

Blade and Takeoff Trajectory Optimization of a Propeller-Driven Electric Aircraft with Acoustic Constraints

Daniel Ingraham* Robert Falck†
NASA Glenn Research Center, Cleveland, Ohio, 44135

In this work, a multi-disciplinary toolchain is described and used to optimize the takeoff trajectory of a electrified general aviation aircraft, subjected to acoustic constraints. The Dymos multi-disciplinary optimal control library is used to optimize the trajectory, with propeller aerodynamic and acoustic models provided by blade element momentum theory (CCBlade.jl) and acoustic analogy (AcousticAnalogies.jl) codes, respectively. Each model is implemented in the OpenMDAO framework, with all derivatives calculated either analytically or via automatic differentiation tools. The toolchain is applied to a hypothetical electrified form of the Cirrus SR20 and compared to the conventional piston-driven form.

Nomenclature

L	=	vehicle lift
T	=	thrust
D	=	vehicle drag
W	=	gross vehicle weight
m	=	gross vehicle mass
sos	=	speed of sound
μ_r	=	runway coefficient of friction, taken as 0.02 in this work
μ	=	air dynamic viscosity
v or tas	=	vehicle true airspeed
γ	=	flight path angle
h or alt	=	altitude/vehicle vertical position
r	=	range/vehicle horizontal position
α	=	vehicle angle of attack
P_{engine}, P_{motor}	=	engine/motor shaft power
ω	=	propeller rotation rate
t	=	time
ρ	=	air density
θ	=	propeller pitch
p_{rms}	=	root-mean square of acoustic pressure
p_{ref}	=	reference acoustic pressure, $20\mu Pa$
ϕ	=	propeller twist distribution
E_{total}	=	total energy expended by the aircraft during takeoff

I. Introduction

Noise is a critical limiting factor in the design of many types of aircraft. For example, Urban Air Mobility/Advanced Air Mobility (UAM/AAM) vehicles, meant to operate in densely-populated areas, will need to keep the noise they emit at a level acceptable to the general public[1], and a vehicle's takeoff and landing noise must meet certain criteria* before operating in airports, and is regulated by government agencies such as the Federal Aviation Administration in the United States. If new aircraft are to be acceptable to the general public and meet regulations, designers need tools that allow them to quickly determine the effect design changes have on vehicle aerodynamic and acoustic performance.

*Research Aerospace Engineer, Member AIAA.

†Research Aerospace Engineer, Member AIAA.

*<https://www.faa.gov/noise/levels>

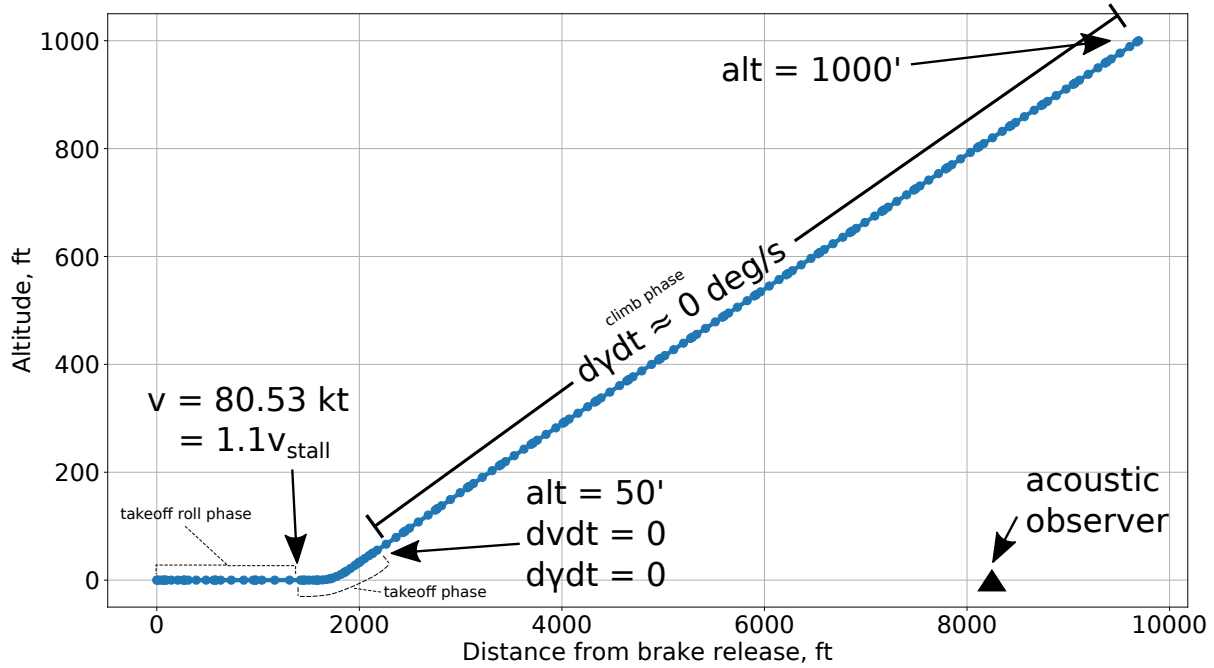


Fig. 1 Diagram of the three-phase takeoff trajectory

The trajectory of an aircraft has a significant effect on many aspects of its performance, especially on the acoustic performance, where noise source reduction may be in tension with both the aerodynamic performance of the aircraft and the noise experienced by an observer on the ground. Ideally, the trajectory would be designed concurrently with other features. Multi-disciplinary optimal control libraries make this feasible, as seen in applications related to urban or advanced air mobility[2, 3] and commercial supersonic aircraft[4]. An aircraft's trajectory also has a strong effect on the noise experienced by observers on the ground, but including acoustic constraints in a multidisciplinary trajectory optimization can be problematic[5]. The goal of this work is to perform a takeoff trajectory optimization of a general aviation aircraft, subject to both takeoff performance and acoustic constraints, where the acoustic predictions is provided by physics-based models. Specifically, a hypothetical electrified version of the Cirrus SR20, inspired by a recent study of Berton and Nark[6], will be studied. In that work, the authors considered the potential noise benefits of replacing a piston engine with an electric motor in a general aviation aircraft. By exploiting the power characteristics of the electric motor—namely, that a motor can operate at near maximum power over a broad range of shaft speeds—Berton and Nark showed that lowering the propeller shaft speeds while simultaneously increasing the blade pitch reduced the noise experienced by a ground observer during takeoff by 12dB, without significantly affecting the climb rate. Normally powered by a Continental Motors IO-360-ES reciprocating engine, the SR20 was hypothetically modified to use an electric motor producing the same maximum shaft power at sea level as the IO-360-ES. All other aspects of the aircraft (weight, aerodynamics, and propeller design) were unchanged in the Berton and Nark work.

This current effort differs from Berton and Nark's work in the following ways:

- Rather than evaluate a range of propeller rotation rates ω decided a priori, the single ω for the constant-speed propeller is a design variable in an optimization process;
- An implementation of Blade Element Momentum Theory is used to optimized the propeller shape (chord and twist along the length of the blade), rather than specifying a constant blade design;
- The OpenMDAO framework is used to link an propeller aerodynamic model, propeller acoustic model, and optimal control framework together and perform the optimization, using either analytic derivatives or derivatives obtained via algorithmic differentiation (AD) in conjunction with a gradient-based optimizer.

II. Problem Statement and Computational Approach

Figure 1 describes the takeoff trajectory that is optimized in this work. The trajectory is divided into three phases: takeoff roll, takeoff, and climb. The takeoff roll phase begins with the vehicle at rest and zero altitude, and ends once it has accelerated to 80.53kn, 10% above the stall speed of the aircraft. The vehicle’s velocity during the takeoff roll phase is obtained by integrating the ODE

$$m\dot{v} = T - D - \mu_r (W - L) . \quad (1)$$

No control variables are present in this phase—the aircraft’s angle of attack α is fixed at zero throughout.

The takeoff phase occurs after the takeoff roll phase, which is governed by the flight ODE, here the three degree of freedom dynamics from Bryson et al.[7]:

$$\begin{aligned} \dot{r} &= v \cos \gamma \\ \dot{h} &= v \sin \gamma \\ m\dot{v} &= T \cos \alpha - D - W \sin \gamma \\ m v \dot{\gamma} &= T \sin \alpha + L - W \cos \gamma . \end{aligned} \quad (2)$$

The vehicle is controlled by the angle of attack α . As shown in Figure 1, the altitude at the end of the takeoff phase is constrained to be equal to 50 ft (the height of a theoretical obstacle), and the time derivatives of v and γ at the same point in time are constrained to zero (for a constant climb rate in the next phase). Additionally, the second derivative of α with respect to time was constrained to be between $\pm 5 \text{ deg/s}^2$ to prevent “ringing” (large unphysical oscillations) in the α control variable.

The climb phase is governed by the same flight ODE (Eq. 2) as the takeoff phase. To achieve a nominally constant climb rate, the flight path angle rate $\frac{d\gamma}{dt}$ is constrained to be between -0.01 deg/s and 0.01 deg/s throughout the length of this phase. The final altitude is typically constrained in this work to be equal to 1000 ft, as shown in Figure 1, but, as will be seen, was increased in some cases. The angle of attack is again the only control variable in this phase, like the previous “takeoff” phase.

Also shown in Figure 1 is the location of a ground observer, 2500 meters downfield of the initial position of the aircraft. This is where the noise emitted by the SR20 will be predicted, mimicking the noise certification procedure used for general aviation aircraft.

