

Optimal Control Strategies for Trajectory Optimization with Applications to Continuous Solar Flight

Ashwin Balakrishna

abalakri@caltech.edu

November 2013

A version of this paper was submitted to the 2014 Intel Science Talent Search Competition

The goal of this research is to develop a high performance system for trajectory optimization of continuously flying solar aircraft. Here, during the day solar power is used to propel the aircraft and to store energy in batteries. The battery energy is then used for continued flight at night. The objective is to find the minimum battery size needed for continuous operation by optimizing several control variables. Traditional methods, where the differential equations that describe flight motion are solved in an inner loop while an outer loop performs the optimization of the control variables, are inefficient for this problem because the boundary conditions are not only unknown, but are required to be identical due to cyclic operation. Furthermore, state variable constraints are difficult to handle with the sequential approach. To address this issue, a computationally efficient trajectory optimization method using orthogonal collocation on finite elements is first developed. After testing on a glider range optimization problem, this method was applied to the solar aircraft trajectory optimization problem. The system developed is robust, computationally efficient, and can be used to optimize and control multi-purpose solar aerial vehicles. Plans are underway to design and build a solar UAV using this optimization system.

Introduction

This paper addresses two important problems. First, a general method to solve flight trajectory optimization problems is developed. Second, this method is applied to create a comprehensive optimization system to achieve *continuous* solar flight in an energy efficient manner. Continuous flight is achieved by cyclic operation, where the trajectory is repeated indefinitely, typically every 24 hours. The word *continuous* is used in the theoretical sense, as

continuous or perpetual flight is not achievable in practice due to degradation of batteries and aircraft components over time.

The importance of flight trajectory optimization has been recognized in both general aviation and space applications; a review of applications and methods is presented in [1]. The development of better methods for solving these optimization models can have substantial impact on the aerospace industry. The prevalent class of algorithms for solving these problems are largely sequential in nature, where the differential equations that describe flight motion are solved in an inner loop while an outer loop performs the optimization of the control variables. These methods can be computationally expensive as they require repeated solution of the differential equations for each guess of the control variable in addition to calculation of gradients for the optimizer [2]. The algorithm may also terminate if the differential equation solver fails at intermediate guesses of the control variables given by the optimizer. For the optimization of solar aircraft, these sequential methods face unique challenges because the boundary conditions are not only unknown, but are required to be identical due to cyclic operation. Furthermore, constraints on the state variables cannot be enforced. Solving the differential equations and performing the optimization simultaneously can address these drawbacks [3, 4], but this requires good initialization of the state variables. In this research, I build upon a simultaneous solution method called orthogonal collocation on finite elements [4] to develop a robust trajectory optimization system with an effective initialization strategy.

The simultaneous method mentioned above can be applied to many flight trajectory optimization problems. The application to solar flight is uniquely interesting from an optimal control perspective since the available power is time dependent. The recent completion of the cross continental flight of the Solar Impulse [5] has sparked renewed interest in solar aviation. The design of solar vehicles with batteries that can offer continuous flight has also seen a lot of interest [6, 7, 8]. Here, during the day solar power is used to propel the aircraft and to store energy in batteries. The battery energy is then used for continued flight at night. Such aircraft offer unique opportunities, especially for unmanned flight, and can be used for communications, imagery, surveillance, and assistance during natural disasters. While the design of such aircraft has seen a lot of interest, there has been little reported in terms of the trajectory optimization for these aircraft. One notable attempt [9] has been made in this regard; however, the battery mass does not appear to be considered in their work, results are given for only one scenario, and computational performance results are not reported.

I address this gap by developing a reliable and computationally efficient system that can determine the optimal trajectory for continuous solar flight. The primary goal is to minimize the mass of the battery needed while allowing sufficient energy to make it through the night. To do this, the development of an efficient algorithm is critical, which is the first step of the research. In the second step, a detailed mathematical model for the solar aircraft is built and the solution algorithm developed in the first step is applied. This system enables a user to find the optimal trajectory for continuous flight given input parameters such as payload, battery efficiency, altitude limits, latitude, day of year, and other aircraft specifications. In addition, limiting cases have also been studied to precisely establish the latitude range within which continuous solar flight is theoretically possible.

This paper is composed of two sections. The first section will discuss the solution methodology and the test results on a glider range maximization problem proposed in [10], which is used as a benchmark for evaluating the method. The second section will describe the formulation and solution of the solar aircraft trajectory optimization problem.

Section I – Development of Solution Algorithm for the Optimal Control Problem

The general optimal control problem (OCP) is presented as follows, where Z_{obj} is the

$$\text{Maximize } Z_{obj}(z(t), y(t), u(t), p) \text{ when } \left\{ \begin{array}{l} F\left(\frac{dz}{dt}, z(t), y(t), u(t), p\right) = 0 \\ G(z(t), y(t), u(t), p) = 0 \\ z_L \leq z(t) \leq z_U \\ y_L \leq y(t) \leq y_U \\ u_L \leq u(t) \leq u_U \\ p_L \leq p \leq p_U \\ z(0) = z_0 \end{array} \right. \quad (OCP)$$

objective function, F is the set of differential equations, G is the set of algebraic equations, $z(t)$ is the set of state variables, $y(t)$ is the set of algebraic variables, $u(t)$ is the set of control profiles to be optimized, and p is the set of time-invariant parameters that are optimized in order to maximize Z_{obj} . Subscripts L and U refer to the lower and upper bounds respectively.

The first step in the solution algorithm is to assume that $z(t)$ can be expressed as a polynomial of order N within the integration domain. The Lagrange interpolating polynomials are a convenient form to express $z(t)$ in terms of its values at time point t_j , $j \in [0, N]$, as shown below:

$$z(t) = \sum_{j=0}^N z(t_j) L_j(t) \quad (1) \quad \text{where} \quad L_j(t) = \prod_{\substack{k=0 \\ k \neq j}}^N \frac{t - t_k}{t_j - t_k} \quad (2)$$

$$\text{Differentiating } z(t), \text{ we get} \quad \frac{dz}{dt}(t) = \sum_{j=0}^N z(t_j) \dot{L}_j(t) \quad (3)$$

$$\text{where} \quad \dot{L}_j(t) = \frac{\sum_{\substack{k=0 \\ k \neq j}}^N \left(\prod_{\substack{m=0 \\ m \neq j \\ m \neq k}}^N (t - t_m) \right)}{\prod_{\substack{k=0 \\ k \neq j}}^N (t_j - t_k)} \quad (4)$$

