle presented in this paper represents the first application of Aviary to an aircraft design problem.

*Banner Quality Management Inc.
**Analytical Mechanics Associates Inc.

Nomenclature

Acronyms

AATT	Advanced Air Transport Technology
AAVP	Advanced Air Vehicles Program
APU	Auxiliary Power Unit
ARMD	Aeronautic Research Mission Directorate
CDISC	Constrained Direct Iterative Surface Curvature Aerodynamic Design Software
DOE	Design of Experiments
EPFD	Electric Powered Flight Demonstration
FLOPS	Flight Optimization System
GASP	General Aviation Synthesis Program
GASPy	General Aviation Synthesis Program with Python
GTOW	Gross Takeoff Weight
IASP	Integrated Aviation Systems Program
LEAPS	Layered and Extensible Aircraft Performance System
ODEs	Ordinary Differential Equations
TACP	Transformational Aeronautic Concepts Program
TOC	Top of Climb
TTBW	Transonic Truss-Braced Wing
TTT	Transformational Tools and Technologies
SFNP	Sustainable Flight National Partnership
SLS	Sea Level Static
UAM	Urban Air Mobility
VLM	Vortex Lattice Method
XDSM	eXtended Design Structure Matrix

I. Introduction

Global commercial aviation supports tens of thousands of flights per day. To minimize costs and limit ecological impact, next-generation aircraft designs are focused on decreased fuel consumption. While many different aircraft designs have been proposed to achieve this goal, a configuration of particular interest to NASA is the Transonic Truss-Braced Wing (TTBW). From 2008 to 2022, NASA conducted preliminary design studies of the TTBW and followed up with multiple wind-tunnel tests of different models to characterize the aerodynamics [1]. These tests were supported by NASA's Sustainable Flight National Partnership (SFNP) and Green Aviation Technology Development program. In 2023, NASA awarded Boeing with a contract to build and fly a TTBW demonstrator. NASA's goal with the demonstrator is to show a reduction in fuel burn compared to traditional wing designs, when combined with advancements in propulsion, materials, and systems architecture [2]. Since the original design studies in 2008, new optimization tools like OpenMDAO [3] have been developed and refined to tightly couple aerodynamics, structures, propulsion, and other subsystems. This paper will seek to leverage those new optimization tools to perform gradient-based design and optimization of a TTBW concept vehicle.

The most striking difference between the TTBW and traditional aircraft is the elongated wing supported by a truss which can be seen in Figure 1. In addition, the wing chord is shorter, leading to a thinner profile. These changes significantly increase the aspect ratio of the wing and they decrease induced drag. The truss on the wing is added to provide additional aeroelastic stiffness and ensure that the thinner wing can support the required loads during emergency maneuvers. Additional benefit is provided by mounting the wings higher on the fuselage, allowing for a larger bypass engine to be attached underneath the wings. Traditional tube-and-wing aircraft have landing clearance constraints that prevent engine diameter from increasing. This limits the bypass ratios that can be achieved on those vehicles. The combination of wing profile changes in addition to engine efficiency increases are expected to be the largest contributors to the fuel-reduction claims of a TTBW configuration. Extending the wing span of the vehicle creates problems with docking the aircraft at terminals designed for shorter wings. This may precipitate the need to create a fold in the wing that can be articulated to stow the wing tips during docking.

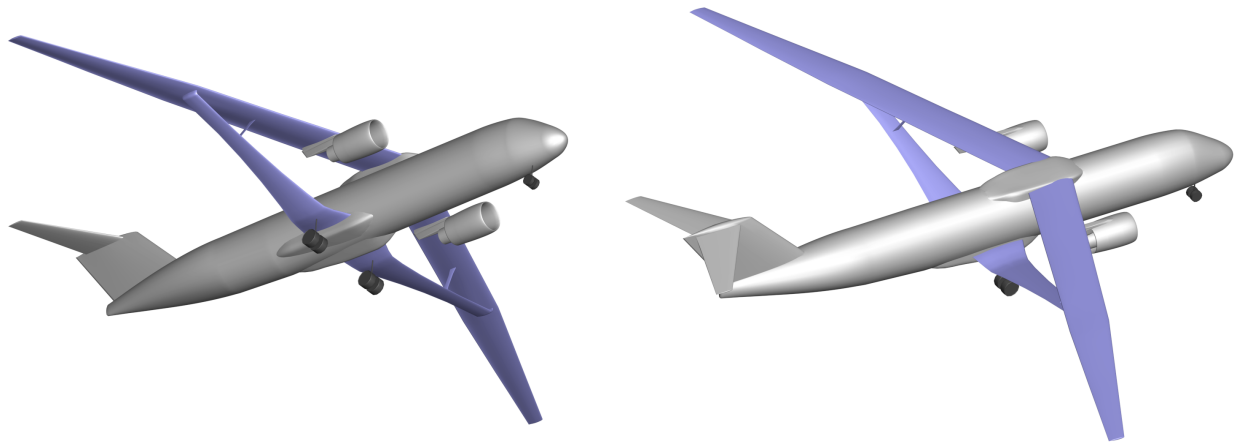

Fig. 1 NASA TTBW Tech Collector geometry.

Previous tightly-coupled novel aircraft designs using OpenMDAO [3] and Dymos [4] have been conducted. These past applications focused on a tiltwing turboelectric vehicle [5, 6], a quadrotor Urban Air Mobility (UAM) vehicle [7, 8], as well as the X-57 [7]. In those studies, numerous subsystems including mass, structure, mission, thermal, and propulsion (including battery, converters, electrical wires, and gearboxes), were sized for a specific mission. For each study, a custom run script was created to execute pre-mission analysis (analysis of subsystems that affect the mission but do not change with time) and mission analysis (analysis of subsystems that do change with time, including trajectory). In contrast to those studies, the study presented in this paper used Aviary to create a formal, simplified run script.

The Aviary open-source aircraft conceptual design tool standardizes the way that subsystems and users interact with aircraft design. At its core, Aviary performs gradient-based optimization using the OpenMDAO library. The fundamental physics of tube-and-wing aircraft from two legacy aircraft-design tools have been implemented into Aviary. General Aviation Synthesis Program (GASP) and Flight Optimization System (FLOPS) are both storied design tools still used in government and industry for preliminary aircraft design. They both contain first-order tube-and-wing

aircraft sizing and analysis tools for weight, structures, aerodynamics, and propulsion. Over the past several years there has been an effort to update these tools which lead to the creation of General Aviation Synthesis Program with Python (GASPy) [9] and Layered and Extensible Aircraft Performance System (LEAPS) [10] (a python version of FLOPS). With Aviary, there was an opportunity to amalgamate those two codes with best practices learned over the last decade. This enables users to perform gradient-based, tightly-coupled optimization with OpenMDAO while using the basic equations and relationships created by GASP and FLOPS, which they are already familiar with. Further details on the Aviary tool will be described in future work.

The studies presented in this paper cover optimization of a TTBW model over two different model configurations. The first configuration considers only advanced aerodynamic and propulsion systems. The second configuration augments the first by including electrification of the engine. Both configurations use the same objective function of minimum fuel burn for a reference mission consisting of climb, cruise, and descent phases. A minimum fuel burn objective roughly translates to minimizing reoccurring costs for a vehicle.

