Optimization of a UAV for Wildfire Management

Nathaniel S. Hargan

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Robert L. West Jr., Chair
Michael Philen
Pinar Acar

March 12th, 2026
Blacksburg, Virginia

Keywords: UAV Design, Multidisciplinary Design Optimization, Finite Element Modeling,
Wildfire Management

Optimization of a UAV for Wildfire Management

Nathaniel S. Hargan

(ABSTRACT)

Wildfires pose a significant environmental risk, cause substantial economic damage, and are a danger to human life. Uncrewed aerial vehicles (UAVs) have the potential to support hotshot crews to combat wildfires more effectively. UAVs are valuable due to their low cost and high maneuverability. UAVs with the ability to carry large payloads can effectively transport fire retardant or propellant to create controlled burns remotely, mitigating the risk of uncontrolled wildfires. A UAV designed for this application should be able to carry a large payload, have enough battery capacity to remain in flight for the extent of a mission, and be rigid enough to resist vibration. The UAV must be large enough for the propellers and electrical components needed to lift the payload. The goal of this project is to design a conceptual model of an octocopter UAV and to examine the design space to find an optimal solution. The UAV has a 2-meter wheelbase and is designed to carry a 45kg payload. The UAV model is an 1/8-symmetric sector model of the full UAV, represented as a finite element model, and is used to estimate the deflection, stress, fatigue life, frequency response, and damage. The UAV is modeled as a 3D Timoshenko beam finite element model. The mass and mass moment of the UAV are minimized using non-linear programming.

Optimization of a UAV for Wildfire Management

Nathaniel S. Hargan

(GENERAL AUDIENCE ABSTRACT)

Wildfires pose a significant environmental risk, cause substantial economic damage, and are a danger to human life. Uncrewed aerial vehicles (UAVs) have the potential to support hotshot crews to combat wildfires more effectively. UAVs are valuable due to their low cost and high maneuverability. UAVs with the ability to carry large payloads can effectively transport fire retardant or propellant to create controlled burns remotely, mitigating the risk of uncontrolled wildfires. A UAV designed for this application should be able to carry a large payload, have enough battery capacity to remain in flight for the extent of a mission, and be rigid enough to resist vibration. The UAV must be large enough for the propellers and electrical components needed to lift the payload. The UAV has a 2-meter wheelbase and is designed to carry a 45kg payload. The UAV model is an 1/8-symmetric sector model of the full UAV and is used to estimate the deflection, stress, fatigue life, frequency response, and damage.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of Need

In 2025, the Palisades wildlands fire caused significant loss of life and property, killing over 25 people, burning 23,000 acres of land, and forcing thousands to evacuate. Thousands of buildings were destroyed in the fire [7]. The scale of this disaster highlighted the need for tools for hotshot crews to support firefighting operations.

Uncrewed aerial vehicles (UAVs) have the potential to provide critical assistance in wildfire management. Currently, the most common use of UAVs in wildfire management is scouting. Another application is the deployment of "dragon eggs," incendiary devices used by hotshot crews to start controlled burns, which can help reduce the risk of more dangerous wildfires. Dragon eggs are golf ball-sized, fire-starting payloads [14]. UAVs that are capable of carrying large payloads could allow hotshot crews to start controlled burns while remaining at a safe distance.

Current UAVs often fail to provide sufficient flight times for missions in wildfire management [6], which highlights the need for a computational framework to optimize a conceptual design of a UAV to achieve extended flight times.

1.2 Hypothesis Statement

A computational framework can be created that utilizes multidisciplinary design optimization in order to improve the conceptual design of an octocopter UAV. The mission profile defines the kinematics and energy requirements for a potential mission. The internal forces in the UAV and power consumption calculations can be derived from the mission profile. A beam model of the UAV is created using Timoshenko beam theory and is used in conjunction with the internal forces to derive constraints for the optimization model, such as displacement, stress, frequency, damage, and energy. This framework can be used to rapidly test different UAV design concepts to test their viability for stated mission profiles, while satisfying design constraints.

1.3 Objective

The primary goal of this project is to develop a framework to optimize the conceptual design of a UAV capable of supporting wildfire operations. The UAV must be able to carry a 45kg payload, so that the UAV can transport dragon eggs across a wildfire zone. The mission profile for this project is based on a potential operation during the 2025 Palisades wildfire. The UAV must traverse a total distance of approximately 5.6 kilometers, corresponding to the span from the northern to the southern edge of the active fire area.

In order to accomplish this goal, a tool to model the mission profile will be created. Using the mission profile tool, forces from the propellers can be estimated, as well as the energy required to accomplish the mission profile. The drone geometry will be modeled from a set of design variables. A finite element model will be created to calculate stresses within the vehicle using forces derived from the mission profile.

The UAV conceptual design is subject to five constraints to ensure structural stability and mission viability. These constraints include structural deflection, stress, energy, frequency response, and fatigue. Subject to these constraints, the design of the UAV is optimized to minimize the mass and mass moment of inertia.

This should create a framework to enable the user to rapidly evaluate the validity of UAV designs for the application of supporting wildfire operations.

1.4 Scope

This thesis investigates the structural optimization of a conceptual octocopter UAV frame intended for wildfire management missions. The UAV is modeled as a radially symmetric eight-arm multirotor configuration. Structural behavior is represented using a finite element formulation derived from Timoshenko beam theory. To reduce computational cost, a 1/8-symmetric slice model of the full UAV is used. The imposed boundary conditions restrict the solution to symmetric vibration modes. The frame consists of arms, struts, and a segmented hub modeled using annular and I-beam cross-sections. The carbon fiber material is modeled as homogeneous and quasi-isotropic. Fatigue material properties are limited to aluminum.

Loading is derived from a predefined mission profile. Thrust is calculated from mission profile kinematics. The mass response, force from gravity, and drag are all modeled as lumped at the center of mass. All forces are modeled as vertical, and UAV tilt is not considered.

The optimization problem minimizes a weighted combination of frame mass and mass moment of inertia subject to constraints on displacement, stress, frequency response, fatigue, and mission energy consumption. Optimization is performed using SLSQP and differential evolution algorithms.

1.5 Outline

- Chapter 1 introduces the project, gives the motivation behind it, and specifies the scope of the project.
- Chapter 2 covers the existing literature around UAV optimization and mission profile modeling.
- Chapter 3 describes the design of the UAV.
- Chapter 4 describes the mission profile and derived force and energy calculations.
- Chapter 5 explains the finite element tool that was designed for the project.
- Chapter 6 covers the optimization tool as well as the formulation of each of the constraints.
- Chapter 7 describes the resulting optimized design for two different configurations of the optimization problem.
- Chapter 8 concludes the project, summarizes it, and gives recommendations on where to move forward with the project.

Chapter 2

Review of Literature

2.1 Multidisciplinary Design Optimization for UAVs

Multidisciplinary design optimization (MDO) addresses engineering design problems that involve multiple interacting disciplines which must be considered simultaneously. Rather than optimizing subsystems in isolation, MDO seeks a system-level optimum that accounts for interdisciplinary coupling and trade-offs, making it a central approach in the design of complex engineering systems.

A range of computational frameworks has been developed to support MDO. OpenMDAO is an open-source Python framework for multidisciplinary design, analysis, and optimization developed by NASA. It is primarily intended for gradient-based optimization problems [10].

UAV design is inherently multidisciplinary, including structures, aerodynamics, propulsion, power, and controls. These interactions motivate the use of MDO methods. Several tools for UAV design have been built on top of OpenMDAO. OpenAeroStruct provides a multidisciplinary model for UAV analysis and optimization. Jasa et al. applied OpenAeroStruct to optimize the design of a UAV with 5 design variables and subject to constraints on lift and yield strength [12]. OpenAeroStruct uses the SNOPT algorithm for optimization, a sequential quadratic programming method to solve nonlinear constrained problems.

FAST-UAV by Pollet et al. [15] has been developed as a Python-based tool for MDO of

UAV designs. It is built on the FAST-OAD and OpenMDAO frameworks. FAST-UAV was applied to optimize the design of octocopters, hexacopters, and quadcopters. Four possible objective functions were used: total mass of the UAV, energy consumption of the UAV, flight endurance, and flight range. Several constraints were imposed on the vehicle, including maximum bending stress, power requirements, and limitations on its geometric properties. Twelve different sizing coefficients are used as design variables for the multirotor design. FAST-UAV uses the SLSQP algorithm for optimization. These methods only have linear constraints and no constraints related to the frequency of the vehicle.

2.2 Mission Profile Design

The mission profile describes the kinematics of an aerial vehicle over the course of a mission. This is crucial to model a mission profile to optimize the design of a UAV so that it can be determined whether the mission is feasible.

The Mission Planning Tool is developed by NASA [16] and is designed to model mission profiles. The Mission Profile Tool optimizes the design of a user-specified mission profile using OpenMDAO to minimize the total power used. The mission profile is used to calculate battery state of charge, electrical component temperatures, thrust, and power usage at all phases of flight. The mission profile is divided into six phases: taxi, motor runup, takeoff, climb, cruise, and descent. The dynamics of the UAV are calculated in each phase.

Chapter 3

UAV Design

This chapter describes the UAV geometry, design variables, material properties, and design parameters used in the project.

3.1 UAV Geometry

The baseline UAV configuration used in this study follows a previously developed firefighting UAV design [13]. The UAV is modeled as a simplified octocopter frame, as shown in Figures 3.1–3.3, with individual frame components highlighted. The frame consists of three primary components: arms, struts, and a central hub. The arms and struts are modeled as beam elements with annular cross-sections. The hub is composed of two octagonal plates connected by a radially symmetric web, shown in Figure 3.4, consisting of eight web sections spanning from the center to the edges of the plates. All connections between arms, struts, and hub are modeled as rigid.

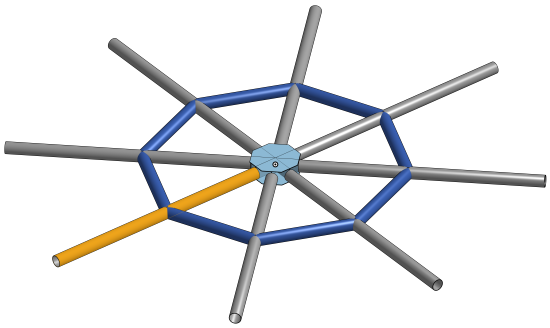


Figure 3.1: Optimized UAV Model with Highlighted Arm

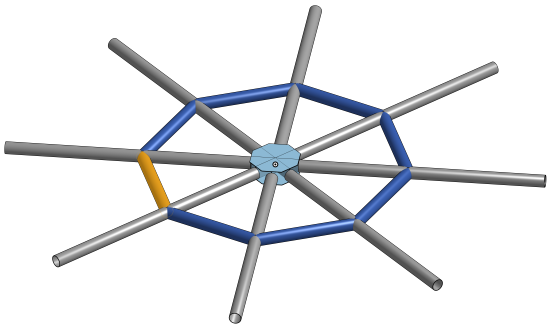


Figure 3.2: Optimized UAV Model with Highlighted Strut

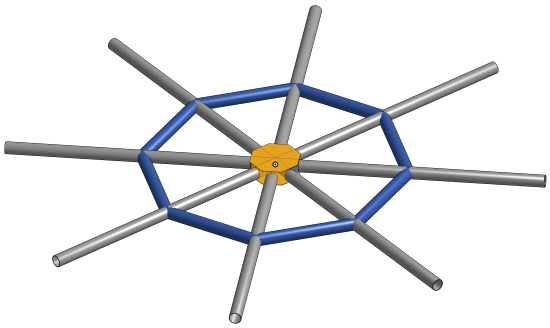


Figure 3.3: Optimized UAV Model with Highlighted Hub

(a) Top Section View (b) Isometric View

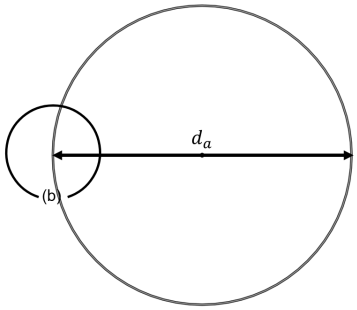
Figure 3.4: Optimized UAV Model Hub Cross-Section View

3.1.1 Design Variables

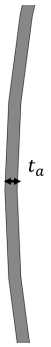
Eight design variables were selected for this analysis and are displayed in Table 3.1. Figure 3.5 shows dimensions for each design variable.

Table 3.1: Design Variable Descriptions

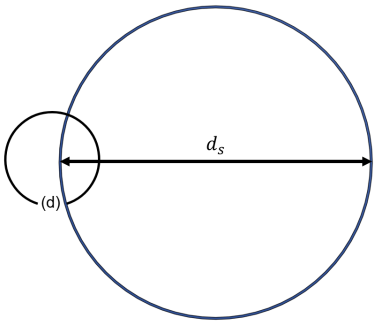
Variable	**Symbol**
Arm outer diameter	d_a
Arm wall thickness	t_a
Strut outer diameter	d_s
Strut wall thickness	t_s
Strut distance from hub center	r_s
Hub flange thickness	t_f
Hub web thickness	t_w
Hub radius	r_h



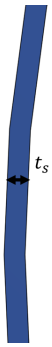
(a) Arm Outer Diameter (d_a)



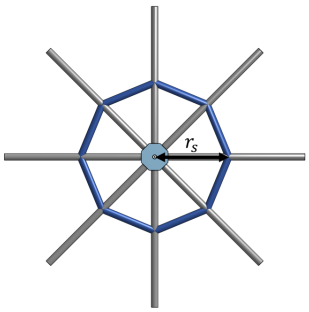
(b) Arm Thickness (t_a)



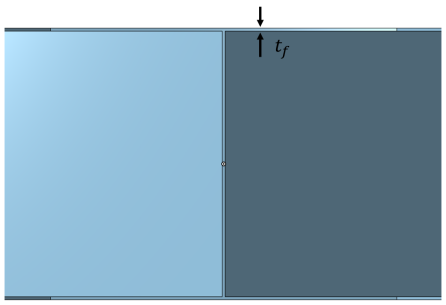
(c) Strut Outer Diameter (d_s)



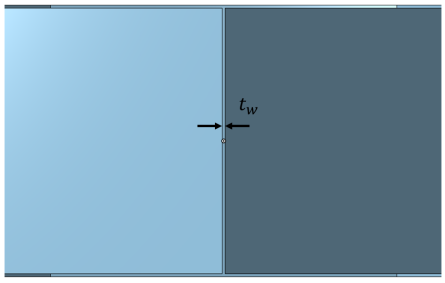
(d) Strut Thickness (t_s)



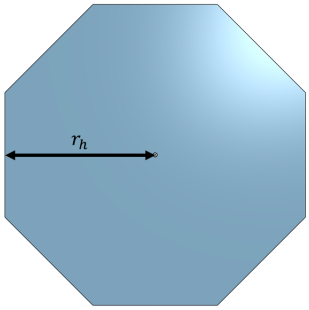
(e) Strut Distance from Center (r_s)



(f) Hub Flange Thickness (t_f)



(g) Hub Web Thickness (t_w)



(h) Hub Radius (r_h)

Figure 3.5: Design Variable Visualizations on UAV model

3.1.2 Design Parameters

The wheelbase and hub height are two parameters of the UAV frame that were not selected as design variables. The inner hub height is set to 64mm, which represents the minimum height required to fit the proposed battery between the hub top and bottom plates. It should be noted that the actual batteries would not fit within this frame model, because the hub web would intersect the battery. Further work could be done to refine the mesh, allowing for more realistic battery storage.

The wheelbase was set as a parameter since increasing beyond the minimum will not increase the objective function, as increasing the wheelbase increases mass and the mass moment of inertia while also causing more stress on the vehicle. The distance from the center of the UAV to the propeller is calculated to be 2254.6mm. This is calculated by finding the minimum distance required to fit all 8 propellers with a radius of 406.4mm with a propeller spacing of 25mm.

3.1.3 Mass and Payload

The UAV carries a payload of 45 kg, which is located at the center of the vehicle. The mass of the frame (m_{frame}) is the mass of the beam model calculated from the UAV geometry and the mass density of the carbon fiber. It is shown in Equation 3.1. The mass of the payload is 45kg. The mass of the two batteries ($m_{\text{batteries}}$) is 8.26kg total.

$$m_{\text{total}} = m_{\text{frame}} + m_{\text{payload}} + m_{\text{batteries}} \tag{3.1}$$

3.1.4 Material Properties

The material selected for all UAV frame components (struts, arms, and hub beams) is a braided carbon fiber composite. For the purposes of this project, the carbon fiber is modeled as homogeneous and isotropic. Future work could include obtaining experimental data for the Young's moduli, Poisson's ratios, and shear moduli along the three principal material directions in order to more accurately model the carbon fiber as an orthotropic material. The properties listed in Table 3.2 correspond to data obtained from ALLRed Corporation for carbon fiber braided round tubes.

Table 3.2: Material Properties

Property	Symbol	Value	Units
Young's Modulus	E	19000	MPa
Tensile Yield Strength	σ_{yield}	345	MPa
Posson's Ratio	ν	0.3	-
Mass Density	ρ	12179	kg/m^3

The shear modulus is calculated from the relationship between tensile strength and Poisson's ratio and the shear modulus, $G = \frac{E}{2(1+\nu)}$. Due to a lack of fatigue information for the specific carbon fiber composites, fatigue material data is taken from aluminum 7075-T6[4]. The data used is shown in Table 3.3. The completely reversed stress amplitude was calculated from the equation $\sigma_{ar} = \sigma'_f(2N_e)^b$.

Table 3.3: Material Properties for Fatigue

Property	Symbol	Value	Units
Fatigue strength coefficient	σ'_f	1466	MPa
Fatigue strength exponent	b	-0.143	-
Fatigue strength cycles	N_e	1.00E+08	-

A Woehler curve for the fatigue analysis was created using the Pylife Library [5]. The completely reversed stress amplitude is the load level at the endurance limit for the Woehler curve. The fatigue strength cycles is the cycle number at the endurance limit for the Woehler curve. The fatigue strength exponent is used for the slope of the Woehler curve.

Chapter 4

Mission Profile

This Chapter describes the mission profile generation. The calculation of energy and power over the duration of the mission are also described.

4.1 Units

The finite element model described in Chapter 5 and the mission profile generation described in Chapter 4 have no units. Therefore, inputs into and outputs out of the FEA model and mission profile generation system require a consistent unit system. Processes within the project operate with the unit system specified in Table 4.1 unless otherwise specified.

Table 4.1: Unit System

Quantity	**Unit**
Mass	$N \cdot s^2/mm$
Pressure / Stress	MPa (N/mm^2)
Length	mm
Velocity	mm/s
Acceleration	mm/s^2
Force	N
Time	s
Power	mW
Current	mA
Mass Moment of Inertia	$N \cdot s^2 \cdot mm$
Frequency	rad/s
Angle	rad

4.2 Mission Profile Generation

A model of the mission profile was created to represent a potential mission with continuous position, velocity, and acceleration as functions of time. The mission is defined as a series of connected mission profile segments, with constraints imposed on position, velocity, and acceleration at the nodes between segments. Unconstrained nodal values are calculated using basic kinematic relationships. Between segment nodes, values are interpolated using kinematic equations for projectile motion, assuming that acceleration varies linearly with time.

The mission profile is based on a potential operation during the 2025 Palisades wildfire. The total mission distance is estimated to be 5.6 km, corresponding to the distance from the northern edge to the southern edge of the active fire area. Using elevation maps of the surrounding terrain, the required altitude change to traverse this path is estimated to be 460m [3]. Time steps were selected such that the maximum downrange velocity remained below 12m/s.

The climb, cruise, and descent phases are each described in two intervals to avoid over-constraining the mission profile. Figure 4.1 shows altitude as a function of downrange position, Figure 4.2 shows altitude versus time, and Figure 4.3 shows downrange position with respect to time. The constraints applied to the mission profile are summarized in Table 4.2, where the duration represents the time between spline intervals. Blank entries indicate that no constraint was applied.

