Institute Supérieur de l'Aéronautique et de l'Espace

Master in Aerospace Engineering

Aircraft and Space Actuation Systems

Multirotor Sizing

Professor:
Mr. Marc Budinger, Mr. Félix Pollet, Mr. Scott Delbecq

Authors:
Deeksha KOTA, Akash SHARMA

1 Introduction

The field of unmanned aerial vehicles (UAVs), commonly known as drones, has seen rapid advancements in recent years. One of the key challenges in the design and development of drones is to accurately size their various components such as the propulsion system, battery, and frame based on specific objectives and constraints. This project aims to address this challenge by using an optimization algorithm for the accurate sizing of components in a multi-rotor drone. These objectives are referred to as sizing scenarios while the constraints are called design drivers which set the limitations to the optimization problem.

The optimization algorithm focuses on accurately sizing five main components of a multi-rotor drone: the propeller, the motor, the battery, the Electronic Speed Controller (ESC), and the frame. Each of these components plays a crucial role in the performance of the drone, and their sizes need to be optimized to meet the specific objectives and constraints of the design. To achieve this, a sizing code is developed for each component. This code takes into account the specific characteristics and requirements of the component and determines the optimal size based on the objectives and constraints specified by the user. Once the sizing codes for all the components have been developed, a gradient search optimization algorithm is used to iteratively find the optimal sizes for all the components. This approach will help design engineers in the drone industry to optimize the size of each component according to their specific use case, ultimately leading to better-performing and more efficient drones.

2 Sizing Methodology

The initial step in determining the characteristics and components of UAVs is referred to as **preliminary sizing**, which includes identifying the main characteristics of the drone and choosing components based on specific scenarios. These scenarios, which are a part of the design specification process, can include transient and endurance criteria such as maximum climbing speed and the ability to hover for an extended period. The final design must take into account all of these scenarios, which can affect the selection of components through design drivers.

Sizing scenarios refer to specific conditions or requirements that the drone must be able to meet or perform under. These scenarios are used to inform the design and selection of components for the UAV. Examples of sizing scenarios include maximum rate of climb, hover flight endurance, maximum speed, payload capacity, and range. These scenarios are used to define the specifications for the UAV and will help to ensure that the final design can meet the intended operational requirements.

Design drivers are factors or constraints that influence the design of a system. In the context of UAVs, design drivers refer to the specific requirements or scenarios that must be taken into consideration during the design process. These drivers can include factors such as weight, aerodynamics, power and energy, stability and control, and performance criteria such as speed, range, and payload capacity. Design drivers help to ensure that the final design of the UAV can meet the intended operational requirements and perform as desired in the various scenarios.

In the given assignment, We consider two sizing scenarios, **hover** and **takeoff**. The design drivers and the sizing scenarios chosen are shown in the table 1.

Components	Design drivers	Parameters involved	Sizing scenarios
Propeller	Max thrust	Diameter tip speed	Takeoff
	Efficiency	Pitch	Hover
Motor	Temperture rise	Nominal torque, Resistance	Hover
	Max. voltage	Torque constant, Resistance	Takeoff
Electronic Speed Controller	Temperature rise	Max power, Max current	Takeoff
Battery	Energy	Voltage, Capacity	Hover (autonomy)
	Power	Voltage, Max discharge rate	Takeoff
Frame	Stress	Mechanical strength	Takeoff

Table 1: Design drivers and sizing scenarios with focus on hover and takeoff scenarios

The basic principle of sizing the parameters of the drone in this assignment is to use an **optimization algorithm** to minimize or maximize a particular objective, given some constraints. The reason to use the optimizer is to solve the three main problems encountered during sizing, namely, **under-constrained singularities,**

over-constrained singularities, and **algebraic loops**.

1. Under-constrained singularity occurs when there is not enough information to uniquely determine the parameter.
2. Over-constrained singularity suffers from too many constraints so that no value of that parameter can satisfy all of them.
3. Algebraic loops occur as a result of a circular dependency between variables in the equations where parameters depend on each other, where the value of the variable cannot be determined without knowing the value of the other variables.

These issues can be solved by introducing the optimizer by adding a variable to update at each iteration or adding a constraint to the algorithm.

In order for the algorithm in the optimizer to accurately size all parameters, it requires estimation models of the components as well as models of the scenarios for which the drone is being sized. These estimation models required to relate different physical parameters are not always readily available or easily derivable. Therefore, two main strategies are used to obtain estimation models, namely, **scaling laws** and **regression models**.