The layout of the rest of the paper is as follows: Sec II.A provides an overview on how the TTBW subsystems are connected to each other. Then details on each subsystem will be covered. Sec II.B describes the geometry modeling used for the aircraft, Sec II.C presents details on how aerodynamics was added to the study, Sec II.D details electrical system sizing calculations, Sec II.E describes the turbine engine integration, Sec II.F details mass and structural analysis, and Sec II.G describes the Ordinary Differential Equations (ODEs) used to perform the dynamic analysis. Sec III describes the two demonstration problems, including desired mission, and other general settings and constraints. Sec III.D discusses the results and challenges. Lastly, Sec IV presents the final conclusions from the study.

II. Methods

A. Overview

The configuration for the TTBW model is best described through the eXtended Design Structure Matrix (XDSM) shown in Figure 2. XDSMs represent a simple way to display coupling between large-numbers of subsystems which would otherwise be overly complex if displayed in a flow-diagram. Inputs to each subsystem are shown on the vertical axes, outputs from the subsystem are shown on the horizontal axes. The green boxes in the XDSM represent the individual subsystem analysis that was performed. The orange ovals represent solvers. In Figure 2 it shows the user supplied inputs which help configure the propulsion and electrical system. Aerodynamic design of the wing is loaded from Open Vehicle Sketch Pad (OpenVSP) [11] and sent to VSPAERO [12]. VSPAERO outputs a tabular data table which is sent into the mission analysis. This table is used to determine lift and drag based on angle of attack during the mission. The optimizer adjusts the design mass-flow going into the propulsion design performed by pyCycle [13]. Additionally, the optimizer can adjust a number of parameters related to the electrical system which is only relevant for the study that includes electrical system sizing. Based on a guess for Gross Takeoff Weight (GTOW), a summation of mass from propulsion, electrical, and aerodynamics is added to the mass calculation of all the other elements in the aircraft (i.e. landing gear, fuselage, passengers, cargo). Finally, configuration parameters for all of these subsystems are sent to the mission analysis. Mission analysis flies the aircraft through it's prescribed mission and returns the amount of fuel burned during the flight as well as values for constraints including battery energy consumed. The total battery energy constraint imposed prevents the energy consumed from exceeding 80% of the total battery energy available. The optimizer then iterates on the design variables in order to minimize the final fuel burn of the mission.

The terminology that describes different parts of the optimization and mission is described below. Pre-mission analysis (also known as static analysis [8]), calculates qualities of the aircraft that do not change in time (i.e. wing length, turbine mass, battery energy density). The pre-mission XDSM shown in Figure 2 represents the information flow prior to setting up mission analysis (which is detailed in Sec II.G). In this example, VSPAERO and OpenVSP are only run once during the optimization. Further iterations of design variables by the optimizer do not affect these calculations. Mission analysis (also known as dynamic analysis [8]), performs time-dependant calculations (i.e. how much fuel is burned per second, how the battery state of charge changes over time). It is also possible to have post-mission analysis but this is not necessary for our examples.

In the subsequent sections, each subsystem that is part of the TTBW analysis is introduced. Some of these subsystems such as Mass (Sec II.F) and Geometry (Sec II.B) are intrinsic to Aviary and are built out of the standard weight relationships for tube-and-wing aircraft that were specified in FLOPS. Aviary is designed to have streamlined connections between external analysis codes. External subsystems integration into Aviary is an essential part of the TTBW model presented in this paper. External subsystems like VSPAERO and pyCycle [5], provide higher-fidelity

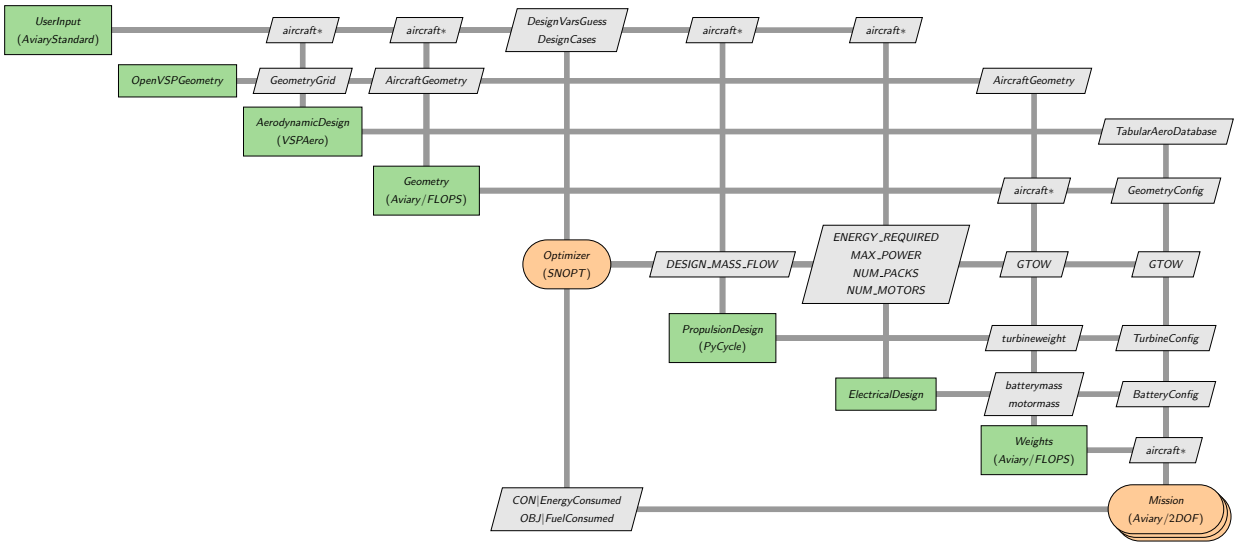

Fig. 2 Pre-Mission eXtended Design Structure Matrix (XDSM). The information that all of these calculations feed into mission is also shown. For configurations without batteries, the electrical block's inputs and outputs are removed.

analysis than Aviary's baseline subsystems (table lookup). This does increase cost and complexity of the optimization and demonstrates the capability of integrating external subsystems.

B. Geometry

The TTBW geometry used in this work, shown in Figure 1, is modeled in OpenVSP [11, 12]. This non-proprietary configuration referred to as the TTBW Tech Collector has been developed by NASA in an effort to allow future vehicle technology studies, open publication of results, and easier collaboration with parties outside of NASA. The wing and strut airfoil sections were independently optimized with the Constrained Direct Iterative Surface Curvature Aerodynamic Design Software (CDISC) [14] to reduce induced, wave, and profile drag. This involved span load tailoring and airfoil shaping to reduce shock strength and flow separation, respectively, at an altitude of 43,000 ft and Mach 0.8. The wing was not co-optimized in this paper with the other subsystems (propulsion, electrical) but this could be considered in future research. The geometry model also includes conceptual geometry representations of subsystems such as avionics, crew and passenger compartments, forward and aft cargo compartments, Auxiliary Power Unit (APU), control surfaces, hydraulics, leading and trailing edge wing spars, fuel tanks, nose landing gear, and main landing gear. These subsystem geometry models and their corresponding weight values calculated by Aviary, including fuel weight states, can be used to more accurately estimate the center of gravity of the aircraft at any point along the flight.