Table 4.2: UAV Mission Profile Constraints

Segment	**Initial**	**Takeoff**	**Climb 1**	**Climb 2**	**Cruise 1**	**Cruise 2**	**Descent 1**	**Descent 2**	**Land**
Duration (s)	N/A	12	10	10	25	25	10	10	15
Position X (m)	0	–	–	500	2800	5100	–	–	5600
Position Y (m)	0	–	–	560	560	560	–	–	0
Velocity X (m/s)	0	–	–	–	–	–	–	–	0
Velocity Y (m/s)	0	–	–	–	–	–	–	–	0
Acceleration X (m/s^2)	0	–	–	0	–	0	–	–	0
Acceleration Y (m/s^2)	0	–	–	0	–	0	–	–	0

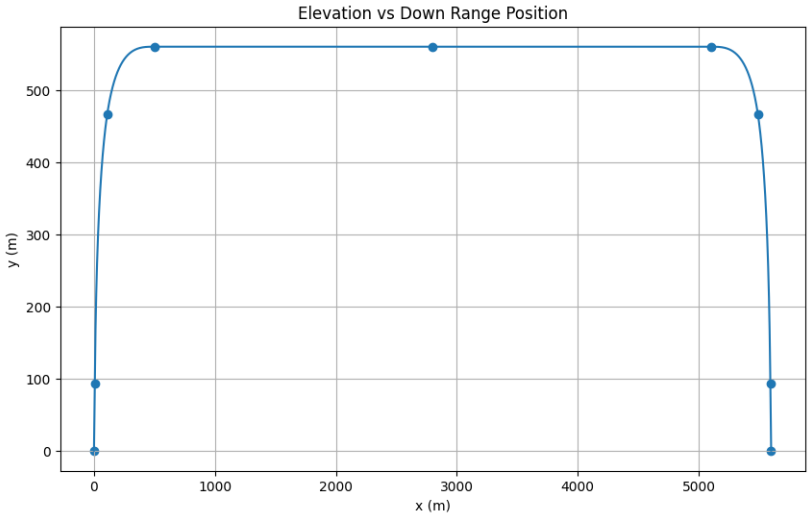


Figure 4.1: Downrange Position vs. Altitude

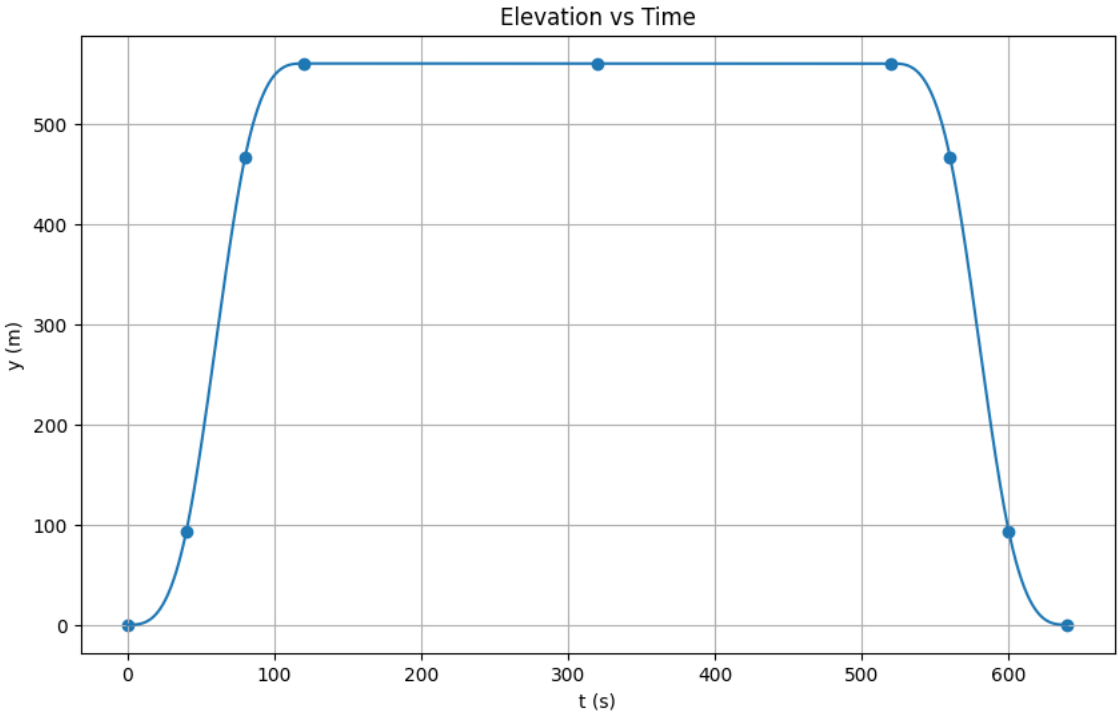


Figure 4.2: Altitude vs. Time

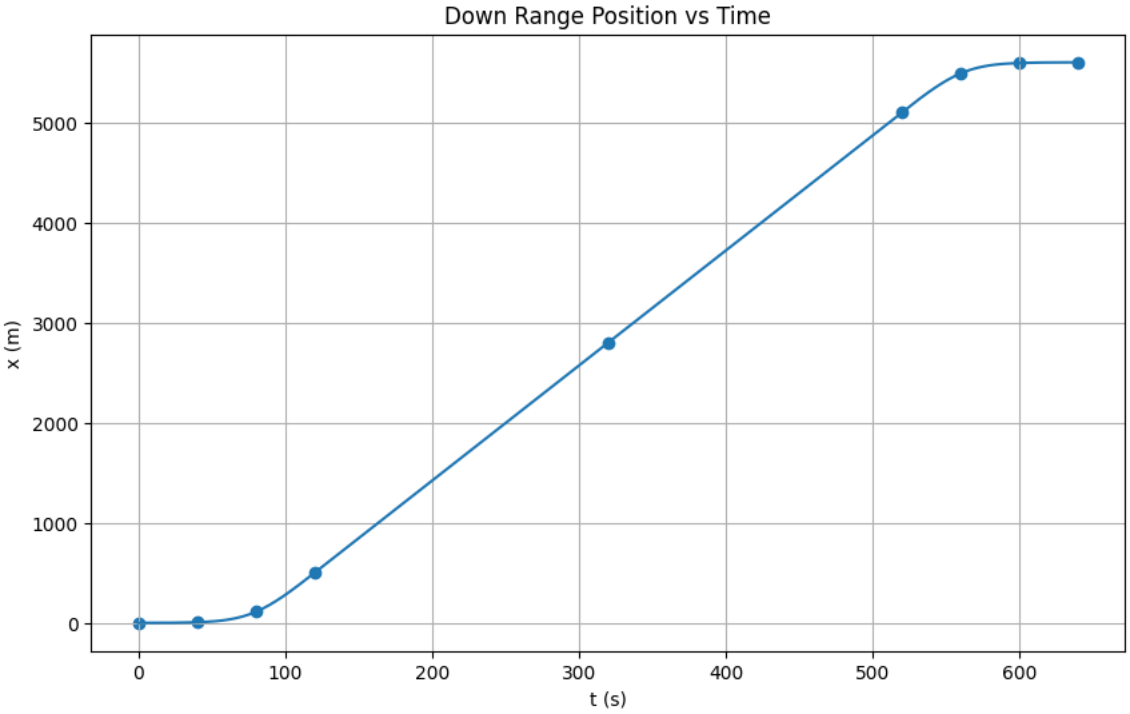


Figure 4.3: Downrange Position vs. Time

The data flow diagram (DFD) for the mission profile generation process is shown in Figure 4.4. The constraints require the node at which the constraint will be applied, the value of the constraint, and the type of constraint. The type of constraint describes the axis the constraint is on and the kinematic quantity that is being constrained. The constraints are then assembled into a kinematic matrix for the entire mission profile. Each node refers to a specific time set by the user. Equation 4.1 describes the relationship between the constrained kinematic quantities ($\mathbf{x}_u$) and the unknown kinematic quantities ($\mathbf{x}_p$). The kinematic matrix is split into two parts: the unconstrained kinematic equations matrix ($\mathbf{A}_k$) and the constrained kinematic equations matrix ($\mathbf{B}_k$).

$$\mathbf{A}_k\mathbf{x}_p + \mathbf{B}_k\mathbf{x}_u = \mathbf{0}, \tag{4.1}$$

The solution of the unknown kinematic quantities is found using the numpy.linalg.lstsq function.

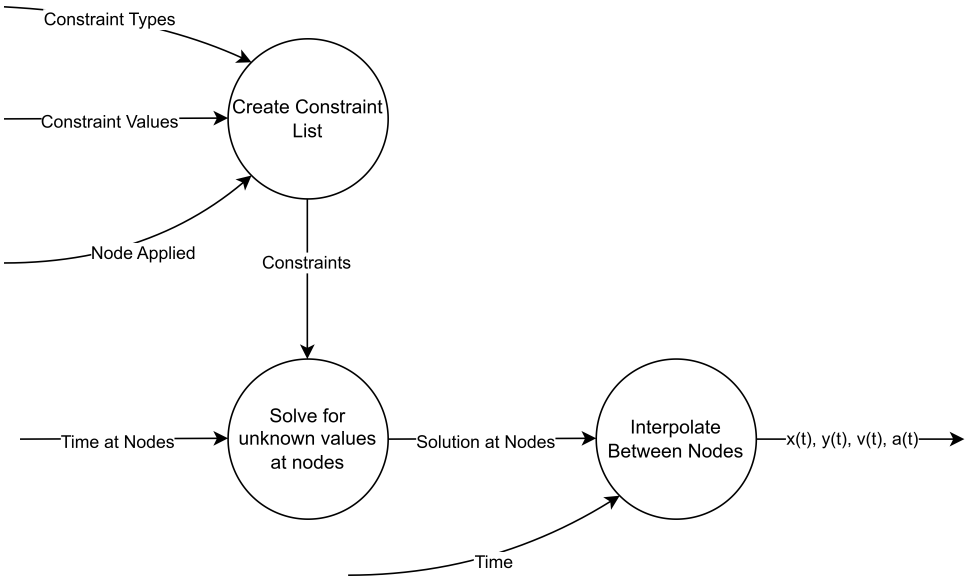


Figure 4.4: DFD of the Mission Profile Generation Process

4.2.1 Forces

This section describes the forces modeled on the vehicle. Figure 4.5 shows the free-body diagram of the UAV. The force applied by a single propeller is F_{thrust}. The force from drag is F_{drag}. All force from gravity (W) is assumed to be at the center of gravity. The weight is described with the equation $W = m_{\text{total}}g$. The combined weight of the batteries, frame, and payload is m_{total}. The acceleration due to gravity is $g = 9.8066\text{m}\,\text{s}^{-2}$. The magnitude of the torque for each propeller is τ.

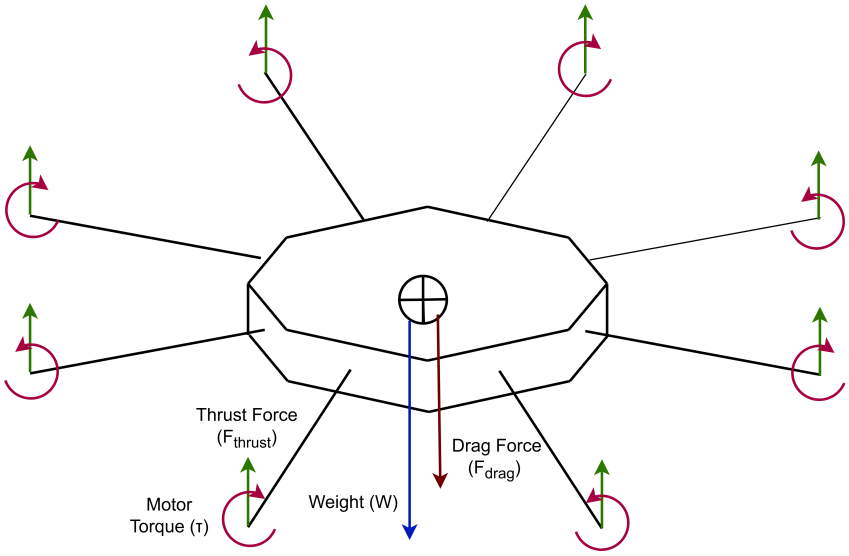


Figure 4.5: Free Body Diagram of the UAV

Equations 4.2 and 4.3 describe the moment and force balances. The sum of all moments is assumed to be zero. The sum of all forces in the x and y directions is also zero. The z-direction is upward.

$$\sum \mathbf{M} = I\alpha = \mathbf{0} \tag{4.2}$$

$$\sum F_x = 0, \quad \sum F_y = 0 \tag{4.3}$$

Force from drag is calculated with Equation 4.4. The vertical velocity of the mission profile at a given point in time is $v_z(t)$. The direction of the velocity is $\text{sgn}(v_z(t))$. s_a is an estimate of the projected surface area of the top of the UAV. This is obtained by summing the projected surface of every beam and hub segment. The surface area is a function of the design variables. $\rho = 1.2250 \text{kg m}^{-3}$ is the air density. The drag coefficient used is $c_d = 0.512$.

$$F_{\text{drag}}(t) = -\frac{1}{2}\,\text{sgn}(v_z(t))\,v_z(t)^2\,s_a\,\rho\,c_d \tag{4.4}$$

The formula for the forces in the z-direction is shown in Equation 4.5. The vertical acceleration in the mission profile is $a_z(t)$. The force from thrust over the duration of the mission profile is given as $F_{\text{thrust,dynamic}}(t)$.

$$\sum F_z = m_{\text{total}} * a_z(t) = -W + F_{\text{drag}}(t) + 8F_{\text{thrust,dynamic}}(t) \tag{4.5}$$

The equation for thrust over the duration of the mission profile is shown in Equation 4.6. This equation is derived from Equation 4.5. In the case where the UAV is in hover, $a_z(t)$ and $v_z(t)$ are equal to zero, and this is described as $F_{\text{thrust,hover}}(t)$. This is shown in Equation 4.7.

$$F_{\text{thrust,dynamic}}(t) = \frac{m_{\text{total}} * a_z(t) + W - F_{\text{drag}}(t)}{8} \tag{4.6}$$

$$F_{\text{thrust,hover}}(t) = \frac{W}{8} \tag{4.7}$$

To simulate the pre-liftoff and post-landing conditions, a rev-up and rev-down period were added to the thrust forcing function. The ramp-up phase is modeled as a linear, increasing thrust from zero to the thrust required for takeoff, while the ramp-down phase is modeled as a linear decrease in thrust to zero. The rev-down period was modeled as a linearly decreasing force until no force is applied. A duration of five seconds is used for both periods.

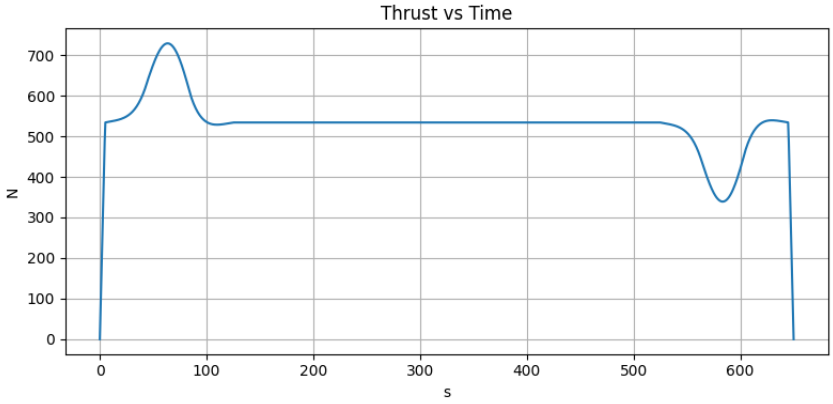


Figure 4.6: Thrust vs. Time Plot

4.3 Energy Consumption

The energy consumption of the UAV over the mission profile is calculated for the energy constraint. The data flow diagram for this energy consumption algorithm is shown in Figure 4.4. Once the thrust function is obtained, as described in Subsection 4.2.1, it is passed to the thrust to power calculation described in Subsection 4.3.2. This model computes the power at each time step. The resulting power profile is then integrated to determine the total energy consumption, as discussed in Subsection 4.3.4. This calculation requires the total energy capacity of the battery system.

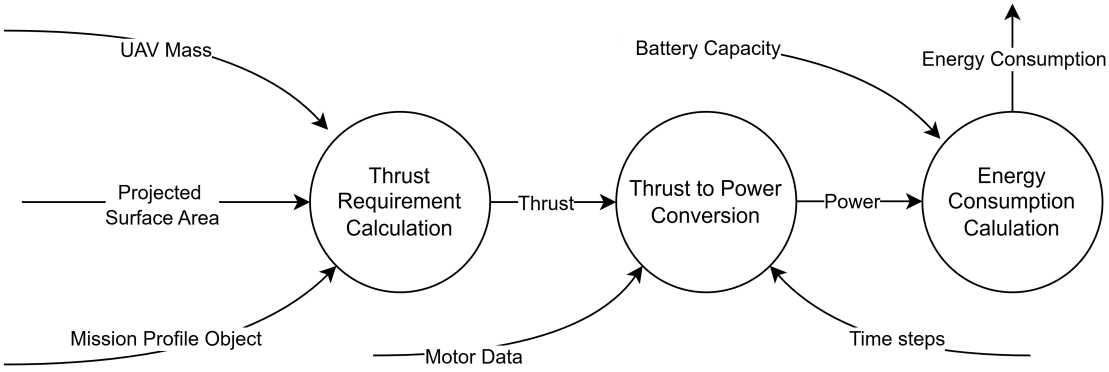


Figure 4.7: DFD of the Energy Consumption Calculation

4.3.1 Batteries

The battery used is the UAV Lipo Battery Pack [2]. Two of these batteries are connected in series, resulting in a total system voltage of 45.6 V. The energy capacity ($E_{capacity}$) of the battery system is calculated by multiplying the total system voltage by the amp-hours of the battery. The batteries have the dimensions of 263x122x64mm. The centroids of the batteries are positioned at (0, 86.75, 0) and (0, -86.75, 0) in the global coordinate system.

4.3.2 Power Calculation

The motor used for vehicle propulsion is the U8 Lite L Efficiency Multirotor UAV Motor KV95 [20]. The propeller used is the G32*11 [21]. Test data from the supplier is shown in Table 4.3 for the thrust, power, and RPM for the pair of motor and propeller. This data was used to generate Equation 4.8 and Equation 4.9.