Figure 2 shows a simplified eXtended Design Structure Matrix (XDSM) of the proposed trajectory optimization—one that does not include aspects related to the pseudospectral collocation method that Dymos uses to integrate the aircraft equation of motion. Along the top of the diagram are the optimization design variables: the angle of attack control α , the propeller rotation rate ω , and the propeller chord and twist distributions. The objective of the optimization, represented by $\text{OBJ}|E_{\text{total}}$ in figure 2, will be to minimize E_{total} , the total energy expended by the aircraft to reach a target altitude (typically 1000 feet in this work), and the primary constraint will be the overall sound pressure level ($\text{CON}|OASPL$) experienced by the observer on the ground. Additional constraints related to the dynamics of the trajectory depicted in figure 2 are represented by $\text{CON}|trajectory$ in figure 2 (the final boundary constraint on the true airspeed in the takeoff roll phase, the path constraint on the second time derivative of α in the roll phase, the final boundary constraint on altitude in the climb phase, etc.). The oblong “Integrator (Dymos)” element in figure 2 represents a generic time integration algorithm, whose function is to take state time derivatives (here, the time derivatives of r , alt , v , and γ) and integrate them in time to obtain the states themselves for the combination of design variables provided by the optimizer. These state rates are computed by the equations of motion (Eqs. 2 and 1), discussed previously and represented by the “EOM” box in figure 2. The “Atmos” “Aero”, and “Prop” elements upstream of “EOM” represent the atmospheric, aerodynamic, and propeller models in this work, and will be described in turn below, in addition to the (quite simple) piston engine and electric motor models.

A. Engine and Motor

The simple model of the Continental Motors IO-360-ES piston engine and hypothetical electric motor from Berton and Nark[6] will be used here. Specifically, the shaft power at wide open throttle and sea level for the IO-360-ES will be solely a function of engine RPM, and will be modeled by

$$P_{ICE} = \frac{200 - 140}{2700 - 2100} (\omega - 2100) + 140, \quad (3)$$



where ω is the rotation rate in revolutions per minute (ranging from 2100 rev/min and 2700 rev/min) and P_{ICE} is the shaft power in horsepower. The shaft power of the hypothetical electric motor will be a constant 200 horsepower, i.e.,

$$P_{EMOTOR} = 200 \quad (4)$$

with identical restrictions on ω .

B. Atmosphere Model

The atmosphere model used in this work are provided by the `USatm1976Comp` class from Dymos, which is an implementation of the US Standard Atmosphere, 1976[8]. The role of the atmosphere model, as shown in figure 2, is to calculate the atmospheric density ρ , speed of sound so_s , and dynamic viscosity μ from the altitude at each point along the trajectory, and provide these properties to the downstream models.

C. Aircraft Aerodynamics

Aircraft lift and drag coefficients for the SR20 were taken from Berton and Nark [6], where a “handbook method” from Roskam[9] was used to estimate lift and drag as a function of angle of attack, and then adjusted to match the rates of climb and takeoff field length information given in the SR20 Pilot’s Operating Handbook. In the model, the lift and drag coefficients are interpolated using SciPy’s cubic interpolation function, called from OpenMDAO’s `MetaModelStructuredComp`[†], and are then dimensionalized to lift and drag forces the dynamic pressure and SR20 reference area. These forces are then passed to the Equations of Motion 1 and 2.

D. Propeller Aerodynamics

The propeller aerodynamics are predicted here using blade element momentum theory (BEMT). BEMT begins by dividing the blade spanwise into a finite number of radial stations, or “blade elements.” Each blade element is treated as an independent airfoil with associated lift and drag curves. A control volume is then formed that encloses the propeller and the flow far upstream and downstream. By applying a momentum balance to this control volume, and including the interaction between the fluid and blade elements via airfoil lift and drag data, the velocity induced by the propeller in the axial and circumferential directions can be predicted. The induced velocities can then be used to calculate desired propeller performance metrics, such as the shaft power, the torque and thrust, and propeller efficiency.

The BEMT implementation used here will be `CCBlade.jl`[‡][10, 11], an open-source differentiable propeller and wind turbine aerodynamic prediction tool written in the Julia programming language. In a typical BEMT, a system of two nonlinear equations are solved for the axial and circumferential components of the velocity induced by each propeller blade element. `CCBlade.jl` reduces this two-equation system with one equation for the inflow velocity angle, which is more easily solved using bracketing methods (in `CCBlade.jl`, the Brent method available in the `FLOWMath.jl`[§] package). The derivatives will be calculated using the `ForwardDiff.jl`[¶][12] Julia package, called via `DifferentiationInterface.jl`^{||}[13] with the differentiation of implicit parts of the calculation efficiently handled by `ImplicitAD.jl`^{**}[15].

The Cirrus SR20 uses a constant-speed propeller, where the propeller pitch is automatically adjusted during flight to maintain a rotation rate set by the pilot. In this work, the propeller pitch at each point in the trajectory is adjusted such that the shaft power required for the propeller as predicted by `CCBlade.jl` matches the shaft power produced by either the piston engine (Eq. 3) or electric motor (Eq. 4), i.e.,

$$R(\theta) = P_{\text{propeller}}(\theta, \omega, \dots) - P_{ICE/EMOTOR}(\omega) = 0. \quad (5)$$

Equation 5 is solved here using the Brent method available in the `FLOWMath.jl`. Figure 4 shows the thrust and power of an arbitrary blade design (plotted in figure 3) as a function of propeller pitch and RPM for 1, 50, and 100 kts (true airspeeds spanning the range expected during the trajectory optimization). The colored dashed lines in the power plots indicate the output power the IO-360-ESI engine is capable of producing at a given RPM—solving Eq. 5 is equivalent

[†]https://openmdao.org/newdocs/versions/latest/features/building_blocks/components/metamodelstructured_comp.html

[‡]<https://github.com/byuflowlab/CCBlade.jl>

[§]<https://github.com/byuflowlab/FLOWMath.jl>

[¶]<https://github.com/JuliaDiff/ForwardDiff.jl>

^{||}<https://github.com/JuliaDiff/DifferentiationInterface.jl>, see also reference [14]

^{**}<https://github.com/byuflowlab/ImplicitAD.jl>

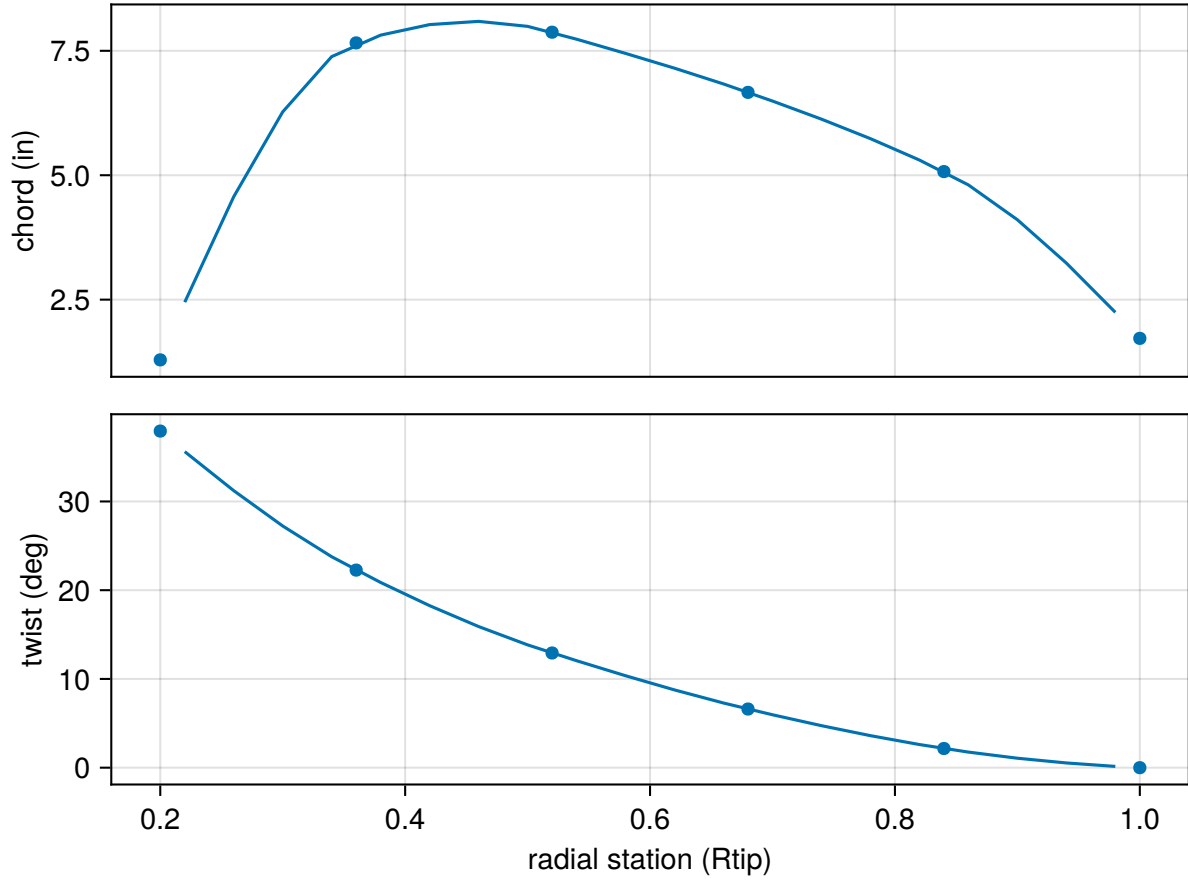


Fig. 3 Baseline blade design used in this work. The circular markers indicate the Akima spline control points that are used as design variables in the optimization, and the lines are the interpolated chord and twist distributions passed to the propeller aerodynamic and acoustic models.