The integration of geometry within Aviary was performed through an OpenVSP wrapper using the OpenVSP Python-based API [15]. This OpenMDAO component allows the user to easily declare input and output variables that can be used to drive geometry optimization or trade studies, and can also be used to automatically output geometry data required by other disciplines within Aviary. In addition, several OpenVSP capabilities like planar slicing, surface grid generation, and wetted area and volume calculations are also supported through the user interface. The computation of some of these quantities replaces the Aviary-computed values through a feature called "overriding", where any internally computed value can be replaced with a new, more accurate value that can come from an external subsystem.

C. Aerodynamics

The TTBW mission analysis requires estimation of aerodynamic performance (lift and drag) along the mission trajectory. Aviary includes drag estimation methods based on empirical equations for conventional subsonic transport aircraft. However, these methods are not suitable for non-conventional configurations such as the TTBW concept, so the external aerodynamics tool VSPAERO is used instead.

VSPAERO [12] is a NASA-developed potential flow solver driven using the OpenVSP Python-based API. VSPAERO computes aerodynamic forces by utilizing either the Vortex Lattice Method (VLM) or the panel method on OpenVSP-

modeled aircraft geometries. For this work, the aerodynamic analysis of the TTBW is conducted using VLM. This method uses simplified representations of the geometries, where lifting bodies are represented by flat surfaces along their camber lines and other bodies are decomposed into cruciform cross sections. Although VLM is an inherently inviscid method, VSPAERO also provides estimates of parasite drag. For lifting bodies, these estimates are based on empirical regressions of NACA 0012 airfoil data. For non-lifting bodies, the estimated skin friction drag is calculated based on flat plate boundary layer theory, and estimated form and transonic drag are calculated using other empirical relations. VSPAERO also accounts for compressibility effects by applying the 2nd-order Karman-Tsien Mach correction [16]. Since the full TTBW OpenVSP model contains more detail than is reasonable to include in VLM analysis, VSPAERO analysis is performed using only a subset of the geometry components. This OpenVSP subset consists of the fuselage, wing, truss, horizontal tail, and vertical tail components, as shown in Figure 3.

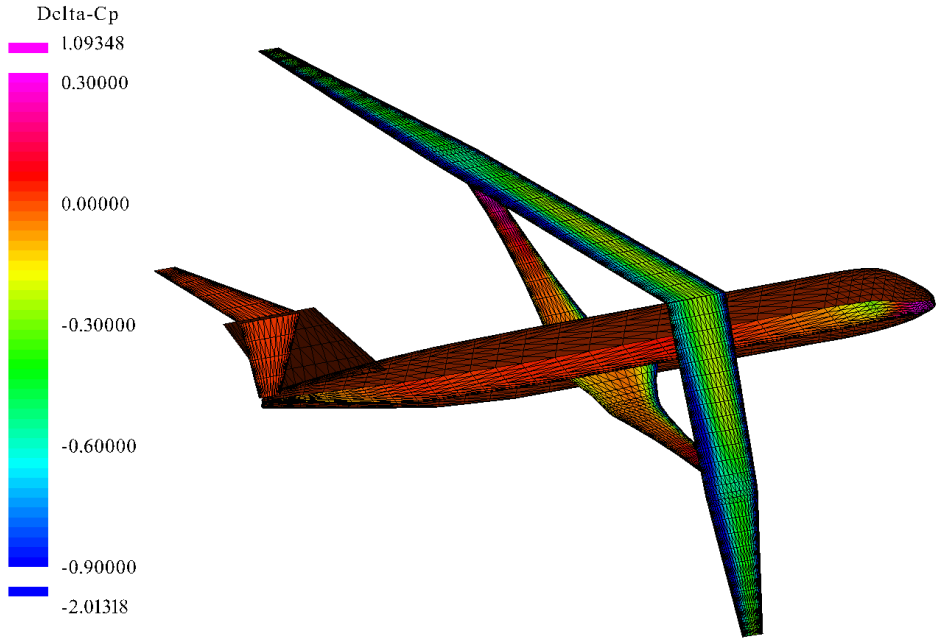

Fig. 3 TTBW VSPAERO grid and solution.

The lift and drag coefficient tables for the TTBW are generated during the Aviary pre-mission phase by executing VSPAERO as an external subsystem. Three 1-dimensional arrays are specified as options to this aerodynamic subsystem to define sampling values for altitude, Mach number, and angle of attack. Within the subsystem, the three arrays are first used to construct a grid of sample points corresponding to each combination of these three parameters. At each sample point, a standard atmosphere model is used to calculate the Reynolds number, then VSPAERO analysis is performed using the angle of attack, Mach number, and Reynolds number as input parameters. The analysis results are compiled into two 3-dimensional arrays containing lift and drag coefficients which are passed to Aviary for use in the mission analysis phase. This data is listed as the "TabularAeroDatabase" in Figures 5 and 6 as an input to the aerodynamics calculation component in an Aviary mission. Since aerodynamic analysis cases are independent, the aerodynamic subsystem can reduce the total computational time of drag polar generation by executing each instance of VSPAERO on an individual processor. This is achieved by using an OpenMDAO subproblem with a design of experiments (DOE) driver for parallel execution through Message Passing Interface (MPI).

In mission analysis, the two tables containing the coefficients of lift and drag are used to train metamodels. The lift and drag coefficients are interpolated at the requested altitudes, Mach numbers, and angles of attack. The equations of motion used in the mission analysis require lift coefficient as an input, so a solver based on Newton's method is added to vary angle of attack until the lift matches the weight of the aircraft. The interpolation method is a 2nd-order Lagrange interpolant that is built into OpenMDAO. This algorithm has been optimized for fixed 3-dimensional tables, and is vectorized to simultaneously compute all values in a single phase.

D. Electrical

One of the demonstration problems described in this paper contains an electrical system including a battery, motor and converter. The following section will describe their configuration and application to the TTBW optimization problem.

The motor model makes use of a performance map that relates speed and torque to efficiency. Characteristics of the map used on the TTBW are consistent with a permanent magnet motor designed for high efficiency at a large range of torque and speed values *. Speed and torque values are scaled to align the TTBW maximum power, that the optimizer controls, with the motor design point. Maximum torque and speed are set as constraints to keep the motor away from the motor torque limit and the motor speed limit governed by the onset of flux weakening. This analysis does not consider the thermal management system (TMS) of the motor.