Table 4.3: Published Data

Throttle	Thrust (kgf)	RPM	Power (W)
40%	2.492	1519	190
45%	3.087	1681	257
50%	3.672	1826	328
55%	4.217	1966	409
60%	4.878	2110	508
65%	5.578	2252	618
70%	6.251	2383	740
75%	6.83	2504	853
80%	7.587	2636	1008
90%	9.144	2873	1375
100%	10.611	3068	1794

To find the power at any point along the curve, the quadratic function of best fit was found for thrust vs power data using the second-order polynomial trend line in Excel. Equation 4.8 shows the resulting thrust to power relationship. $P(t)$ is the power over time in units of watts. $T(t)$ is thrust in units of kilograms force. Figure 4.8 is a plot of the published data and the resulting trend line.

$$P(t) = 11.743\,T(t)^2 + 42.302\,T(t) + 15.871 \tag{4.8}$$

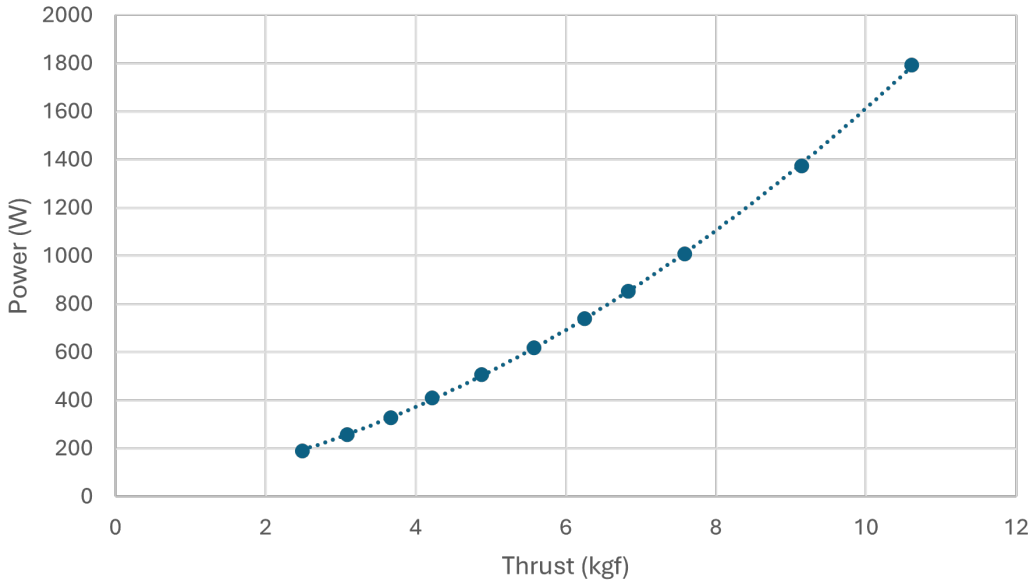


Figure 4.8: Power Vs. Thrust Curve

4.3.3 Angular Velocity Calculation

For the perturbation model and frequency constraint, the angular speed of the propeller over the duration of the mission ($\omega(t)$) was calculated. Equation 4.9 was created by finding the quadratic function of best fit of the RPM vs Thrust data from the supplier's test data. This equation converts the thrust ($T(t)$) in units of kilograms force into angular velocity with the units of RPM. Table 4.9 contains the line of best fit.

$$\omega(t) = -9.3523 \times 10^{-6}\, T(t)^2 + 312.11\, T(t) + 805.73 \tag{4.9}$$

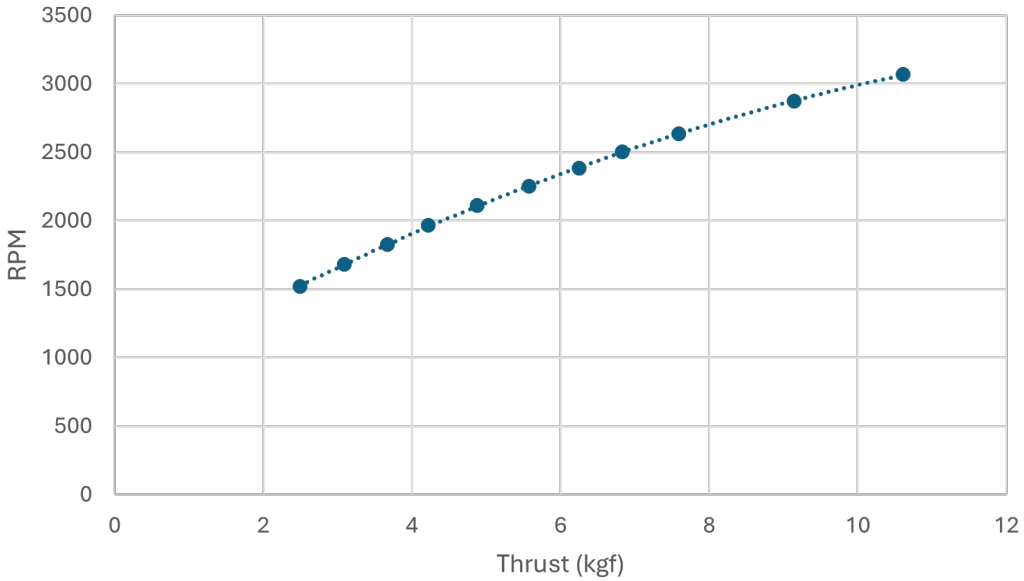


Figure 4.9: Angular Velocity vs. Thrust Curve

4.3.4 Energy Consumption Calculation

The energy consumed over the mission duration is computed by subtracting the time integral of the power draw from the initial energy capacity of the battery system. Equation 4.10 describes the relationship between the energy remaining in the battery and the power consumed. The total energy remaining between both batteries as a function of time is $E_{remaining}(t)$. The energy stored by a fully charged battery is $E_{capacity}$. The time since the start of the mission is t.

$$E_{\text{remaining}}(t) = E_{\text{capacity}} - \int_0^t P(t)\,dt \tag{4.10}$$

The integral of the power is calculated numerically using scipy.integrate.cumulative_trapezoid. The number of time steps is a number set when running the optimization function. The total

span of time calculated is the duration of the mission.

4.3.5 Perturbation Calculation

A forcing function modeling the oscillations caused by propeller perturbations was also required for the transient analysis in Subsection 5.3.3. The desired function is a sinusoidal wave with a frequency matching double the propeller frequency, as the propeller has two blades. The amplitude of the sinusoidal forcing function is modeled as 1% of the magnitude of the thrust. Equation 4.11 shows the completed equation for the perturbation forcing function.

$$F_{\text{perturbation}}(t) = 0.01 \sin\left(2\omega(t)\, t\right) F_{\text{thrust}}(t) \tag{4.11}$$

4.3.6 Torque Calculation

Torque is calculated using Equation 4.12. The motor reaction torque is assumed to be static. The maximum value for power $P(t)$ over the mission profile is P_{max}. The maximum value for angular velocity $\omega(t)$ over the mission profile is ω_{max}. This is an overestimation as the torque formulation does not take into account losses in the motor.

$$\tau = P_{\text{max}} * \omega_{\text{max}} \tag{4.12}$$

Chapter 5

Finite Element Analysis

This chapter discusses the architecture of the FEA model designed to be run by the optimization algorithm. The finite element model is designed so that it can be easily adapted for different frame models and design variables.

5.1 Beam Model

A beam model contains the coordinates of the nodes and the element definitions for the beam system. Figure 5.2 shows a data flow diagram of the process of generating the beam model. Each node is given a name as well as a set of three-dimensional coordinates within the global coordinate system. The nodes that define the elements are shown in Figure 5.1. Each element is created with these three nodes: the *i-node*, the *j-node*, and the *k-node*. The *i-node* is the start of the beam, the *j-node* is the end of the beam, and the *k-node* describes the orientation of the cross-section. The *i-node* and *j-node* are defined as points within the global coordinate system, while the *k-node* is defined with a vector within the global coordinate system which determines the direction of the top face of the beam element. Results from the beam model were checked against an FEA model in Abacus to ensure the validity of results.

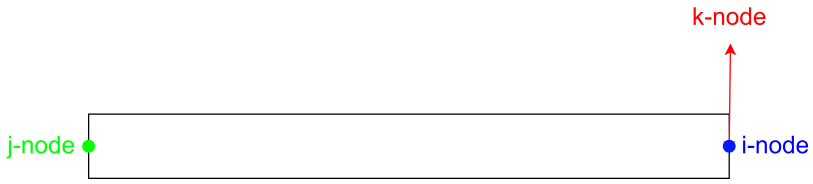


Figure 5.1: Diagram of Nodes for Beam Elements

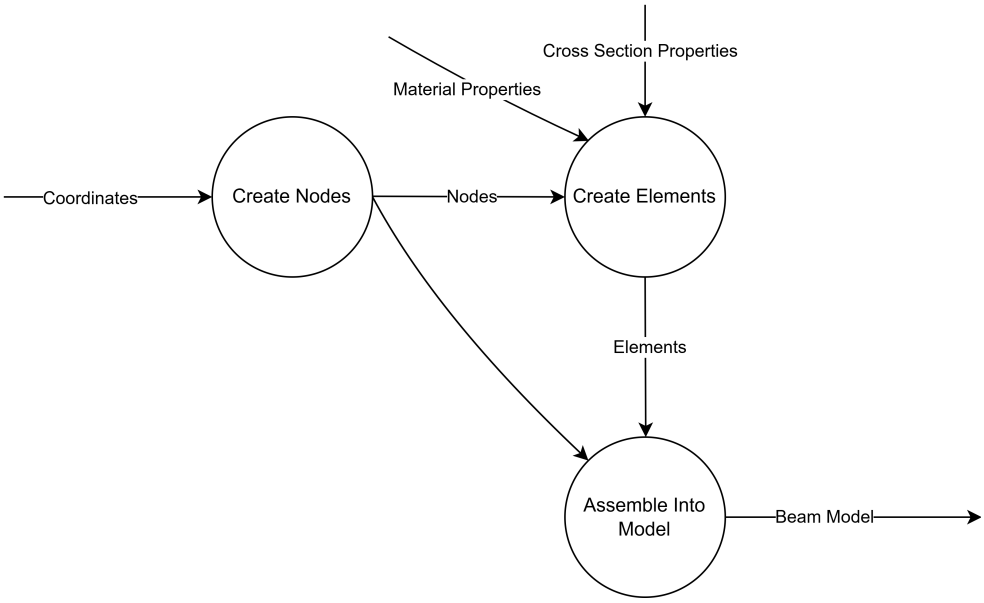


Figure 5.2: DFD of the Creation of the Beam Model

A set of material property data and cross-section properties data is assigned to each element in the model. The section properties include the type of cross-section of the beam element and the dimensions of that cross-section. The two cross-section types used in this thesis are annular and I-beam sections. From these cross-section dimensions, the relevant properties are calculated such as: cross-sectional area, second moment of area in both planar directions, second polar moment of area, torsional constant, transverse shear deflection constant in both directions on the cross-section plane, and the circumscribed radius of the cross-section [18][19].

The transverse shear deflection constants are calculated assuming Poisson's ratio is zero.

The equations used for the transverse shear deflection constants are obtained from work by Shames and Dym [17].

The material properties required for the beam model are the mass density, elastic modulus, Poisson's ratio, tension yield strength. The values are retrieved from the supplier's data in Table 3.2. The fatigue strength coefficient, exponent, and cycles are required for fatigue analysis. The shear modulus (G) is calculated from the elastic modulus (E) and Poisson's ratio (ν) using the relationship shown in Equation 5.1.

$$G = \frac{E}{2(1+\nu)}. \tag{5.1}$$

5.1.1 Beam Element Formulation

For each beam element, a stiffness and mass matrix within the local coordinate system must be created. The stiffness matrix is developed from Timoshenko beam theory so shear deformation effects are taken into account. Timoshenko beam theory is more accurate than Euler–Bernoulli beam theory, which is desired for the use in transient and dynamic analysis. Each beam element has two nodes, each with three displacement and three rotational degrees of freedom. The stiffness matrix ($\mathbf{K}$) is described in Equations 5.3 - 5.7. It requires the length of the beam (L), second moments of area (I_y and I_z), the torsional constant (K), and the transverse shear deflection constants (K_y and K_z). The material properties required are the elastic modulus (E) and shear modulus (G). The shear factor calculations used in Equation 5.4-5.7 are described in Equation 5.2.

$$\phi_y = \frac{12EI_z}{GAK_yL^2}, \qquad \phi_z = \frac{12EI_y}{GAK_zL^2} \tag{5.2}$$

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{5.3}$$

$$k_{11} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{(1+\phi_y)L^3} & 0 & 0 & 0 & \frac{6EI_z}{(1+\phi_y)L^2} \\ 0 & 0 & \frac{12EI_y}{(1+\phi_z)L^3} & 0 & -\frac{6EI_y}{(1+\phi_z)L^2} & 0 \\ 0 & 0 & 0 & \frac{GKL}{L} & 0 & 0 \\ 0 & 0 & -\frac{6EI_y}{(1+\phi_z)L^2} & 0 & \frac{(4+\phi_z)EI_y}{(1+\phi_z)L} & 0 \\ 0 & \frac{6EI_z}{(1+\phi_y)L^2} & 0 & 0 & 0 & \frac{(4+\phi_y)EI_z}{(1+\phi_y)L} \end{bmatrix} \tag{5.4}$$

$$k_{12} = \begin{bmatrix} -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{(1+\phi_y)L^3} & 0 & 0 & 0 & \frac{6EI_z}{(1+\phi_y)L^2} \\ 0 & 0 & -\frac{12EI_y}{(1+\phi_z)L^3} & 0 & -\frac{6EI_y}{(1+\phi_z)L^2} & 0 \\ 0 & 0 & 0 & -\frac{GKL}{L} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{(1+\phi_z)L^2} & 0 & \frac{(2-\phi_z)EI_y}{(1+\phi_z)L} & 0 \\ 0 & -\frac{6EI_z}{(1+\phi_y)L^2} & 0 & 0 & 0 & \frac{(2-\phi_y)EI_z}{(1+\phi_y)L} \end{bmatrix} \tag{5.5}$$

$$k_{21} = k_{12}^T \tag{5.6}$$

$$k_{22} = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{(1+\phi_y)L^3} & 0 & 0 & 0 & -\frac{6EI_z}{(1+\phi_y)L^2} \\ 0 & 0 & \frac{12EI_y}{(1+\phi_z)L^3} & 0 & \frac{6EI_y}{(1+\phi_z)L^2} & 0 \\ 0 & 0 & 0 & \frac{GKL}{L} & 0 & 0 \\ 0 & 0 & \frac{6EI_y}{(1+\phi_z)L^2} & 0 & \frac{(4+\phi_z)EI_y}{(1+\phi_z)L} & 0 \\ 0 & -\frac{6EI_z}{(1+\phi_y)L^2} & 0 & 0 & 0 & \frac{(4+\phi_y)EI_z}{(1+\phi_y)L} \end{bmatrix} \tag{5.7}$$

The mass matrix ($\mathbf{M}$) is a three-dimensional lumped mass matrix described in Equation 5.8. This requires the total mass of the beam (m), which is described by the equation $m = \rho l A$ where ρ is the mass density of the material, l is the length of the beam, and A is the cross-sectional area of the beam. The mass moment of inertia tensor in the local coordinate system contains the principal moments of inertia I_{xx}, I_{yy}, and I_{zz}.

$$\mathbf{M} = \frac{1}{2}\begin{bmatrix} M_{11} & 0 \\ 0 & M_{22} \end{bmatrix}, \quad M_{11} = M_{22}\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \tag{5.8}$$

The annulus cross-section uses the mass moment of inertia tensor for a thick-walled cylindrical tube. The equation for the mass moment of inertia tensor for the annulus is shown in Equation 5.9. The mass of the beam (m), inner radius (r_1), and outer radius (r_2) are the required parameters.

$$\mathbf{I} = \begin{bmatrix} \frac{1}{2}m(r_2^2 + r_1^2) & 0 & 0 \\ 0 & \frac{1}{12}m(3(r_2^2 + r_1^2) + L^2) & 0 \\ 0 & 0 & \frac{1}{12}m(3(r_2^2 + r_1^2) + L^2) \end{bmatrix} \tag{5.9}$$

The I-beam element uses mass moment of inertia tensors of three rectangular cuboids assembled using the parallel-axis theorem. The equation for the mass moment of inertia tensor of a rectangular prism is shown in Equation 5.10 [23]. The mass of the beam (m), length (l), height (h), and width (b) are the required parameters. Equation 5.11 describes the rotation of the moment inertia tensor into the global coordinate space. Equation 5.12 describes how the parallel axis theorem is used to find the mass moment of inertia of the I-beam. Mass moments of inertia are calculated for the bottom flange (I'_{bot}), web (I'_{web}) and top flange (I'_{top}). The vector from the center of the beam cross-section to the center of the top rectangular prism cross-section is given by $d_{\mathbf{top}}$. The vector from the center of the beam cross-section to the center of the bottom rectangular prism cross-section is given by $d_{\mathbf{bottom}}$. The identity matrix is I_3.

$$\mathbf{I} = \begin{bmatrix} \frac{1}{12}m(h^2 + b^2) & 0 & 0 \\ 0 & \frac{1}{12}m(h^2 + L^2) & 0 \\ 0 & 0 & \frac{1}{12}m(L^2 + b^2) \end{bmatrix} \tag{5.10}$$

$$\mathbf{I}' = \mathbf{T}\,\mathbf{I}\,\mathbf{T}^T \tag{5.11}$$

$$\mathbf{I} = \mathbf{I'}_{\text{top}} + \mathbf{I'}_{\text{web}} + \mathbf{I'}_{\text{bot}} + m_t\left((\mathbf{d}_{\text{top}} \cdot \mathbf{d}_{\text{top}})\mathbf{I}_3 - \mathbf{d}_{\text{top}}\mathbf{d}_{\text{top}}^T\right) + m_t\left((\mathbf{d}_{\text{bot}} \cdot \mathbf{d}_{\text{bot}})\mathbf{I}_3 - \mathbf{d}_{\text{bot}}\mathbf{d}_{\text{bot}}^T\right) \tag{5.12}$$

5.1.2 UAV Slice Model

The UAV slice model is the beam model of a 1/8-symmetry model of the octocopter used for finite element analysis. The slice model is shown in Figure 5.3.

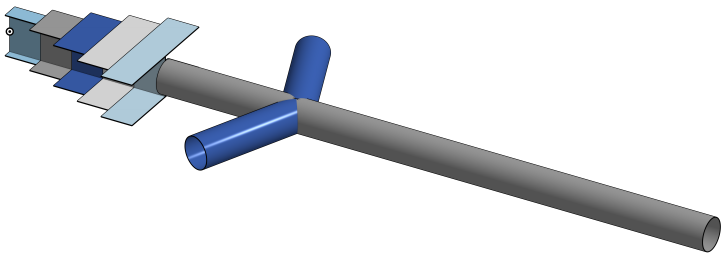


Figure 5.3: UAV Slice Beam Model Visualization

The finite element model is evaluated repeatedly by the optimization algorithm, so the slice model was created to reduce computation time. A representation of the nodes and elements of this model is shown in Figure 5.4. The model is 1/8th of the model of the entire UAV described in Section 3.1, split along the struts. All forces applied for the optimization algorithm are applied to the outer node, as that is the location of the propeller.

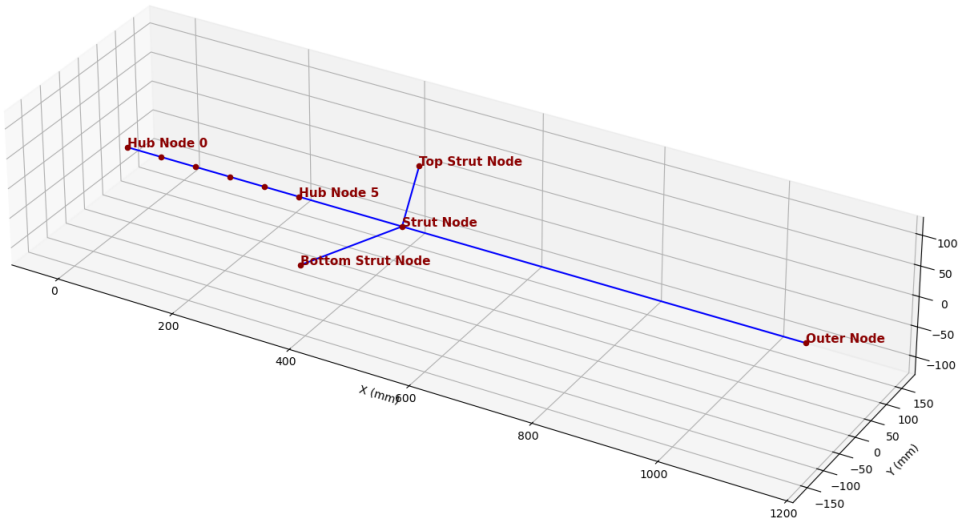


Figure 5.4: UAV Slice Node Visualization

The hub segment is represented as a series of repeated I-beam cross-sections. The process of splitting the hub into sections is demonstrated in Figure 5.5. The width of each I-beam cross-section in the slice model is the width at the centroid of the corresponding trapezoidal area of the hub's top plate. The number of hub segments is input by the user.

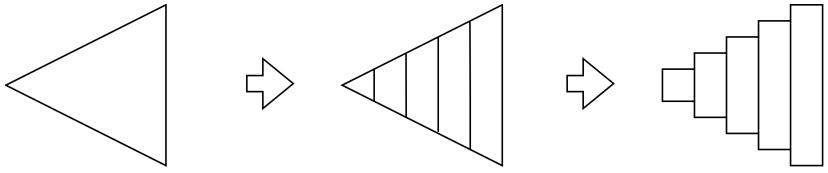


Figure 5.5: Hub Segmentation Example

The 1/8th-symmetric beam model of the UAV is generated automatically by Python code from the design variables, material properties associated with all the components, the wheelbase calculation, and hub height parameter.

5.1.3 Constraints

The beam system requires constraints to be applied to solve the vector of known displacements. Each constraint references a node where the constraint is applied. A type and a value are required for each constraint. The constraint type refers to which direction or rotation the node should be constrained in. The constraint value is the value to which the direction or rotation is set to.

There is also an option to set a local coordinate system to orient the constraint. This requires a vector in the global coordinate system, which determines the x-direction of the constraint, and a vector that determines the y-direction. The direction vectors of the resulting coordinate system are the transformation matrix. If the constraint is applied in the global coordinate system, the transformation matrix is set to the identity matrix.