By substituting the value for dz/dt from equation 3 into the optimal control problem (OCP), we can convert the OCP into a set of algebraic equations which are satisfied at special time points t_j , called collocation points. These points are set to be located at the roots of N^{th} order orthogonal polynomials within the integration interval. By using this collocation procedure, exact accuracy in the solution of the differential equation is guaranteed as long as the state variable is a polynomial of order $2N-1$ or less [11]. Thus, the OCP is transformed into a non-linear programming (NLP) model, where $u(t_j)$ is optimized to determine $z(t_j)$ and $y(t_j)$. In order to ensure integration accuracy, the integration domain is divided into several finite elements, which allows for the use of a lower order polynomial within each finite element. Continuity of the state variables between the finite elements is ensured. This method, also known as orthogonal collocation on finite elements, has been used in chemical engineering applications [2, 4]. I implemented the discretization procedure described above to convert the OCP to its equivalent NLP in the General Algebraic Modeling System (GAMS) modeling language [12]. To account for situations in which the end of the integration horizon (some time T_f) is unknown, the integration horizon was normalized to a [0, 1] domain. To do this, a variable $\omega = \frac{t}{T_f}$ was created, where t is the time in seconds counted from the beginning of the integration interval. In this way, the start and end of each interval is clearly defined. This also enables use of orthogonal polynomial roots on the [0, 1] domain as collocation points regardless of the size of the integration interval. However, the chain rule must be used to convert $\frac{dz(t)}{dt}$ to $\frac{1}{T_f} \frac{dz}{d\omega}$. In the implementation, the differential

equations are written with respect to the scaled ω variable, and thus the total integration horizon can still be optimized. The generated NLP is then solved using CONOPT to maximize the objective function. CONOPT is an efficient implementation of the generalized reduced gradient method and has better relative performance for models that are very nonlinear and in which achieving feasibility is difficult [13]. The implementation allows for flexibility in many algorithmic parameters such as the initialization strategy, polynomial order for approximation, choice of orthogonal polynomial roots, and number of finite elements. The algorithm was first tested on a well-known problem in order to evaluate the best parameters for the solution method.

Algorithm Performance Analysis and Improvement

Rigorous testing of the method was done on the well documented glider range maximization problem [10], which was also used to tune the algorithm for general use. In this problem, the lift coefficient is optimized to maximize the horizontal range of a glider that has a known thermal updraft ahead of it. The glider launches from a 1000 m cliff with a known initial speed. The aim is to cover maximum horizontal distance while dropping no more than 100 m vertically. The mathematical model for this problem is shown in the equations below [10]:

Hang Glider Model (HG-1)

Variables

x	horizontal position (m)	$\mu_a(x)$	thermal updraft (m/s)
y	vertical position (m)	c_L	lift coefficient
v_x	x component of velocity (m/s)	c_D	drag coefficient
v_y	y component velocity (m/s)	L	lift (N)
v_r	net velocity (m/s)	D	drag (N)
η	flight angle (rad)	t, t_f	time (s), total flight time (s)

Parameters

$x(0)$	initial x position	0 m
$y(0)$	initial y position	1000 m
$v_x(0)$	x component of initial velocity	$13.23 \frac{m}{s}$
$v_y(0)$	y component of initial velocity	$-1.288 \frac{m}{s}$

$y(t_f)$	final position in y direction	900 m
S	wing area	14 m^2
m	mass of hang glider	100 kg
ρ	air density	$1.13 \frac{\text{kg}}{\text{m}^3}$
g	gravitational acceleration	$9.81 \frac{\text{m}}{\text{s}^2}$

Objective: Maximize $x(t_f)$ by finding the optimal lift coefficient profile $c_L(t)$ subject to:

Equations

$$\frac{dx}{dt} = v_x \quad (5)$$

$$\frac{dy}{dt} = v_y \quad (6)$$

$$\frac{dv_x}{dt} = \frac{1}{m} (-L \sin(\eta) - D \cos(\eta)) \quad (7)$$

$$\frac{dv_y}{dt} = \frac{1}{m} (L \cos(\eta) - D \sin(\eta) - mg) \quad (8)$$

$$\eta = \arctan\left(\frac{v_y - \mu_a(x)}{v_x}\right) \quad (9)$$

$$v_r = \sqrt{v_x^2 + (v_y - \mu_a(x))^2} \quad (10)$$

$$L = \frac{1}{2} c_L \rho v_r^2 S \quad (11)$$

$$D = \frac{1}{2} c_D \rho v_r^2 S \quad (12)$$

$$c_D = 0.034 + 0.069662(c_L)^2 \quad (13)$$

$$\mu_a(x) = 2.5 \left(1 - \left(\frac{x}{100} - 2.5\right)^2\right) e^{-\left(\frac{x}{100} - 2.5\right)^2} \quad (14)$$

$$c_L \leq 1.4 \quad (15)$$

$$v_x(0) = v_x(t_f), \quad v_y(0) = v_y(t_f) \quad (16)^1$$

¹ This constraint is eliminated in model HG-2 described later in the paper

The thermal updraft profile, $\mu_a(x)$ is shown in Fig. 1, which also shows small downdraft regions at the boundaries of the thermal. The peak updraft is 250 m from the launch point. This problem was solved with a second order polynomial approximation over 500 equally spaced finite elements using Radau [14] collocation points. The key findings are as follows:

- a) The method I implemented solved the HG-1 model easily in just 9.74 seconds (240 solver iterations) on a Lenovo T430 machine with an Intel i5 2.5 GHz processor and 4.00 GB of memory running Windows 7 Professional². A maximum range of 1247.99 m was obtained, which compares favorably with the range of 1247.60 m obtained in [10] and 1248.03 m obtained in [3]. The optimal profile of the lift coefficient shown in Fig. 2 matches the profile reported in [10] and [3], confirming the accuracy of the discretization. In Fig. 2, the lift coefficient rises to take advantage of the thermal and then returns to a lower value after the thermal passes, which enables reduction of lift induced drag. The position charts are shown in Fig. 3 and 4. The horizontal position of the glider changes roughly linearly while the vertical position indicates a dip in altitude before the thermal and then an increase in the altitude due to the thermal winds. Finally, the altitude decreases roughly linearly until the end of the flight. I did not encounter any of the initialization difficulties reported in [10] and my method did not require the complex derivation of adjoint equations described in [10]. The velocity profile was initialized with just the initial launch values, the y position profile with the initial vertical position, and the x position profile with a value of 10 m. The lift coefficient was initialized at 0.7.

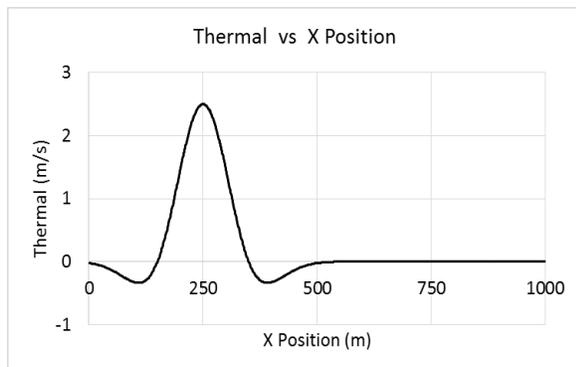


Fig. 1 Thermal Updraft vs. Horizontal position (x)