2.1 Scaling Laws

Scaling laws describe the functional relationship between two physical parameters that scale with each other over a significant interval. These relationships are used to relate parameters without a detailed design using a reference parameter. The scaling ratio x* is defined as $x^* = \frac{x}{x_{ref}}$ where x_{ref} is the parameter under reference and x is the parameter under study.

2.2 Regression Models

Regression models are statistical methods to determine relationships between 2 variables based only on data points. This is particularly useful when the relationship between variables is not known and when significant manufacturers' data is available. Based on how the data points arrange themselves on a graph, one can opt for a linear regression model which approximates a linear function between the 2 variables, or a logarithmic regression model where a power law is obtained.

3 Sizing Codes for Components

3.1 Mass computation

An initial estimate of the mass is required to start the sizing of components, but in order to have the total weight, we need the weight of all the components. Thus we encounter an **algebraic loop** in the sizing procedure. To solve this issue, an oversizing coefficientk_os is introduced. The solution includes the use of an oversizing coefficient k_{os} [1-10] as a design variable and an additional constraint that the total load must be greater than or equal to the total mass M_{tot}. This ensures that the drone has the capability to lift both the load and itself. As we are trying to minimize the total mass M_{tot}, the oversizing coefficient k_{os} will be as small as possible, resulting in the constraint becoming an equality ($M_{total_{load}} = M_{tot}$). This is one method of addressing an algebraic loop.

```
#specification
M_pay = 4.0 #[kg] payload mass

#design variable updated by optimiser
k_os = 3.2 # over sizing coefficient on the load mass

#equations
M_total = k_os * M_pay  # [kg] Estimation of the total mass (or equivalent weight of dynamic scenario)
```

3.2 Propeller Sizing

There are four unknowns in this system are the diameter of the propeller, the rotational speed, the pitch ratio, and the thrust coefficient. From these unknowns, other parameters such as mass, power, and torque can be determined. It can be seen that there are three equations and four unknowns, making the system **under-constrained singularity**. To make the system solvable, the pitch-to-diameter ratio called beta is taken as an

input and a design variable which will be updated by the optimizer at each iteration.
The design of the propeller is driven by the maximum thrust it can deliver. Three estimation models are obtained for the thrust and pressure coefficients by regression and the mass of the propeller by scaling law. parameter nD, the linear velocity of the propeller was overconstrained hence, a slow-down propeller coefficient, k_{nD} has been introduced in the inequality to obtain a value of nD and hence solve **over-constrained singularity**. This velocity is constrained by a maximum value so that the tips of the propeller blades do not break the sound barrier which would result in additional drag. The optimizer, then, iterates through the values of k_{nD} to converge to the optimum value.

```
# Specifications
M_pay = 4.0 #[kg] payload mass
rho_air=1.18# [kg/m^3] Air density
ND_max=105000./60.*.0254 #[Hz.m] Max speed limit (N.D max) for APC MR propellers
a_to = 0.25 * 9.81 # [m/s**2] acceleration
N_arm = 4 # [-] number of arms
N_pro_arm = 1 # [-] number of propeller per arm (1 or 2)

# Reference parameters for scaling laws
D_pro_ref=11.*.0254# [m] Reference propeller diameter
M_pro_ref=0.53*0.0283# [kg] Reference propeller mass

# Design variables updated by the optimizer
k_os = 3.2 # over sizing coefficient on the load mass
k_ND = 1.2 # slow down propeller coef : ND = NDmax / k_ND
beta_pro = .33 # pitch/diameter ratio of the propeller

# SCENARIOS
M_total = k_os * M_pay  # [kg] Estimation of the total mass (or equivalent weight of dynamic scenario)
N_pro = N_pro_arm * N_arm  # Propellers number
F_pro_to= M_total * (9.81 + a_to) / N_pro #[N] Thrust for 1 propeller during Take Off
F_pro_hov= M_total * (9.81) / N_pro #[N] Thrust for 1 propeller during hover

#Equations:
# Estimation models for propeller aerodynamics
C_t = 4.27e-3 + 1.44e-1 * beta_pro # Thrust coef with T=C_T.rho.n^2.D^4 with estimation model for ct=f(beta)
C_p = -1.48e-3 + 9.72e-2 * beta_pro # Power coef with P=C_p.rho.n^3.D^5 with estimation model for cp=f(beta)

# Propeller selection with take-off scenario
nD = ND_max/k_ND #equality due to the addition of oversizing coefficient
D_pro = math.sqrt(F_pro_to/(rho_air*nD**(2)*C_t)) # [m] Propeller diameter
n_pro_to = nD/D_pro # [Hz] Propeller speed
Omega_pro_to = (2*math.pi/60)*n_pro_to # [rad/s] Propeller speed

# Estimation model for mass
M_pro = M_pro_ref*(D_pro/D_pro_ref)**3 # [kg] Propeller mass

# Performance in various operating conditions
# Take-off
P_pro_to = C_p*rho_air*(nD)**3*D_pro**2 # [W] Power per propeller
T_pro_to = P_pro_to/Omega_pro_to  # [N*m] Propeller torque
# Hover
n_pro_hov = math.sqrt(F_pro_hov/C_t*rho_air*D_pro**4) # [Hz] hover speed
Omega_pro_hov = (2*math.pi/60)*n_pro_hov # [rad/s] Propeller speed
P_pro_hov = C_p*rho_air*(n_pro_hov)**3*D_pro**5 # [W] Power per propeller
T_pro_hov = P_pro_hov/Omega_pro_hov # [N*m] Propeller torque
```