The center of the slice model is fixed in all 6 degrees of freedom as all forces are at either the center of the UAV or at the end of the arm, so the forces at the center can be modeled as a reaction. For the constraints at the struts, the constraint transformation was used to create a symmetric constraint about the vector along the struts. Each symmetric constraint is at the end of the strut. Therefore, the only modes analyzed with the slice model are the symmetric modes. However, the 1/8-symmetric constraints could be changed to asymmetric to resolve the asymmetric modes. Each of the nodes in the hub are constrained symmetrically about the y-plane. Table 5.1 shows each constraint in the UAV slice model from Figure 5.4. The x-direction of the coordinate system of the bottom strut constraint is r_{bot}, which is equal to the vector from the strut node to the bottom strut node. The x-direction of the coordinate system of the top strut constraint is r_{top}, which is equal to the vector from the strut node to the top strut node. The parameters r_{bot} and r_{top} are shown in Figure 5.6.

Table 5.1: Constraints in the UAV Slice Model

Location	Constraint	Direction Vector	k-node
Hub node 0	Encastre	–	–
Upper strut Node	X symmetry	r_{top}	$[0\ 0\ 1]^T$
Bottom strut node	X symmetry	r_{bot}	$[0\ 0\ 1]^T$
Hub node 1	Y symmetry	–	–
Hub node 2	Y symmetry	–	–
Hub node 3	Y symmetry	–	–
Hub node 4	Y symmetry	–	–
Hub node 5	Y symmetry	–	–

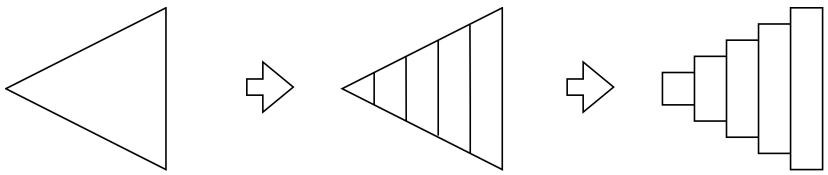


Figure 5.6: I-Beam Cross-section

5.2 Beam Stresses

Once the global displacement vector is calculated, the vector is subdivided into sections corresponding to the different elements that are then transformed back into the local coordinate space so that the C^1 continuous hermitian interpolation functions can be calculated [8]. Equations 5.13 - 5.17 describe the interpolation functions. The length is L, the position along the beam is x, and the displacement at a given point on the beam is $u(x)$. The displacement and rotation at $x = 0$ are u_1 and θ_1. The displacement and rotation at $x = L$ are u_2 and θ_2.

$$u(x) = N_1(x)\,u_1 + N_2(x)\,\theta_1 + N_3(x)\,u_2 + N_4(x)\,\theta_2 \tag{5.13}$$

$$N_1(x) = 1 - 3\frac{x^2}{L^2} + 2\frac{x^3}{L^3} \tag{5.14}$$

$$N_2(x) = x\left(1 - 2\frac{x}{L} + \frac{x^2}{L^2}\right) \tag{5.15}$$

$$N_3(x) = 3\frac{x^2}{L^2} - 2\frac{x^3}{L^3} \tag{5.16}$$

$$N_4(x) = x\left(-\frac{x^2}{L^2} + \frac{x}{L}\right) \tag{5.17}$$

Two different shape functions are generated, one representing the displacement in the y-direction and another representing the displacement in the z-direction. The shape functions allow for displacement and rotations to be interpolated at any point within the beam.

Using the shape functions, the bending stress can be calculated at any point along the beam. Equation 5.18 describes the relationship between displacement and strain. The shape functions are assembled into $\mathbf{B}$. The bending strain is $\epsilon_{\text{bending}}$. The formula used to calculate stress is shown in Equation 5.20. The maximum nominal bending stress in the beam in a given bending axis is σ_{bending}). The derivative of rotation along the beam axis is $\frac{d\theta(x)}{dx}$ and is obtained from the shape functions. E is the Young's Modulus. h is the radius of the circumscribed cross-section. This causes an overestimation of the stress and was selected to simplify the calculation. This causes the estimated stress in the I-beams to be higher than reality. The bending stress is calculated for both bending axes, and the resulting maximum

stress is taken by taking the Euclidean norm of both vectors.

$$\epsilon_{\text{bending}} = \mathbf{B}\,\mathbf{u} \tag{5.18}$$

$$\sigma_{\text{bending}} = E\epsilon_{\text{bending}} \tag{5.19}$$

$$\sigma_{\text{bending}} = -hE\frac{d\theta(x)}{dx} \tag{5.20}$$

5.3 Solvers

The FEA model was created with three separate solvers. The static solver, which is used to calculate the response given a static loading, the dynamic solver, which calculates the dynamic response of the UAV slice model's nodes with a force at a given frequency, and the transient solver, which calculates the transient dynamic response given a time-changing force.

5.3.1 Static

The static solver calculates the reactions, displacements, and rotations given a specific set of point loads and moments in static equilibrium. The scipy.linalg.solve function is used to solve for unknown displacements. The displacements (U_x, U_y, and U_z), rotations (θ_x, θ_y, and θ_z), forces (F_x, F_y, and F_z), and moments (M_x, M_y, and M_z) for the entire beam system are calculated from the stiffness matrix, constraints, and known forces.

The shape functions and stresses for each beam in the state of static stress are calculated using the formulation discussed in Section 5.2. Figure 5.7 shows the data flow diagram for the static solver.

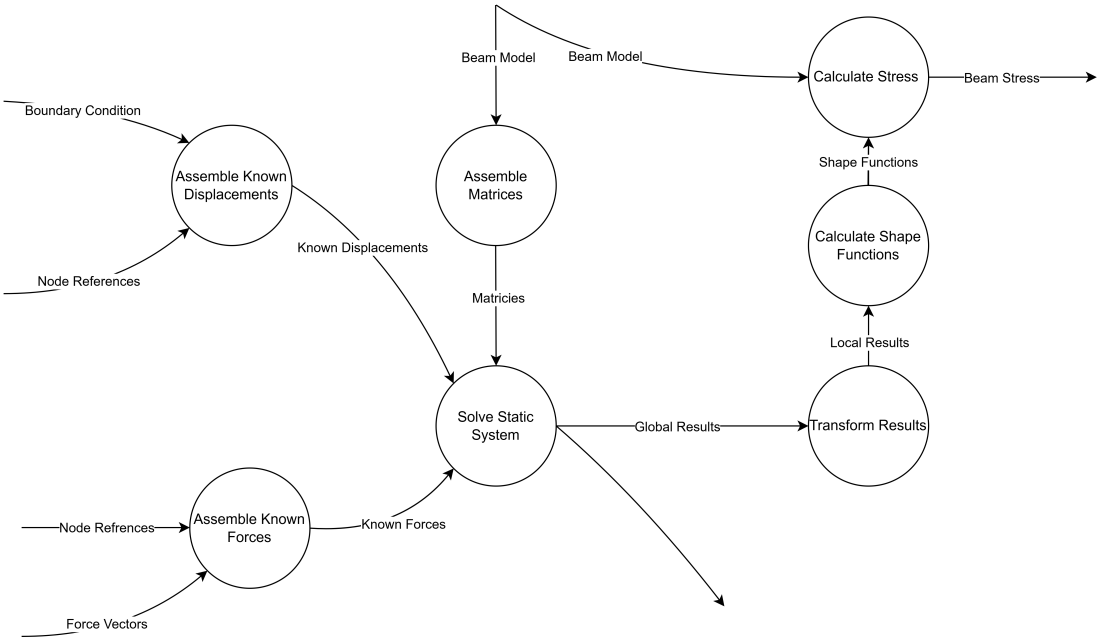


Figure 5.7: DFD of the Static Solver

5.3.2 Dynamic

The dynamic solver solves the amplitude of reactions, displacements, and rotations given a set of harmonic loads and moments. The solver analyzes steady-state harmonic behavior to compute the frequency response of the system. The constraints are static and have a frequency of zero. Each force and moment applied is assumed to have the same frequency and phase. The dynamic stiffness matrix is calculated from the stiffness and mass matrices using Equation 5.21. This equation assumes damping is zero. $\mathbf{K_d}$ is the dynamic stiffness matrix. $\mathbf{K}$ is the static stiffness matrix and $\mathbf{M}$ is the mass matrix. ω is the frequency of the loads and moments. Equation 5.22 describes the relationship between the dynamic stiffness

matrix, displacement, force, and time.

$$\mathbf{K_d} = \mathbf{K} - \omega^2 \mathbf{M} \tag{5.21}$$

$$\mathbf{K}_d \, \mathbf{u} \, e^{i\omega t} = \mathbf{F} \, e^{i\omega t} \tag{5.22}$$

The scipy.linalg.solve function is used with the dynamic stiffness matrix to solve for unknown displacement amplitudes. The harmonic stress amplitude is calculated for each element using the formulation in Section 5.2. Figure 5.8 shows the data flow diagram for the dynamic solver.

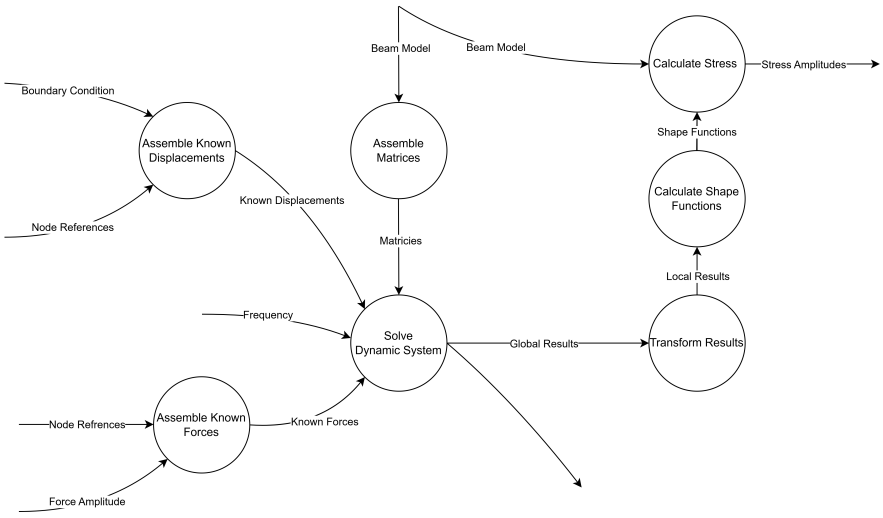


Figure 5.8: DFD of the Dynamic Solver

5.3.3 Transient

The transient solver finds the reactions, displacements, and rotations over time, given a user-defined time-varying force and moment vectors. The relationship between force and the kinematics of the system is described in Equation 5.23. The force vector is $f(t)$. The

mass matrix is **M**. The Newmark integration method is used in order to solve the dynamic stiffness matrix shown in Equation 5.21. The Newmark integration method is a semi-implicit integration method that is used for structural dynamics. Equation 5.24 and Equation 5.25 are the update equations for the Newmark integration method. The inputs are a time step interval (ΔT), the time step count (T_n), and α and δ as constants for Newmark integration. The time step interval is a constant. The displacement at a given time step is u_n.

$$\mathbf{M}\,\ddot{u}(t) + \mathbf{K}\,u(t) = f(t) \tag{5.23}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t\,\dot{\mathbf{u}}_n + \Delta t^2\left[\left(\frac{1}{2} - \beta\right)\ddot{\mathbf{u}}_n + \beta\,\ddot{\mathbf{u}}_{n+1}\right] \tag{5.24}$$

$$\dot{\mathbf{u}}_{n+1} = \dot{\mathbf{u}}_n + \Delta t\,[(1-\gamma)\ddot{\mathbf{u}}_n + \gamma\,\ddot{\mathbf{u}}_{n+1}] \tag{5.25}$$

Each force added to the system is described as a vector of generalized displacements and rotations at each time step. The Newmark integration method is used to estimate the displacement, rotations, and reactions at each timestep. There is a constant duration between each timestep, which is determined by the timestep interval. The stresses are calculated for each timestep. Figure 5.9 shows the data flow diagram for the transient solver.

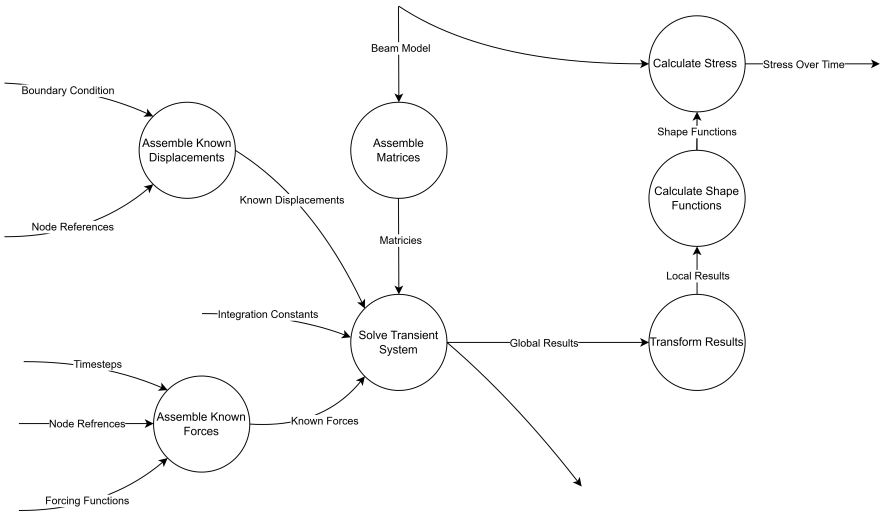


Figure 5.9: DFD of the Transient Solver

5.4 Mass Analysis

The total mass of the frame (m_{frame}) of the UAV was calculated geometrically, by calculating the volume of each beam and multiplying it by the density of the associated material. The mass of each battery and the payload were added to obtain the total mass of the vehicle (m_{total}).

The mass moment of inertia tensor for the UAV frame is calculated. The mass moment of inertia is used in the objective function of the optimization problem. The calculation only takes into account the mass moment of inertia of the UAV frame, and not the mass moment of inertia of the batteries or the payload.

The mass moment of inertia of the system is calculated by summing the mass moment of inertia of each beam in the finite element model. The calculation for mass moment of inertia uses the full UAV model and has the hub subdivided into individual I-beams. The principal moments of inertia are calculated by taking the eigenvalues of the tensor. The eigenvalues

are solved using np.linalg.eig. The mean of all the principal moments is calculated for the objective function.

Chapter 6

Optimization

This chapter describes the optimization problem formulation and how the optimization architecture was designed. It describes the bounds, optimization function, and constraints.

The optimization problem is formulated as follows:

$$\min F(X) \tag{6.1}$$

$$X = (d_a, t_a, d_s, t_s, r_s, t_f, t_w, r_h) \tag{6.2}$$

subject to constraints:

$$g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \ldots, J \tag{6.3}$$

$$x_i^L \leq x_i \leq x_i^U, \quad i = 1, 2, \ldots, N \tag{6.4}$$

The SLSQP algorithm was selected for this optimization problem. Scipy.optimize.minimize algorithm's SLSQP method is used [22]. The error tolerance is set to 1e-6.

If the frequency or fatigue constraints are active, SLSQP can not be used due to the nonlinear

discontinuous nature of these constraints. Instead, the differential evolution algorithm is used with a tolerance of 1e-6, a population size of 15, a mutation constant between 0.5 and 1, and a recombination constant of 0.7. Scipy's scipy.optimize.differential_evolution method is used.

To save on computation time, constraints can be made active or inactive by the solver, as well as setting design variables to constants if the boundaries are active. The next sections describe the formulation of the objective function, the bounds on design variables, and the constraints.

6.1 Objective Function

The objective function is formulated in order to minimize both the mass and the mass moment of inertia of the UAV frame. The objective function is described in Equation 6.5. The mass of the frame given a set of design variables is M_{frame}. The mean principal moment of inertia of the frame for a given set of design variables is $I_{\text{frame,mean}}$. Scaling factors S_i and S_m were added to keep both mass and mass moment around the same order of magnitude. The scaling factor for mass (S_m) is set to 10^3 and the scaling factor for mass moment of inertia is 10^{-3}. The weighting factor (W_m) for mass is 2/3, and the weighting factor for mass moment of inertia (W_i) is 1/3.

$$F(X) = \frac{W_m M_{\text{frame}}(X)}{S_m} + \frac{W_i I_{\text{frame,mean}}(X)}{S_i} \tag{6.5}$$

The data flow diagram showing how the objective function is calculated is shown in Figure 6.1. The dimensions of the drone are calculated using a set of design variables and parameters. The mass is calculated geometrically from the geometry and material properties.

The beam model of the full UAV with the segmented hub from the slice model described in Subsection 5.1.2 is created. The mean principal mass moment of inertia ($I_{\text{frame,mean}}$) for this beam model is calculated using the method discussed in Section 5.4.

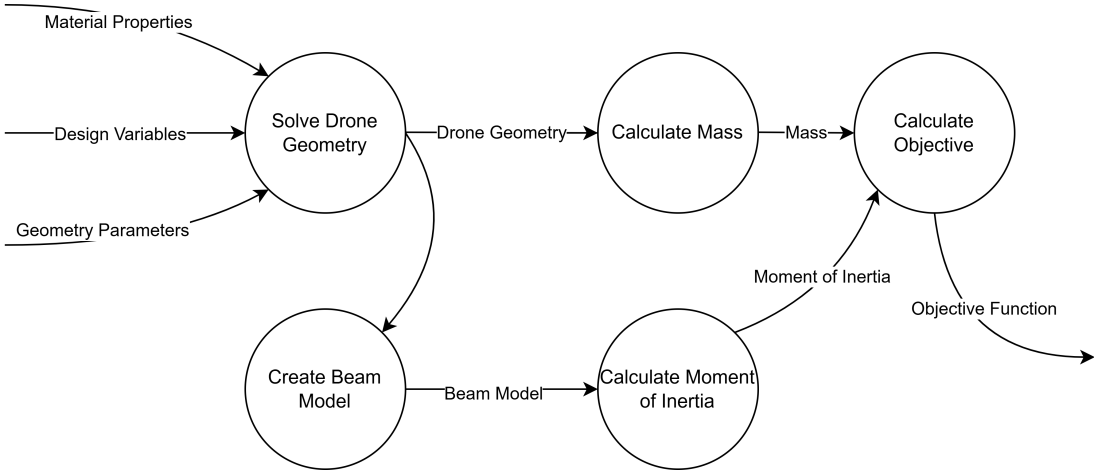


Figure 6.1: DFD of the Objective Function

6.2 Constraints

This section describes the constraints applied to the UAV model. There are upper and lower bounds on each of the design variables, a displacement constraint to limit the maximum displacement while the UAV is in hover, a constraint to ensure the stress is below yield strength, a frequency constraint to prevent vibration, an energy constraint to ensure that the UAV has enough energy to complete the mission, and a fatigue constraint to prevent long-term damage to the vehicle.

6.2.1 Boundaries

Each design variable has a lower and upper bound applied to it. Table 6.1 describes each bound. The arm diameter (d_a), arm thickness (t_a), strut diameter (d_s), and strut thickness

(t_s) bounds were obtained through the minimum and maximum pipe diameter and thickness for braided carbon fiber found on AllRED's Dragonplate Website [1]. The hub flange thickness (t_f) and hub web thickness (t_w) bounds were taken from the minimum and maximum thickness of carbon fiber sheets. The maximum bounds on hub radius (r_h) and strut distance from the center (r_s) are the distance from the center of the UAV to the propellers, so the struts and hub stay within the frame. The minimum constraint of 197.5mm for hub radius (r_h) and strut distance (r_s) is the minimum hub radius for a circle to circumscribe both batteries with their given positions and dimensions.

Table 6.1: Optimization Bounds

Variable	**Description**	**Lower Bound (mm)**	**Upper Bound (mm)**
d_a	Arm outer diameter	9	50
t_a	Arm wall thickness	0.254	3.175
d_s	Strut outer diameter	9	50
t_s	Strut wall thickness	0.254	3.175
r_s	Strut distance from hub center	197.5	1127.3
t_f	Hub flange thickness	0.7	12.7
t_w	Hub web thickness	0.7	12.7
r_h	Hub radius	197.5	1127.3

6.2.2 Displacement

The displacement constraint limits the maximum allowable displacement while the UAV is in hover. The data flow diagram of calculating the stress constraint is shown in Figure 6.2. The drone geometry is the set of all relevant coordinates, material properties, and cross-section geometry for the UAV. From the drone geometry, the mass of the frame is calculated. The

force used is the force on the propeller from the UAV in hover. This force points upward and is on the outer node of the segment model.