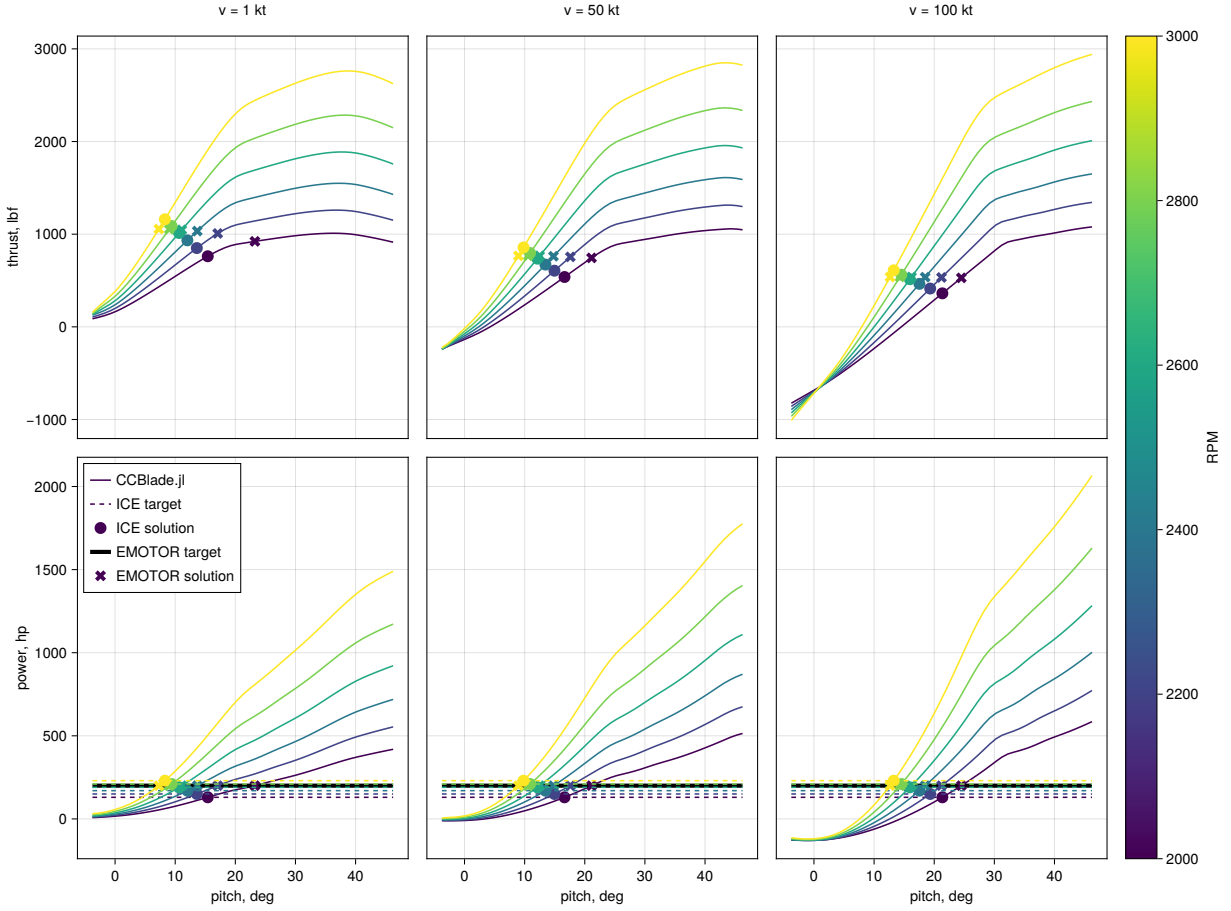


Fig. 4 Thrust and shaft power produced by the baseline blade design (figure 3) as a function of propeller pitch and RPM for 1, 50, and 100 knots. The colored dashed lines in the power plots indicate the output power the piston engine (internal combustion engine, ICE) is capable of producing at a given RPM, and the thick solid black line indicates the output shaft power of the electric motor model (which is not sensitive to RPM). Colored circular markers indicate the point at which the ICE shaft power matches the CCBlade.jl shaft power, and colored X markers indicate the same for the electric motor shaft power.

to finding the intersection between the dashed and solid colored lines of the same RPM (color) in the power plots of figure 4. These intersections are indicated by the circular markers, and the thrust value that corresponds to each intersection is indicated by similarly colored markers in the thrust plots. For the electric motor, the output shaft power is not a function of RPM and is indicated by the single horizontal black line in figure 4. The intersection between the CCBlade.jl power predictions and the electric motor shaft power is indicated by the colored X-shaped markers in the power plots, and the thrust corresponding to this intersect is indicated by similar markers in the thrust plots. We see that, as expected, increasing either RPM or pitch tends to increase the propeller’s power and thrust, and increasing forward speed tends to decrease the thrust, as increasing the inflow velocity will tend to reduce the local angle of attack at each blade station, unloading the blade. The thrust produced by the propeller powered by the electric motor is much less sensitive to RPM, and so we would expect that, for a fixed blade design, varying the RPM for the electric motor case would have little effect on the trajectory. When solving Eq. 5, we find it useful to search for a root by first evaluating $R(\theta)$ for a high pitch value, and then for progressively small pitch values, looking for a sign change relative to the first $R(\theta)$ evaluation. This bracketing interval is then passed to the Brent solver to find the root.

The propeller chord and twist distribution in this work was parameterized by Akima splines, again provided by the FLOWMath.jl package, and then interpolated onto a denser radial grid before being passed to CCBlade.jl and AcousticAnalogies.jl for the aerodynamic and acoustic predictions. These Akima spline control points (shown as circular markers for the baseline blade design in figure 3) were included as design variables in the trajectory optimization. Using the propeller shaft power balance of Eq. 5 with an optimization that includes the full propeller twist distribution as a design variable can pose challenges to the optimizer. The pitch θ in Eq. 5 has the effect of uniformly increasing the twist along the blade, i.e. the effective twist for a given operating point is actually $\phi + \theta$, where ϕ is the radial twist distribution. This means that, when ϕ is a design variable, there is a search direction the optimizer can take that will have no effect on any of the model outputs, namely if each ϕ is increased or decreased by a uniform amount (and all other design variables are left unchanged). For example, a uniform increase of 3° in ϕ with no change in any of the other propeller model inputs will lead to the propeller power balance to reduce the pitch θ by 3° , and yield the same thrust and noise outputs. This confuses the optimizer in isolated propeller optimizations, and did seem to cause problems with the takeoff trajectory optimizations that are the subject of this work. To remedy this, when including the blade twist as a design variable, the Akima twist control point at the tip was fixed to 0° , eliminating this pathological search direction.

Smoothed NACA 0012 lift and drag coefficients based on XFOIL data^{††} were used, along with the local Reynolds number and Mach number corrections provided by CCBlade.jl. Like what is actually included on the SR20, the propeller was assumed to have three blades, a 74-inch tip diameter, and a 14.8 inch hub diameter.

The objective of the trajectory optimization in this work will be to minimize E_{total} , the total energy expended by the aircraft to reach an altitude of 1000 feet. Since the shaft RPM and thus shaft power is constant throughout the takeoff trajectory, E_{total} can be calculated simply via

$$E_{\text{total}} = Pt_{\text{final}} \quad (6)$$

where t_{final} is the total duration of the takeoff trajectory, and P is the shaft power provided by either the piston engine Eq. 3 or electric motor Eq. 4. (Had the power varied throughout the trajectory, the energy rate could have been added to the equations of motion and integrated by Dymos, but that is unnecessary complication here.)

E. Propeller Acoustics

The tonal noise radiated by the propeller is predicted using an acoustic analogy, specifically Farrasat’s 1A formulation[16, 17] of the Ffowcs-Williams Hawking (FWH) approach[18]. Like all acoustic analogies, the FWH method begins with the Navier-Stokes equations, and algebraically manipulates them into a propagation term and one or more source terms. In the case of the FWH equation, the propagation term is the unsteady wave equation for acoustic pressure, and three source terms are identified: the monopole, dipole, and quadrupole terms. The FWH equation is solved using generalized functions and the free-space Green’s function for the wave equation, which require the evaluation of a surface integral of the monopole and dipole terms, and a volume integral of the quadrupole term. In propeller and rotor acoustic analysis, the integration surface is often taken as the blade surface, and the volume integral associated with the quadrupole term is ignored. Farrasat’s F1A formulation, the form of the FWH approach used here, is algebraically equivalent to the original FWH method, but switches the order of the surface integral and a temporal derivative, which is numerically beneficial.

A compact version Farrasat’s 1A formulation of the FWH, including the compact approximation to the so-called monopole term[19, 20] is used in this work: the integration surface (here, the propeller blade) usually associated with the

