Team 10
OPTIMAL DESIGN OF QUADCOPTOR

By

Preeti Vaidya (1207863185)

Gaurav Pokharkar (1207644278)

Nikhil Sonawane (1207802748)

MAE 598

Final Report

Abstract:

This project was undertaken to study quadcopters or quad rotors which are like miniature helicopters with four rotors. Quad copters use two sets of fixed pitch propellers, two clockwise and two counter clockwise. These use variations of rpm to control torque. Control of vehicle motion is achieved by altering the rotation rates of one or more discs, thereby changing its torque load and thrust/lift characteristics. These devices are becoming increasingly popular in recent times as they offer good maneuverability, increased payload carrying capacity with simple mechanics and moderate cost as compared to actual helicopters or other UAVs. They have vast applications ranging from surveillance, disaster aid and rescue to crop survey in different parts. Current quadcopter models have persistent issues in vertical flights, torque induced control issues and high energy consumptions. Therefore, this project was outlined to study different quadcopter systems to increase the energy efficiency in the same power for given payload capacity and size constraints.

Table of Contents

Table of Figures

Design problem statement

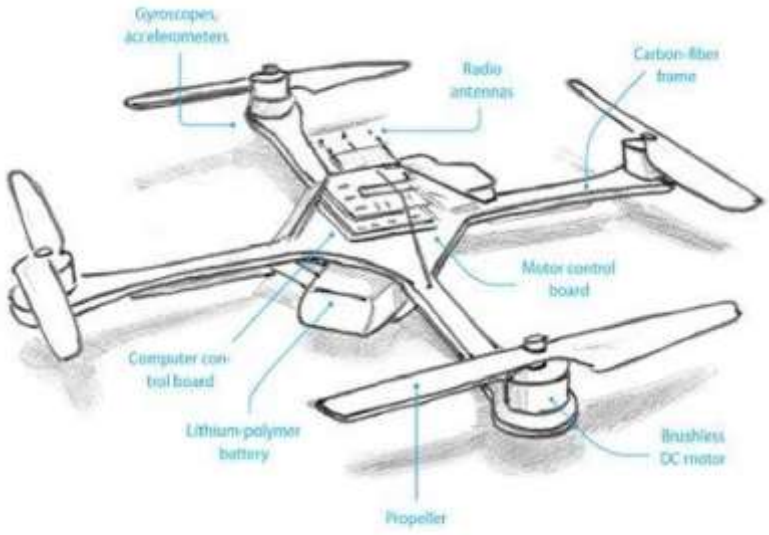
As stated in the abstract current quadcopters models face issues like unsteady flight, failure of structural components and high energy consumptions during flight. These issues limit large scale use of quadcoptors. Keeping this in mind, the design problem focuses on increasing the efficiency of operation of quadcopter by improving efficiency of propulsion system, minimizing the overall weight of the frame and optimum placement of different components in a quadcopters.

This project proposes a practical method to handle the design problem of a small-scale rotorcraft by combining the theoretical knowledge of the system and the result of a system level optimization analysis. The method is driven by the application. We define a target size and weight for the system and the setup a model to choose the best components to be used and estimate iteratively the most important design parameters.

The subsystems we will be working upon are broadly classified into following parts-

1. Optimization of the frame design
2. Optimization of propulsion system
3. Component placement optimization and path optimization

Optimization of each sub-system independently will lead to sub-optimization of the system. As all the systems are inter linked and interconnected, design of one system will affect the design of another system. Hence, to optimize the performance of the quadcopter it necessary to study the effect of all the systems on the overall performance and do trade-offs wherever required in order to get overall optimal solution.



Subsystem scope:

- Optimization of the frame design
    - Analysis of frame design with different types of cross section
    - Selection of optimal frame configuration
- Optimization of propulsion system
    - Rotor
    - Motor
    - Battery
- Component placement optimization and path optimization
    - Optimal placement of objects on frame

Subsystem 1: Optimization of the frame design (Gaurav Pokharkar)

The frame is the basic structure which holds all the components of the quadcopter i.e. motors, battery, control circuits, camera etc. Due to this reason its design becomes an important parameter affecting the performance of the quadcopter. The strength of the frame, its weight, its material, its configuration and various other factors are to be considered while designing the frame. We aim at optimizing the frame so as to increase strength and reduce its weight and manufacturing cost which would increase the operation time, minimize overall weight of the quadcopter giving it robustness and rigidity and also it would be economical to manufacture.