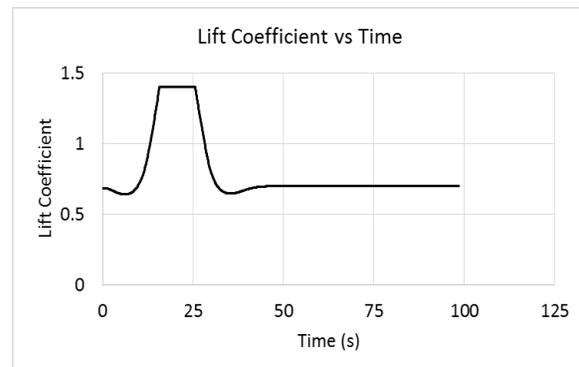


Fig. 2 Optimal Lift Profile for HG-1

² All computation times in this paper are reported using this computer

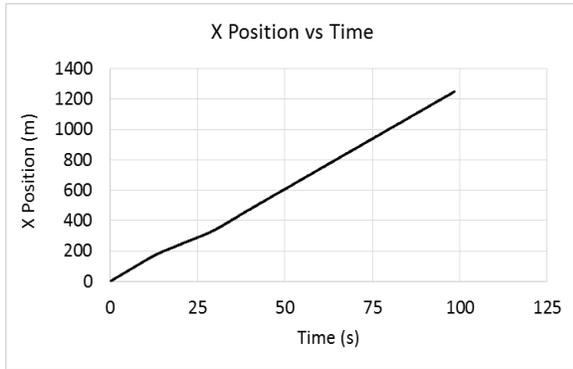


Fig. 3 Horizontal Position vs. Time for HG-1

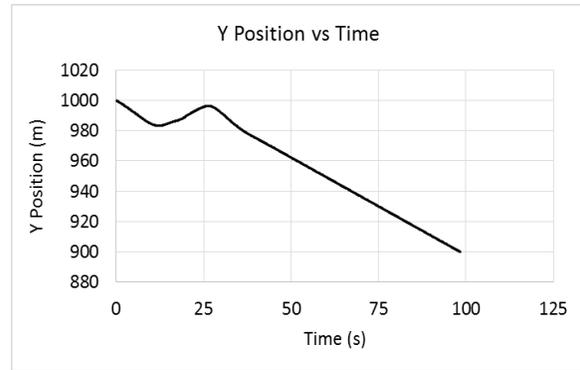


Fig. 4 Vertical Position vs. Time for HG-1

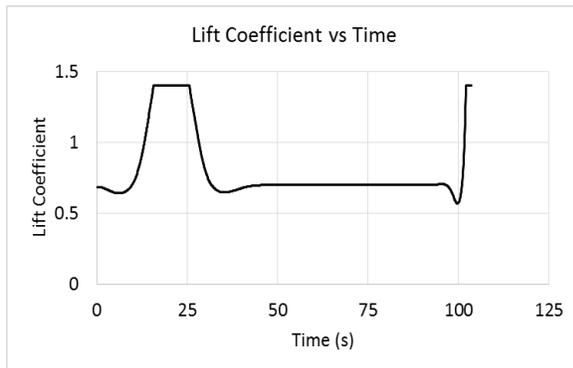


Fig. 5 Optimal Lift Profile HG-2

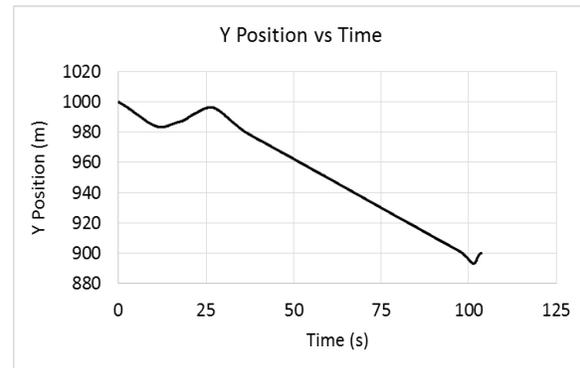


Fig. 6 Vertical position vs. time for HG-2

- b) Having solved the HG-1 model easily, more studies were conducted. To explore new opportunities to maximize the range, a model HG-2 was created by eliminating equation 16, thus allowing the final velocity to be different from the initial launch velocity. Interestingly, the HG-2 model yielded a much higher range (1311.17 m) compared to the HG-1 model value of 1247.99 m. I believe this is because of the sharp increase in the lift coefficient (Fig. 5) near the end of the flight, which I dub as a ‘mad-dash’ to get a final burst of horizontal distance while still ending the flight above the required 900 m altitude mark (Fig. 6). Since this rise in the lift coefficient is short lived, the increased drag has a limited negative impact on the range. Although this ‘mad-dash’ in the lift coefficient would likely affect the stability of the glider, it shows the efficacy of the method in solving models with sharp changes in the control profile. A graph of the vertical position of the glider is shown in Fig. 6, which shows the effects of the final surge in the lift coefficient on the altitude.
- c) The HG-2 model described above required 1714 solver iterations (99.7 sec) compared to the 240 solver iterations (9.74 sec) for the simpler HG-1 model. I investigated this issue further and found an initialization strategy that improved computational performance, which I call the 2-solve method. This idea involves fixing

control variables at nominal values³, generating the profiles for the state variables, and then starting the optimization. By using this 2-solve method, the HG-2 model was solved with a combined 139 solver iterations (10.5 sec). Further studies with the HG-2 model confirmed that the solution is consistently more reliable when the 2-solve method is used.

- d) Regarding polynomial order, second order polynomial approximation within each finite element resulted in the best computational performance without sacrificing solution quality. For the HG-1 model, the difference in the range obtained by moving from second order to third order polynomial approximation was just 0.000024%, with a 392% increase in computation time. In addition, I tried Legendre and Radau roots for the collocation [14], and found that because Radau collocation provides roots at the end of the integration interval, no interpolation is needed to get the values at the end of the integration interval and thus it is better adapted to enforce state variable continuity across finite elements.

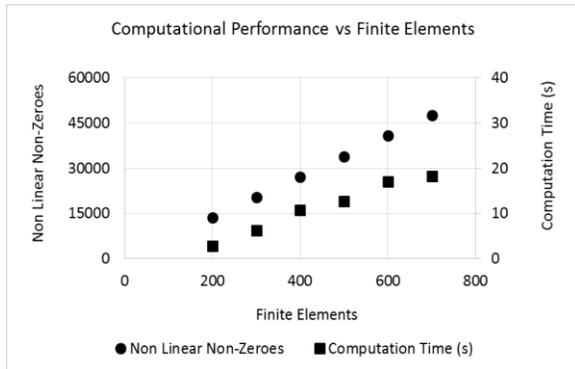


Fig. 7 Computation Time vs. Finite Elements

As the method requires sufficient finite elements for accuracy, I experimented and found that the solution stays practically identical when more than 200 finite elements are used, confirming the precision of the discretization. All the tests above were run using 500 finite elements and the objective function for the HG-1 model experienced negligible change (+0.000008%) when adding 100 more finite elements. In Fig. 7, the computational performance is shown for different finite element trials. As the number of finite elements is increased, the degrees of freedom increase linearly along with the number of non-linear non-zeroes (Fig. 7) in the equation matrix. However, the increase in computation time is relatively moderate when the number of finite elements is increased.

- f) I initially believed that trigonometric functions would inhibit the performance of the solver due to their oscillatory nature. To investigate this, trigonometric functions were replaced with their equivalent ratios (for example $\sin(\theta) = v_y / \sqrt{v_x^2 + v_y^2}$). However, the resulting model was harder to converge, and the HG-2 model

³ The computation time did not vary substantially for different settings of the nominal values

was numerically infeasible when the trigonometric functions were replaced. Therefore, I decided to keep the original formulation of the model.