^{††}<http://airfoiltools.com/airfoil/details?airfoil=n0012-il>

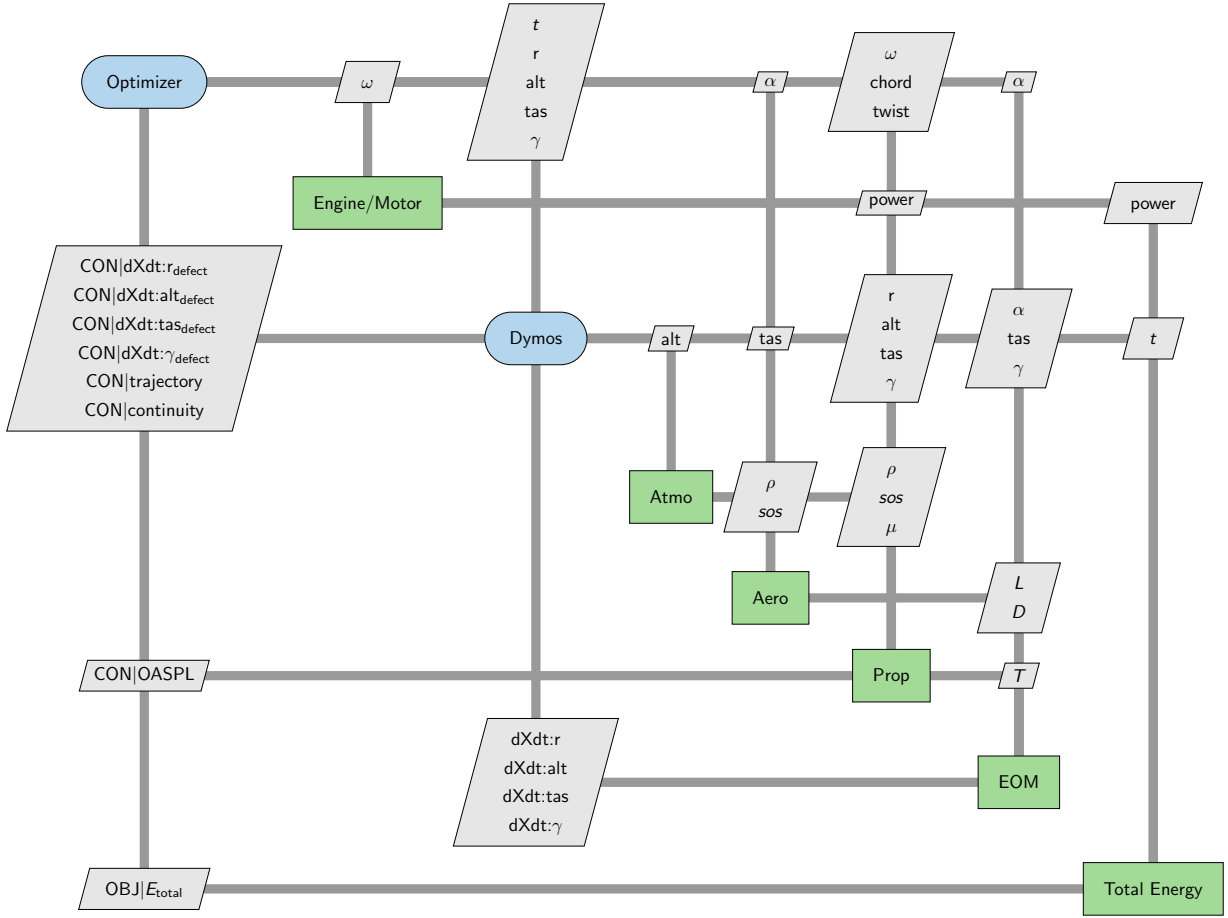


Fig. 5 eXtended Design Structure Matrix (XDSM) of the takeoff trajectory, showing details relevant to the Dymos optimal control library

method is replaced with a radial lifting line on which the net aerodynamic forces at each radial station act. The compact form of the F1A equation assumes that the original integration surface is elongated and that the acoustic observer distance is much larger than the characteristic thickness of the blade—both of these criteria are likely to be satisfied by a farfield noise prediction of a propeller blade. Besides reducing the computation cost of the non-compact F1A method, the inputs needed for the compact F1A calculation directly correspond to the outputs provided by blade element momentum theory, which is quite convenient. The compact F1A implementation will be provided by `AcousticAnalogies.jl`^{‡‡}, and differentiated (as part of the propeller model) via the `ForwardDiff.jl` and `DifferentiationInterface.jl` packages, like the propeller aerodynamic model.

The overall sound pressure level (OASPL) will be the noise metric used in this work, which is the mean-square of the acoustic pressure expressed in decibels, i.e.,

$$\text{OASPL} = 10 \log_{10} \frac{p_{\text{rms}}^2}{p_{\text{ref}}^2} \quad (7)$$

where p_{ref} is the usual reference pressure of $20 \mu\text{Pa}$.

	Variable/Function	Size
minimize	total energy consumption	
with respect to	propeller rotation rate	1
	chord distribution	6
	twist distribution	5
	angle of attack control	36
	pseudospectral state and time inputs	≈ 370
subject to	observer OASPL	60
	trajectory constraints	125
	pseudospectral constraints	336

Table 1 Full trajectory optimization problem statement

F. Flight Dynamics and Optimal Control

In this work, these equations of motion will be integrated by Dymos^{§§}[21], a general-purpose ODE solver that uses techniques from optimal control theory. Specifically the Radau pseudospectral method will be used to express the trajectory optimization as a nonlinear optimization problem, where the ODE state variables r , h , γ , and v are optimization design variables, and state rates as calculated by the interpolating polynomial of the pseudospectral method are constrained to equal those calculated via the right-hand side of each Equation of Motion 1 and 2 (the “collocation defects”). The XDSM of figure 5 shows this diagrammatically. In the XDSM, the time and states are now design variables set by the optimizer, and the collocation defects $\text{CON}|dXd t : r_{\text{defect}}$ through $\text{CON}|dXd t : \gamma_{\text{defect}}$ are calculated by Dymos and provided to the optimizer as constraints, in addition to continuity constraints (e.g. the time and states at the end of a given phase should match the start of the next phase).

Not shown in figure 5 is a detail related to the takeoff roll phase. Because the takeoff roll phase does not contain any controls, it is especially amenable to being integrated using Dymos’s `solve_segments` functionality^{¶¶}, where the collocation defects are solved for using an OpenMDAO NewtonSolver. In this work, `solve_segments` was used to integrate the takeoff roll phase, starting from the initial time level.

G. Optimization Framework

Each of the above models are linked together with the OpenMDAO optimization framework. OpenMDAO is an open-source Python implementation of the modular analysis and unified derivatives[22] (MAUD) approach to multidisciplinary design, analysis, and optimization (MDAO). OpenMDAO allows the application developer to express an analysis as a connected series of components, where each component uses inputs to calculate not only the outputs, but also the derivatives of the outputs with respect to the inputs. It then assembles these partial derivatives together into a total derivative, i.e. the derivative of the problem objective and constraints with respect to the design variables. These total derivatives are passed by OpenMDAO to a gradient-based optimizer to find the solution to the optimization problem. The OpenMDAO framework is used here to couple together the propeller and trajectory model codes described above, and the pyOptSparse^{***}[23] Python package is used to call SNOPT[24], a sparse nonlinear optimizer.

III. Results

Table 1 summarizes the “full” optimization problem that is the focus of this work. Before discussing the results of the “full” optimization, however, a series of simplified cases are presented here.

^{¶¶}<https://github.com/OpenMDAO/AcousticAnalogies.jl>

^{§§}<https://github.com/OpenMDAO/dymos>

^{¶¶}https://openmdao.org/dymos/docs/latest/getting_started/collocation.html#satisfying-defects-with-a-nonlinear-solver

^{***}<https://github.com/mdolab/pyoptsparse>

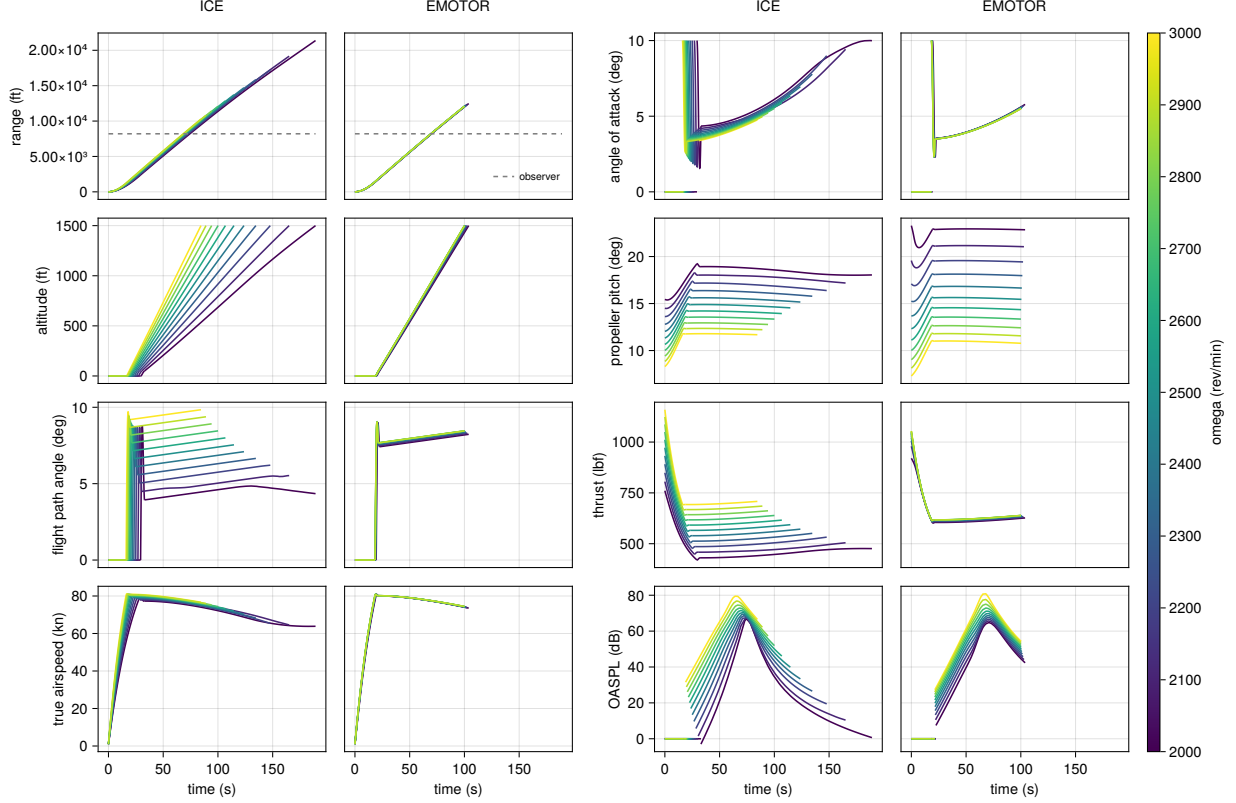


Fig. 6 Case I time histories

A. Case I: Fixed blade design, RPM sweep, no noise constraint

The first simplified case (“Case I”), summarized by figures 6 and 7 will be identical to that described by table 1, but will 1) use a fixed blade chord and twist, 2) not include an OASPL constraint, 3) sweep the propeller rotation rate instead of including it as a design variable in the optimization. Figure 6 shows the time histories of a number of different variables associated with this case, for both the piston-driven (ICE) and electric motor (EMOTOR) power plants. The state time histories for the ICE and EMOTOR cases are shown in the first and second column, respectively, with subsequent subplots showing the angle of attack control, the propeller pitch and thrust, and finally the OASPL for the ICE and EMOTOR settings. We see that, for the electric motor with its shaft power that is insensitive to RPM, varying the RPM has little effect on the trajectory itself: as the RPM is decreased, the propeller model increases the propeller pitch to match the shaft power provided by the motor, which also maintains thrust and thus essentially the same trajectory path as the higher RPM values. For the piston-driven cases, with its shaft power directly proportional to RPM, decreasing the RPM causes the propeller thrust to decrease sharply, reducing the flight path angle and climb rate. For both the ICE and EMOTOR cases, however, the noise experienced by the ground observer sharply decreases with reduced RPM. The strength of the propeller tonal noise source will of course decrease significantly with decreased RPM, but it is interesting to note that the noise level on the ground was reduced for the ICE case, where the aircraft was significantly *closer* to the observer for the low-RPM values.