Secondly, the motors and rotor rotation causes vibrations in the quadcopter frame. It is necessary to check the frame for vibrations analysis to ensure that applied frequency does not coincide with the natural modes of frequency of the frame. If this happens it will lead to resonance.

Subsystem 2: Optimization of propulsion system (Preeti Vaidya)

Rotor, motor and battery constitute the major components of the propulsion system. Quadrotor derives its thrust from the rotation of rotor blades such as a helicopter does. This subsystem mainly aims at introducing rotor aerodynamics and identifying methods to improve the thrust producing capacity of a quadrotor, hence improving its performance and efficiency. A fundamental and feasible route for realizing this performance improvement is identified by examining rotor selection based on theoretical models, such as the combined momentum and blade element theory, and on experimental data extracted from propulsion tests performed in the laboratory.

The rotor optimization can be based either on the blade element theory or combined momentum or by performing propulsion tests. Each propulsion unit must be able to lift its own weight, one quarter of the electronics, structure and payload weight, while supplying a residual amount of

thrust sufficient for hover stability and maneuverability. Much of the design effort falls in the proper combination of the batteries, motor, and propellers to produce an efficient propulsion unit. Hence, the following propulsion component analysis is to be performed to help in their proper selection

Subsystem 3: Component placement optimization (Nikhil Sonawane)

Optimal placement of objects on frame

Various components placed on the frame contribute to the balancing of the quadcopter which is the most important factor to be worked upon. The components mounted will not only add to the weight of the system but also affect the center of gravity which has to coincide with frame center of gravity for maximum stability. Thus figuring out the distance at which the components are placed and their exact position would play a vital role in optimizing the design of quadcopter.

Nomenclature

m: Mass of the quadcopter (kg)

P: Payload to be carried (kg)

Mb: Battery weight (kg)

I: Mass moment of inertia (kg.mm^2)

Subsystem 1: Optimization of the frame design (Gaurav Pokharkar)

m: Mass of the quadcopter (Kg)

F: Thrust force due to motor (N)

C_m: Manufacturing cost ($/m^3)

C_r: Cost of raw material ($/kg)

C_p: Cost for precision ($)

L: Length of the arm (m)

M: Moment due to force (F) (N-m)

f: Frequency of the motors (Hz)

f1, f2, f3, f4, f5: First five natural frequencies of the cantilever beam (Hz)

σ: Bending stress (N/m^2)

I: Area moment of inertia of the cross-section (m^4)

L: Length of the arm (m)

ρ: Density of the material (kg/m^3)

E: Young's modulus (N/m^2)

σ_t: Tensile strength of the material (N/m^2)

σ_D: Design strength of the material (N/m^2)

FOS: Figure of safety

W_R: Weight of raw material (kg)

V_R: Volume of material removed (m^3)

W_F: Final weight of arm (kg)

OD: Outer Diameter of the arm (m)

ID: Inner Diameter of the arm (m)

A: Cross-section area of the beam (m^2)

y: Distance of the outer most filament from the center (OD/2) (m)

Subsystem 2: Optimization of propulsion system (Preeti Vaidya)

T_1, T_2, T_3, T_4: Thrust force developed by rotors (N)

T: Total thrust force developed (N)

M: Torque required by individual propellers (Nm)

n: Rotational speed of the rotors (rad/sec)

N: Rotational speed of the rotors (rpm)

V: translational velocity of quadcopter (m/s)

D: diameter of propeller or rotor

ρ: density of air (kg/m^3)

Cp: co-efficient of power

Ct: co − efficient of thrust T

Cq: co − efficient of torque M

J: advance ratio

α: angle of attack (radians)

θ: geometric pitch angle (radians)

φ: local inflow angle

p: pitch of the propeller blade

Vinf: forward velocity of the quadcopters

V_2: angular flow velocity

a: axial inflow factor

b: angular inflow factor

Cl: Lift co-efficient

Cd: Drag co-efficient

t: time of flight to cover specified area (min)

L: Maximum allowable length of arms (mm)

B: maximum width of the quadcopter (mm)

H: maximum height of the quadcopter (mm)

h: Height at which the quadcopter is traversing

C: total cost for manufacture of quadcopter ($)

g: acceleration due to gravity (m/s^2)

Vw: wind velocity (m/s)

Subsystem 3: Component placement optimization (Nikhil Sonawane)

Mx :moment about X axis (N-mm)

My: moment about Y axis (N-mm)

Xcg: X coordinate of cg of a component

Ycg: Y coordinate of cg of a component

g: Inequality constraints

h: equality constraints

Subsystem 1: Optimization of structural frame design (Gaurav Pokharkar)

Mathematical model

Objective

The main aim of the optimization problem is to reduce the overall weight and cost of manufacturing of the frame of the quadcopter.To minimize both cost and weight of the quadcopter we combine with higher weight assigned to the weight of the quadcopter. The objective function is as follows:

min 0.25 *(C_m* V_R + C_p + C_r * W_R) +0.75*(W_F)

Material was assumed to be as Aluminum as it light in weight and also has good tensile strength.