The above findings were used to set the tuning parameters for the solution algorithm. In summary, the default settings include second order polynomial approximation with Radau collocation points and the 2-solve method for effective initialization. The trigonometric functions in the force balance equations are to be kept as is. The simultaneous method I implemented addresses the drawbacks in the sequential method. Repeated solution of the differential equations for each iteration of the control variable guesses is eliminated because the solution of the differential equations and the optimization are performed simultaneously. Tests on the glider model show that the method has attractive computational performance.

Section II – Development of Solar Aircraft Trajectory Optimization Model

Having established the numerical methods, I sought to apply it to the trajectory optimization of continuous solar flight. Here, solar power is used for propulsion as well as charging the batteries during the day. This battery energy is then used for continued flight during the night. Perpetuity is enforced by setting the boundary conditions to be the same within the repetitive solar cycle (24 h). Therefore, the state variables are required to have the same values at the beginning and end of the cycle. The maximum energy stored in the battery is minimized, thus effectively minimizing the battery mass. The mathematical model is synthesized by combining the equations for atmospheric effects, flight dynamics, solar flux, battery operation, and panel performance. In this model, the hardware parameters (panel efficiency, panel mass density, battery energy density) are based on currently available technology. The source of this information is referenced in the parameter list. The model for each of the phenomena described above is shown below.

Atmospheric Effects

Temperature, Pressure, and Air Density as functions of height [15].

Variables

T	temperature (K)	h	height from sea level (m)
p	air pressure (Pa)	ρ	air density $\left(\frac{kg}{m^3}\right)$

Parameters

T_0	sea level standard temperature	288.15 K
-------	--------------------------------	----------

L_T	temperature lapse rate	$0.0065 \frac{K}{m}$
p_0	sea level atmospheric pressure	101325 Pa
g	surface gravitational acceleration	$9.81 \frac{m}{s^2}$
M	molar mass of dry air	$0.0289644 \frac{kg}{mol}$
R	ideal universal gas constant	$8.31447 \frac{J}{mol K}$

Equations

$$T = T_0 - L_T h \quad (17)$$

$$p = p_0 \left(1 - \frac{L_T h}{T_0}\right)^{\frac{gM}{RL_T}} \quad (18)$$

$$\rho = \frac{pM}{RT} \quad (19)$$

Flight Dynamics

The flight dynamics equations are from [16]. Equations for the drag coefficient are regressed from values provided in [6]. Solar panel mass is calculated based on a mass density of $0.840 \frac{kg}{m^2}$ as provided in [17].

Variables

v	velocity ($\frac{m}{s}$)	$m_{payload}$	mass of payload (kg)
θ	angle to horizontal (rad)	m_{panel}	mass of solar panel (kg)
D	drag (N)	$m_{battery}$	mass of battery (kg)
T_p	thrust (N)	E_{batmax}	maximum energy stored in battery (kJ)
L	lift (N)	c_L	lift coefficient
s	horizontal distance (m)	S	wing planform area (m^2) & solar panel area (m^2)
t	time (s)	c_D	drag coefficient
m	total mass of aircraft (kg)		

Parameters

$m_{airframe}$	mass of airframe	136 kg
E_D	battery energy density [18]	1260 $\frac{kJ}{kg}$
c_{D_0}	drag equation constant	0.0108
k_{d_1}	drag equation constant	0.0011
k_{d_2}	drag equation constant	0.0127

Equations

$$m \frac{dv}{dt} + mg \sin(\theta) + D - T_p = 0 \quad (20)$$

$$mv \frac{d\theta}{dt} + mg \cos(\theta) - L = 0 \quad (21)$$

$$\frac{dh}{dt} - v \sin(\theta) = 0 \quad (22)$$

$$\frac{ds}{dt} - v \cos(\theta) = 0 \quad (23)$$

$$c_D = c_{D_0} + k_{d_1} c_L + k_{d_2} (c_L)^2 \quad (24)$$

$$L = \frac{1}{2} c_L \rho v^2 S \quad (25)$$

$$D = \frac{1}{2} c_D \rho v^2 S \quad (26)$$

$$m_{panel} = 0.840S \quad (27)$$

$$m_{battery} = E_{batmax} / E_D \quad (28)$$

$$m = m_{airframe} + m_{payload} + m_{panel} + m_{battery} \quad (29)$$

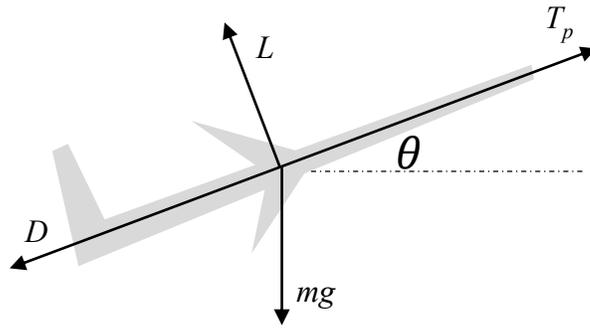


Fig. 8 Free Body Diagram of Airplane

Solar Flux, Battery, and Panel Performance

The equations governing the flow of power from the sun to the propeller and battery are written below. A schematic of this flow is shown in Fig. 9. This section includes the calculation of the incident solar flux [19], panel power

generation, and the differential equation for battery energy. Effects of latitude, day of year, time of day, air mass attenuation [19], and efficiencies of mechanical devices [9] are taken into account.

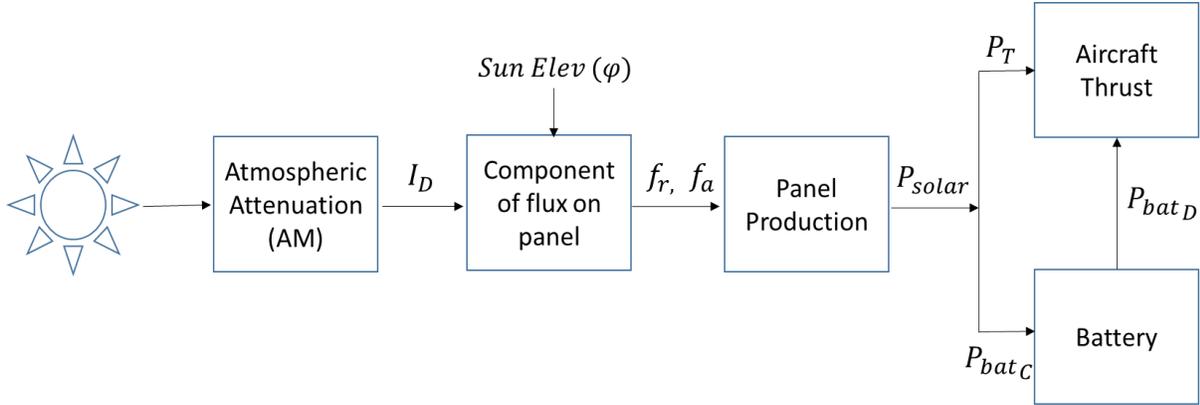


Fig. 9 Power Flow Chart

Variables

σ	Sun hour angle (<i>rad</i>)	f_r	raw solar flux on panel ($\frac{W}{m^2}$)
ϕ	declination angle (<i>rad</i>)	f_a	adjusted solar flux ($\frac{W}{m^2}$)
d	day of the year	P_{solar}	total power generated by panel (<i>W</i>)
φ	elevation angle (<i>rad</i>)	P_T	total power needed for the aircraft (<i>W</i>)
δ	latitude (<i>rad</i>)	E_{bat}	energy stored in battery (<i>kJ</i>)
f_s	solar flux on horizontal panel without atmospheric effects ($\frac{W}{m^2}$)	P_{bat_D}	power discharged from battery (<i>W</i>)
I_D	solar flux within atmosphere ($\frac{W}{m^2}$)	P_{bat_C}	power used to charge battery (<i>W</i>)
AM	air mass effect factor		