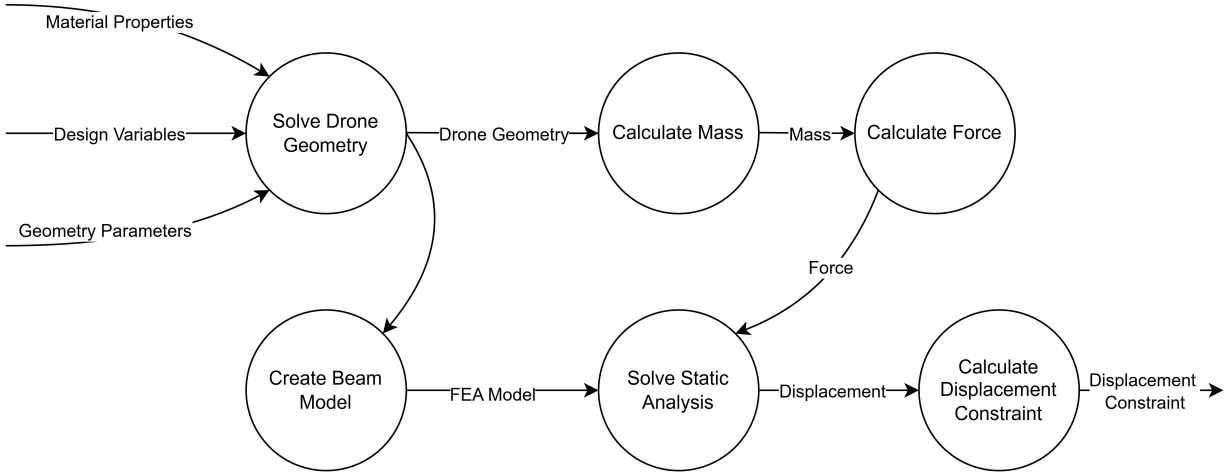


Figure 6.2: DFD for the Displacement Constraints

The beam model is the UAV slice model discussed in Subsection 5.1.2. The force is applied to the outer node and points upward. This force is equal to the force necessary for the UAV to hover ($F_{\text{thrust,hover}}$) as described in Equation 4.7. The static solver discussed in Subsection 5.3.1 was used to calculate the displacement (d_{outer}) at the outer node of the 1/8-segment model. The formula for the displacement constraint is shown in Equation 6.6. The maximum allowable displacement is $d_{allowable}$.

$$g_1(X) = d_{\text{outer}} - d_{\text{allowable}} \leq 0 \tag{6.6}$$

6.2.3 Stress

The stress constraint limits the maximum allowable stress while the UAV is in hover. The data flow diagram of calculating the displacement constraint is shown in Figure 6.3.

The beam model is the UAV slice model discussed in Subsection 5.1.2. The force is applied

to the outer node and points upward. This force is equal to the force necessary for the UAV to hover ($F_{\text{thrust,hover}}$) as described in Equation 4.7. The moment (τ) from motor torque described in Subsection 4.3.6 is applied at the outer node. The static solver discussed in Subsection 5.3.1 was used to calculate the stress (σ_{center}) at the hub node 0 of the 1/8-segment model. This node is at the center of the UAV and was selected because it is likely the location of maximum stress.

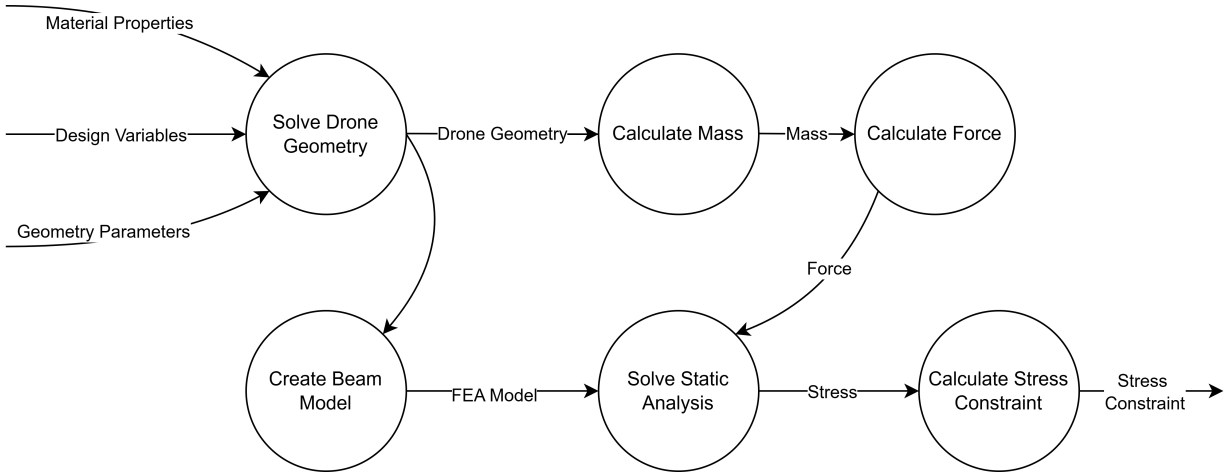


Figure 6.3: DFD for the Stress Constraints

Equation 6.7 shows the calculation for the allowable stress. The tension yield strength of the material is σ_{yield}. The factor of safety associated with the stress constraint is FOS. The factor of safety is a parameter of the analysis. Equation 6.8 shows the calculation of the constraint.

$$\sigma_{\text{allowable}} = \frac{\sigma_{\text{yield}}}{\text{FOS}} \tag{6.7}$$

$$g_2(X) = \sigma_{center} - \sigma_{\text{allowable}} \leq 0 \tag{6.8}$$

6.2.4 Frequency

The objective of the frequency constraint is to ensure that the frequency of excitation developed by the propeller during the mission does not result in a high displacement amplitude in the vehicle. This is accomplished by calculating the maximum admittance of the frequency response function over a range of values. The admittance is the amplitude of the displacement at the outer node per unit force.

The data flow diagram for the frequency is shown in Figure 6.4. First, the beam model of the UAV slice discussed in Subsection 5.1.2 is created. The dynamic model is run repeatedly with frequencies over a range of values. This range is calculated by from multiplying the maximum and minimum values of the angular frequency over the mission profile calculated in Equation 4.9 by two. These are the maximum and minimum excitation frequencies of the propellers. The rev-up and rev-down sections are not considered for this calculation. A user-defined number of points is sampled across the range between the minimum and maximum excitation frequencies. The dynamic model is run at a frequency on each point along that range. The amplitude of the force is the force required to keep the UAV in hover ($F_{\text{thrust,hover}}$) described in Equation 4.7.

The maximum displacement amplitude between all run dynamic models ($d_{amplitude}$) is found. Equation 6.9 is the formulation of the constraint. The allowable admittance frequency response function is represented $Y_{\text{allowable}}$. The amplitude of the displacement at the outer node with the given forcing function is $d_{\text{amplitude,outer}}$.

$$g_3(X) = \frac{d_{\text{amplitude,outer}}}{F_{\text{thrust,hover}}} - Y_{\text{allowable}} \leq 0 \tag{6.9}$$

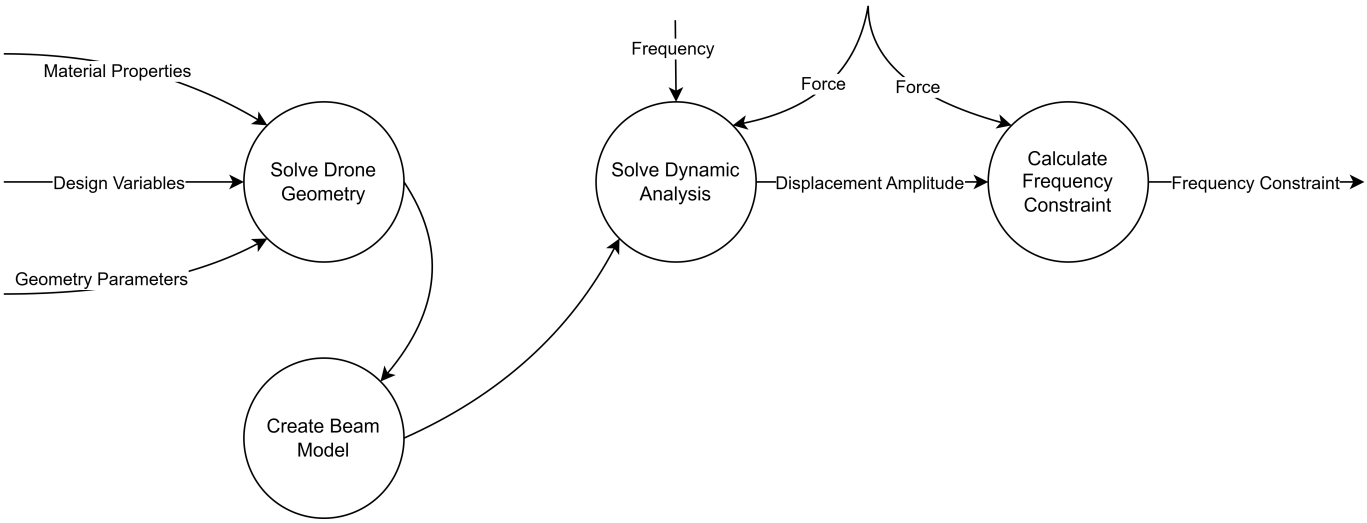


Figure 6.4: DFD of the Transient Solver

6.2.5 Damage

The damage constraint is imposed in order to limit damage caused by the vibrations in the vehicle. The data flow diagram of the damage calculation is shown in Figure 6.5. The mission profile, total mass, and surface area are used to calculate the thrust function ($F_{\text{thrust,dynamic}}$) shown in Equation 4.6. The angular frequency function is calculated with the thrust function by applying Equation 4.9. The thrust function is then used to calculate the perturbation force ($F_{\text{perturbation}}(t)$) by applying Equation 4.11. The beam model used is the slice model discussed in Subsection 5.1.2. Both the perturbation force ($F_{\text{perturbation}}(t)$) and thrust force ($F_{\text{thrust,dynamic}}$) are upward forces and are applied to the outer node of the slice model.

The number of time-steps for the transient solver is set by the user. The constants used for Newmark-beta integration are $\beta = 0.25$ and $\gamma = 0.5$. The results for the bending stress are taken for each timestep at hub node 0.

Once the stress function is obtained, the rainflow counting algorithm is applied to calculate the damage. The Rainflow module in the PyLife library [5] is used and it uses the four-point

Rainflow-counting algorithm to calculate constant amplitude stress reversals from a time-varying load. The total damage accumulated over the mission is calculated from the stress reversals using the Woehler curve data and the LoadCollective class in the PyLife library. The Pylife library uses the stress-life model to estimate the damage at given stress levels.

Equations 6.10 - 6.15 describe the damage calculation given a time-varying stress [9]. The stress ratio is represented by R in Equation 6.10. The maximum and minimum stress for a given stress cycle are $\sigma_{\max,i}$ and $\sigma_{\min,i}$. Equations 6.11 and 6.12 calculate the reversed stress amplitude for a given stress cycle ($\sigma_{ar,i}$). The number of cycles (N) at the given state of stress is calculated using the relation in Equation 6.13. The fatigue strength exponent is b. The fatigue strength coefficient is σ'_f. Equation 6.14 calculates the damage for a given cycle (D_i). The number of cycles at a given stress state is N_i. The number of cycles until failure is N_{fi}. Miner's rule for cumulative fatigue damage is used as shown in Equation 6.15. D is the cumulative damage taken in a single mission.

$$R_i = \frac{\sigma_{\min,i}}{\sigma_{\max,i}} \tag{6.10}$$

$$\sigma_{a,i} = \frac{1 - R_i}{2}\,\sigma_{\max,i} \tag{6.11}$$

$$\sigma_{ar,i} = \sqrt{\sigma_{\max,i}\,\sigma_{a,i}} \tag{6.12}$$

$$N_{fi} = \frac{1}{2}\left(\frac{\sigma_{ar,i}}{\sigma'_f}\right)^{\frac{1}{b}} \tag{6.13}$$

$$D_i = \frac{N_i}{N_{fi}} \tag{6.14}$$

$$D = \sum_i D_i \tag{6.15}$$

One limitation of this analysis is that fatigue material data is limited to aluminum 7075, as fatigue data for the carbon fiber composite would need to be obtained experimentally. Equation 6.16 describes the constraint. B_f is the blocking factor.

$$g_4(X) = D\,B_f - 1 \leq 0 \tag{6.16}$$

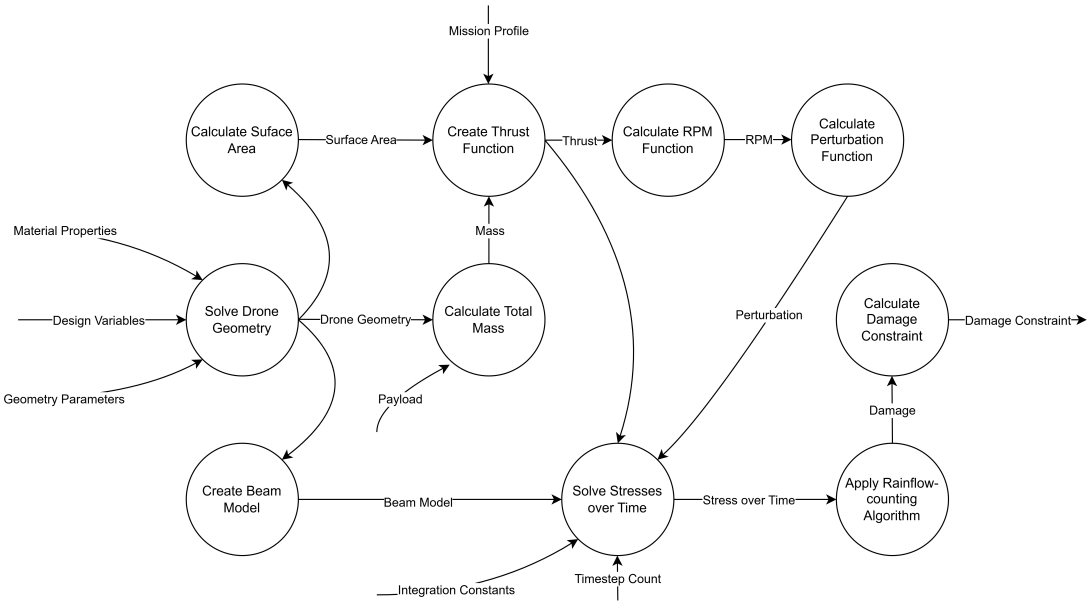


Figure 6.5: DFD of the Damage Solver

6.2.6 Energy

The objective of this constraint is to keep battery energy above desired levels for the duration of the mission. The data flow diagram for the energy constraint is shown in Figure 6.6. The energy constraint uses the energy consumption calculation in Subsection 4.3.4 to determine the total energy consumed during the mission. Equation 6.18 and Equation 6.17 describe the energy constraint. The fraction of energy spent is equal to $E_{\text{used,fraction}}$. The total energy used during the mission (E_{used}) is equal to $E_{capacity} - E_{remaining}(t_{\text{final}})$. The time at the end of the mission profile is t_{final}. The energy capacity of the battery is $E_{capacity}$. $E_{\text{remaining,fraction}}$ is a parameter that is equal to the fraction of energy remaining that is desired after landing.

$$E_{\text{used,fraction}} = \frac{E_{\text{used}}}{E_{\text{capacity}}} \tag{6.17}$$

$$g_5(X) = E_{\text{used,fraction}} + E_{\text{remaining,fraction}} - 1 \leq 0 \tag{6.18}$$

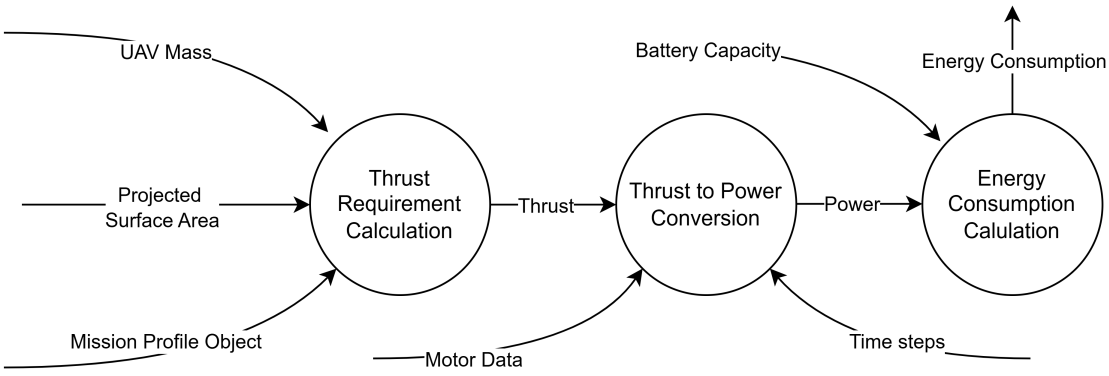


Figure 6.6: DFD of the Energy Constraint Calculation

Chapter 7

Results

This chapter discusses the results generated. The framework designed in this thesis is flexible and allows the user to rapidly test various design configurations. The study initially applied a constraint on deflection of 50mm or approximately 4.43% deflection per span. The results for this constraint are shown in Section 7.3.

The 50mm deflection constraint was determined to allow for excessive deformation due to the high deflection peaks during the transient analysis and the excessively thin UAV arms. The design was iterated to have 10mm of deflection based on this analysis. The results for the 10mm section are shown in Section 7.4.

7.1 Initial Guess

The initial guess used for optimization analysis is shown in Table 7.1. The same initial guess was used for both the 50mm deflection and 10mm deflection designs.

Table 7.1: Initial Design Variable Values

Variable	**Description**	**Initial Guess (mm)**
d_a	Arm outer diameter	50
t_a	Arm wall thickness	2.5
d_s	Strut outer diameter	40
t_s	Strut wall thickness	0.254
r_s	Strut distance from hub center	475
t_f	Hub flange thickness	3
t_w	Hub web thickness	2
r_h	Hub radius	300

7.2 Constants

Table 7.2 shows what the constants for the constraints discussed in Section 6.2 are set to for the 50mm deflection design. The constraint formulation that each constant is part of is shown in the first column. In the 10mm deflection case, the allowable deflection limit is changed to 10 mm. The stress factor of safety selected is 4.

Table 7.2: Constraint Constants for 50mm deflection design

Constraint	Description	Variable	Value
Displacement	Allowable deflection limit	$d_{\text{allowable}}$	50 mm
Stress	Allowable stress	$\sigma_{\text{allowable}}$	86.25 MPa
Frequency	Allowable admittance	$Y_{allowable}$	1.5
Energy	Remaining energy fraction	$E_{\text{remaining,fraction}}$	0.2
Fatigue	Blocking factor	B_f	10,000

Table 7.3 shows the constants associated with the objective function discussed in Section 6.1. The mass scaling factor and mass moment of inertia scaling factor are set so that both are on the order of magnitude of one within the consistent unit system described in Section 4.1. The weighing coefficients are set so that mass is valued more by the objective function than the mass moment of inertia.

Table 7.3: Objective Function Constants

Constant	Variable	Value
Mass scaling factor	s_m	10^{-3}
Mass moment scaling factor	s_I	10^3
Mass weighting coefficient	w_m	$\frac{2}{3}$
Mass moment weighting coefficient	w_I	$\frac{1}{3}$

Simulation parameters used during the optimization process are summarized in Table 7.4. The hub is subdivided into five sections for the finite element model. The hub is subdivided into twenty-five sections for mass moment calculation in the objective function. The FEA model only has five sections to save computation time. 1000 time steps are used in the numerical integration of power consumption to compute the total energy expenditure and resulting battery charge depletion over the mission profile.

During frequency constraint evaluation, the admittance response is sampled at five evenly distributed frequencies in the range of propeller frequencies. The range of propeller frequencies is from the minimum to the maximum frequency experienced by the motor forcing function. The number of transient timesteps is determined by taking the 640 second mission profile duration and multiplying double the maximum excitation frequency. This is so that each oscillation is represented.