Figure 7 shows the total energy objective and maximum OASPL as a function of propeller RPM for Case I. Perhaps counter-intuitively, the figure shows that increasing the RPM and thus energy consumption rate of the piston engine actually decreases the amount of energy required to climb to the 1000 ft altitude, as the increases in energy consumption were offset by the decrease in the duration of the takeoff trajectory. As expected from the previous figure 6, the peak OASPL is strongly proportional to the RPM. Additionally, the noise reductions associated with reduced RPM for the electric motor case come with mild energy consumption penalties, as might be expected from Berton & Nark’s earlier work[6].

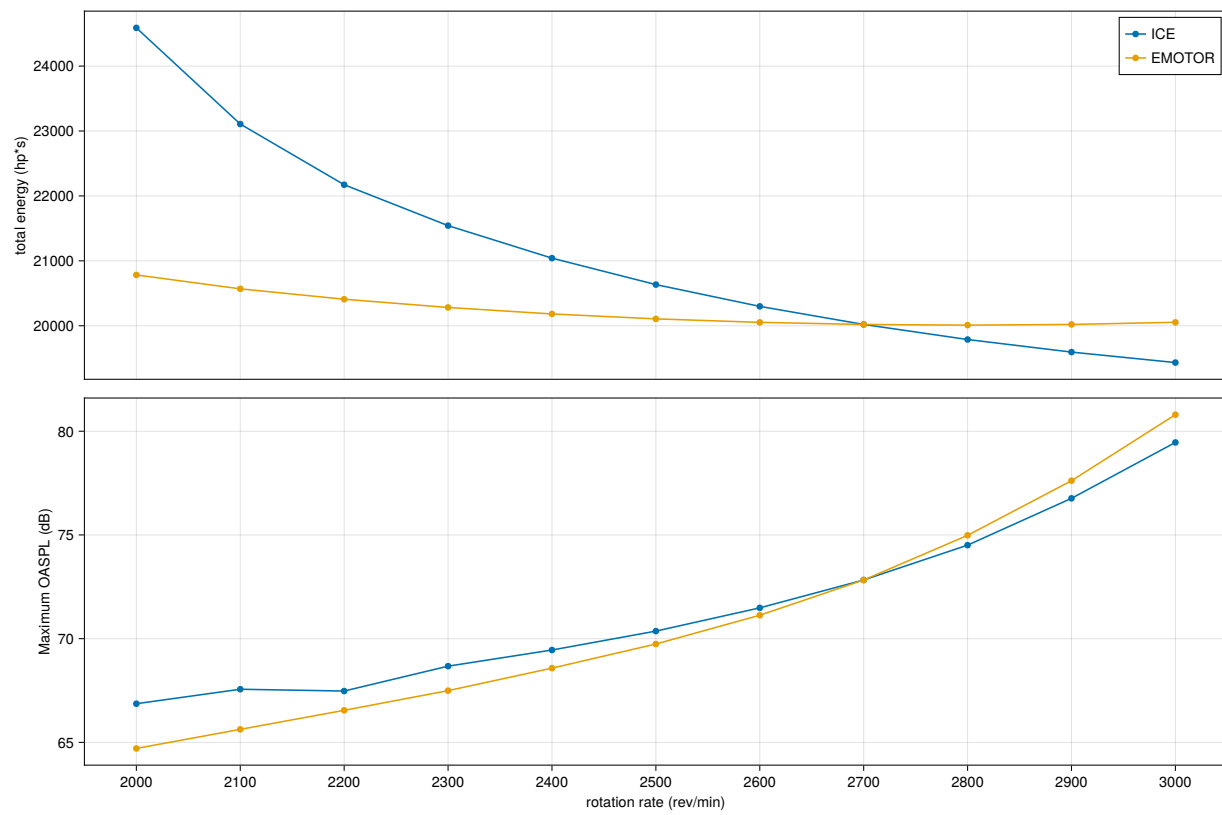


Fig. 7 Case I total energy and OASPL vs RPM

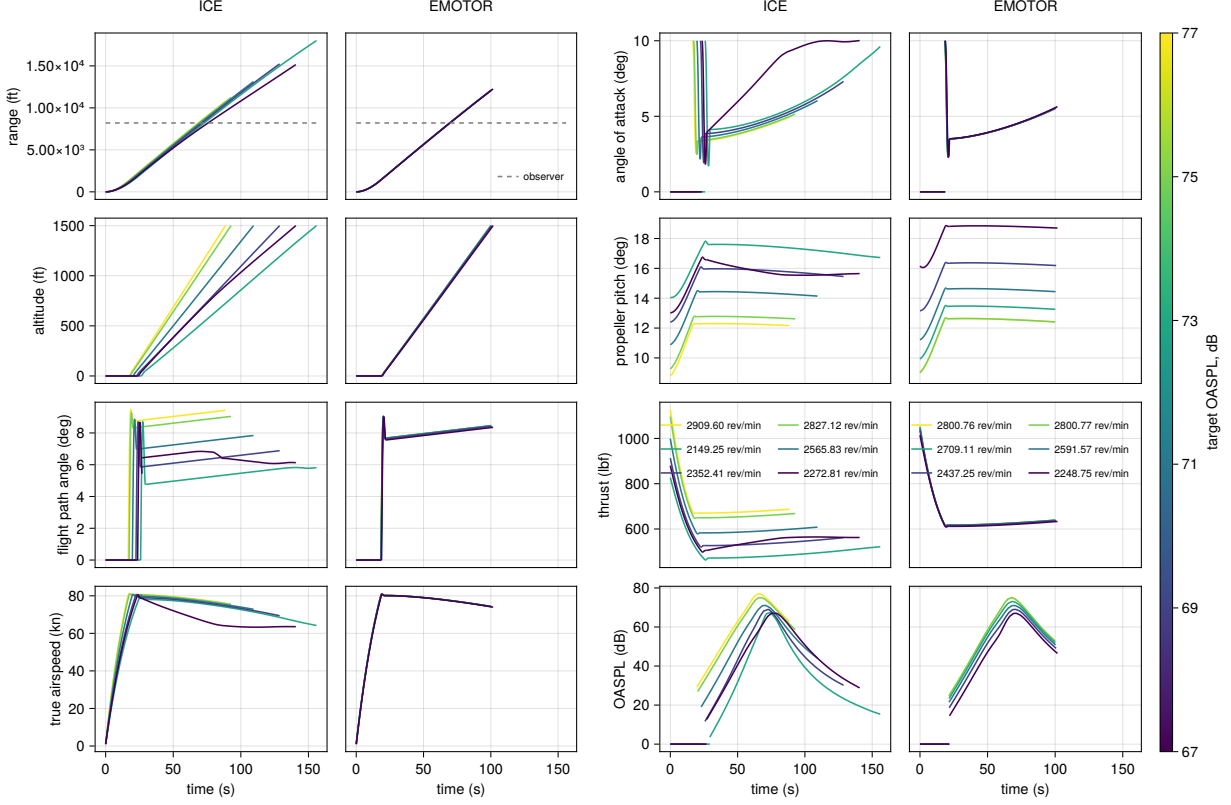


Fig. 8 Case II time histories

B. Case II: Fixed blade design, RPM design variable, with acoustic constraint

The next case (“Case II”) also includes a fixed blade design, but instead of sweeping the propeller rotation rate, the rotation rate will be included as a design variable in the optimization, and the value of an OASPL constraint will instead be swept. Figure 8 shows the time histories of this case in a form similar to that of figure 6, but with the color bar indicating the value of the OASPL constraint instead of RPM. The rotation rates for each case are indicated by the colored legends in the thrust subplots. The results show that, for the electric motor case, the optimizer decided to decrease the RPM to meet the OASPL constraint, as might be expected from the previous case. The results for the ICE case are a bit more complicated. For the most part, the optimizer also decided to decrease RPM, maintaining a consistent true airspeed and giving up flight path angle to meet the noise constraint. For the most restrictive acoustic constraint, however, the optimizer opted to significantly reduce the true airspeed to maintain a higher flight path angle, perhaps directing the noise away from the observer at a critical portion of the trajectory. It does appear, for this most restrictive acoustic constraint value, the trajectory is at least very close to violating the path constraint on climb phase’s $\frac{dy}{dt}$. The optimizer did report this and all optimizations shown in the figure as feasible, however, possibly indicating that this path constraint is poorly scaled.

Figure 9 shows the relationship between total energy, overall sound pressure level, and RPM for Case II. As expected from the previous figure 8, the EMOTOR results are very similar to the case with swept RPM. The ICE results also follow the trend of the swept RPM case, though perhaps not as smoothly. In general the ICE optimizations were more difficult to converge, especially for the lower RPM values (which are technically outside of the valid RPM range of the IO-360-ES motor).