Parameters

e_{sol}	efficiency of solar panel [17]	0.295
e_{prop}	efficiency of propeller	0.716
P_{int}	power for fixed internal components	100 <i>W</i>
e_{bat}	efficiency of battery	0.960

Equations

$$\sigma = \left(-180 + \frac{15t}{3600}\right) \frac{\pi}{180} \quad (30)$$

$$\phi = \sin^{-1} \left(\sin \left(\frac{23.45\pi}{180} \right) \sin \left(360\pi \frac{(d-81)}{65700} \right) \right) \quad (31)$$

$$\varphi = \sin^{-1}(\sin(\phi) \sin(\delta) + \cos(\phi) \cos(\delta) \cos(\sigma)) \quad (32)$$

$$f_s = 1353 \sin(\varphi) \quad (33)$$

$$AM = \frac{1}{\sin(\varphi) + 0.50572 \left(6.07995 + \frac{180}{\pi} \varphi \right)^{-1.6364}} \quad (34)$$

$$I_D = 1353 \left[(1 - ah) 0.7^{AM^{0.678}} + ah \right] \quad (35)$$

$$f_r = I_D \sin(\varphi) \quad (36)$$

$$f_a = \max(0, f_r) \quad (37)$$

$$P_{solar} = (f_a)(S)(e_{sol}) \quad (38)$$

$$P_T = P_{int} + \frac{vT_p}{e_{prop}} \quad (39)$$

$$\frac{dE_{bat}}{dt} + \frac{0.001P_{bat_D}}{e_{bat}} - 0.001P_{bat_C}e_{bat} = 0 \quad (40)$$

Power Balance Constraints

$$\text{The battery charging rate must be less than available solar power: } P_{bat_C} < P_{solar} \quad (41)$$

$$\text{The battery discharge cannot exceed power needed by more than 1 W: } P_{bat_D} < P_T + 1 \quad (42)$$

$$\text{Total Power Balance must be satisfied at all times: } P_{bat_C} + P_T < P_{solar} + P_{bat_D} \quad (43)$$

Additional constraint to force discharge power to zero when sufficient solar flux is available:

$$P_{bat_D} = 0 \text{ when solar flux above atmosphere } (f_s) \text{ is greater than } 1000 \frac{W}{m^2} \quad (44)$$

Constraints on Control Profiles

$$1.35 < c_L < 1.5 \quad (45)$$

$$5 < T_p < 500 \quad (46)$$

Boundary Condition Equations to Enforce Perpetuity

$$v_{beg} = v_{end}, \theta_{beg} = \theta_{end}, h_{beg} = h_{end}, E_{bat_{beg}} = E_{bat_{end}} \text{ where the subscripts } beg \text{ and } end \text{ refer to} \quad (47)$$

the beginning and end of the repetitive 24 hour cycle.

Objective Function

$$\text{Min } Z_{obj} = E_{bat_{max}} \approx E_{bat}(t_{sunset}), \text{ where } t_{sunset} \text{ denotes point at which the elevation angle is zero as} \quad (48)$$

the sun is descending to the horizon.

In reality, the point of maximum stored energy occurs slightly before sunset. At sunset, the solar flux is zero and in the period leading to sunset, some battery energy is used up by the aircraft. Determining the exact point of this maximum storage and then minimizing this value is computationally expensive. However, this determination is not necessary. When the energy stored at sunset is minimized, so too is the maximum energy stored, thus practically achieving the same objective with vastly improved computational performance. The cycle is assumed to be a 24 hour cycle and is discretized into 500 equally spaced finite elements. The values of the control variables (T_p , C_L , P_{bat_C} , and P_{bat_D}) were optimized within each finite element. In addition, to prevent bang-bang control profiles, the control variables were held constant within each finite element and constraints on their variation between finite elements were enforced as shown below:

$$\left| T_{P_{i+1}} - T_{P_i} \right| < 2 \quad (49) \quad \left| C_{L_{i+1}} - C_{L_i} \right| < 0.01 \quad (50)$$

$$\left| P_{bat_{C_{i+1}}} - P_{bat_{C_i}} \right| < 25 \quad (51) \quad \left| P_{bat_{D_{i+1}}} - P_{bat_{D_i}} \right| < 25 \quad (52)$$

The subscript i refers to the finite element index. These constraints were effective in reducing the multiple optimal solutions that were encountered in my initial optimization trials. The entire set of equations above was discretized with the procedure described in Section I and implemented in GAMS. Second order polynomials with Radau collocation points and 500 finite elements were chosen for the optimization. Furthermore, the 2-solve initialization strategy was used to improve computational performance.

Results and Discussion

Having completed the development and implementation of the mathematical model, various case studies were performed whose results are shown in Table 1 below. I chose a default case (**bolded** in Table 1) in order to compare against other cases. Each case modifies only the indicated parameter from the default, making it easy to establish causal links. The *minimum* amount of energy storage needed at sunset is reported for each case. The

solution method is quite robust, as the set of diverse cases described above were solved with ease. To understand how variables such as velocity, flight angle, thrust, battery energy, and power usage change over the duration of the 24 hour (86400 s) cycle, graphs are shown for the default case (**bolded** in Table 1) in Fig. 10-16.

Table 1: Summary of case studies and results

Minimum required battery capacity (in kJ) is reported for each case

1. <u>Altitude Range (m)</u>	Min. Battery Size	2. <u>Day of Year</u>	Min. Battery Size
1,000 to 6,000	11470 kJ	79 (Spring Equinox)	14039 kJ
1,000 to 8,000	7832 kJ	172 (Summer Solstice)	7731 kJ
1,000 to 10,000	5943 kJ	180	7832 kJ
		265 (Fall Equinox)	13938 kJ
		355 (Winter Solstice)	20624 kJ
3. <u>Latitude</u>	Min. Battery Size	4. <u>Solar Panel Efficiency</u>	Min. Battery Size
0° (Equator)	13755 kJ	22%	8025 kJ
37° N (San Francisco)	7832 kJ	29.5%	7832 kJ
5. <u>Payload (kg)</u>	Min. Battery Size	6. <u>Battery Efficiency</u>	Min. Battery Size
0	7832 kJ	50%	16145 kJ
30	10761 kJ	75%	10242 kJ
50	13057 kJ	96%	7832 kJ