Table 7.4: Simulation Parameters

Parameter	**Value**
Hub sections (fea)	5
Hub sections (moment)	25
Transient timesteps	115200
Energy timesteps	1000
Frequency range sample count	5

7.3 50mm Deflection Design

The beam model representation of the result with the 50mm constraint is shown in Figure 7.1. Due to the frequency and damage constraints being inactive constraints, the SLSQP optimization algorithm was used. The algorithm iterated 27 times. Only the displacement constraint was evaluated at each iteration. The time it took the optimization algorithm to run was 128 seconds.

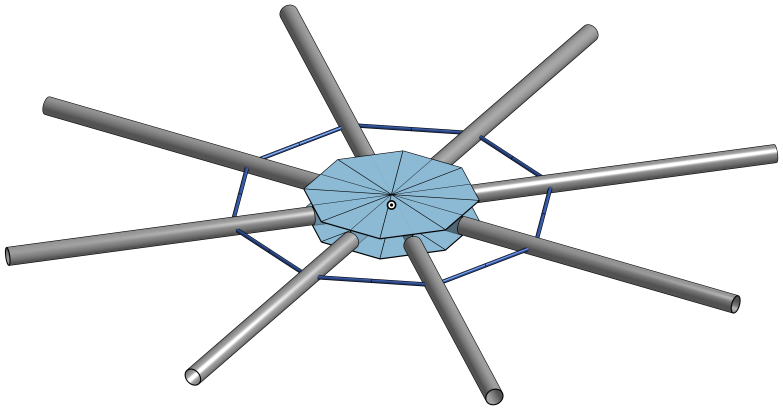


Figure 7.1: Visualization of 50mm Deflection Design

Table 7.5 shows the resulting design variables for the 50 mm deflection design. The minimum bounds for strut distance, strut thickness, hub flange thickness, and hub web thickness are active. The upper bound for the arm diameter is also active.

Table 7.5: Resulting Design Variables for 50mm Deflection Design

Parameter	Value (mm)
Arm diameter	50
Arm thickness	0.336
Strut diameter	9
Strut thickness	0.254
Strut distance	474
Hub radius	228
Hub flange thickness	0.7
Hub web thickness	0.7

Table 7.6 shows the objective function comparison between the initial guess and result for the 50mm design.

Table 7.6: Objective Function Comparison for 50mm Deflection Design

Result	Initial guess	Optimized	Percent change
Frame mass (kg)	6.27	0.916	-86
Frame mass moment of inertia ($kg \cdot m^2$)	1.24	0.183	-85
Objective	4.68	0.672	-86

Table 7.7 shows the constraint violation values. If the constraint violation values are greater than zero, the constraint is being violated. The constraint violation values are $g_1(X)$, $g_2(X)$, $g_3(X)$, $g_4(X)$, and $g_5(X)$ as described in the equations in Subsection 6.2. Displacement is the only active constraint. This shows that the design is driven by the stiffness of the UAV Frame. This design is significantly improved from the initial guess.

Table 7.7: Constraint Violation for 50mm Deflection Design

Constraint	Variable	Violation
Displacement	$g_1(X)$	-1.51E-08
Stress	$g_2(X)$	-1.12E-01
Frequency	$g_3(X)$	-9.73E-01
Damage	$g_4(X)$	-1.00E+00
Energy	$g_5(X)$	-1.29E-01

The constraint on deflection was determined to be too lenient. The arm, struts, and hub are very thin. The transient analysis done for the damage calculation yielded a maximum deflection of 90.0mm across the duration of the mission and a maximum stress of 130MPa. The 90mm of deflection along the UAV arm is 7.98% deflection per span, which was determined to be too high.

7.4 10mm Deflection Design

The beam model representation of the result with the 10mm deflection constraint is shown in Figure 7.2. Due to the frequency and damage constraints being inactive constraints, the SLSQP optimization algorithm was used. The algorithm iterated 43 times. Only the displacement constraint was evaluated at each iteration. The time it took the optimization algorithm to run was 135 seconds.

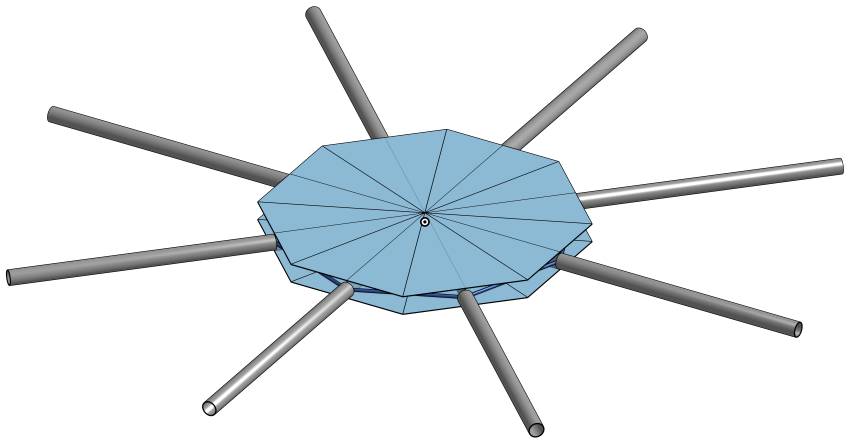


Figure 7.2: Visualization of 10mm Deflection Design

Table 7.8 shows the resulting design variables for the 10 mm deflection design. The optimal value for the arm diameter is at the upper bound. The optimal values for hub flange thickness and strut diameter are at the lower bound. The arm thickness, strut diameter, strut distance, hub radius, and hub web thickness are all not affected by the bounds on the design variables.

Table 7.8: Resulting Design Variables for 10mm Deflection Design

Design variable	Value (mm)
Arm diameter	50
Arm thickness	1.26
Strut diameter	9.04
Strut thickness	0.254
Strut distance	497
Hub radius	497
Hub flange thickness	0.7
Hub web thickness	1.18

Table 7.9 shows the objective function comparison between the initial guess and the result. Both the mass and the mass moment of inertia were decreased.

Table 7.9: Objective Function Values for 10mm Deflection Design

Result	Initial guess	Optimized	Percent difference
Frame mass (kg)	6.27	3.53	-44
Frame mass moment of inertia ($kg \cdot m^2$)	1.24	0.697	-44
Objective	4.68	2.52	-45

Table 7.10 shows the constraint violations for the 10mm design. Only the deflection constraint is active. The design for the 10mm deflection case is also stiffness-driven.

Table 7.10: Constraint Violation for 10mm Deflection Design

Constraint	Variable	Violation
Displacement	$g_1(X)$	8.45E-08
Stress	$g_2(X)$	-6.71E-01
Frequency	$g_3(X)$	-9.92E-01
Damage	$g_4(X)$	-1.00E+00
Energy	$g_5(X)$	-6.32E-02

7.4.1 Displacement and Stress

A plot of the deflection across the vehicle for the static analysis is shown in Figure 7.3. The maximum deflection is 10mm as the deflection constraint is active.

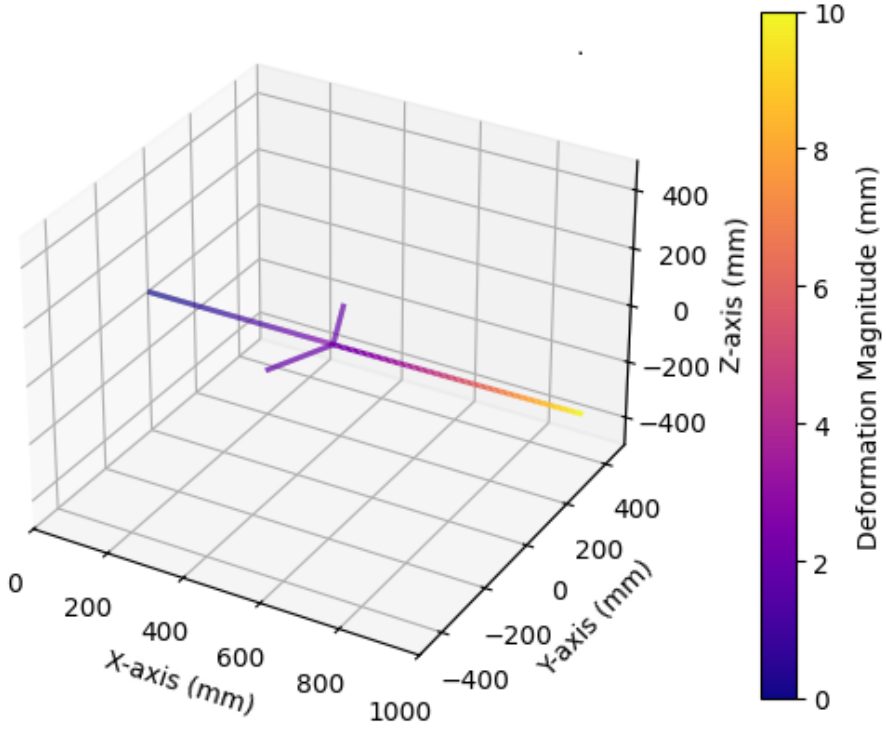


Figure 7.3: Deflection for 10mm Deflection Design

The stresses at each node within the model for the static analysis are provided in Table 7.11. The corresponding node references are shown in Figure 5.4. This shows that the point of maximum stress is at hub node 0, which is at the center of UAV model. This is partly due to the stress in the I-beam being overestimated due to using the radius of the circumscribed cross-section for the distance from the neutral axis, where the bending stress is measured (h). Another reason for this overestimation is that the hub is modeled with beam elements. If the hub were modeled with shell elements, this would allow for a symmetric constraint along the edge of the hub section. This would reduce stress in the hub due to decreased rotation. In a more refined model, the point of maximum stress concentration would likely be where the arm meets the hub (hub node 5).

Table 7.11: Stress Table for 10mm Deflection Design

Element Name	Start Node	Start Node Stress (MPa)	End Node	End Node Stress (MPa)
Hub beam 0	Hub node 0	40.8	hub node 1	35.3
Hub beam 1	Hub node 1	28.7	hub node 2	26.1
Hub beam 2	Hub node 2	24.6	hub node 3	22.8
Hub beam 3	Hub node 3	21.5	hub node 4	20.1
Hub beam 4	Hub node 4	18.8	hub node 5	17.5
Center beam	Hub node 5	19.5	strut node	19.5
Outer beam	Strut node	19.5	outer node	0
Strut element top	Strut node	1.56	strut node top	1.56
Strut element bottom	Strut node	1.56	strut node bottom	1.56

A plot of the stress distribution is shown in Figure 7.4.

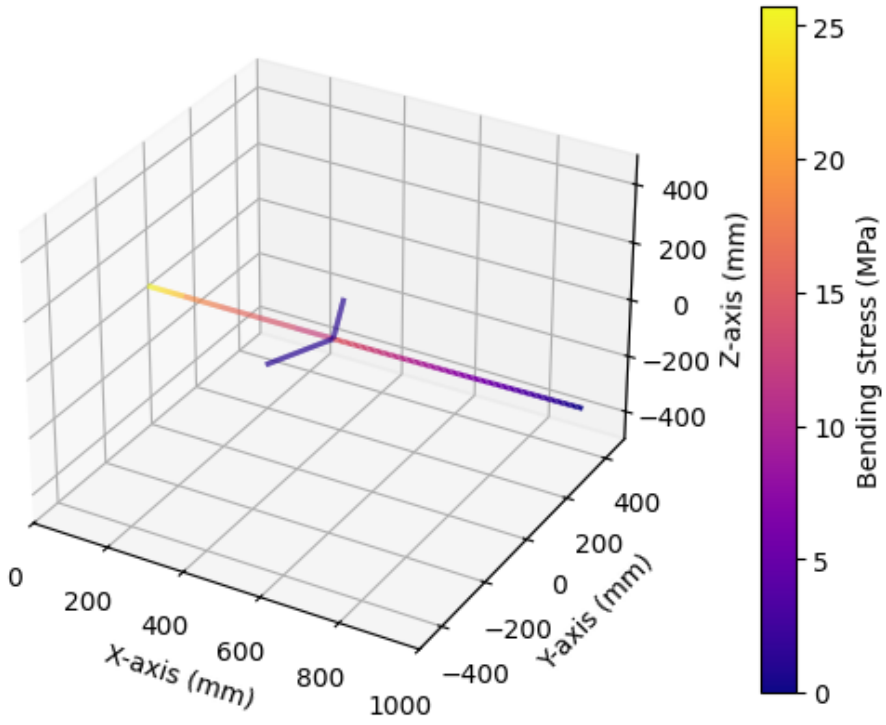


Figure 7.4: Stress Distribution for 10mm Deflection Design

7.4.2 Frequency

The first five natural frequencies are shown in Table 7.12. The frequency response from a lateral harmonic force is shown in Figure 7.5. The first, second, and third lateral modes are the peaks in the frequency response function plot.

Table 7.12: First Five Natural Frequencies for 10mm Deflection Design

Mode	Frequency (Hz)	Description
1	40.59	First transverse mode
2	64.29	First lateral mode
3	128.4	Second transverse mode
4	306.4	Second lateral mode
5	322.0	Third transverse mode

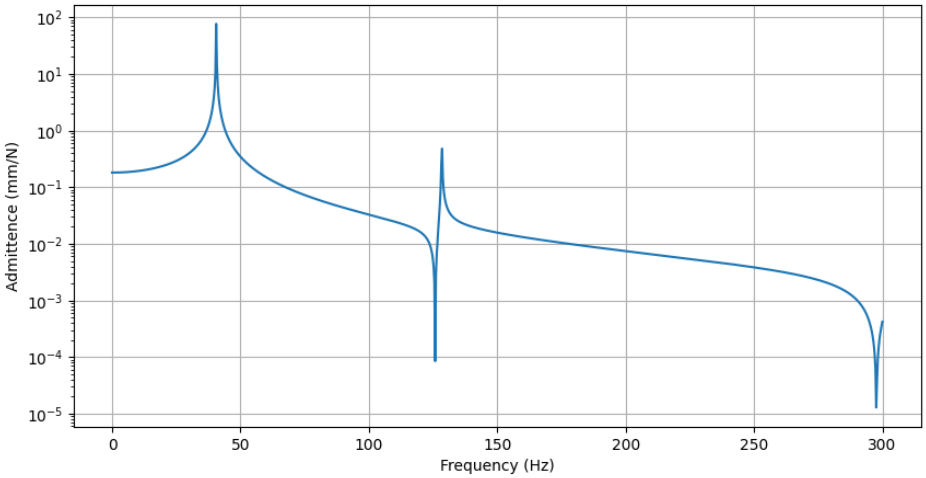


Figure 7.5: Admittance Frequency Response Function

The propulsion frequency operating range is between 81.1Hz and 90.2Hz. The frequency constraint has the potential of having a resonance peak between samples within the frequency range. Figure 7.6 shows the admittance over operating frequencies of the vehicle. There is no frequency peak between samples.

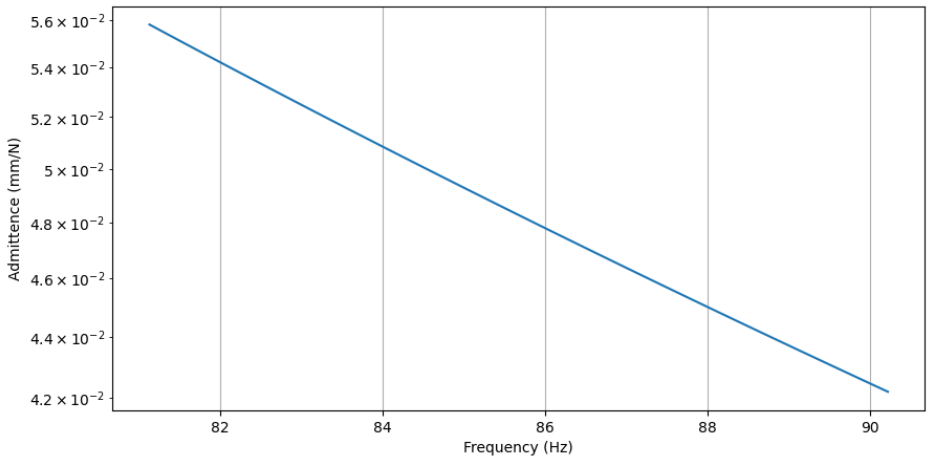


Figure 7.6: Structural Admittance Response in Operational Frequency Range

7.4.3 Energy

Results related to energy for the 10mm configuration are shown in Table 7.13. The maximum rotational speed required of the propeller to achieve the mission profile does not exceed the manufacturer's maximum motor rotational speed of 3068rpm.

Table 7.13: Energy Results for 10mm Deflection Design

	Value
Total energy capacity (kWh)	1.82
Final energy remaining (kWh)	0.48
Mission duration (s)	640
Min RPM (RPM)	806
Average RPM (RPM)	2561
Peak RPM (RPM)	2707
Allowable RPM (RPM)	3068

The energy capacity over time plot is shown in Figure 7.7. The total energy starts at (Ecapacity) 1.82kW then descends to 0.48kWh. This descent appears linear because gravity is the primary force that the thrust force must overcome.

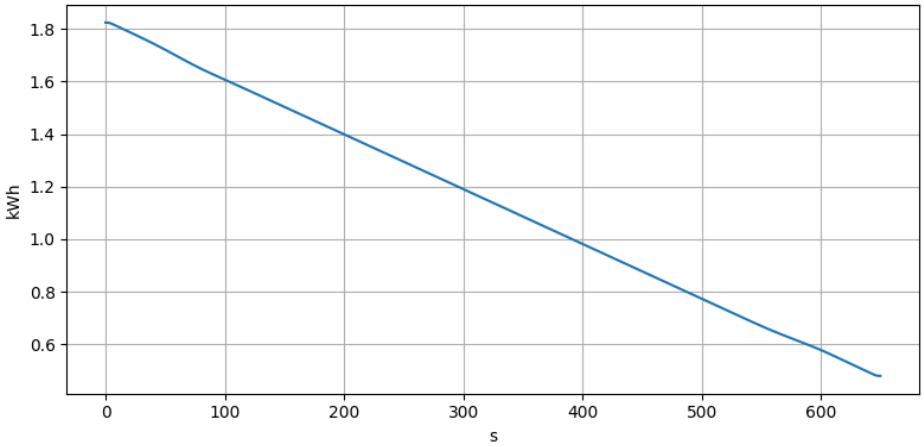


Figure 7.7: Energy Capacity over time

The power over time plot is shown in 7.8. The period of increased power usage at around 50 seconds is primarily caused by the drag force pushing the UAV downward. This occurs during the period where the UAV is rapidly ascending. The period of decreased power at around 570 seconds of usage is primarily caused by the drag force pushing the UAV upward. This occurs during the period where the UAV is rapidly descending.

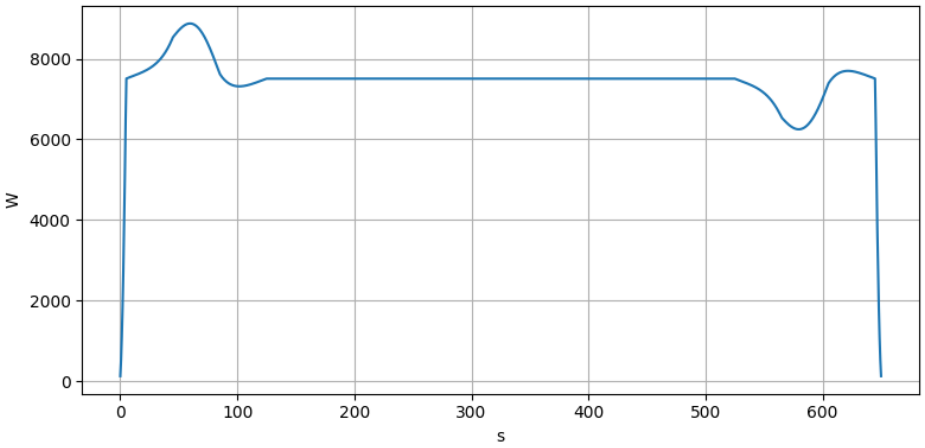


Figure 7.8: Power Over Time

Rotor speed and thrust response are shown in Figures 7.9 and 7.10. The periods of increased and decreased RPM and Thrust occur because of the drag force.