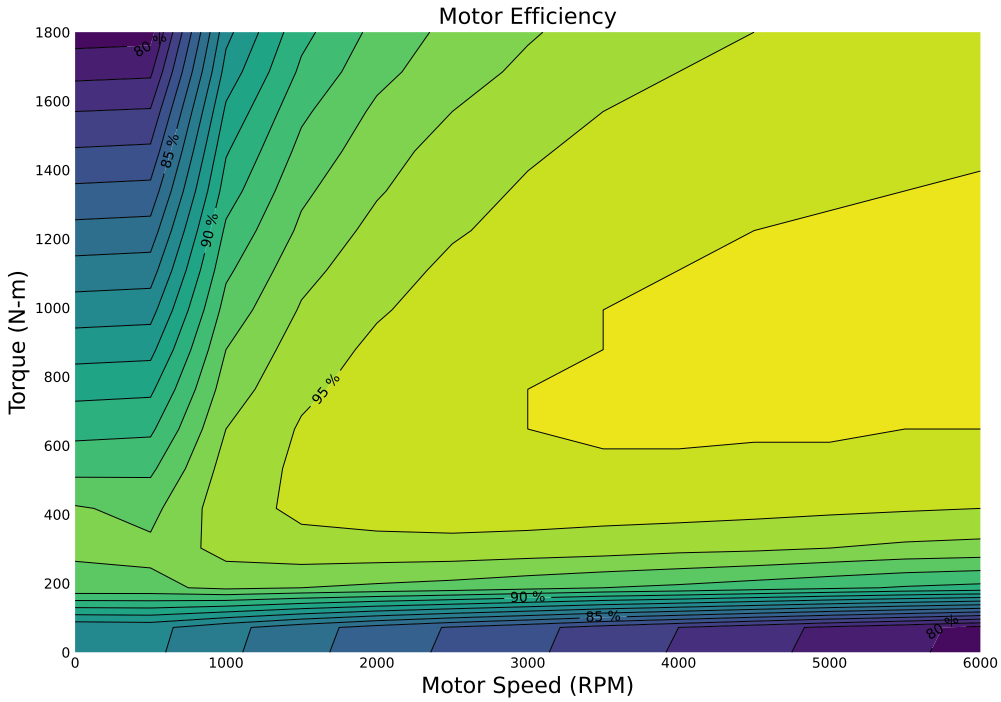

Fig. 4 Electric motor efficiency as a function of torque and speed.

The converter model is based on a simple 1D interpolation table, with values given below in Table 1. The efficiency calculation uses basic linear interpolation using OpenMDAO's `MetaModelStructuredComp` class. The motor operates at a fixed 1000 V, so as battery voltage drops due to decreasing state-of-charge (SOC), the converter incurs an increasingly large efficiency penalty to boost the battery voltage up to the motor voltage. The battery system is designed to have a nominal voltage in-line with the motor voltage, and the battery voltage will monotonically decrease throughout the mission. To prevent numerical errors if the battery voltage starts higher than the motor voltage, efficiency values are included in the table for voltages greater than 1000 V, even though optimizations should not be operating in this range of the table.

Table 1 Converter Efficiency as a Function of Input Voltage from the Battery.

Input Voltage (V)	15	166.97	347.09	527.2	707.31	887.43	1000	1100	1300	1500
Efficiency (%)	35.89	50.97	68.91	76.68	81.35	84.81	98.36	84.81	81.35	76

The battery model is a low fidelity representation of the system for simplicity and ease of integration. Future work will be dedicated to integrating a battery model of higher fidelity. Voltage is set to linearly decrease from 900 V to 700 V over the course of climb. The battery total energy usage in kWh is calculated by determining output amperage

*https://github.com/nasa/NPSS-Power-System-Library/blob/master/include/STARCABLmotorGenerator.map

demand from the motor and multiplying by the battery voltage. Margin is added to ensure the battery does not get discharged below 20% during climb. There is an inherent assumption that the battery is sized to handle the amount of amperage required. However there are no constraints for this in the model. The battery pack energy density is assumed to be 2,000 (Wh/kg). This reflects roughly a 5x multiple on current state-of-the-art battery pack designs. Considering the number of years required to test, manufacture, and integrate a TTBW aircraft into the national air space, this target energy density is aggressive but useful for the purposes of this study.

E. Propulsion

A two spool turbofan engine was chosen for the propulsion system, and was modeled using pyCycle [13]. For simulations with electrification, a motor was connected to the low-pressure shaft and injected power onto the shaft during climb. The power injected on the shaft was chosen by the Aviary optimizer and constrained to be between zero and 746 kW (1,000 hp). The Aviary optimizer also had access to the mass flow at cruise and imposed minimum rate of climb constraint on all points of cruise. This constraint is significant because the electric portion of the propulsion system was not used during cruise, and thus there was a potential that the engine would be sized the same regardless of the size of the electrical system. Key engine design parameters are found in the Table 5 in the results. The engine was sized based off the design information at top of climb. A key variable to note is throttle. Aviary considered throttle from 0% to 100% where 0% was the minimum throttle required for the engine at idle. The engine may still produce net thrust when throttle is set to 0%. This is different than other systems where throttle is defined on a scale of zero thrust to max thrust. For normal engines, the minimum throttle value is dependent on the altitude. For the case of this engine, it is 12% of the maximum thrust below 427m (1400 ft) and 5% of the maximum thrust above 427m. To account for the different representations of throttle a sigmoid activation function was used to switch out the minimum throttle value depending on altitude and an equation was created to convert the throttle into a 0% to a 100% range.

The engine is formatted in a way that makes it easier to integrate into Aviary. Engine design information is in the pre-mission section of the Aviary tool (Figure 2) and the off-design cases for an engine are in the mission portion of Aviary (Figures 5 & 6). An Aviary-based data bus was used to connect required variable outputs from design to required variable inputs in off-design. The mission portion of the pyCycle code had to be reformatted to run points in parallel and to return a vectorized set of data. This is due to the difference in how information is passed in Aviary mission (as arrays) and how information is typically passed for pyCycle off-design cases (single instances). pyCycle models (and engine cycle analysis models in general) are sensitive to the initial guesses of their state variables, and are usually "ramped up" or consecutively run at points close together in the flight envelope in order to converge successfully. When pyCycle models are run in parallel, "ramping up" is not possible, and points that are far from their initial guesses may fail to converge. To overcome this problem, guessing tables for some of the state variables were created, and these tables covered the value of those variables throughout the entire flight envelope. The variables chosen for initial guessing tables are located in Appendix IV.A and the values are dependent on the Mach number, altitude, throttle percent, and the amount of power added to the low pressue shaft by the motor. These tables allowed the pyCycle mission section to pick guess information closer to the point being run. For the most part, this solved the nonconvergence issue, however, there were still some points that did not converge when running the optimization. When a nonconvergent point occurred during optimization, all variables in the model were reset to their default values, the guess matrix method was applied again, and the point was rerun. This reset approach successfully converged the corner cases that could not be converged using the guess tables alone, and allowed the engine model to run successfully throughout the entire optimization.

One specific note regarding the propulsion system is that there were a few points in the engine's results where the engine's stall margin was lower than desired, and the engine could potentially have been at risk of stalling. However, these corner cases only occurred when the engine was near a low power setting, and any potential stall could easily be alleviated by rerouting additional engine bleed air, a change which would minimally affect the results at the affected points. Thus, the stall margin issue was considered an acceptable deviation and was not addressed for this work.

F. Mass

The weights equations that were used to compute the majority of component masses for the TTBW model were derived from FLOPS conventions. These equations are empirically-derived regression equations for the weight of individual components, which are summed into vehicle-level weight totals. Aviary includes Python translations of these regressions that produce exact matches of FLOPS results, with the exception that units have been changed to mass. These mass relations are designed to match conventional cantilever wings and not the TTBW concept. Therefore,

calibration factors were used to adjust the mass of components used in the wing weight buildup to match pre-existing TTBW analysis. Because wing geometry is fixed for this analysis, wing mass does not change between solver iterations. This allows for the calibration method used to remain accurate during optimization.