3.3 Motor Sizing

The efficiency of the motor is dependent on the temperature rise which occurs due to the amount of current delivered to it. Therefore, the motor sizing is driven by this temperature rise as well as the maximum voltage. The maximum thrust delivered, the mass of the motor and the resistance are optimized using scaling laws.
There are 2 inequalities defined based on the scenario models. The first one is $T_{nom} \geq T_{hover}$ where the nominal thrust must be greater than or equal to the hover thrust. This is to ensure that the drone is able to perform maneuvers. The second inequality is $T_{max} \geq T_{takeoff}$. This inequality exists to make sure that the motor has the capacity to lift its total weight as well as the weight of the payload and takeoff. While T_{max} is a function of T_{norm}, the system is **over-constrained singularity**
In this case, we have only defined an over-sizing coefficient; k_mot to calculate T_{norm}. Further use the estimation model to find T_{max} and check the constraint for takeoff. The torque constant is calculated using the estimated battery voltage and the rotational speed of the propellers. A new design variable called the motor speed oversizing coefficient is introduced in this case. The maximum torque, resistance and mass of the motors are computed by means of scaling laws.

```
# Reference parameters for scaling laws
# Motor reference
# Ref : AXI 5325/16 GOLD LINE
T_nom_mot_ref = 2.32  # [N.m] rated torque
T_max_mot_ref = 85./70.*T_nom_mot_ref # [N.m] max torque
R_mot_ref = 0.03  # [Ohm] resistance
M_mot_ref = 0.575 # [kg] mass
```

```python
K_T_ref = 0.03 # [N.m/A] torque coefficient
T_mot_fr_ref = 0.03 # [N.m] friction torque (zero load, nominal speed)

# SCENARIOS from the propeller
T_pro_to=0.5#[N.m] Propeller Torque during takeoff
Omega_pro_to=400.0#[rad/s] Propeller speed during takeoff
T_pro_hov=1.0#[N.m] Propeller Torque during hover
Omega_pro_hov=0.22#[rad/s] Propeller speed during hover

#Design variables to optimize
k_mot = 1. # over sizing coefficient on the motor torque
k_speed_mot = 1.2 # adaption winding coef on the motor speed
k_vb = 1. # oversizing coefficient for voltage evaluation

#Equations:
# Nominal torque selection with hover scenario
T_nom_mot = k_mot*T_pro_hov # [N*m] Motor nominal torque per propeller - equality due to the addition of the oversizing coefficient to optimize

# Torque constant selection with take-off scenario
U_bat_est = k_vb*1.84*P_pro_to**(0.36)  # [V] battery voltage estimation
K_T = U_bat_est /(k_speed_mot*Omega_pro_to) # [N*m/A] or [V/(rad/s)] Kt motor

# Estimation models
M_mot = M_mot_ref*(T_nom_mot/T_nom_mot_ref)**(3/3.5)# [kg] Motor mass
R_mot = R_mot_ref*(K_T/K_T_ref)**2*(T_nom_mot/T_nom_mot_ref)**(-5/3.5) # [ohm] motor resistance
T_mot_fr = T_mot_fr_ref*(T_nom_mot/T_nom_mot_ref)**(3/3.5)  # [N*m] Friction torque
T_max_mot = T_max_mot_ref*(T_nom_mot/T_nom_mot_ref)  # [N*m] Max. torque

# Performance in various operating conditions
# Hover current and voltage
I_mot_hov = (T_mot_fr+T_pro_hov)/K_T  # [A] Current of the motor per propeller
U_mot_hov = K_T*Omega_pro_hov + R_mot*I_mot_hov  # [V] Voltage of the motor per propeller
P_el_mot_hov = I_mot_hov*U_mot_hov  # [W] Hover : electrical power
# Takeoff current and voltage
I_mot_to = (T_mot_fr+T_pro_to)/K_T  # [A] Current of the motor per propeller
U_mot_to = K_T*Omega_pro_to + R_mot*I_mot_to # [V] Voltage of the motor per propeller
P_el_mot_to = I_mot_to*U_mot_to # [W] Takeoff : electrical power
```