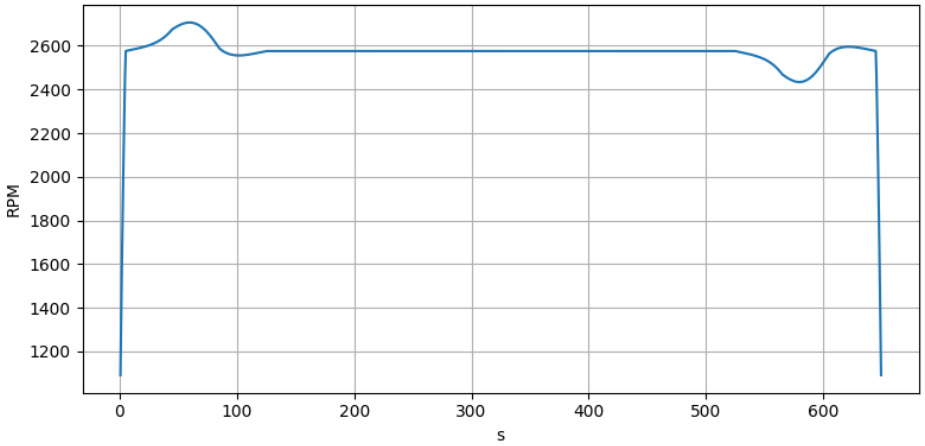


Figure 7.9: Rotor Speed During Operation

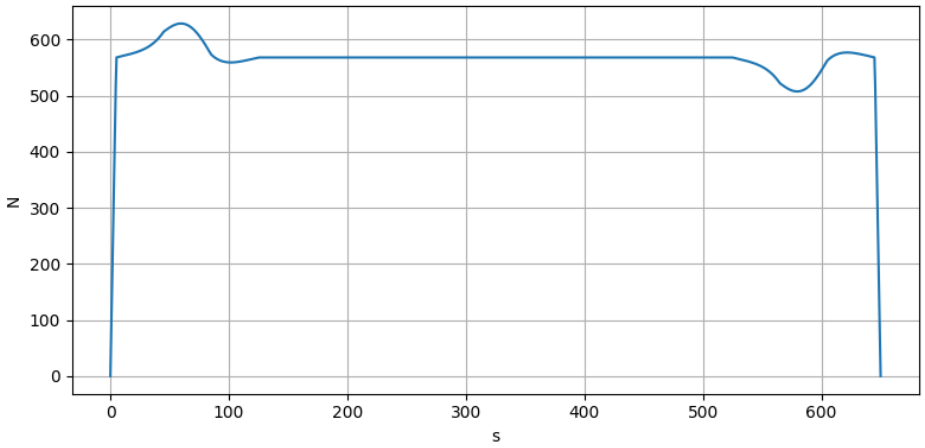


Figure 7.10: Thrust Profile

7.4.4 Damage

During the transient analysis for the case of 10 mm of deflection for the UAV, no damage was accumulated. This is due to the structural forces not being high enough to cause fatigue and the fatigue data for aluminum being stronger than carbon fiber. The maximum deflection during the transient analysis 14.4mm or 1.27% deflection per span. The maximum stress during the transient analysis is 37.1 MPa.

7.4.5 Design Space

The SLSQP solver does not ensure convergence to a global minimum. To evaluate local behavior, the design space was examined. The interaction between the arm radius and arm thickness design variables is shown in Figures 7.11 and 7.12. The white dot in Figure 7.11 and 7.12 is the location of the optimal result. The design space plots are sections around the set of optimal design variables with respect to the indicated variables. The contour plots represent the constraints. A value of zero or less on the contour plot means the constraint is

being met. The contour line where the constraint value is 0 is the boundary between where the constraint is active and inactive. The dotted line is on the region of the viable design space for that constraint. More design space plots are in Appendix A. The blue contour plot shows many discontinuities within the frequency constraint.

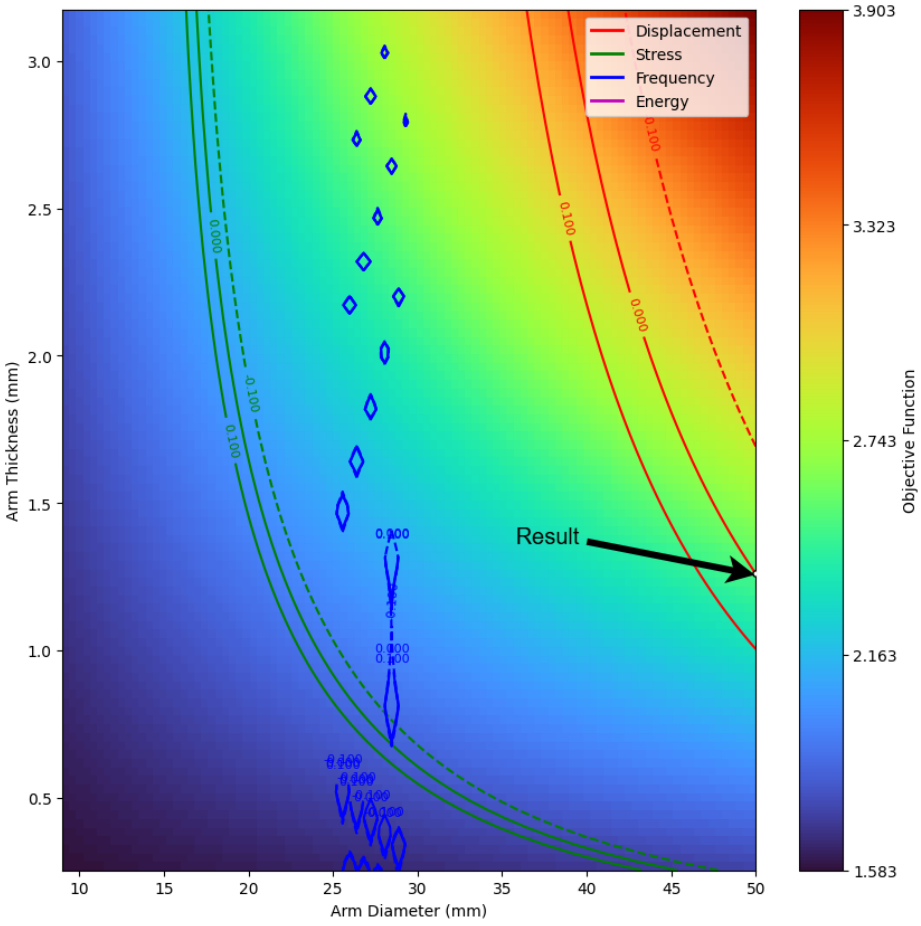


Figure 7.11: Design Space Between Arm Radius and Arm Thickness

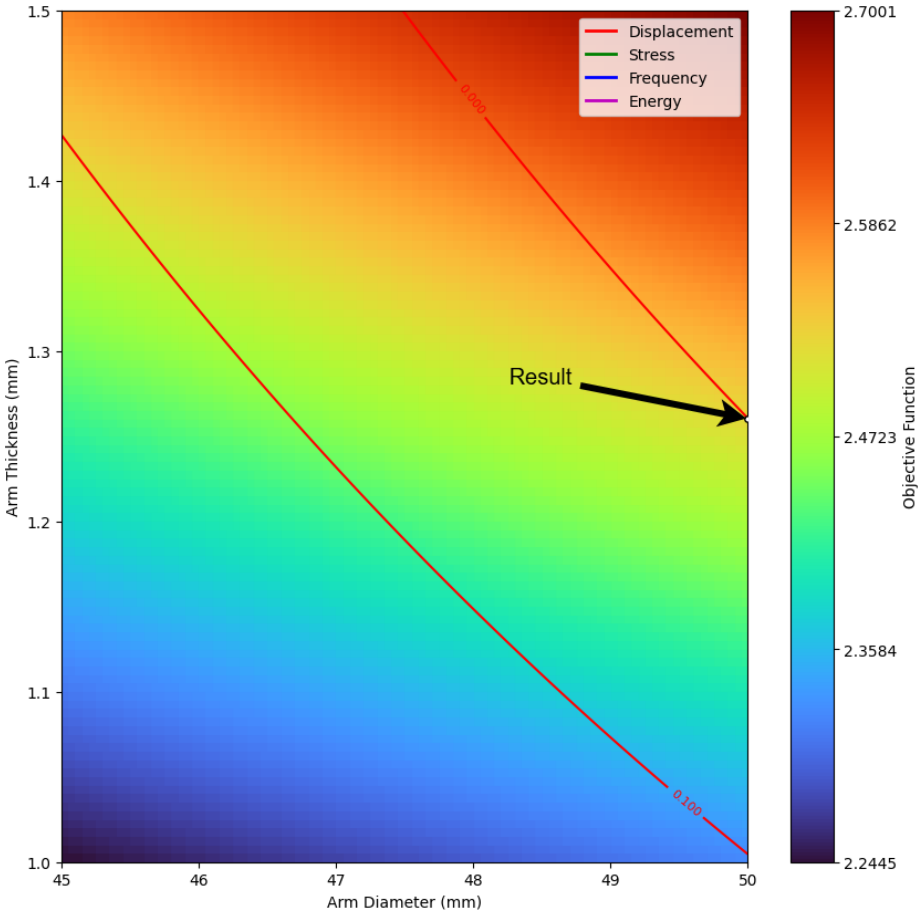


Figure 7.12: Zoomed Design Space Between Arm Radius and Arm Thickness

7.4.6 Sensitivity Analysis

Table 7.14 shows the sensitivity analysis for the 10mm deflection result. All combinations of design variables that do not have active bounds were varied by 0.1% in both the positive and the negative directions. The strut radius design variable was not considered.

The red cells are values where a constraint is violated or the objective function is worse than the optimum. This shows that the result is valid, as changing the solution in any direction makes the solution invalid or less optimal.

	Case 0	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12	Case 13	Case 14	Case 15	Optimal Point
Arm diameter (mm)	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Arm thickness (mm)	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2656	1.2681	1.2668
Strut diameter (mm)	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03	9.03
Strut thickness (mm)	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254	0.254
Strut distance (mm)	497.38	497.38	498.38	498.38	497.38	497.38	498.38	498.38	497.38	497.38	498.38	498.38	497.38	497.38	498.38	498.38	497.88
Hub radius (mm)	497.20	497.20	497.20	497.20	498.19	498.19	498.19	498.19	497.20	497.20	497.20	497.20	498.19	498.19	498.19	498.19	497.70
Hub flange thickness (mm)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Hub web thickness (mm)	1.1255	1.1255	1.1255	1.1255	1.1255	1.1255	1.1255	1.1255	1.1278	1.1278	1.1278	1.1278	1.1278	1.1278	1.1278	1.1278	1.1266
Deflection	1.0016	1.0008	1.0016	1.0008	1.0027	1.0019	0.9994	0.9986	1.0014	1.0006	1.0014	1.0006	1.0025	1.0017	0.9992	0.9984	0.9984
Stress	0.3015	0.3015	0.3015	0.3015	0.3012	0.3012	0.3012	0.3012	0.3013	0.3013	0.3013	0.3013	0.3011	0.3011	0.3011	0.3011	0.3011
Displacement amplitude	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076	0.0076
Energy	0.9367	0.9367	0.9367	0.9367	0.9369	0.9369	0.9369	0.9369	0.9367	0.9367	0.9367	0.9367	0.9369	0.9369	0.9369	0.9369	0.9369
Damage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Objective	2.5164	2.5183	2.5164	2.5183	2.5210	2.5229	2.5209	2.5228	2.5169	2.5188	2.5169	2.5188	2.5215	2.5234	2.5214	2.5233	2.5198

Table 7.14: Sensitivity Analysis

Chapter 8

Conclusions, Summary, and Recommendations

8.1 Summary

A multidisciplinary design optimization tool was developed for the conceptual design of a wildfire management octocopter. The objective was to minimize frame mass and mass moment of inertia while maintaining structural integrity and ensuring a desired mission profile is feasible.

A mission profile model was developed based on kinematic equations for linear acceleration. This mission profile was used to calculate power, energy, thrust, and propeller frequency, and motor perturbation force during the mission.

The structural response was evaluated using a finite element model based on Timoshenko beam theory. This model had three types of forces: static forces, harmonic forces, and transient forces. The structural response results were used to evaluate constraints on maximum stress, allowable deflection, frequency response, and fatigue life.

The optimization tool uses the mission profile data and finite element model. The optimization problem minimized mass and mass moment of inertia subject to deflection, stress, frequency, energy, and fatigue constraints.

The tool is designed to allow the user to rapidly test various design configurations. Designs with a constraint on deflection of 50mm and 10mm of deflection were created. Only the displacement constraint was active for both designs. The 50mm deflection constraint design resulted in a solution with overly thin walls, so the design of the 10mm constraint was created. Both designs resulted in the struts being very thin and of low diameter.

This project successfully integrates mission modeling, structural analysis, energy estimation, and optimization into a computational framework. This framework allows potential UAV designs to be rapidly evaluated.

8.2 Conclusions

This research was successful in developing a structural optimization framework for a wildfire management octocopter. The optimization tool developed during this study can be used to rapidly test potential UAV configurations.

A tool to model the mission profile was created. This mission profile tool was used to model the power usage and energy stored over the duration of the mission profile. The mission profile was also used to find the thrust required to lift the vehicle.

The finite element analysis tool was developed in order to allow an optimization algorithm to quickly solve structural analysis problems. The finite element tool finds the stress and displacement given a set of static forces. The FEA tool can also find stress and displacement solutions given harmonic and transient forces. The FEA tool can also be used to find natural frequencies of a beam system.

An optimization tool was developed to minimize the UAV mass and mass moment subject to constraints on displacement, stress, frequency response, fatigue, and energy. The constraints

were developed from the mission profile and finite element analysis tools.

A design was created with a constraint on deflection of 50mm. The optimization function was run to reduce the mass of the frame from 6.27kg to 0.92kg. The mass moment of inertia was reduced from $1.24kg \cdot m^2$ to $0.183kg \cdot m^2$. This constraint was determined to be too lenient due to the thin walls of the design, so a design with a deflection constraint of 10mm was created. The mass was reduced from $6.27kg$ to $3.53kg$. The mass moment of inertia was reduced from $1.24kg \cdot m^2$ to $0.70kg \cdot m^2$. Both resulting designs are dictated by the stiffness of the UAV frame. This demonstrates the ability of the developed framework to rapidly evaluate UAV concepts. Figure 7.2 shows the resulting frame. Tables 7.8 and 7.9 show the resulting design variables and objective function.

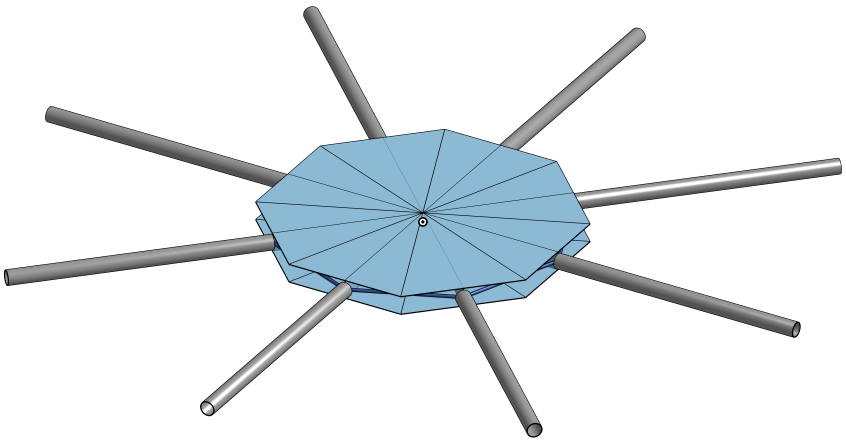


Figure 8.1: Visualization of 10mm Deflection Design (repeated)

Repeated Table 7.8: Resulting Design Variables for 10mm Deflection Design

Design variable	Value (mm)
Arm diameter	50
Arm thickness	1.26
Strut diameter	9.04
Strut thickness	0.254
Strut distance	497
Hub radius	497
Hub flange thickness	0.7
Hub web thickness	1.18

Repeated Table 7.9: Objective Function Values for 10mm Deflection Design

Result	Initial guess	Optimized	Percent difference
Frame mass (kg)	6.27	3.53	-44
Frame mass moment of inertia ($kg \cdot m^2$)	1.24	0.697	-44
Objective	4.68	2.52	-45

8.3 Recommendations

Work can be done to further improve the model presented in this thesis. This section presents recommendations to proceed with this work, refining the finite element model, mission profile, frequency constraint, and objective function.

8.3.1 Finite Element Model Refinement

The UAV finite element mesh may be further refined to improve the accuracy of the constraints. In particular, improved representation of the joints between structural members would allow for a more accurate representation of the frequency response. Shell elements can be included in the finite element model so that the hub surface can be modeled as continuous. Additionally, the hub should be remodeled in order to account for the storage of electrical components and the battery. Different configurations of the design can also be tested.

8.3.2 Mission Profile Refinement

The current mission profile does not account for vehicle tilt or wind. The mission profile could also use real mission data in order to design a more accurate profile, allowing for a more accurate energy constraint and more accurate perturbation and thrust forcing functions. The increased accuracy of the perturbation and thrust forcing functions would increase the accuracy of the damage constraint.

8.3.3 Frequency Constraint

Both strut thickness and diameter are driven to their lower bounds. This suggests that the present loading model does not sufficiently represent the structural contribution of the struts. Horizontal loading in the dynamic harmonic model to represent changes in motor torque could be included in the analysis. The frequency constraint is also limited in scope. Currently, only selected vibration modes are considered. Asymmetric modes as well as full-vehicle modes should be incorporated into the frequency constraint. The frequency constraint

may require to be modeled as a continuous constraint so that SLSQP can be used when it is an active constraint.

8.3.4 Material and Fatigue Modeling

More accurate Woehler curve data is needed to improve fatigue life predictions. The aluminum 7075 material model used is overly stiff compared to the carbon fiber composite. The aluminum used has an elastic modulus of 71.7GPa, where the carbon fiber composite has an elastic modulus of 19GPa. The stiffness-to-density ratio of aluminum is $2.55 \times 10^7 \ \frac{m^2}{s^2}$. The stiffness-to-density ratio for the carbon fiber composite is $1.56 \times 10^7 \ \frac{m^2}{s^2}$. Carbon fiber composite material property data may be obtained so that the structure may be modeled as orthotropic. Incorporating fully orthotropic material behavior would improve the stress and fatigue constraints.

8.3.5 Preventing High Surface Area Solutions

The optimization algorithm produced a solution with a high projected surface area for the 10mm displacement constraint case. The hub radius increased significantly compared to the initial design. If the energy constraint were active, aerodynamic drag would likely have a greater influence on the optimal configuration. A more accurate drag coefficient estimation based on experimental data for the UAV configuration could improve the energy constraint. Incorporating power consumption or drag directly into the objective function would help prevent the optimization from converging to high projected surface area solutions.

Bibliography

[1] Dragonplate carbon fiber composites. https://dragonplate.com/.

[2] Tattu 40000mah 6s uav lipo battery. https://genstattu.com/tattu-40000mah-6s-10c-22-8v-high-voltage-uav-lipo-battery-pack-with-as150-as150-plu

[3] Arcgis map viewer. https://www.arcgis.com/apps/mapviewer/index.html?layers=ba0631e5417f4c5f8c02a834d121bd81.

[4] Matweb aluminum 7075-t6 and 7075-t651. https://matweb.com/search/datasheet.aspx?matguid=4f19a42be94546b686bbf43f79c51b7d.

[5] pylife. https://pylife.readthedocs.io/en/latest/README.html.

[6] Moulay A. Akhloufi, Andy Couturier, and Nicolás A. Castro. Unmanned aerial vehicles for wildland fires: Sensing, perception, cooperation and assistance. *Drones*, 5(1), 2021. ISSN 2504-446X. doi: 10.3390/drones5010015. URL https://www.mdpi.com/2504-446X/5/1/15.

[7] Camille Baker and William B. Davis. Maps: Tracking southern california wildfires. *The New York Times*, July 2025. URL https://www.nytimes.com/interactive/2025/01/08/weather/los-angeles-fire-maps-california.html.

[8] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and Applications of Finite Element Analysis, Fourth Edition*. 2002.

[9] Norman E. Dowling, Stephen L. Kampe, and Milo V. Kral. *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue*. Pearson, Hoboken, NJ, 5 edition, 2019. ISBN 9780134606545.

[10] Justin S. Gray, John T. Hwang, Joaquim R. R. A. Martins, Kenneth T. Moore, and Bret A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi: 10.1007/s00158-019-02211-z.

[11] Nathaniel Hargan. Uav_optimization. https://github.com/NathanielHargan/UAV_Optimization, 2026. Accessed: 2026-02-19.