C. Case III: Propeller chord and twist design variables, RPM sweep, no acoustic constraint

Figure 10 shows the time histories for Case III, where we again sweep the RPM, do not include an OASPL constraint, but *do include* chord and twist design variables in the optimization. The format is similar to that of the previous time

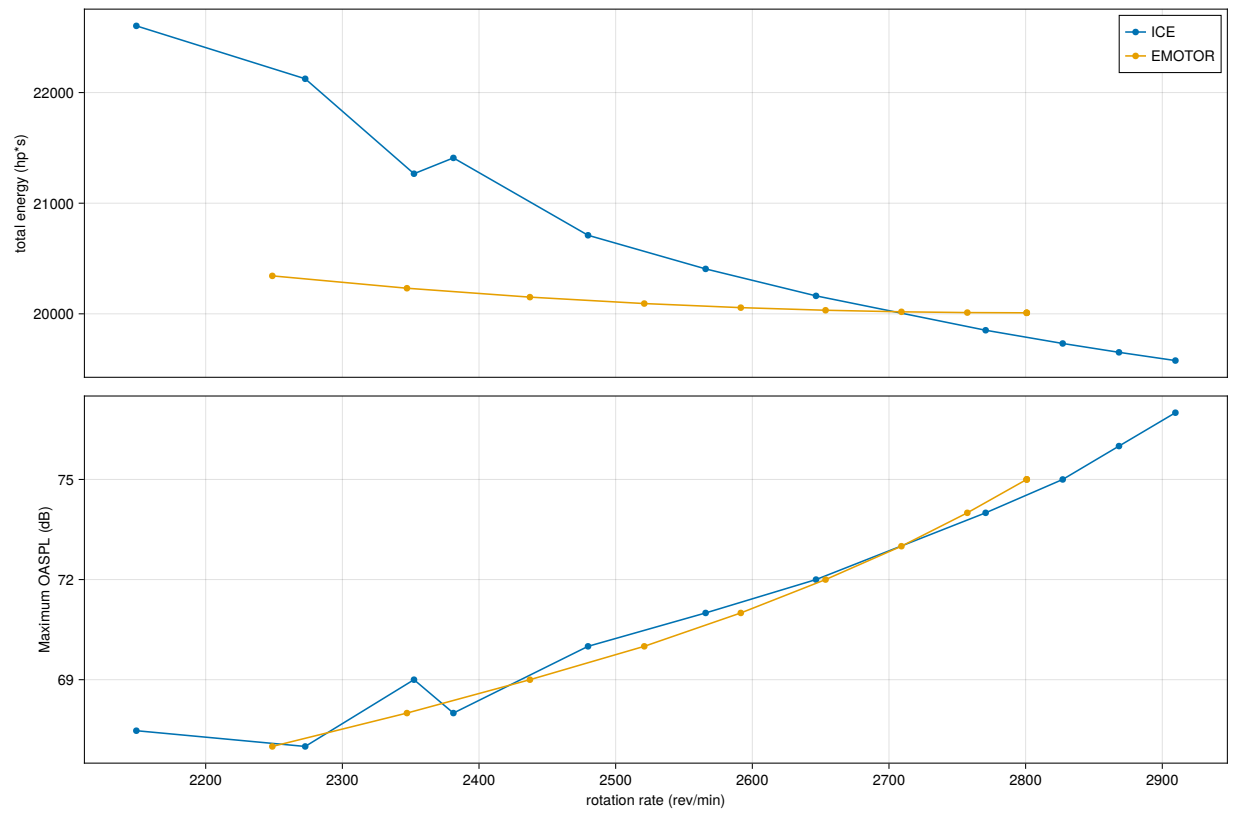


Fig. 9 Case II total energy and OASPL vs RPM

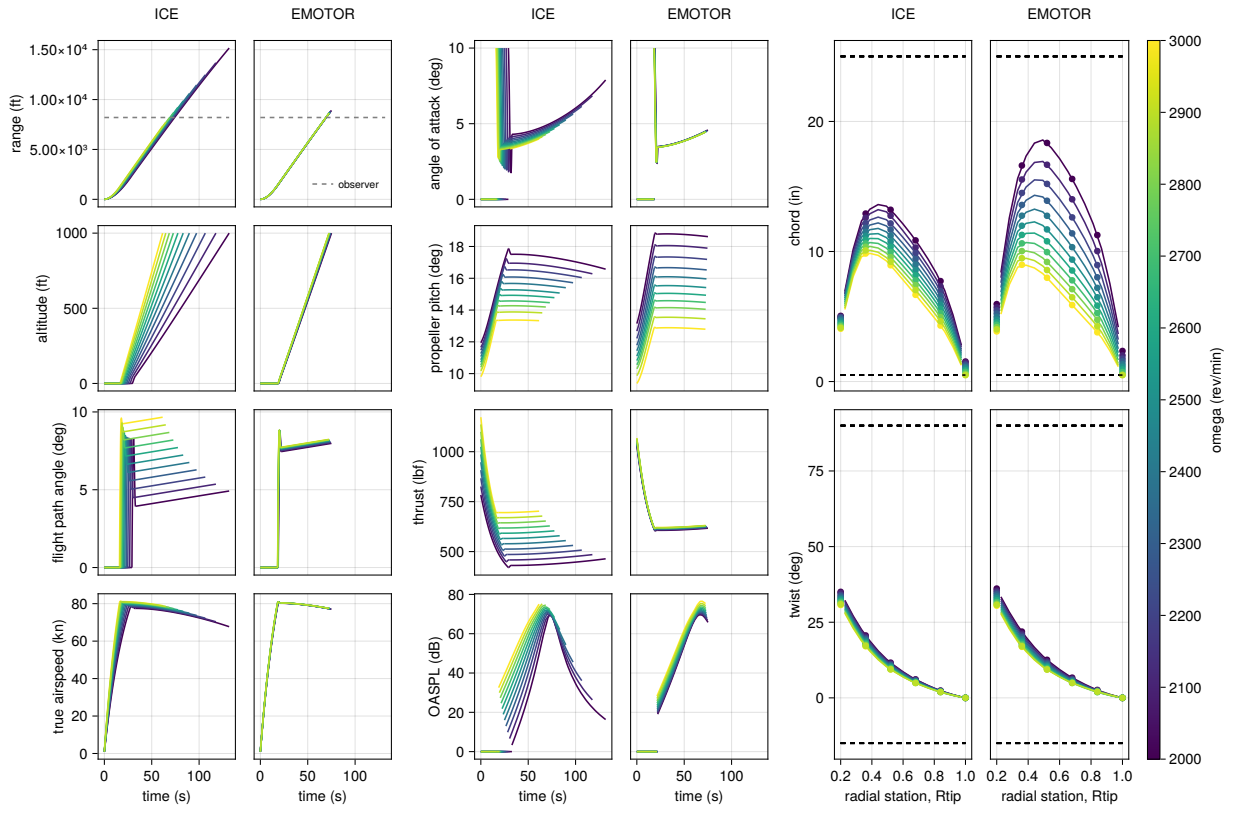


Fig. 10 Case III time histories

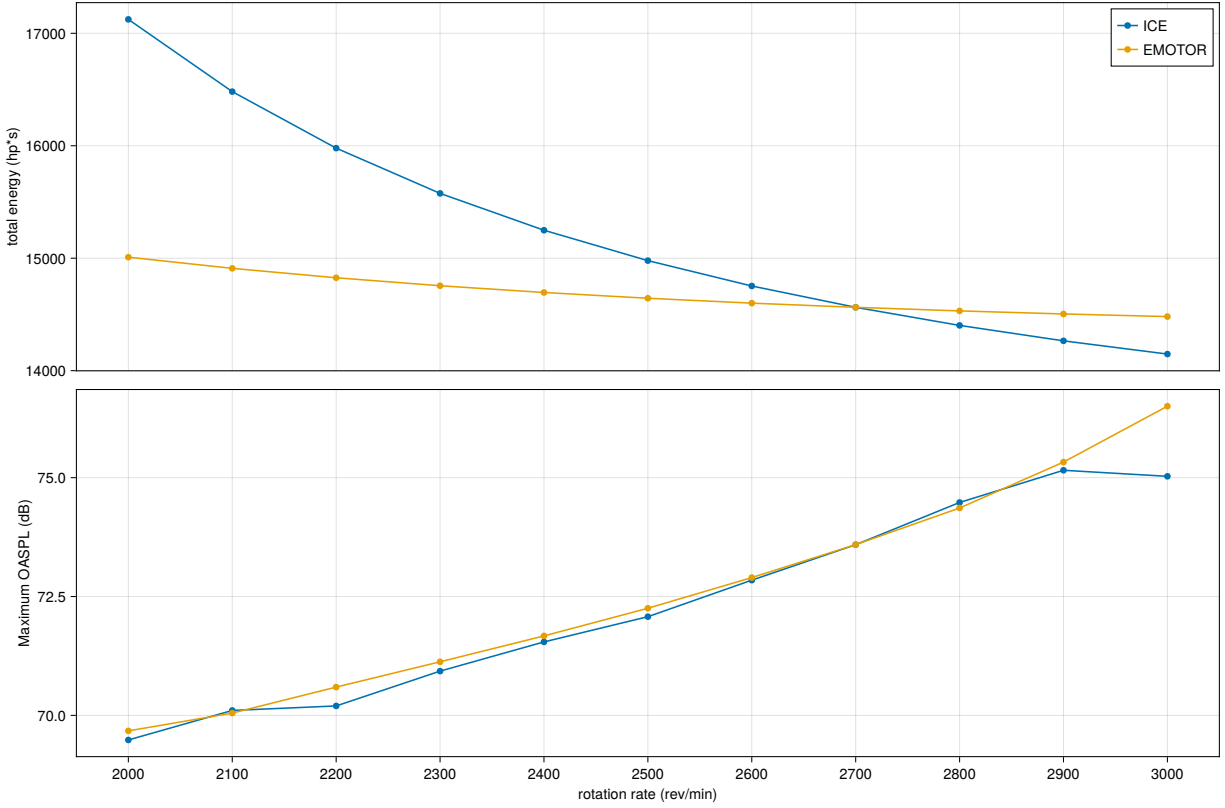


Fig. 11 Case III total energy and OASPL vs RPM

history figures 6 and 8, but now include the chord and twist distributions for the optimizations. We find broadly that the optimizer decided to take the same approach as in the fixed blade design case of figure 6. For both the ICE and EMOTOR cases, the optimizer decided to increase the chord as the RPM was decreased to better absorb the lower shaft power.