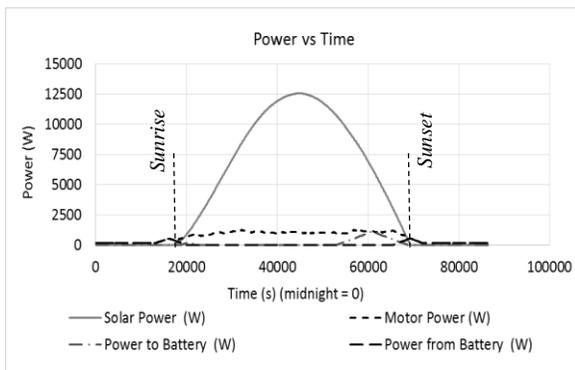


Fig. 10 Power Profiles for Default Case

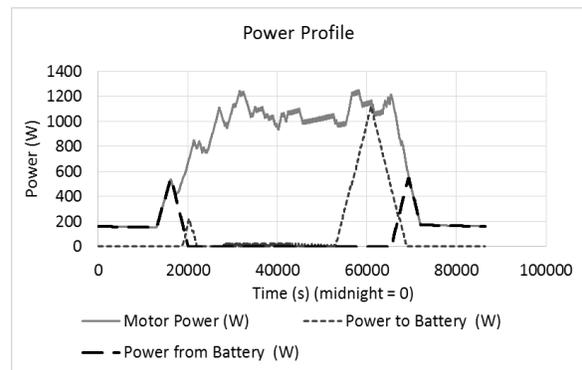


Fig. 11 Zoomed Power Profiles for Default Case

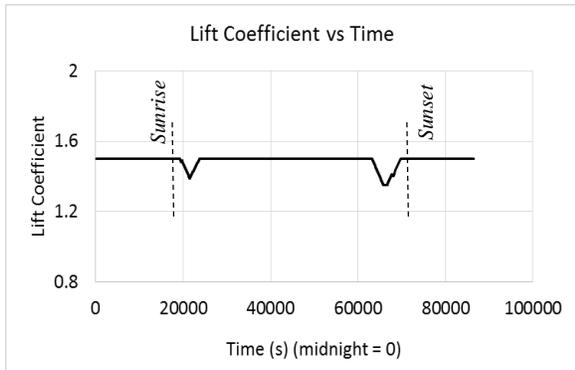


Fig. 12 Lift Coefficient vs. Time for Default Case

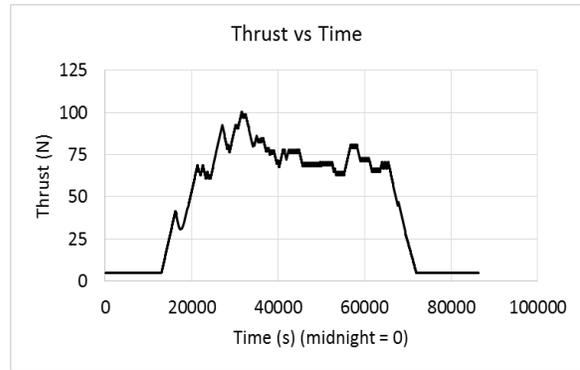


Fig. 13 Thrust vs. Time for Default Case

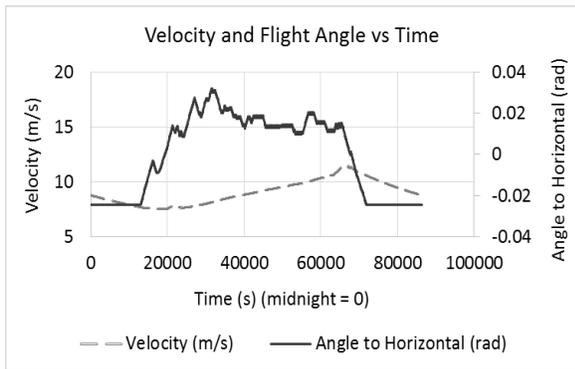


Fig. 14 Velocity and Angle for Default Case

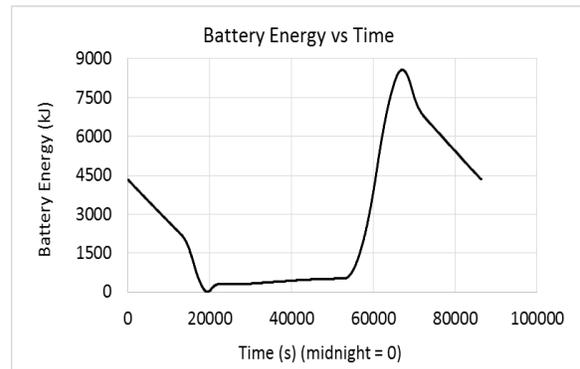


Fig. 15 Battery Energy vs. Time for Default Case

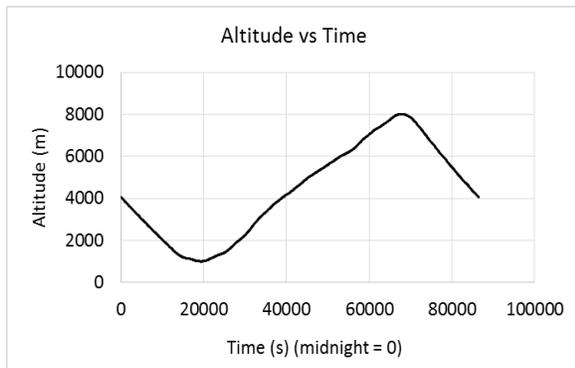


Fig. 16 Altitude vs. Time for Default Case

Figure 10 is interesting as it shows that the solar power available is much greater than that used for powering the plane. This is because the default case is day 180, close to the summer solstice in San Francisco. The power discharge from the battery is quite small at night, and is barely visible in Fig. 10. Therefore, a zoomed version of Fig. 10 without the solar power is shown in Fig. 11. This discharge is so small because the plane has a smooth

descent (Fig. 16) while providing the minimum required thrust of 5 N. Periods of peak discharge occur before sunrise when the plane has hit the lower altitude limit and around sunset to preserve the altitude (Fig. 16). Figures 13 and 14 show that the flight angle is directly correlated to the thrust control variable. The angle is negative until sunrise and then increases as solar power is available. The velocity changes little throughout the flight, although it hits a low around sunrise and then increases with thrust and altitude.

Figure 15 shows the battery energy steadily decreasing from sunset to sunrise due to the power discharge to the propeller. Since the maximum battery energy is minimized, the battery reaches dead storage level (indicated by 0 kJ) slightly after sunrise, when there is enough solar power to provide sufficient thrust. The charging of the battery occurs mainly towards the end of the day, as seen in the battery energy chart in Fig. 15. However, I ran some trials by forcing an earlier start to the charging and realized that the objective function did not change, indicating the presence of multiple solutions. The reason for this is simple. As there is so much excess solar power available on the default day, which is in the middle of summer, there are multiple charging patterns that result in the same objective function. If the model were run on a day with less sunlight, such as the winter solstice, the solver would have far less freedom and charging would begin earlier on in the day. Figure 17 shows the battery energy graph when the model is run on the winter solstice. Predictably, with less sunlight, the battery must start charging much sooner in the day to ensure that it has sufficient energy for the night. As constraints on the model are tightened, the issue of multiple solutions becomes far less problematic. However, the solution method performs reliably even if the objective function is flat over a range of values.