Constraints

The thrust due to motor act at the tips of the arms and the weight of the quadcopter acts at the center of the frame. Hence the arms behave as cantilever beam. Initially assuming to be as 8-9 N

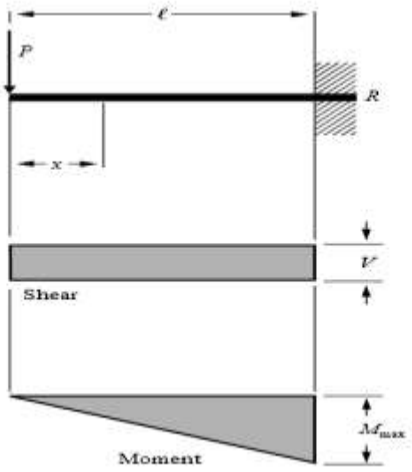


Figure 1 Cantilever Beam

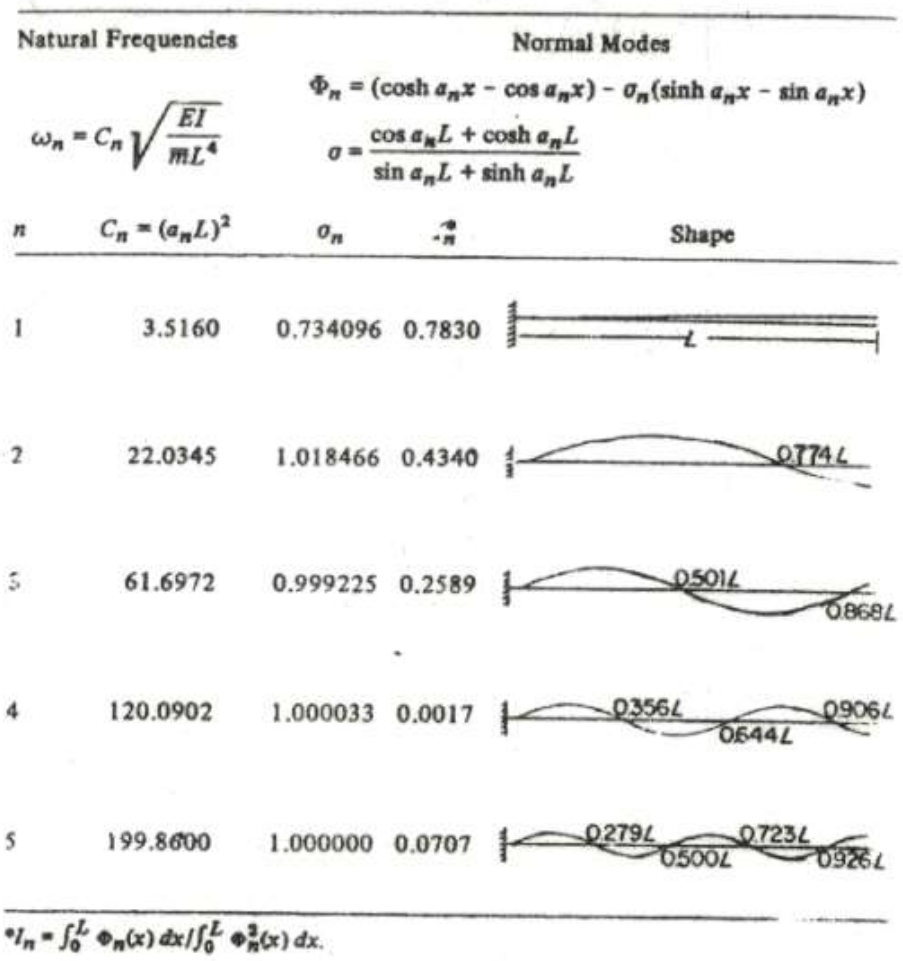
The bending stress due to the thrust generated by motor should be less than the design stress. Assuming the factor of safety (FOS) to be about 2.