The mass of propulsion and electrical subsystems were accounted for through addition of new analysis. The engine model mass was scaled based on design mass flow. Motor mass was estimated using simple scaling relationships with max required power. Battery mass was estimated based on an energy density and the total amount of energy required for the mission. Other electrical components masses such as cables or thermal management systems were not included in this study.

G. Mission and Trajectory

The mission for the example problems consists of a climb phase, a cruise phase and a descent phase. Starting at 15.2m (50ft) off the ground and 0.3 Mach climb went up to 11,582m (38,000ft) and Mach 0.8. This mission analysis did not include a takeoff phase and assumed the climb phase began at 15.2m. The aircraft then performs a steady-level cruise at 11,582m. Finally the descent phase brings the aircraft back down to 0.0m altitude and Mach 0.3. The total distance traveled during the mission from climb through descent is 6320 km (3412.7 nautical miles). Integrating detailed takeoff and landing to this mission analysis will be incorporated in future work.

The XDSM for all of these phases is shown in Figures 5 & 6. In this configuration, the atmosphere subsystem calculates speed of sound based on input altitude. Mach number is calculated based on speed of sound and aircraft velocity. Internal to aerodynamics, the angle of attack (alpha) is solved for. This is performed by varying alpha inside of the aerodynamic subsystem until lift equals weight. The resulting drag is output to the equations of motion and combined with the thrust from propulsion to determine the resulting change in altitude, range, and specific energy. These values are fed back into Dymos, which uses collocation methods to solve for all the states in the phase [4]. A small angle assumption is used to decouple aerodynamics from propulsion in this case to enable us to avoid having to run pyCycle in the alpha-solving loop.

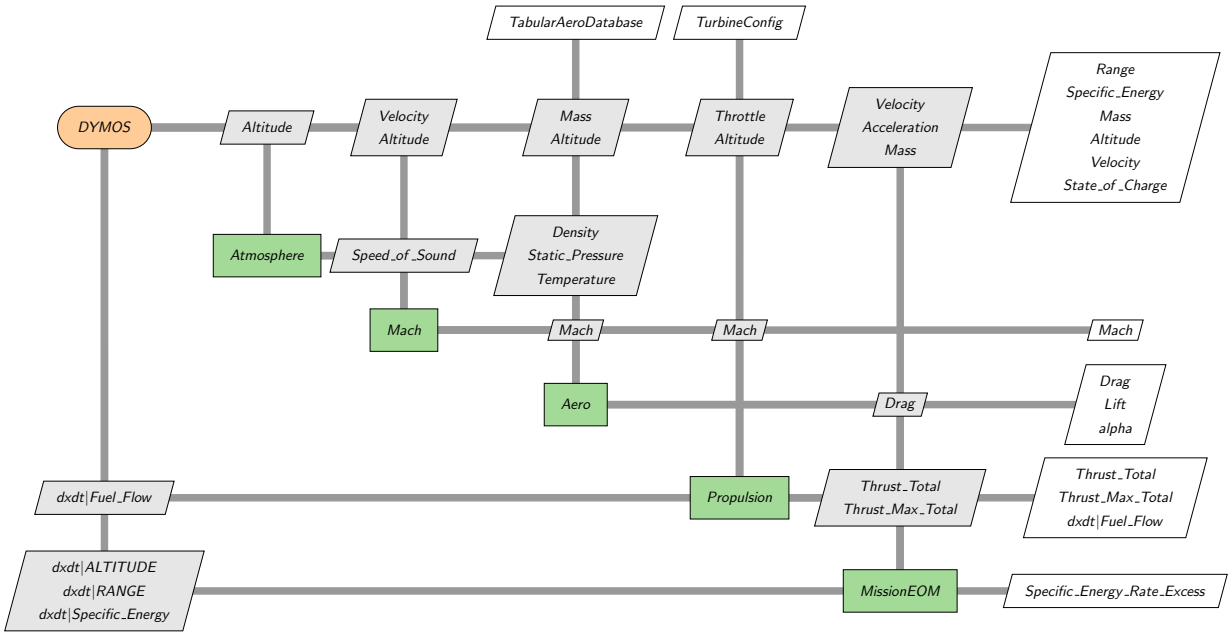

Fig. 5 Climb, Cruise, and Descent Phase XDSM. Outputs to the right are made available to the optimizer and made available to the user for graphing.

Electrification of the turbine is present in only the climb phase of the second demonstration problem (Sec III.B). The modified climb phase XDSM with electrical subsystem is shown in Figure 6. The key difference in the electrified climb phase is that pyCycle sends RPM values from the low-pressure shaft of the engine to the electrical subsystem.

Additionally, the amount of power injected by the electrical system onto the low-pressure shaft is dictated by the optimizer, preventing the need for separate loop-closure between pyCycle and the electrical system.

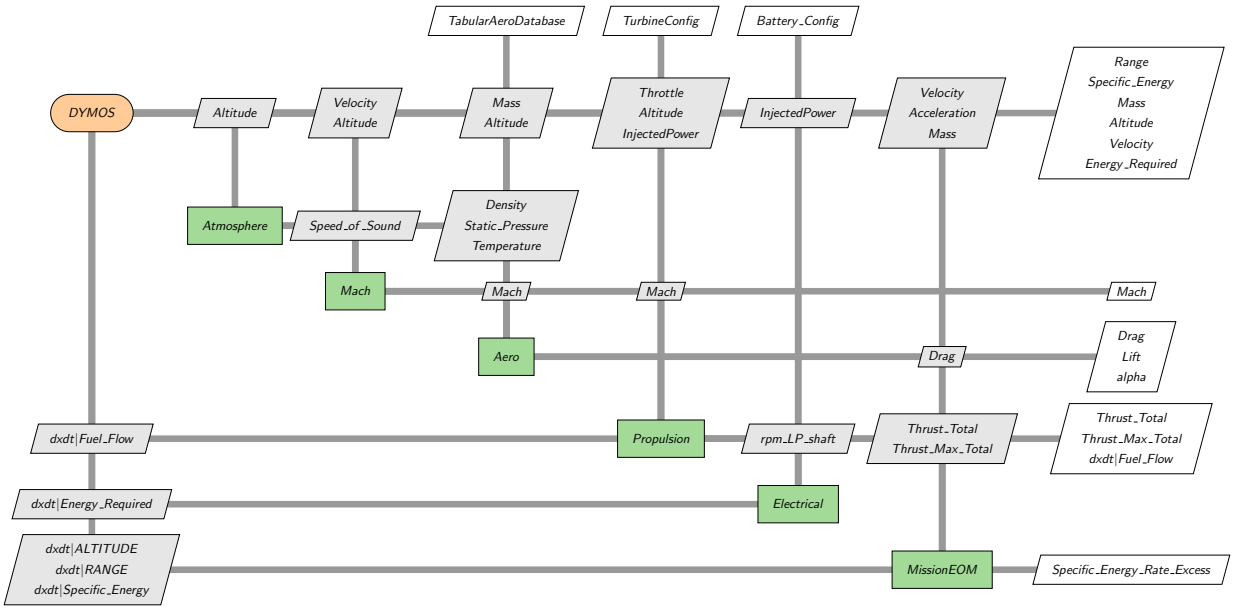

Fig. 6 Electrified Climb phase XDSM. Outputs to the right are made available to the optimizer and made available to the user for graphing.