[12] John P. Jasa, John T. Hwang, and Joaquim R. Martins. Open-source coupled aerostructural optimization using python. *Structural and Multidisciplinary Optimization*, 57(4):1815–1827, Feb 2018. doi: 10.1007/s00158-018-1912-8.

[13] Nicholas James Newton. Development of tools for conceptual design of a wildland firefighting uav. Master's thesis, Virginia Polytechnic Institute and State University, Falls Church, Virginia, 2023.

[14] Laura Parker, 2024. URL https://education.nationalgeographic.org/resource/drones-shoot-fireballs-help-control-wildfires/.

[15] Félix Pollet. *Design optimization of unmanned aerial vehicles: a multidisciplinary approach with uncertainty, fault-tolerance, and environmental impact assessments*. PhD Thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Mar 2024. URL http://www.theses.fr/2024ESAE0013/document.

[16] Sydney L. Schnulo, Jeff Chin, Robert D. Falck, Justin S. Gray, Kurt V. Papathakis, Sean C. Clarke, Nickelle Reid, and Nicholas K. Borer. Development of a multi-segment mission planning tool for sceptor x-57. *2018 Multidisciplinary Analysis and Optimization Conference*, Jun 2018. doi: 10.2514/6.2018-3738.

[17] Irving H. Shames and Clive L. Dym. *Energy and Finite Element Methods in Structural Mechanics*. 1985.

[18] StructX. Hollow circle (annulus) - geometric properties. https://structx.com/Shape_Formulas_014.html, .

[19] StructX. Square i-beam - geometric properties. https://structx.com/Shape_Formulas_013.html, .

[20] T-Motor. U8 lite l efficiency multirotor uav motor kv95. https://store.tmotor.com/product/u8lite-l-kv95-u-efficiency.html, .

[21] T-Motor. G32*11 prop-2pcs/pair. https://store.tmotor.com/product/g32x11prop-2pcs-1pair-glossy-carbon-fiber.html, .

[22] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[23] Matthew West. Moments of inertia. https://dynref.engr.illinois.edu/rem.html, 2017.

Appendices

Appendix A

Design Space Plots

Additional variable interactions are shown in Figures A.1, A.2, A.3, A.4, A.5, and A.6. The white dot is the result. They are all sections of the design space of the 10mm result.

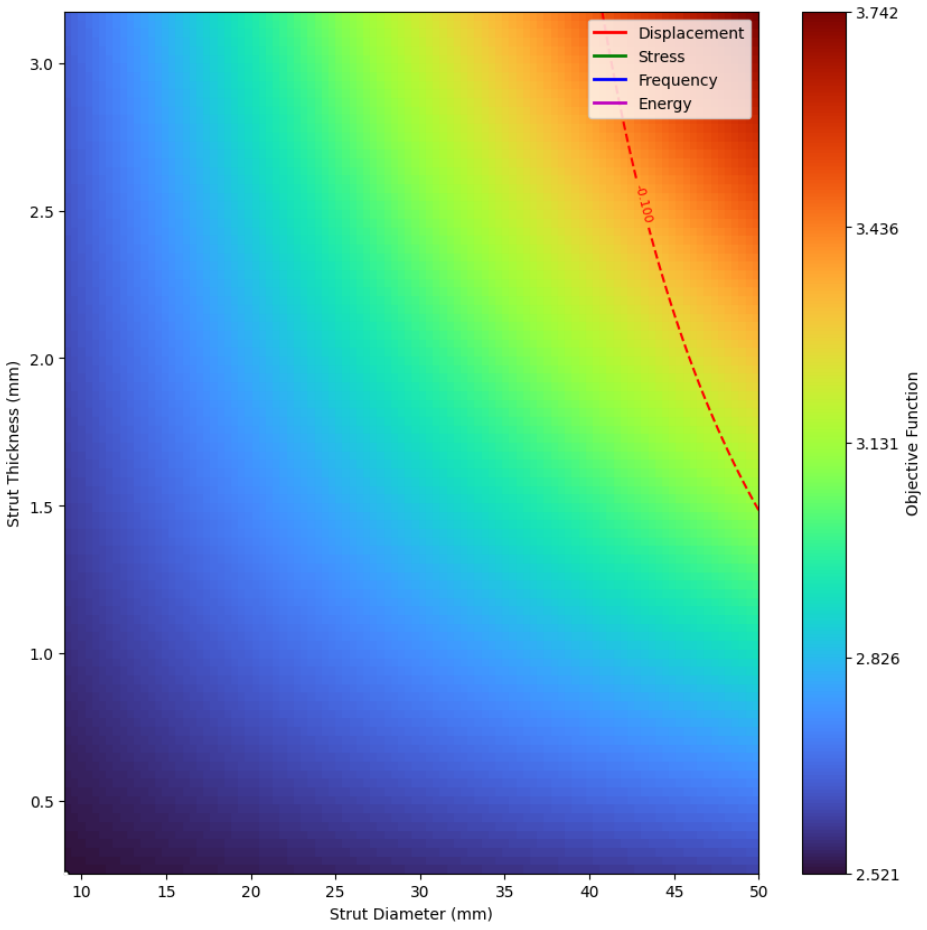


Figure A.1: Strut Thickness vs Strut Diameter Design Space for the 10mm Results

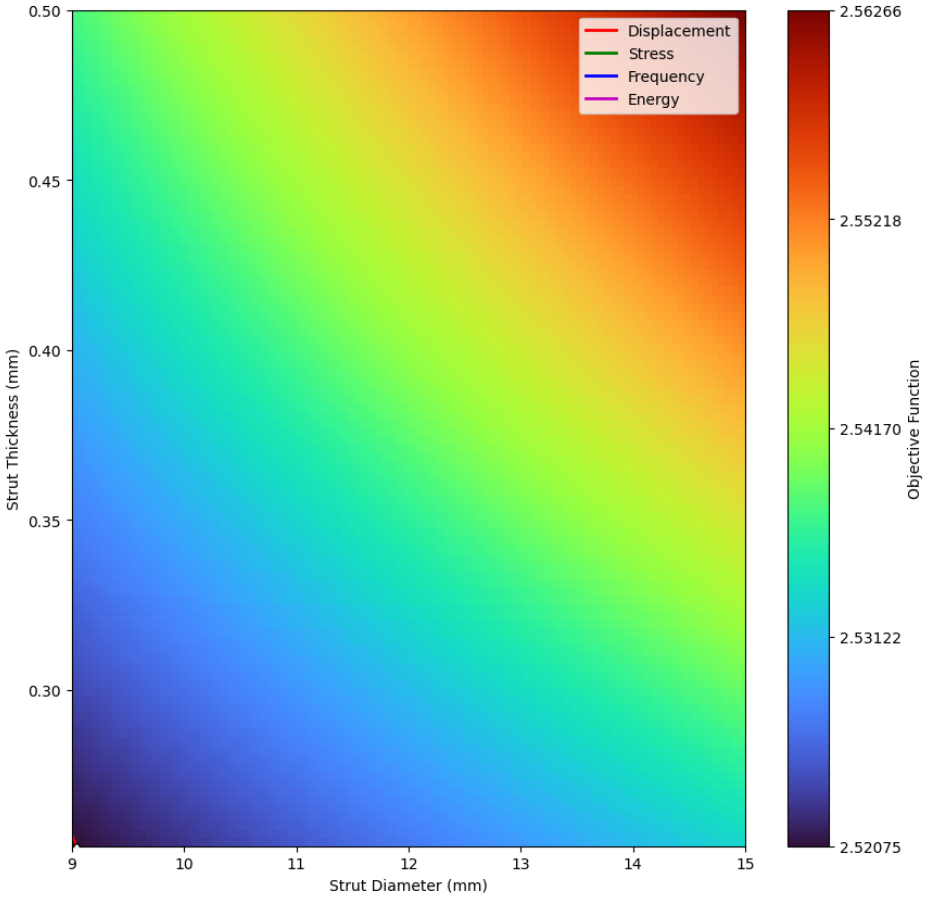


Figure A.2: Strut Thickness vs Strut Diameter Design Space for the 10mm Results Zoomed

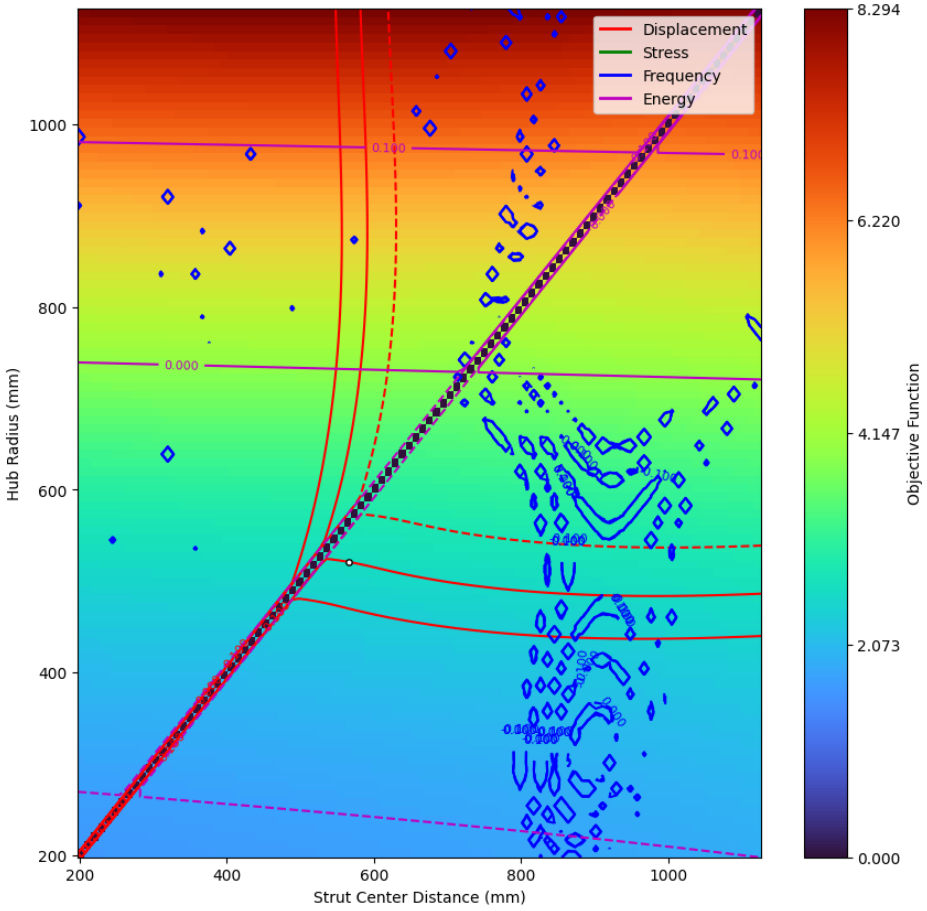


Figure A.3: Hub Radius vs Strut Distance Design Space for the 10mm Results

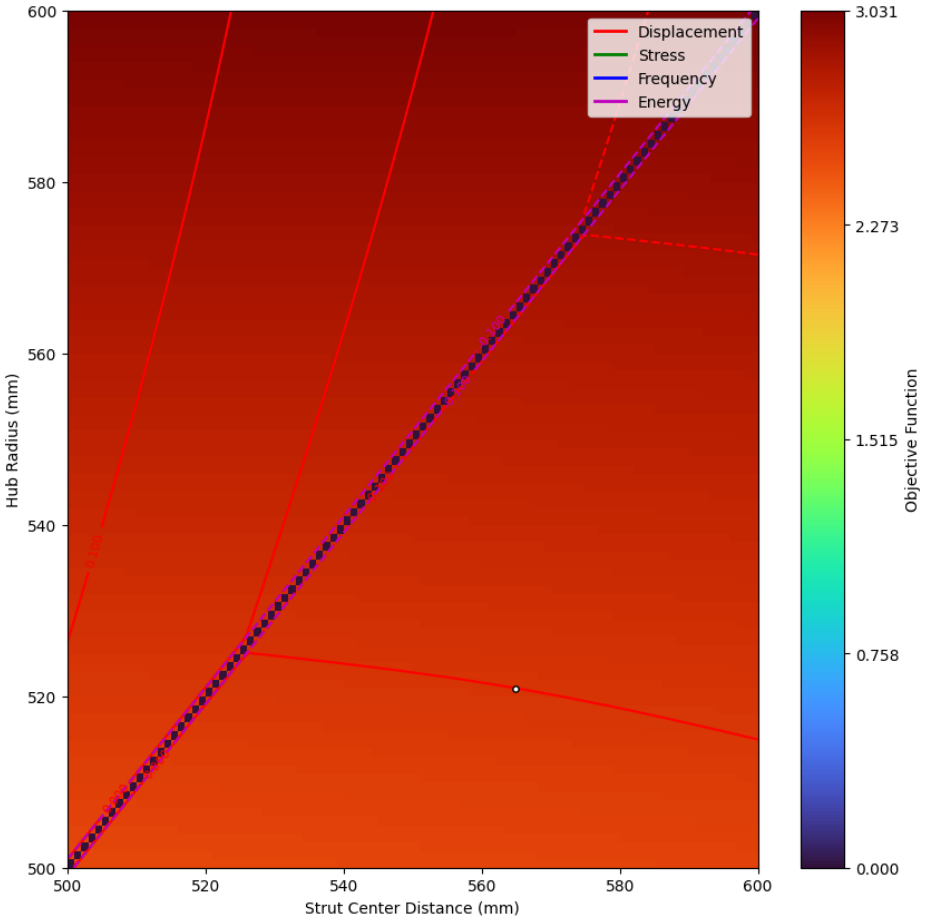


Figure A.4: Hub Radius vs Strut Distance Design Space for the 10mm Results Zoomed

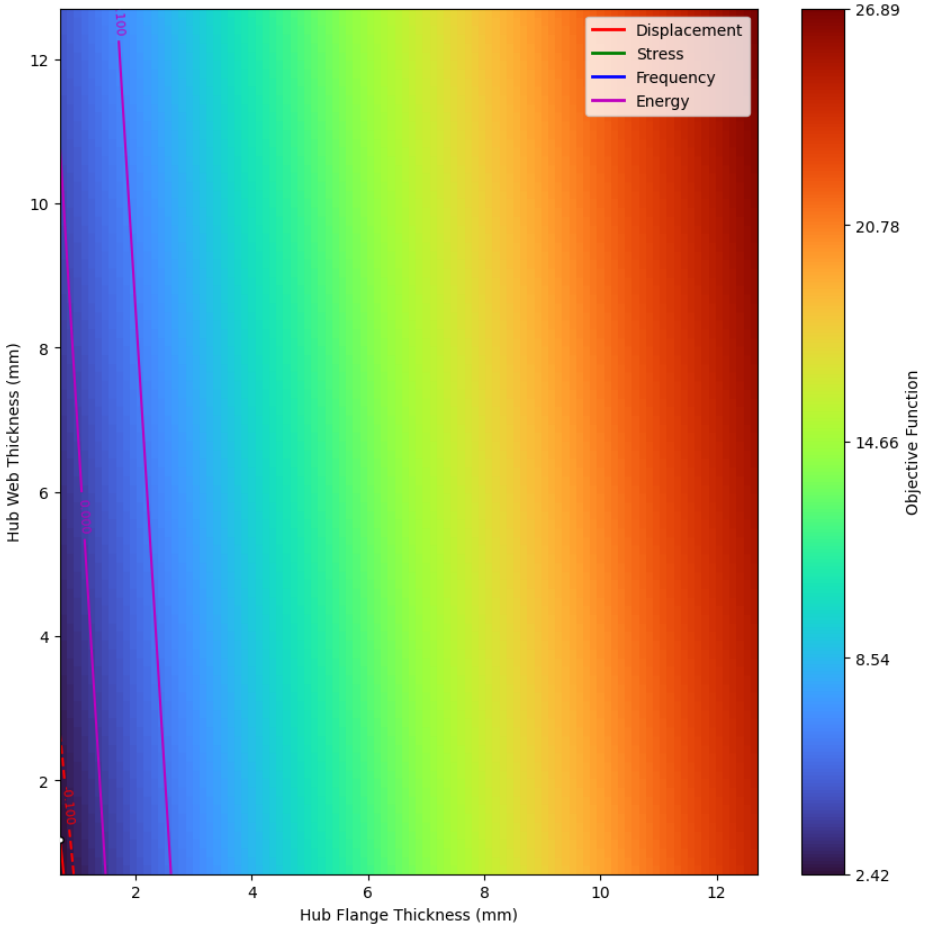


Figure A.5: Hub Flange Thickness vs Hub Web Thickness Design Space for the 10mm Results

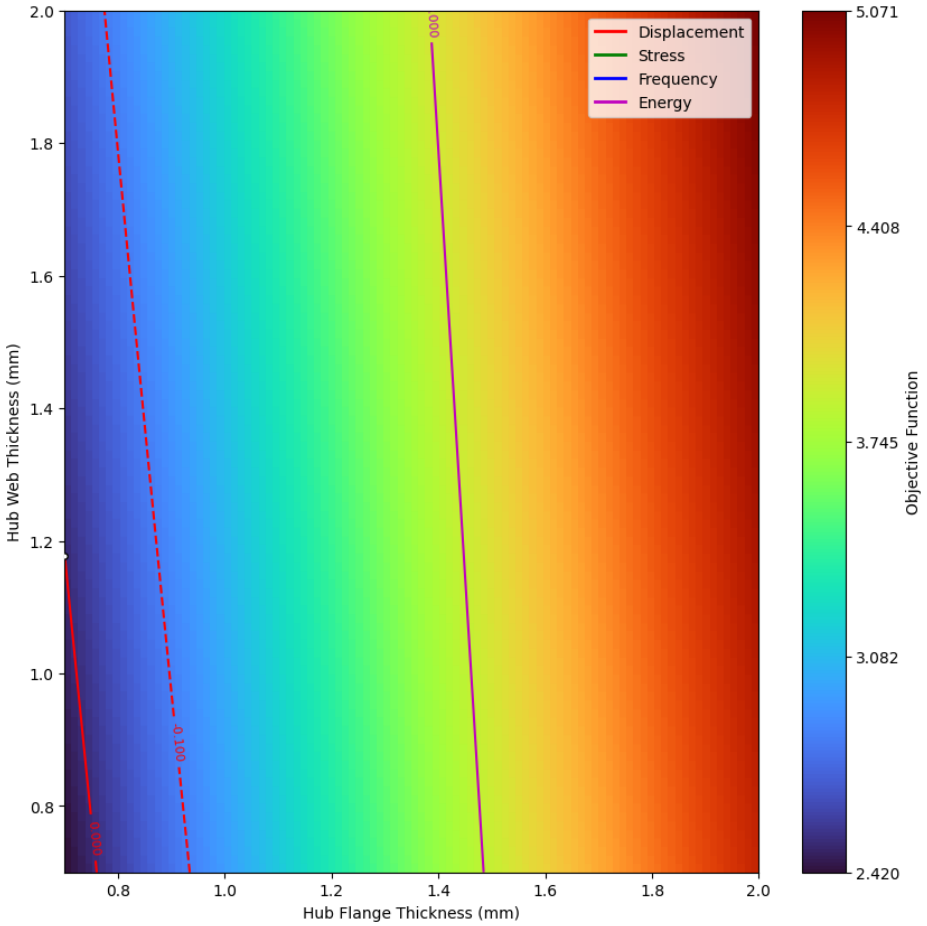


Figure A.6: Hub Flange Thickness vs Hub Web Thickness Design Space for the 10mm Results Zoomed

Appendix B

Code Architecture

This appendix discusses the architecture of the code created for this project. All source code developed for this research is available in the UAV_Optimization GitHub repository [11]. Plots and results are generated with the UAV_Optimize Jupyter notebook. Modules are included within the Scripts folder. The contents are listed below.

- The beam_mass module calculates the moment of inertia for a given beam_system object.
- The drone_mass module calculates the mass of the UAV given a drone_geometry object.
- The mission_profile module generates the mission profile.
- The drone_power module contains all calculations related to the UAV's energy consumption during the mission.
- The optimization module implements the optimization framework.
- The drone_geometry module calculates geometric properties of the UAV.

Within the Finite_Elements folder, there are several modules listed below.

- The cross_section_properties module calculates cross-section properties for a given cross-section.

- The material_properties module contains dictionaries with the necessary material properties.
- The drone_fea module initializes the FEA model for a given drone_geometry object.
- The beam_system module contains the FEA framework.