3.4 Battery and ESC Sizing

The battery is only constrained by the amount of energy it can store and deliver while the electronic speed controller is prone to damage due to high temperature. Therefore, these components are sized according to these constraints. the initial estimate of the battery is done by using the oversizing coefficient k_vb to find the initial voltage estimate with consideration of safety factor.

In the battery sizing calculations, 2 methods can be employed. The first method is the one used for the previous components where an over-sizing coefficient is defined for the scenario model inequality to find the energy, which can then be used to calculate the mass of the battery through the estimation model. The second strategy instead finds the mass of the battery by using the ratio of battery mass to payload mass, k_mb, and the payload mass. The energy capacity of the battery is then calculated instead based on the estimation model.

The ESC calculations are self-explanatory.

```python
# Specifications
N_pro=4.0#[-] Number of propellers
N_pro_arm = 1 # [-] number of propeller per arm (1 or 2)
M_pay=4.0 #[kg] Payload mass

# Reference parameters for scaling laws
# Ref : MK-quadro
M_bat_ref = .329 # [kg] mass
E_bat_ref = 220.*3600.*.329 # [J]
C_bat_ref= 5 # [Ah] Capacity
I_bat_max_ref = 50*C_bat_ref # [A] max discharge current

# Ref : Turnigy K_Force 70HV
P_esc_ref = 3108. # [W] Power
M_esc_ref = .115 # [kg] Mass

#Design variable to optimize
k_mb = 1. # ratio battery/load mass
k_vb = 1. # oversizing coefficient for voltage evaluation

# SENARIOS from Motor
U_bat_est= k_vb*1.84*(P_pro_to)**0.84#[V] Battery voltage estimation
P_el_mot_hov=10.0#[W] Electrical power consumption for one motor during hover
P_el_mot_to=30.0#[W] Electrical power consumption for one motor during takeoff
U_mot_to=12.0#[V] Motor voltage during takeoff

#Design variable
eff_esc = 0.95

# equations
# BATTERY
# Estimation model for Energy selection with payload mass
M_bat = k_mb * M_pay # [kg] Battery mass
```

```
E_bat = 0.8*E_bat_ref * (M_bat/M_bat_ref) # [J] Energy  of the battery (.8 coefficient because 80% use only of the total capacity)

# Estimation models
C_bat = E_bat / U_bat_est # [A*s] Capacity  of the battery
I_bat_max = I_bat_max_ref * (C_bat/C_bat_ref) # [A] Max discharge current
P_bat_max = U_bat_est * I_bat_max  # [W] Max power

# Performance in hover
I_bat_hov = (P_el_mot_hov*N_pro/eff_esc) / U_bat_est # [A] Current of the battery

# ESC
# Power selection with takeoff scenario
P_esc = U_bat_est * I_mot_to # [W] power electronic power (corner power or apparent power)

# Estimation models
U_esc = 1.84*(P_esc)**0.84  # [V] ESC voltage
M_esc = M_esc_ref*(P_esc/P_esc_ref)  # [kg] Mass ESC
```

3.5 Frame Sizing

In the frame sizing calculations, the arms of the frame are modelled as cylinders with an internal and external diameter which are proportional to each other with aspect ratio D_in/D_out for the beam of the frame, K_D. Using the calculated values of the diameters along with the length and the number of the arms, we can calculate the total mass of the arms and subsequently, the total mass of the frame.

```
# Specifications
N_arm=4.0#[-] Number of arms
N_pro_arm=1.0#[-] Number of propellers per arm (1 or 2)

# Reference parameters for scaling laws
sigma_max = 280e6/4. # [Pa] Composite max stress (2 reduction for dynamic, 2 reduction for stress concentration)
rho_s = 1700. # [kg/m3] Volumic mass of aluminum

# SCENARIOS from propeller
D_pro=0.3#[m] Propeller diameter
F_pro_to=1.0#[N] Thrust for one propeller during take off

#Design variables
k_D = .99 # aspect ratio D_in/D_out for the beam of the frame

# Arms selection with propellers size
alpha_sep = 2*math.pi/N_arm  # [rad] interior angle separation between propellers
L_arm = D_pro/2*math.sin(math.pi/N_arm)  # [m] length of the arm

# Tube diameter & thickness selection with take-off scenario
D_out_arm = ((F_pro_to*L_arm*32)/(math.pi*sigma_max*(1-k_D**4)))**(1/3)  # [m] outer diameter of the arm (hollow cylinder)
D_in_arm = k_D*D_out_arm # [m] inner diameter of the arm (hollow cylinder)

# Estimation models
M_arms = math.pi * (D_out_arm**2 - D_in_arm**2)*rho_s*L_arm*N_arm  # [kg] mass of the arms
M_body = M_arms/0.4 # [kg] mass of the body (40% of total mass is the arms)
M_frame = M_arms + M_body    # [kg] total mass of the frame
```