III. Demonstration Problems

To demonstrate the effects of system coupling, the TTBW vehicle is evaluated over a typical long-haul mission with 154 passengers. The mission profile for the study is described in Sec II.G. Two demonstration problems are investigated, a non-electric TTBW mission and an electrified TTBW mission. Results and discussion are presented in Section III.C and III.D respectively.

A. Non-Electric Demonstration Problem

The first demonstration problem is a non-electrified TTBW. The mission XDSM for this configuration is shown in Figure 5. This non-electrified TTBW provides a reference design for discussion on what electrification adds to this study. This optimization includes all of the subsystems mentioned in Section II.A except the electrical subsystem. The non-electric TTBW optimization problem statement is described in Table 2. The optimization objective is to minimize the fuel burn for the mission. The only subsystems that are being re-sized for this first problem are the propulsion system, which is sized by design mass flow, and the weight subsystem, which is sized by the gross takeoff mass. The pseudospectral constraints listed in Table 2 are a result of the collocation method used to solve the mission trajectory.

Results of this optimization are shown in Figures 7, 8, 9, and 10. In Figures 7 and 8 the TTBW altitude profile shows that the aircraft successfully takes off at Mach 0.3 and ascends to cruising altitude at Mach of 0.8. After cruising for the majority of the mission, the aircraft descends and slows back down to Mach 0.3. Both graphs show smooth transitions between the three different phases of flight: climb, cruise, and descent. Minimizing fuel burn is the objective function for the mission and a graph of fuel-flow rate is shown in Figure 9. Maximum fuel-flow happens in climb with around 8000lbm/hour of fuel for a short time and tapering off to 2000lbm/hour once cruise conditions are reached. Fuel-flow drops sharply as the aircraft descends to landing altitude at which point it increases to a modest amount around 1800lbm/hour. The turbine throttle command for this case is shown in Figure 10. The figure shows that in climb, throttle is maxed at 1, decreasing only when cruise is attained. Throttle in descent is zero which indicates that a minimum amount of fuel is going to the engine. Throttle level of zero means the engine is still producing a minimum level of thrust, as was discussed in Section II.E.

Table 2 TTBW Optimization Problem Formulation, Non-Electrified

	Variable or Function	Size	Discipline
minimize	Fuel Burn	1	
with respect to	Design mass flow	1	Gas Turbine
	Gross Takeoff Mass	1	Weights
	Climb Duration	1	Trajectory
	Cruise Duration	1	
	Descent Duration	1	
	Altitude	24	
	Mass	24	
	Range	24	
	Velocity	24	
	Engine Throttle	9	
	Acceleration	9	
subject to	Mass Residual	1	Weights
	Throttle Constraints	24	Propulsion
	Pseudospectral Constraints	155	Trajectory

B. Electric Demonstration Problem

The second demonstration problem is an electrified TTBW model. The mission XDSM for the electrified TTBW is shown in Figure 6. In contrast to the previous study, this study contains the electrical subsystem detailed in Section II.D along with all the other subsystems mentioned previously. This gives the designer the ability to assess the system-level impact of electrification of the vehicle. Minimum fuel burn was the optimization objective for this study, similar to the first study. The optimization problem setup is described in Table 3. Subsystems that are being re-sized in this study include the turbine engine, the gross takeoff mass, the total amount of power required by the battery, and the electrical motor size.

Table 3 TTBW Optimization Problem Formulation, Electrified

	Variable or Function	Size	Discipline
minimize	Fuel Burn	1	
with respect to	Design mass flow	1	Gas Turbine
	Gross Takeoff Mass	1	Weights
	Nominal Energy	1	Electrical
	Motor Max Power	1	
	Climb Duration	1	Trajectory
	Cruise Duration	1	
	Descent Duration	1	
	Climb Electric Shaft power	9	
	Altitude	24	
	Mass	24	
	Range	24	
	Velocity	24	
	Engine Throttle	9	
	Acceleration	9	
subject to	Mass Residual	1	Weights
	Throttle Constraints	24	Propulsion
	Electric Power Constraints	14	Electrical
	Pseudospectral Constraints	155	Trajectory

C. Results

Results for the electrified TTBW optimization are shown in Figures 7, 8, 9, and 10. Additional summary results are presented in Tables 4 and 5. The fuel-flow command in climb is limited to a 2nd order polynomial which explains the curvature in Figure 10. If a higher order polynomial had been chosen for this there is a possibility the throttle command would be more complex. This polynomial order was the same for both demonstration cases. In addition, the electric power injected into the turbine engine was controlled by the optimizer but limited to a 2nd order polynomial, a decision that was made to simplify the problem and assist in optimizer convergence. The optimizer maximized the electric power injected throughout climb to its upper bound of 746kW.

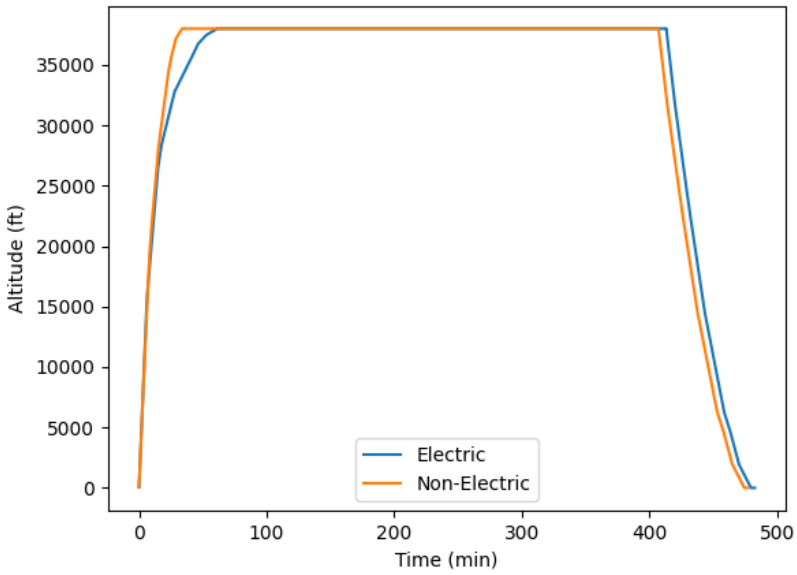

Fig. 7 Altitude Profile TTBW Mission.