Figure 16 shows the altitude decreasing throughout the night to minimize energy use and then increasing throughout the day in order to gain potential energy from solar power. Figure 12 shows that the lift coefficient is relatively constant except for drops when the upper or lower bounds in altitude are reached. At these points, the solar flux is just barely enough to provide sufficient thrust. As the plane reaches the upper altitude limit, the lift coefficient is lowered to minimize lift induced drag. As night proceeds, the lift coefficient increases to reduce the fall rate, thus minimizing the battery energy requirement. Likewise, at the lower altitude limit, the solar flux is just enough to keep the altitude stable, so lowering the lift coefficient reduces the drag, thus minimizing battery use. Once sufficient solar flux is available, the lift coefficient increases to enable increase in altitude. When the model is optimized with all altitude restrictions removed, the two drops in the lift coefficient disappear.

Case 1 in Table 1 shows the effect of the altitude range on the objective function. As the airplane is given more altitude flexibility, the required energy storage steadily decreases. Since the plane is allowed to fly higher, it uses the

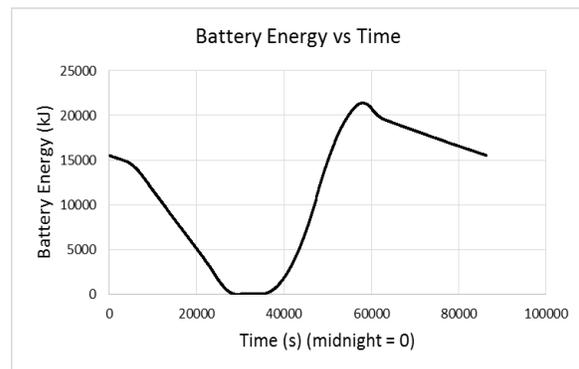


Fig. 17 Battery Energy on Winter Solstice (37° N)

solar energy available in the day to gain altitude. This allows for more room to descend in the night and thus less energy needs to be stored in the battery. For Case 2, the objective function values match what one would expect, as on the spring and fall equinox there is similar amount of sunlight. The summer solstice has maximum sunlight, so minimal energy needs to be stored in the battery. In contrast, for the winter solstice the greatest amount of energy needs to be stored. Figure 18 shows that a higher percentage of solar power is used on the winter solstice than on the default day (Fig. 10). Case 3 shows the effect of latitude. The battery energy storage needed is lower in San Francisco because the default case is the 180th day of the year; therefore, solar flux availability drops as we move south to the equator. For Case 4, the objective function is larger when the panel efficiency drops to 22 % (same as Solar Impulse [5]). Cases 5 and 6 show predictably that as payload increases or battery efficiency reduces, the needed energy storage increases. The battery efficiency study shows a reasonable safety margin, allowing for battery degradation from the default case.

Investigation of Limiting Cases: Results and Methods

I also established the zone of feasibility for continuous operation of the solar aircraft. Since the availability of solar power is the most important factor, I studied cases on the winter solstice in the northern hemisphere, and found that the maximum latitude at which the plane can fly continuously is $47.5^\circ N$, at which the required minimum battery capacity is 24918 kJ. All other parameters were kept fixed as per the default case shown in Table 1. Figure 19 compares the solar power available to the solar power used at latitude $47.5^\circ N$. Comparing this to the graph at latitude $37^\circ N$ (Fig. 18), we can see that the graphs of the power available and the power used almost completely overlap in Fig. 19, as the plane is using nearly all of the solar power available for both propulsion and storing energy for use at night. At latitudes higher than $47.5^\circ N$, solar power is insufficient to sustain continuous flight. Thus, on the winter solstice, this airplane would not be able to fly above Seattle, Washington. Regarding solar panel

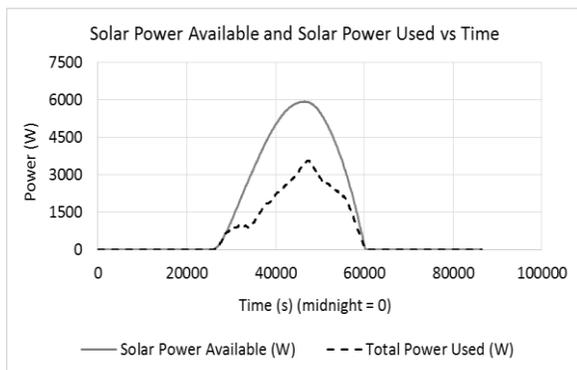


Fig. 18 Power Use at Lat $37^\circ N$ - Winter Solstice

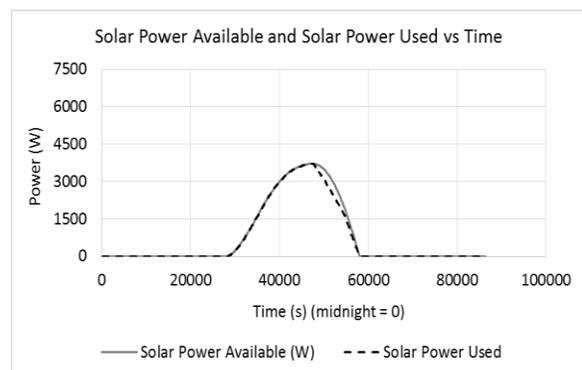


Fig. 19 Power Use at Lat $47.5^\circ N$ - Winter Solstice

efficiency, even at an efficiency as low as 14% with all other parameters kept fixed as per the default case, the airplane could remain in the air continuously on the winter solstice at 37° N, thus enabling use of cheaper solar panels.

Initially, the feasible region studies were conducted by simply changing one parameter, such as the latitude, until the model failed to converge due to insufficient solar power. However, this approach, although successful in my case, can be unreliable. This is because we cannot be certain whether it failed to converge due to truly insufficient solar power or due to solver deficiencies. To improve the reliability of this study, an alternate method was created by adding an ‘artificial sun’ which can provide a power source P_{asun} to the plane. Constraints 41 and 43 are modified to add P_{asun} to the right hand side and the objective function is modified to add the power used from this artificial sun with a large weighting factor as shown in equation 53 below:

$$Z_{obj} = E_{bat}(t_{sunset}) + 1000P_{asun} \quad (53)$$

I determined through trials that a weighting factor of 1000 was sufficient to force the solver to not use this artificial sun unless it is absolutely necessary to ensure feasibility. Besides improving reliability, the other advantage is that by examining how much power from the artificial sun the airplane used, the power deficit beyond the boundaries of feasible operation can be determined.

Computational Results

The simultaneous method I implemented proved robust and computationally efficient for the optimization. All

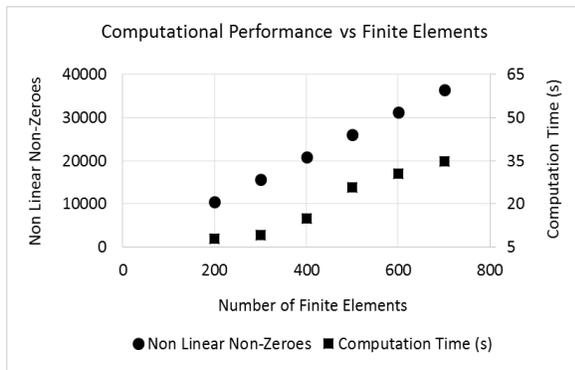


Fig. 20 Computation Time vs. Finite Elements

the cases described above were solved with ease, especially due to the utilization of the 2-solve strategy that was developed after the tests on the HG-2 glider model. Without the 2-solve method, computation times tend to be substantially higher, and in some cases the model fails to converge. The computational performance of the solar aircraft model is reported in Fig. 20. As the number of finite elements increases, the degrees of freedom and

nonlinear non-zeroes increase; however, even for 500 finite elements the computation time is only about 26 seconds. Low computation time is important for the applicability of my method to larger models and allows for quick readjustment of the trajectory should disturbances occur.