4 Optimization problem

An **optimization problem** is a problem that seeks to find the best solution among a set of possible solutions. It involves the selection of inputs (or parameters) that will result in the best performance or outcome for a given objective. Given a set of inputs (or parameters) X, and a scalar function f(x) called the objective function, the optimization problem is to find the input x* that minimizes or maximizes the objective function f(x).

The **objective** of the optimization is the function or criterion that is being optimized and the goal of the optimization process is to find the set of inputs that will minimize or maximize the objective function. For example, an objective in sizing optimization can be to minimize the weight of the drone while satisfying certain constraints. The optimization process would then search for the design or configuration of the UAV that results in the lowest weight while still meeting the performance criteria.

4.1 Optimization Approach

In this assignment, we have considered the **Gradient Search Algorithm** to optimize the objective by updating the parameters in the loop while all the constraints. Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (f) that minimizes a cost function. The cost function represents how far the model's output is from the desired output.

The algorithm works by iteratively moving in the opposite direction of the gradient of the cost function with respect to the parameters. The gradient is the vector of the partial derivatives of the cost function with respect to the parameters.

The basic steps of the gradient descent algorithm are:

1. Initialize the parameters with random values or any other values.
2. Compute the cost function and its gradient.
3. Update the parameters in the opposite direction of the gradient.
4. Repeat steps 2 and 3 until the cost function reaches a minimum or a stopping criterion is met.

4.2 Objective function

The optimization is done for two objectives in the assignment:

1. Total mass of the drone

$$\text{f(x)} = \text{total\_mass} \quad \text{(with} \quad \text{scalar=1} \quad \text{to} \quad \text{minimize)}$$

2. Flight time

$$\text{f(x)} = \text{flight\_time} \quad \text{(with} \quad \text{scalar=-1} \quad \text{to} \quad \text{maximize)}$$

```
# OBJECTIVES
t_hov = C_bat/I_mot_hov # [min] Hover time
M_total_real = M_pay + M_frame + M_bat + (M_esc + M_mot + M_pro)*N_pro # [kg] Total mass
```

4.3 Optimization constraints

As shown in the sizing codes, the following constraints were considered in the optimizer during its iterations.

1. The initial calculation for the total mass was based on a factor for overestimation with respect to payload mass, therefore the final optimized total mass value must be equal to or less than the initial estimate.

$$Mass\_total\_initial - Mass\_total\_real \geq 0 \tag{1}$$

2. The highest torque needed is during takeoff. This torque must be less than or equal to the maximum allowed torque that the motors can provide.

$$Torque\_max - Torque\_takeoff \geq 0 \tag{2}$$

3. The battery supplies power to the motor and would require the highest at takeoff. So the highest voltage at the motor will always be less than or equal to the voltage supplied by the battery.

$$U\_battery - U\_motor \geq 0 \tag{3}$$

4. The maximum power that the motor will require will be during takeoff which should be within the capacity limits of the battery. Therefore, the maximum power that the battery can deliver must be greater than or equal to the power that is delivered to the motors after taking into account efficiency losses. Here, N_{pro} is the number of motors.

$$P\_bat\_max - (P\_el\_mot\_to * N\_pro)/\eta \geq 0 \tag{4}$$

5. The voltage supplied to the electronic speed controller (ESC) comes from the battery. Due to potential losses, the voltage of the ESC is always less than or equal to the voltage supplied by the battery.

$$U\_battery - U\_ESC \geq 0 \tag{5}$$

6. The duration of hovering time specified in the requirements must be less than or equal to the one that is calculated through sizing of the components, which represents the maximum possible hovering time of the drone.

$$t\_hov - t\_hov\_spec \geq 0 \tag{6}$$

7. The MTOW specified in the requirements must be more than or equal to the one that is calculated through sizing of the components, which represents the minimum possible mass of the drone.

$$MTOW - M\_total\_real \geq 0 \tag{7}$$

4.4 Design Variables

An important drawback of the gradient-search algorithm is that optimizing the value of the objective function is dependent on what initial conditions are provided unless it's a convex curve where local minima is equal to the global minimum. If not, based on these values, it may find local minima or global minima, and hence the initial values will play a significant role. The oversizing coefficients are initialized and given a certain range in which they are to be varied as shown in Table 2.