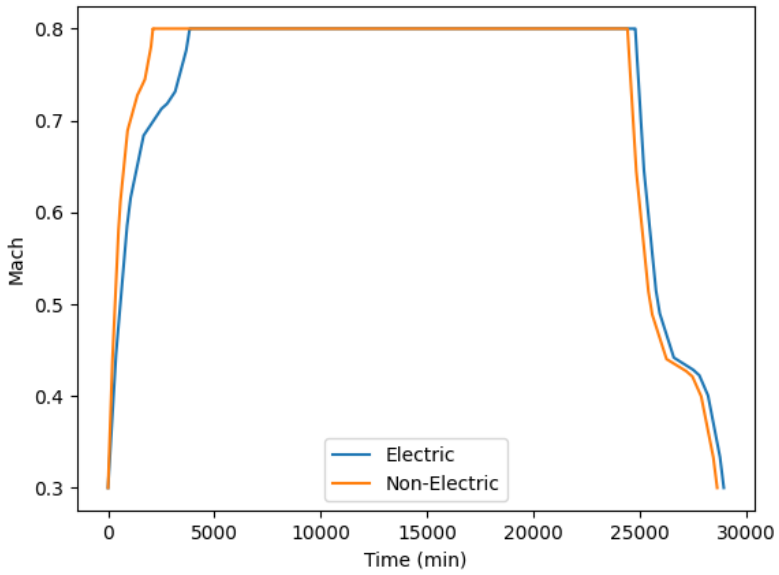

Fig. 8 Mach for TTBW Mission.

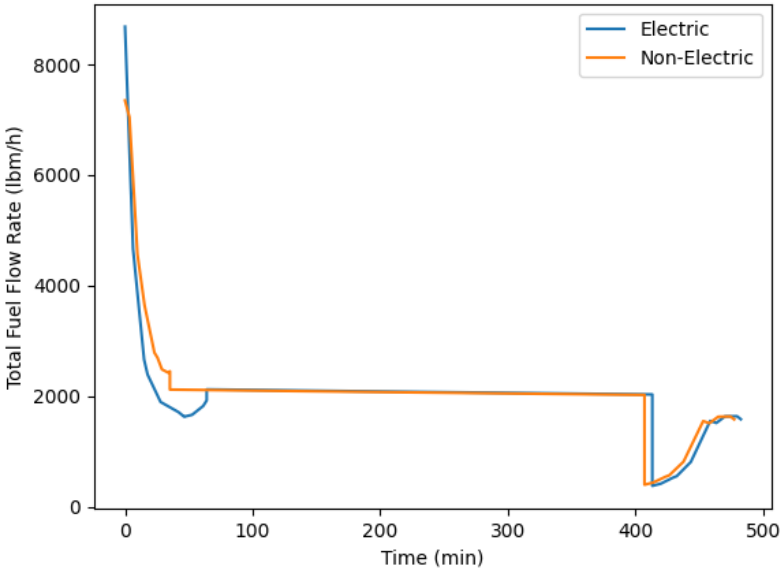

Fig. 9 Fuel Flow Objective for TTBW Mission.

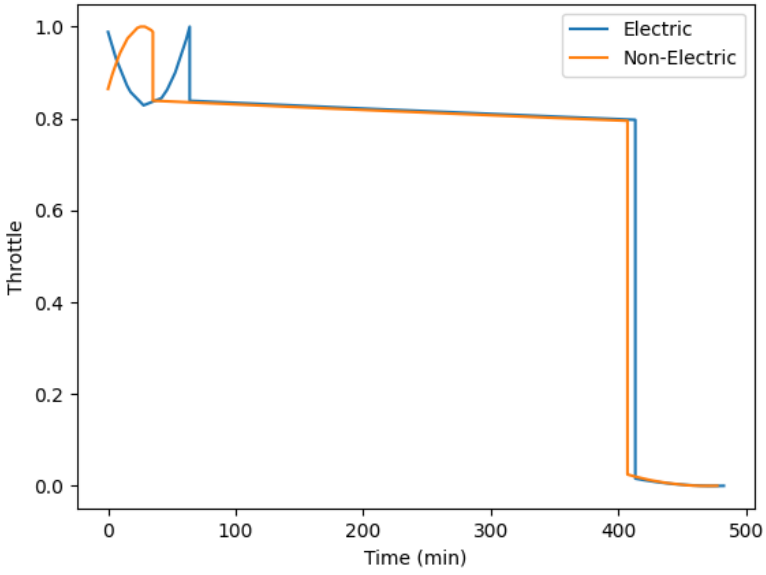

Fig. 10 Turbine Throttle Command for TTBW Mission.

Table 4 TTBW Optimization Results

Optimization	Fuel Burn (kg)	GTOW (kg)	Battery Mass (kg)	Battery Enrgy (kWh)	Battery Energy Density (Wh/kg)	Motor Max Power (kWh)
Non-Electric	7542	61941	0	0	N/A	0
Electrified	7350	62692	1221	611	2000	746

Table 5 Key Pre-Mission Engine Parameters.

Optimization	TOC Overall Pressure Ratio	TOC Bypass Ratio	TOC Net Thrust per Engine (N)	TOC Mass Flow (kg/s)
Non-Electric	57.8	13.6	14700	148
Electrified	57.8	13.6	14700	148

D. Discussion

Electrification of the TTBW produced some differences in the throttle profile during climb. Part of this variation can be attributed to the problem constraints of using only a 2nd order polynomial to control throttle and injected electrical power. This offers limited opportunity for the optimizer to modify the control. Using a higher order control would have decreased these differences. Throttle profiles during cruise and descent are essentially identical in both demonstration problems.

In the electrification study, with a battery energy density of 2,000 (Wh/kg), the optimizer chose to inject the maximum electrical power allowed onto the low-pressure shaft (746 kW). This energy density is significantly higher than state-of-the-art to encourage the optimizer to use some electrical power, because the model showed that current state of the art energy densities were not sufficient for the battery to buy its way on board in this study. Fuel burn did decrease in the electrified TTBW, however, only around 200kg of fuel was saved. The lack of significant fuel-burn improvement is similar to previous studies on electric thrust assist [17].

Turbine engine size did not decrease as anticipated with electrification. Further investigation of this revealed that the primary driver for the engine size is a 300 (ft/minute) climb rate constraint in cruise. Aircraft need to be able to climb quickly even when at cruise altitude and this constraint is implemented in Aviary. In the electrified TTBW study, the engine size was not able to decrease because there was still the need to meet this constraint during cruise, which is a phase of flight where the electrification system is not used. However, when this constraint is removed, the electrified TTBW down-sized the engine to below the size of the non-electric TTBW. Therefore, it is likely that the electrification of engines for climb-assist should consider satisfying the 300 (ft/minute) climb rate in cruise constraint via electric assist as well. This may cost additional battery mass or battery reserve. However, this appears to be one of the essential considerations for decreasing engine size.

For the electrified TTBW problem, the optimizer did choose to maximize the electric power injected, keeping it at 746kW throughout climb. If the battery had been less energy-dense, or if the control order had been higher, it is possible that the optimizer would have chosen much less power injection or chosen a non-uniform power injection profile. This type of behavior was observed in other studies not included in this paper.

The total computational time for this optimization of VSPAERO in pre-mission, as well as pyCycle, electrical, and weight co-design took around 3 hours on 10 hyperthread cores on a standard laptop. Optimization specifically runs multiple analyses based on what values the optimizer chooses to investigate. Further reductions in computation time can be derived from building the VSPAERO tables beforehand and importing directly. This is only possible because the aerodynamics were not optimized during this study. Aviary will support analysis-only missions which will not involve any optimization. It is expected that analysis-only missions will take a fraction of the time of an optimization mission.