Figure 11 show the total energy and maximum OASPL as a function of RPM for Case III. Broadly the trends are similar to the equivalent case with the fixed blade design shown in figure 7, but comparison with figure 11 shows that allowing the optimizer to design the blade's chord and twist significantly decreases the energy consumption and achieves a noise reduction as well.

D. Case IV: Propeller chord and twist design variables, RPM design variable, with acoustic constraint

Figures 12 and 13 show the time histories and total energy/OASPL vs RPM plots for Case IV, the “full” optimization problem, where the rpm, chord, and twist are included as design variables, and the OASPL constraint is swept. Again, this final set of optimizations follows the trend of the previous, though converging the especially the ICE cases was difficult (all optimizations did meet the feasibility criteria and did exhibit a significant increase in optimality, however). Like the fixed blade design case with the RPM design variable and OASPL constraint in Figure 8, a few of the lower-RPM ICE optimizations opted to trade increased angle of attack and decreased airspeed for lower flight path angle, perhaps achieving a mild noise benefit in the process. Most interestingly, the optimizer decided to decrease the chord with decreasing RPM for the ICE case, along with increasing the twist. Overall the quality of these optimizations could likely be improved with more attention paid to the scaling of the design variables and constraints (though significant effort has already been invested at the time of writing).

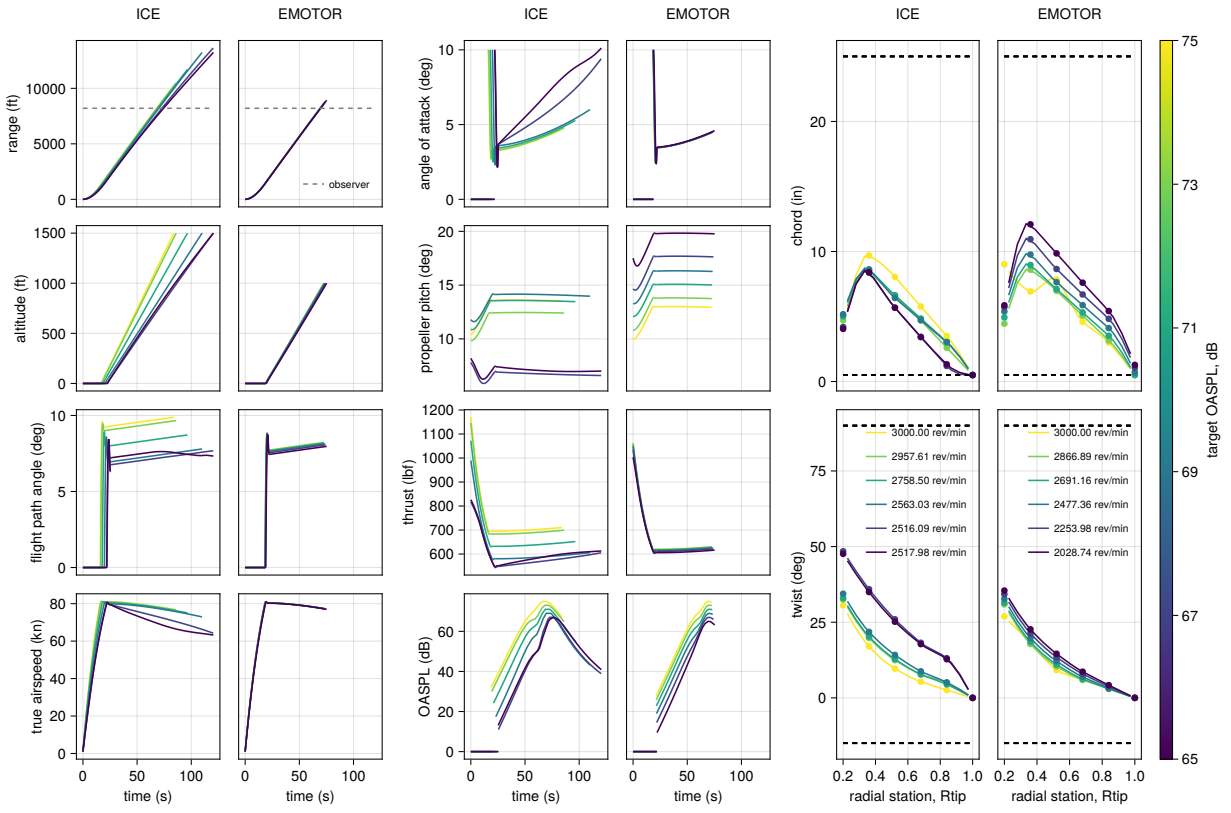


Fig. 12 Case IV time histories

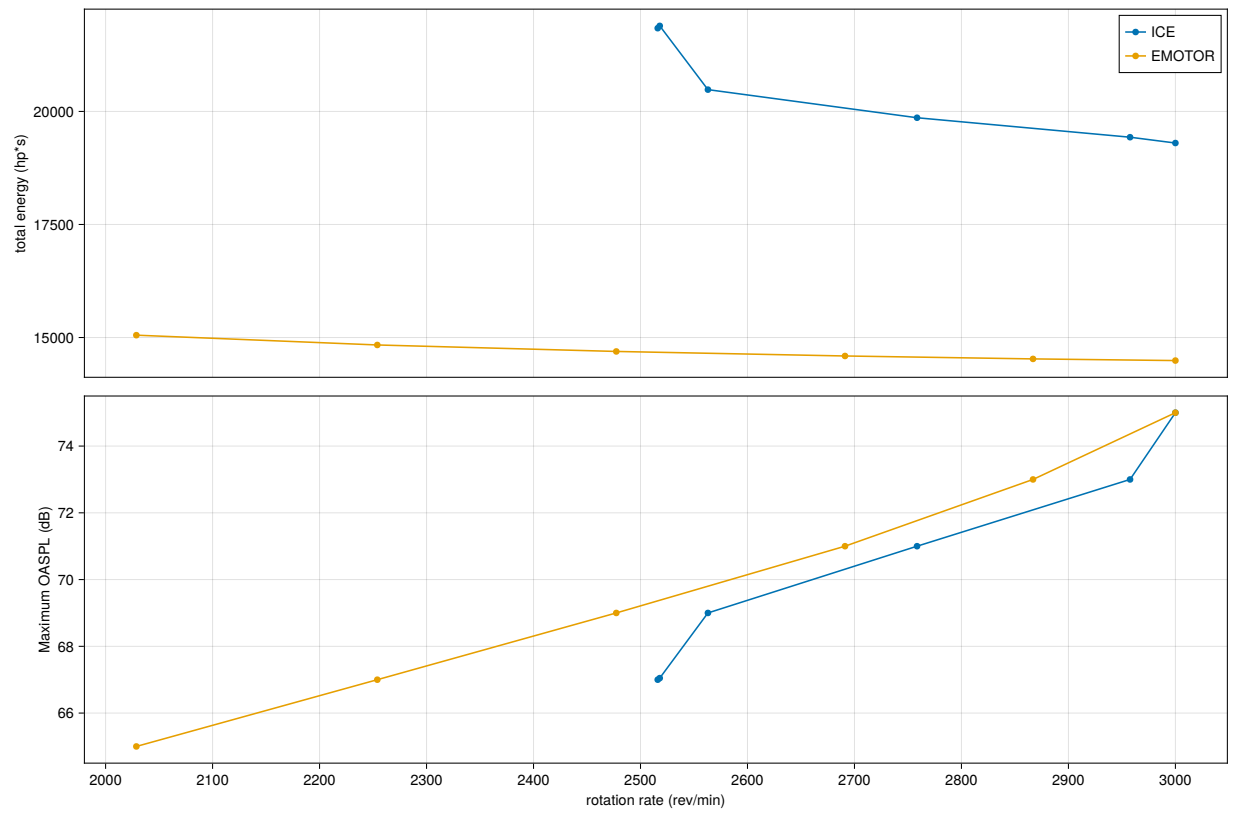


Fig. 13 Case IV total energy and OASPL vs RPM

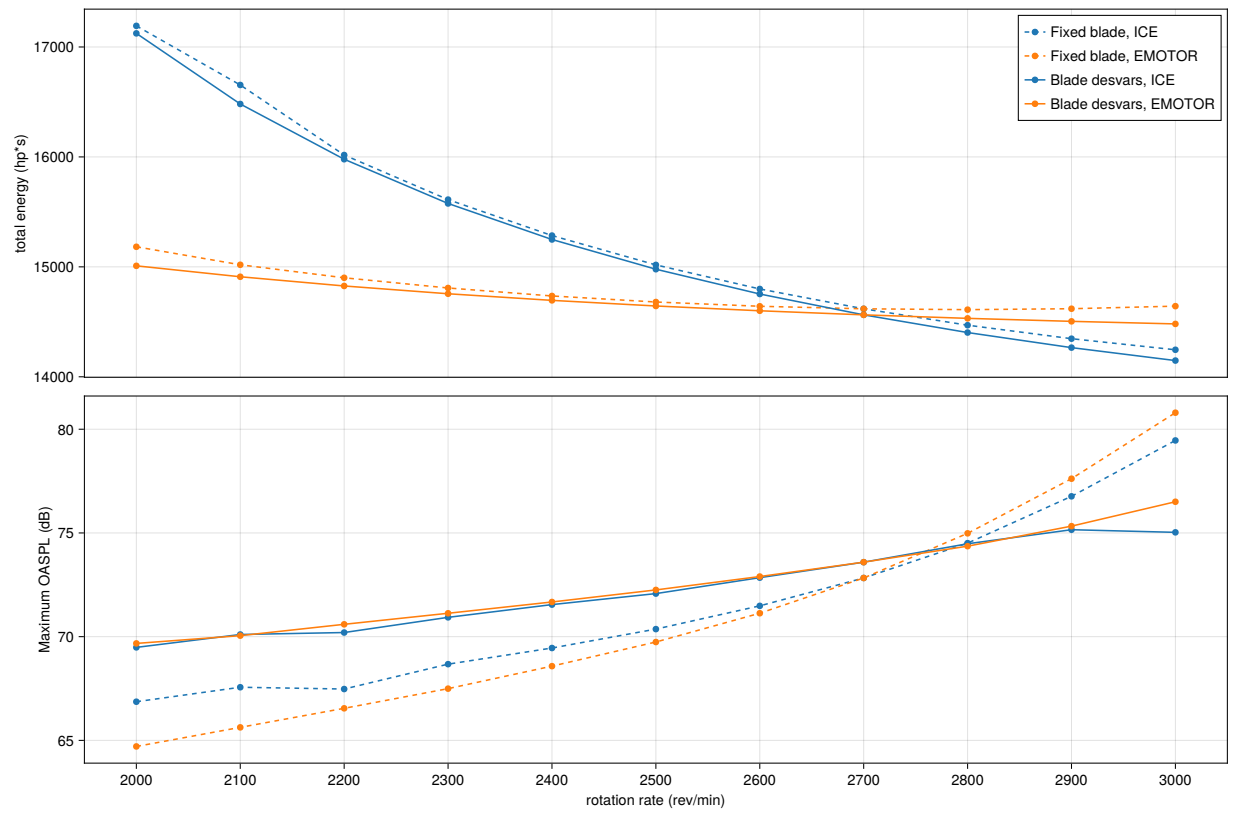


Fig. 14 Case I vs Case III total energy and OASPL vs RPM

E. Fixed-blade vs blade design variable cases

Finally, figure 14 compares the total energy objective^{†††} and maximum OASPL as a function of RPM for both the internal combustion engine and electric motor and both the fixed blade shape and variable blade design cases. The figure shows that, when the optimizer is able to modify the blade shape, decreases in the total energy expended during the takeoff trajectory can be achieved, at the cost of increasing the OASPL noise metric for all but the highest RPM settings. The increased thrust required to shorten the trajectory duration and thus decrease the expended total energy is at odds with noise reduction, of course, though increasing the climb rate, putting the vehicle further away from the ground observer, should be beneficial from a noise reduction perspective. It would be interesting to add a noise constraint to these fixed-RPM optimizations to see if the optimizer is capable of reducing the noise for the cases shown in figure 14, but that was not attempted in this work.

IV. Conclusions

This work presented the results of combining low fidelity two-dimensional vehicle dynamics, propeller aerodynamic and tonal acoustic models, and an optimal control library to simultaneously design a takeoff trajectory and propeller blade of a general aviation aircraft subject to acoustic constraints. The results show that significant decreases in both noise and energy consumption can be achieved through this gradient-based design optimization process, and (as shown in figure 14) allowing the optimizer to modify the blade shape can further reduce the energy consumption of the vehicle. Future work will focus on improving the quality of the “full” optimization case, especially with regards to scaling, and including a broadband noise model to increase the realism of the acoustic constraint.

Acknowledgements

This work is supported by NASA’s Transformational Tools and Technologies (TTT) project, part of the Transformative Aeronautics Concepts Program (TACP). The authors are also grateful for the computational resources provided by NASA’s Commercial Supersonic Technology project, and thank Jeff Berton (NASA Glenn Research Center, retired) for considerable guidance on modeling the Cirrus SR20.

References

- [1] Rizzi, S. A., Huff, D. L., Boyd, D. D., Jr., Bent, P., Henderson, B. S., Pascioni, K. A., Sargent, D. C., Josephson, D. L., Marson, M., He, H. B., and Snider, R., “Urban Air Mobility Noise: Current Practice, Gaps, and Recommendations,” Tech. Rep. NASA/TP-2020-5007433, NASA Langley Research Center, Hampton, VA, United States, October 2020.
- [2] Hendricks, E. S., Falck, R. D., Gray, J. S., Aretskin-Hariton, E., Ingraham, D., Chapman, J. W., Schnulo, S. L., Chin, J., Jasa, J. P., and Bergeson, J. D., “Multidisciplinary Optimization of a Turboelectric Tiltwing Urban Air Mobility Aircraft,” *AIAA Aviation 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. <https://doi.org/10.2514/6.2019-3551>, URL <http://dx.doi.org/10.2514/6.2019-3551>.
- [3] Hendricks, E. S., Aretskin-Hariton, E., Ingraham, D., Gray, J. S., Schnulo, S. L., Chin, J., Falck, R., and Hall, D., “Multidisciplinary Optimization of an Electric Quadrotor Urban Air Mobility Aircraft,” *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, 2020. <https://doi.org/10.2514/6.2020-3176>, URL <http://dx.doi.org/10.2514/6.2020-3176>.