Name	Parameter	initial value	lower bound	upper bound
pitch/diameter ratio of the propeller	beta_pro	0.33	0.3	0.6
over sizing coefficient on the load mass	k_{os}	3.2	1	40
slow down propeller coef	K_nD	1.2	1	10
over sizing coefficient on the motor torque	k_mot	1	1	20
adaption winding coef on the motor speed	k_speed_mot	1.2	1	10
ratio battery/load mass	k_mb	1	0.01	60
oversizing coefficient for voltage evaluation	k_vb	1	1	10
aspect ratio D_in/D_out for the beam of the frame	k_D	0.99	0.05	0.99

Table 2: Design Variables

Comments on the design variables initialization and bounds:

1. The battery voltage oversizing coefficient k_vb is taken to introduce a factor of safety. This coefficient lies between[1-10]. The minimum coefficient value is 1 since the battery estimate has to be equal to the ESC at the least, Hence initialized at 1.

2. The design variable k_mot is bounded between [1 -20]. It has to have a minimum value of 1 as the nominal torque has to be greater than the torque required for hovering, hence initialized at 1.

3. The motor speed oversizing coefficient, k_speed_mot quantifies a compromise between the speed and voltage at the motor is considered between [1-10] and initialized at 1.2 to begin with.

4. The payload mas has no intuitive impact on the mass of the battery and therefore, large bounds for the battery to payload ratio, K_mb were chosen: [0.01-60] and was initialized at 1, meaning the mass of the battery is equal to the payload.

5. The mass oversizing coefficient k_os was taken in [1-40] as the total load must be greater than or equal to the total mass M_tot thus making the lower bound of 1. It was initialized at 3.2 to have an oversized intial estimate.

6. The aspect ratio D_in/D_out for the beam of the frame, k_D was considered between [0.05-0.99] so that D_in doesn't exceed D_out and initialized at 0.99 considered to reduce the structure mass contributed to the MTOW.

7. The slow-down propeller coefficient, K_nD is given a range between [1-10] with a lower bound of 1 as the nD should not exceed nD_max. It was initialized at 1.2 which is slighted lower than nD_max to have good performance.

8. The pitch/diameter ratio of the propeller, beta_pro was given a bound of [0.3 to 0.6] as for a drone, the optimal pitch-to-diameter ratio is typically between 0.4 and 0.6 for a propeller that is optimized for hover and low speed, and 0.7 to 0.9 for a propeller that is optimized for high speed and forward flight.

5 Case studies

For analysis of the optimization results, two main parameters have been chosen as the objectives which the algorithm will optimize, namely, MTOW and hovering time. For both cases, the following parameters are varied and the corresponding maximized MTOW and minimized hovering time is plotted with the masses of the individual components.

- Specified maximum allowable MTOW
- Specified minimum flight time
- Specified payload

Additionally, the original specifications of the optimization algorithm are as follows. The analysis is conducted by varying these parameters one at a time while keeping rest at the original specification.

1. $MTOW \leq 360kg$
2. $FlightTime \geq 18min$
3. $Payloadmass = 4kg$

The analysis is conducted for a quadcopter, so the number of arms of the drone is set at 4.

5.1 MTOW optimization

5.1.1 Variation of specified maximum allowable MTOW

1. The variation of the masses of the components and flight time are observed as the specified MTOW increases.
2. It is observed that at $MTOW \leq 5kg$, the **optimization fails** since the drone cannot sustain a flight time of 18min with a maximum MTOW of only 5kg. The optimization reports that the constraints 3 (eqn-3) and 6 (eqn-6) are not met. This makes sense since the battery sized is too small for the length of hovering time. So, it cannot deliver enough voltage to the motors and the hovering time does not reach 18 min.
3. Once the MTOW is increased in the next data-points, we see that the values do not change at all. This is because the drone calculates the minimum mass it needs to reach in order to sustain a flight time of 18 min, which is around 6kg. This mass is sufficient for satisfying all constraints.