Subsystem design for this study was originally intended to be more complex. There were plans to use the cell-model based on previous work by Chin et al [7]. However, during integration of this battery model, the optimization became unstable and was unable to close (successfully converge). This failure illustrates the need to sufficiently test subsystem models before integration with the larger optimization problem. For subsystems, it is essential to characterize their behavior independent of other subsystems over a large range of input conditions. Because subsystems are often strongly-coupled with each other, passing output from one subsystem into the input of another, it is easy for outputs from one subsystem to drive the subsequent subsystem into a regime where it cannot converge.

IV. Conclusion

This study has demonstrated rapid optimization of a TTBW concept vehicle with varying levels of electrification. All the optimizations were performed in Aviary, a novel tool for preliminary aircraft design. Comparisons of TTBW models from other sources were not included because the focus of the work was to demonstrate the capabilities of Aviary to model a novel aircraft. The electrified TTBW used electric motors and a battery to enhance the power on the low-pressure turbine engine shaft during climb. This enabled a slight decrease in fuel burn despite increasing Gross Takeoff Weight (GTOW). However, the turbine engine was not able to down-size itself as anticipated due to constraints for excess power in cruise. Results were presented for the mission profile and subsystem configuration for each optimization. Primary challenges were subsystem design and testing before inclusion in the larger optimization.

Acknowledgements

The work presented in this paper and the development of the Aviary tool were performed with support from NASA's Transformational Tools and Technologies (TTT), Advanced Air Transport Technology (AATT), and Electric Powered Flight Demonstration (EPFD) Projects. These projects are funded by Transformational Aeronautic Concepts Program (TACP), Advanced Air Vehicles Program (AAVP), and Integrated Aviation Systems Program (IASP) respectively. Funding for those Programs is provided by the Aeronautic Research Mission Directorate (ARMD).

Appendix

A. Initial Guess Matrix Information for Propulsion Mission Analysis

List of all of the state variables used for the initial guessing matrices:

- Burner fuel to air ratio
- Mass flow at flight conditions
- Bypass ratio at splitter
- Mechanical speed on the low-pressure shaft
- Mechanical speed on the high-pressure shaft
- Fan map R-line
- Low pressure compressor map R-line
- High pressure compressor map R-line
- Fan map corrected speed
- Fan stall margin at constant mass flow map corrected speed
- Low pressure compressor map corrected speed
- Low pressure compressor stall margin at constant mass flow map corrected speed
- High pressure compressor map corrected speed
- High pressure compressor stall margin at constant mass flow map corrected speed
- High pressure turbine map corrected speed
- High pressure turbine map pressure ratio
- Low pressure turbine map corrected speed
- Low pressure turbine map pressure ratio
- Core nozzle pressure ratio
- Bypass nozzle pressure ratio

References

[1] Harrison, N. A., Beyar†, M. D., Dickey, E. D., Hoffman, K., Gatlin, G. M., and Viken, S. A., "Development of an Efficient Mach=0.80 Transonic Truss-Braced Wing Aircraft," *AIAA SciTech Forum*, 2020. https://doi.org/https://doi.org/10.2514/6.2020-0011.

[2] Margetta, R., "NASA Issues Award for Greener, More Fuel-Efficient Airliner of Future," , No. 23-008, 2023. https://doi.org/https://www.nasa.gov/press-release/nasa-issues-award-for-greener-more-fuel-efficient-airliner-of-future.

[3] 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, No. 4, 2019, pp. 1075–1104. https://doi.org/10.1007/s00158-019-02211-z.

[4] 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.

[5] Hendricks, E., Falk, R., Gray, J., Aretskin-Hariton, E., Ingraham, D., Chapman, J., Schnulo, S., Chin, J., Jasa, J., and Bergeson, J., "Multidisciplinary Optimization of a Turboelectric Tiltwing Urban Air Mobility Aircraft," *American Institute of Aeronautics and Astronautics*, 2019. https://doi.org/https://doi.org/10.2514/6.2019-3551.

[6] Aretskin-Hariton, E., and Hurst, D. J., "Electrical Cable Design for Urban Air Mobility Aircraft," *American Institute of Aeronautics and Astronautics*, 2020. https://doi.org/https://doi.org/10.2514/6.2020-0014.

[7] Chin, J. C., Aretskin-Hariton, E., Ingraham, D., Hall, D., Schnulo, S. L., Gray, J. S., and Hendricks, E., "Battery Evaluation Profiles for X-57 and Future Urban Electric Aircraft," *AIAA Propulsion and Energy Forum*, 2020. https://doi.org/10.2514/6.2020-3567.

[8] Aretskin-Hariton, E., Bell, M., Schnulo, S., and Gray, J. S., "Power Cable Mass Estimation for Electric Aircraft Propulsion," *AIAA 2021 Aviation Forum*, 2021. https://doi.org/10.2514/6.2021-3021.

[9] Lyons, K., Margolis, B., Bowles, J., Gratz, J., Schnulo, S. L., Aretskin-Hariton, E., Gray, J. S., and Falck, R. D., *Advancement of the General Aviation Synthesis Program Using Python to Enable Optimization-Based Hybrid-Propulsion Aircraft Design*, 2023. https://doi.org/10.2514/6.2023-3224, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-3224.

[10] Welstead, J. R., Caldwell, D., Condotta, R., and Monroe, N., *An Overview of the Layered and Extensible Aircraft Performance System (LEAPS) Development*, 2018. https://doi.org/10.2514/6.2018-1754, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-1754.

[11] Hann, A., "Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design," *48th AIAA Aerospace Sciences Meeting*, AIAA 2010-657, 2010. https://doi.org/10.2514/6.2010-657.

[12] 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.

[13] Gray, J., Chin, J., Hearn, T., Hendricks, E., Lavelle, T., and Martins, J. R. R. A., "Chemical-Equilibrium Analysis with Adjoint Derivatives for Propulsion Cycle Analysis," *Journal of Propulsion and Power*, Vol. 33, No. 5, 2017, pp. 1041–1052. https://doi.org/10.2514/1.B36215, URL https://doi.org/10.2514/1.B36215.

[14] Campbell, R. L., "An approach to constrained aerodynamic design with application to airfoils," Tech. Rep. NASA/TP-3260, National Aeronautics and Space Administration Langley Research Center, Hampton, VA, November 1992. URL https://ntrs.nasa.gov/api/citations/19930003133/downloads/19930003133.pdf.

[15] "Documentation for the OpenVSP API," https://openvsp.org/api_docs/latest/, 2023. Accessed: 2023-11-14.

[16] Von Karman, T., "Compressibility Effects in Aerodynamics," *Journal of the Aeronautical Sciences*, Vol. 8, No. 9, 1941, pp. 337–356. https://doi.org/10.2514/8.10737.

[17] Frederick, Z. J., Hallock, T. J., Ozoroski, T. A., Chapman, J. W., Kuhnle, C. A., and Frederic, P. C., *Design Exploration of a Mild Hybrid Electrified Aircraft Propulsion Concept*, 2023. https://doi.org/10.2514/6.2023-4226, URL https://arc.aiaa.org/doi/abs/10.2514/6.2023-4226.