- [4] Voet, L. J. A., Prashanth, P., Speth, R. L., Sabnis, J. S., Tan, C. S., and Barrett, S. R. H., “Automatic Continuous Thrust Control for Supersonic Transport Takeoff Noise Reduction,” *Journal of Aircraft*, Vol. 61, No. 1, 2024, p. 291–306. <https://doi.org/10.2514/1.c037394>, URL <http://dx.doi.org/10.2514/1.C037394>.
- [5] Ingraham, D., Falck, R., and Gray, J. S., “Blade and Trajectory Optimization of a Propeller -Driven Electric Aircraft with Acoustic Constraints,” *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics, 2020. <https://doi.org/10.2514/6.2020-3141>, URL <http://dx.doi.org/10.2514/6.2020-3141>.
- [6] Berton, J. J., and Nark, D. M., “Low-Noise Operating Mode for Propeller-Driven Electric Airplanes,” *24th AIAA/CEAS Aeroacoustics Conference*, American Institute of Aeronautics and Astronautics, 2018.

^{†††}Case I and Case III used different values (1500 feet and 1000 feet, respectively) for the final altitude of the climb phase, which would significantly affect the total energy calculation. To account for this, a spline for each of Case I’s climb altitude time histories was constructed, and the time at which each trajectory reached 1000 feet was found numerically. This time was then used with Eq. 6 to obtain the total energy values for Case I in figure 14.

- [7] Bryson, A. E., Jr., Desai, M. N., and Hoffman, W. C., “Energy-state approximation in performance optimization of supersonic aircraft,” *Journal of Aircraft*, Vol. 6, No. 6, 1969, p. 481–488. <https://doi.org/10.2514/3.44093>.
- [8] Extension to the Standard Atmosphere (COESA), T. U. S. C., “U.S. Standard Atmosphere, 1976,” Tech. Rep. NASA-TM-X-74335, National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, Washington, DC, United States, October 1976.
- [9] Roskam, J., *Airplane Design, Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics*, 3rd ed., Design, Analysis and Research Corporation, 2000.
- [10] Andrew Ning, S., “A simple solution method for the blade element momentum equations with guaranteed convergence,” *Wind Energy*, 2013, p. n/a–n/a. <https://doi.org/10.1002/we.1636>.
- [11] Ning, A., “Using blade element momentum methods with gradient-based design optimization,” *Structural and Multidisciplinary Optimization*, 2021. <https://doi.org/10.1007/s00158-021-02883-6>, URL <http://dx.doi.org/10.1007/s00158-021-02883-6>.
- [12] Revels, J., Lubin, M., and Papamarkou, T., “Forward-Mode Automatic Differentiation in Julia,” *arXiv:1607.07892 [cs.MS]*, 2016. URL <https://arxiv.org/abs/1607.07892>.
- [13] Dalle, G., and Hill, A., “DifferentiationInterface.jl,” , 2024. <https://doi.org/10.5281/zenodo.11092033>, URL <https://github.com/JuliaDiff/DifferentiationInterface.jl>.
- [14] Schäfer, F., Tarek, M., White, L., and Rackauckas, C., “AbstractDifferentiation.jl: Backend-Agnostic Differentiable Programming in Julia,” *NeurIPS 2021 Differentiable Programming Workshop*, 2021. URL <https://github.com/JuliaDiff/AbstractDifferentiation.jl>.
- [15] Ning, A., and McDonnell, T., “Automating Steady and Unsteady Adjoint: Efficiently Utilizing Implicit and Algorithmic Differentiation,” , 2023. URL <https://arxiv.org/abs/2306.15243>.
- [16] Farassat, F., “Linear Acoustic Formulas for Calculation of Rotating Blade Noise,” *AIAA Journal*, Vol. 19, No. 9, 1981, pp. 1122–1130. <https://doi.org/10.2514/3.60051>.
- [17] Farassat, F., “Derivation of Formulations 1 and 1A of Farassat,” Tech. Rep. NASA/TM 2007-214853, NASA Langley Research Center, Hampton, VA, United States, Mar 2007.
- [18] Ffowcs Williams, J. E., and Hawkings, D. L., “Sound Generation by Turbulence and Surfaces in Arbitrary Motion,” *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 264, No. 1151, 1969, pp. 321–342.
- [19] Lopes, L. V., “Compact Assumption Applied to the Monopole Term of Farassat’s Formulation,” *21st AIAA/CEAS Aeroacoustics Conference*, American Institute of Aeronautics and Astronautics, 2015. <https://doi.org/10.2514/6.2015-2673>.
- [20] Lopes, L. V., “Compact Assumption Applied to Monopole Term of Farassat’s Formulations,” *Journal of Aircraft*, Vol. 54, No. 5, 2017, p. 1649–1663. <https://doi.org/10.2514/1.c034048>.
- [21] Falck, R. D., and Gray, J. S., “Optimal Control within the Context of Multidisciplinary Design, Analysis, and Optimization,” *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. <https://doi.org/10.2514/6.2019-0976>.
- [22] Hwang, J. T., “A modular approach to large-scale design optimization of aerospace systems,” Ph.D. thesis, University of Michigan, 2015.
- [23] Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A., “pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems,” *Journal of Open Source Software*, Vol. 5, No. 54, 2020, p. 2564. <https://doi.org/10.21105/joss.02564>.
- [24] Gill, P., Murray, W., and Saunders, M., “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. <https://doi.org/10.1137/S0036144504446096>.