MTOW specified	5	10	50	100	360
MTOW achieved	5.06	6.07	6.07	6.07	6.07
Propeller mass	0.07	0.14	0.14	0.14	0.14
Motor mass	0.28	0.52	0.52	0.52	0.52
Battery mass	0.64	1.31	1.31	1.31	1.31
ESC mass	0.04	0.04	0.04	0.04	0.04
Frame mass	0.03	0.06	0.06	0.06	0.06
Flight time achieved	8.5	18	18	18	18

Table 3: Data points for variation of MTOW specs for MTOW optimization

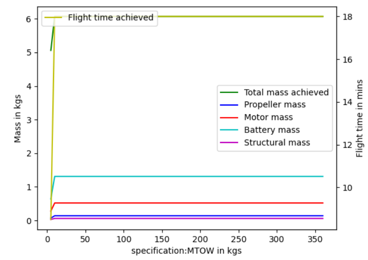

Figure 1: Plot of variation in MTOW specs for MTOW optimization

5.1.2 Variation of specified flight time

1. As the specified flight time is increased, we see a corresponding increase in the MTOW since we need a larger battery to sustain a longer flight time. We can see that the battery mass increases the most with the increased flight time.
2. The optimizer solves the equations considering the minimum flight time to achieve is equal to the specified flight time and minimizes the mass for it.
3. However, when the flight time required is increased to 100min, the **optimization fails** with constraints 1,3,6 not satisfied. The drone does not reach the 100min flight time mark and the battery voltage saturates. Constraint 1 indicates that the over-sizing coefficient k_{os} goes out of its limits of [1 40]. This means that it needs to over-size the drone further than this range to satisfy the flight time requirement.

Flight time specified	5	18	30	50	100
MTOW achieved	4.74	6.07	7.68	12.76	35
Propeller mass	0.06	0.14	0.25	0.56	1.65
Motor mass	0.24	0.52	0.92	2.2	7.81
Battery mass	0.38	1.31	2.36	5.64	19.99
ESC mass	0.04	0.04	0.04	0.06	0.14
Frame mass	0.02	0.06	0.11	0.31	1.47
Flight time achieved	5	18	30	50	73

Table 4: Data points for variation of flight time specs for MTOW optimization

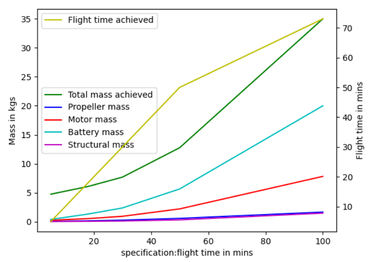

Figure 2: Plot of variation in flight time specs for MTOW optimization

5.1.3 Variation of specified payload mass

1. As expected, the MTOW increases as the payload mass is increased. The MTOW is still minimized for each case since we see the optimizer calculating the minimum specified flight time which would be the case for a minimized MTOW.

2. It is observed that at a payload mass of 185kg, the **optimization fails** with constraints 1,2, and 6 being violated. Constraints 1 and 2 fail because the over-sizing coefficient k_{os} goes out of bounds and the torque needed for takeoff exceeds the limits of the motors. Constraint 6 indicates the flight time did not reach 18min.

Payload specified	4	15	50	100	150	185
MTOW achieved	6.07	23.95	86.09	183	289.07	360
Propeller mass	0.14	0.47	1.3	2.31	3.37	4.19
Motor mass	0.52	2.38	11.04	27.07	46.96	62
Battery mass	1.31	5.54	20.89	46.87	75.29	90.52
ESC mass	0.04	0.16	0.61	1.37	2.2	2.72
Frame mass	0.06	0.41	2.26	6.06	11.25	15.57
Flight time achieved	18	18	18	18	18	17.49

Table 5: Data points for variation of payload for MTOW optimization

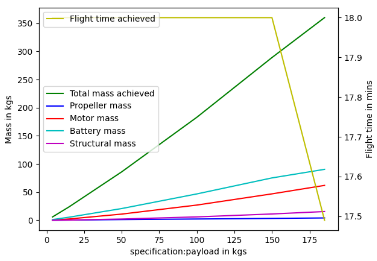

Figure 3: Plot of variation in payload specs for MTOW optimization

5.2 Hover time optimization

5.2.1 Variation of specified maximum allowable MTOW

1. In this trend, it is observed that a flight time of 18min is not achieved with only a maximum MTOW of 5kg and the **optimization fails** with constraints 3 and 6 being violated. A very low MTOW means that the battery is too small and lacks the capacity to supply the total power required.

2. When the MTOW is increased however, the optimizer is able to maximize the flight time without violating any constraints.

3. One important thing to note is that once the MTOW reaches 50kg, there is not much increase in flight time. This is because the weight of the battery is too heavy to compensate for it with the power it is supplying. Thus, the flight time does not change.