Conclusions and Future Work

The solution method I implemented for solving optimal control problems has proven to be robust and computationally efficient. Tests on the glider model were used to tune the algorithm for better performance. The interesting finding of the ‘mad-dash’ for the glider model range optimization demonstrated the method’s ability to find opportunities to increase flight range. Furthermore, the complex solar aircraft optimization runs, in which four control profiles are optimized, take less than 30 seconds for most cases. The robustness and speed of execution that I aimed for were achieved, especially with the 2-solve initialization strategy. By using the simultaneous integration and optimization method, the drawbacks of repeated integration of the differential equations are avoided.

The solar aircraft optimization model formulated is also the first comprehensive model that can be used in two ways. In the first step, the model can be used to determine the minimum battery capacity needed before launch. Following the launch, the model can perform as a control system to ensure that the flight stays on the optimal trajectory for continuous flight. I hope that this model will be used by future researchers to test their solution methods.

There is much more work that needs to be done. On the solver side, the effect of non-linearities on robustness needs to be further investigated. Performance studies of the simultaneous method on trajectory optimization problems involving planetary missions should be done to see how the method could be adapted for space flight. The solar airplane model must be enhanced by accounting for environmental factors such as clouds, rain, wind, and humidity. Regarding multiple solutions, the effect of additional constraints and limits on the control variable profiles must be evaluated. The current model is computationally efficient, but allows for limited motion about the launch point. This is not a major problem since the plane rarely reaches speeds above 50 km/h . However, the effect of the earth’s rotation on the cycle time as well as Coriolis force effects and solar flux changes for more extensive north-south motion should be considered if the range of motion is expected to be wide. The effect of the tilt of the solar panel on the solar flux should also be considered in future work, as this could be a factor in take-off and landing. Furthermore, it is also important to consider better battery power output models in future work. I have recently joined a team of Stanford graduate students to design and build a small solar aircraft using the optimization method and plan to further improve the model and the solution algorithm in the process. I hope this project will help us evaluate the opportunities and practical limitations of continuous solar flight.

Acknowledgements

I am very grateful to the members of the Aerospace Computing Lab at Stanford University. I would like to specially thank Ph.D. student Manuel Lopez for his feedback and encouragement throughout my work. I am also grateful to Professor Antony Jameson, for inviting me to the Aerospace Computing Lab meetings and helping me to learn more about aeronautics. I would also like to thank my AP Physics teacher, Mr. Charles Williams, for his feedback and support.

References

- [1] Bulirsch, R., A. Miele, J. Stoer, and K. Well. *Optimal Control: Calculus of Variations, Optimal Control Theory, and Numerical Methods*. Basel: Birkhäuser Verlag, 1993. Print.
- [2] Biegler, Lorenz T., and Ignacio E. Grossmann. "Retrospective on Optimization." *Computers & Chemical Engineering* 28 (2004): 1169-192. Print.
- [3] Betts, John T. *Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2001. Print.
- [4] Cuthrell, J. E., and L. T. Biegler. "On the Optimization of Differential-algebraic Process Systems." *AIChE Journal* 33.8 (1987): 1257-270. Print.
- [5] "SOLAR IMPULSE - AROUND THE WORLD IN A SOLAR AIRPLANE." *SOLAR IMPULSE - AROUND THE WORLD IN A SOLAR AIRPLANE*. SOLAR IMPULSE, n.d. Web. 10 Aug. 2013. <<http://www.solarimpulse.com/>>.
- [6] Najafi, Yaser. "Design of a High Altitude Long Endurance Solar Powered UAV." Thesis. San Jose State University, 2011. Web. 10 Aug. 2013. <<http://www.engr.sjsu.edu/nikos/MSAE/pdf/Najafi.S11.pdf>>.
- [7] Noth, A., R. Siegwart, and W. Engel. "Autonomous Solar UAV for Sustainable Flight." *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*. Ed. Kimon P. Valavanis. N.p.: Springer Verlag, 2007. 377-405. Print.
- [8] Amos, Jonathan. "'Eternal Plane' Returns to Earth." *BBC News*. BBC, 23 July 2010. Web. 10 Aug. 2013. <<http://www.bbc.co.uk/news/science-environment-10733998>>.
- [9] Sachs, G., J. Lenz, and F. Holzapfel. "Periodic Optimal Flight of Solar Aircraft with Unlimited Endurance Performance." *Applied Mathematical Sciences* 4.76 (2010): 3761-778. Print.
- [10] Bulirsch, R., E. Nerz, J. Pesch, and O. Von Stryk. "Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider." *Optimal Control: Calculus of Variations, Optimal Control Theory, and Numerical Methods*. By R. Bulirsch, A. Miele, J. Stoer, and K. Wells. Basel: Birkhäuser Verlag, 1993. 273-88. Print.

- [11] Villadsen, John, and M. L. Michelsen. *Solution of Differential Equation Models by Polynomial Approximation*. Englewood Cliffs, NJ: Prentice-Hall, 1978. Print.
- [12] Rosenthal, Richard E. "GAMS - A User's Guide." *On-Line Documentation*. GAMS, n.d. Web. 10 Aug. 2013.
<<http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf>>.
- [13] Drud, A. S. "CONOPT--A Large-Scale GRG Code." *INFORMS Journal on Computing* 6.2 (1994): 207-16. Print.
- [14] Weisstein, Eric W. "Radau Quadrature." *MathWorld*. Wolfram, n.d. Web. 10 Aug. 2013.
<<http://mathworld.wolfram.com/RadauQuadrature.html>>.
- [15] Shelquist, Richard. "Equations - Air Density and Density Altitude." *Equations - Air Density and Density Altitude*. N.p., n.d. Web. 10 Aug. 2013. <http://wahiduddin.net/calc/density_altitude.htm>.
- [16] Vinh, Nguyen X., Adolf Busemann, and Robert D. Culp. "Chapter 2: Equations for Flight Over a Spherical Planet." *Hypersonic and Planetary Entry Flight Mechanics*. Ann Arbor: University of Michigan, 1980. 19-28. Print.
- [17] "29.5% NeXt Triple Junction (XTJ) Solar Cells." *SPECTROLAB. A BOEING COMPANY*, 2010. Web. 10 Aug. 2013.
<<http://www.spectrolab.com/DataSheets/cells/PV%20XTJ%20Cell%205-20-10.pdf>>.
- [18] "Lithium Sulfur Rechargeable Battery Data Sheet." *Sion Power*. SION POWER, 3 Oct. 2008. Web. 19 Oct. 2013.
<<http://sionpower.com/pdf/articles/LIS%20Spec%20Sheet%2010-3-08.pdf>>.
- [19] Honsberg, Christina, and Stuart Bowden. "A Collection of Resources for the Photovoltaic Educator." *PVEducation*. N.p., n.d. Web. 10 Aug. 2013. <<http://pveducation.org/>>.