MTOW specified	5	10	50	100	360
MTOW achieved	5.06	10	50	57.43	57.43
Propeller mass	0.07	0.39	2.22	2.49	2.49
Motor mass	0.28	1.5	11.56	13.42	13.42
Battery mass	0.64	3.86	29.53	34.28	34.28
ESC mass	0.04	0.05	0.23	0.27	0.27
Frame mass	0.03	0.2	2.46	2.98	2.98
Flight time achieved	8.5	41.5	68.18	68.27	68.27

Table 6: Data points for variation of MTOW specs for flight time optimization

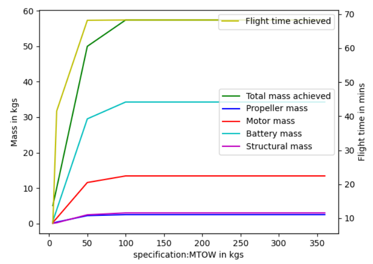

Figure 4: Plot of variation of the MTOW specs for flight time optimization

5.2.2 Variation of specified flight time

1. The minimum specified flight time does not have any effect on the optimized parameters upto a certain point. The optimizer is able to easily satisfy the default minimum flight time condition of 18min and maximize the flight time to about 68min.

2. However, once the minimum specified flight time reaches 100min, the **optimization fails**. The optimizer tries to reach this value by reducing the MTOW, which in this case is not sufficient since the battery is not large enough to sustain such a high flight time. Hence, we observe that constraints 1,3 and 6 have been violated.

Flight time specified	5	18	30	50	100
MTOW achieved	57.43	57.43	57.43	57.43	7.93
Propeller mass	2.49	2.49	2.49	2.49	0.32
Motor mass	13.42	13.42	13.42	13.42	0.94
Battery mass	34.28	34.28	34.28	34.28	2.57
ESC mass	0.27	0.27	0.27	0.27	0.01
Frame mass	2.98	2.98	2.98	2.98	0.09
Flight time achieved	68.27	68.27	68.27	68.27	96.26

Table 7: Data points for variation of flight time specs for flight time optimization

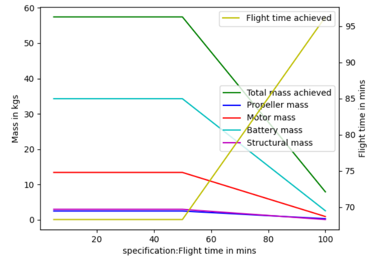

Figure 5: Plot of variation of the flight time specs for flight time optimization

5.2.3 Variation of specified payload mass

1. As we increase the payload mass of the drone, there is an increase in the MTOW but at the cost of maximum flight time which keeps decreasing.

2. At a payload mass of 50kg, the MTOW reaches its maximum value and thus, the components and the battery sizes cannot be increased further to accommodate more flight time. This contributes to a faster decrease of maximum flight time.

3. As the payload mass reaches 185kg, the **optimization fails** violating every constraint. This is because, simultaneously, the MTOW and the minimum flight time parameters have crossed their respective bounds of 360kg and 18min.

Payload specified	4	15	50	100	150	185
MTOW achieved	57.43	189.76	360	360	360	362
Propellers mass	2.49	5.92	8.34	6.59	4.87	4.2
Motor mass	13.42	42.82	53.52	52.54	66.35	62.14
Battery mass	34.28	111.43	188.36	152.81	118.52	92.32
ESC mass	0.27	1.03	2.11	2.3	2.57	2.72
Frame mass	2.98	13.57	27.67	22.76	17.69	15.63
Flight Time achieved	68.27	57.83	47.16	35.13	24.25	17.78

Table 8: Data points for variation of payload specs for flight time optimization

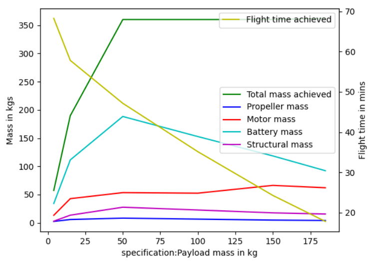

Figure 6: Plot of variation of the payload specs for flight time optimization

6 Conclusion

In conclusion, the assignment focuses on an optimization problem for sizing the components of a drone, specifically for a quadcopter, using the gradient search algorithm. The optimization process is done for two objectives, total mass and flight time and it is done by updating the parameters in the loop while satisfying all the constraints. The results are analyzed through case studies where the variation in MTOW and flight time are noted. The analysis is conducted by varying specified maximum allowable MTOW, specified minimum flight time, and specified payload. The results of the case studies provide useful insights for optimizing the design of a drone to meet certain performance requirements.