As the beam is under bending we know, [1]

$$\sigma = \frac{M * y}{I}$$

Also, the rod can also fail due to resonance when the natural frequency of the rod matches with that of the motor. Hence the rod should be designed in such a way that the natural frequency of the rod is different than the frequency of the motor. [2]

The natural frequency of the cantilever beam is given as: [3]



Natural Frequencies		Normal Modes		
$\omega_n = C_n\sqrt{\frac{EI}{mL^4}}$		$\Phi_n = (\cosh a_n x - \cos a_n x) - \sigma_n(\sinh a_n x - \sin a_n x)$ $\sigma = \frac{\cos a_n L + \cosh a_n L}{\sin a_n L + \sinh a_n L}$		
n	$C_n = (a_n L)^2$	σ_n	$\bar{x}_n$	Shape
1	3.5160	0.734096	0.7830	L
2	22.0345	1.018466	0.4340	0.774L
3	61.6972	0.999225	0.2589	0.501L, 0.868L
4	120.0902	1.000033	0.0017	0.356L, 0.644L, 0.906L
5	199.8600	1.000000	0.0707	0.279L, 0.500L, 0.723L, 0.926L

$^*I_n = \int_0^L \Phi_n(x)\,dx / \int_0^L \Phi_n^2(x)\,dx.$

Figure 2 Natural Frequency Modes of a Cantilever Beam

Some of the common types of cross sections are as follows:

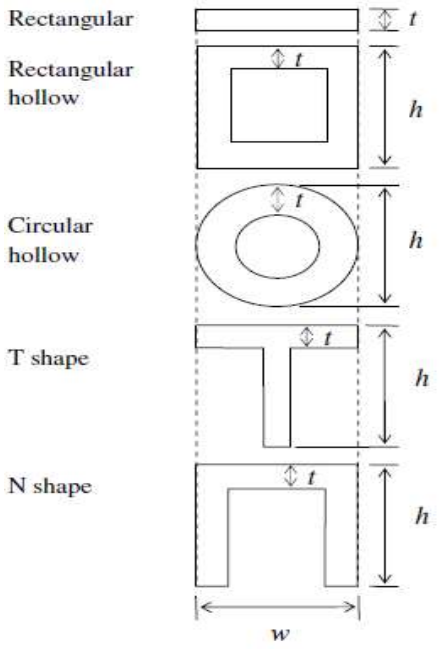


Figure 3 Common cross sections used for frame of quadcopters

SelectingHollow circular tube as the cross section of the arms as it has largest inertia for the same weight/area also it is symmetric.

Hence

$$\mathrm{A} = 0.25 * \mathrm{pi} * (OD^2 - ID^2)$$

and

$$I = \frac{pi * (OD^4 - ID^4)}{64}$$

Assuming that the length of the arms should be at least 200 mm so as to have enough space to mount all the components.

As the material chosen is Aluminum the material properties are as follows: [4]

σ_t = 276 MPa

ρ = 2700 kg/m^3

E = 6.9e10 N/mm^2

Assuming the FOS as 2.

Hence $\sigma_D = \sigma_t/2 \sim 120$ N/mm^2 and $\sigma < \sigma_D$

Assuming the frequency of the motor as 120 rps. Hence the natural frequency of the arm should not fall between 90 rps to 150 rps.

Assuming the cost of manufacturing as:

$C_m = 2$ \$/m^3

$C_p = 0.5$ \$

$C_r = 4$ \$/kg

From manufacturing point of view the minimum thickness of the rod should be 4 mm.

Hence the optimization problem is as follows:

Min 0.25 *(C_m * V_R + (C_p * L/ID) + C_r * W_R) + 0.75 * (W_F)

$$\min 0.25 * \left(C_m * \frac{pi * ID^2 * L}{4} + \left(C_p * \frac{L}{ID} \right) + C_r * \frac{pi * OD^2 * L}{4} \right) + \frac{0.75 * pi * (OD^2 - ID^2) * L * \rho}{4}$$

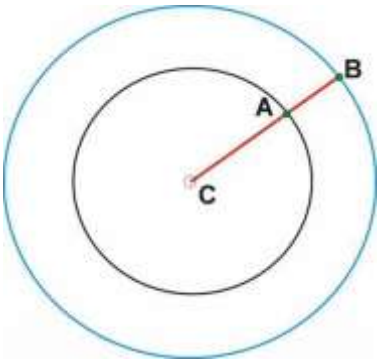


Figure 4 Cross-section of the arm

Where CB is OD and CA is ID

Subject to-

g1:

OD-ID $\geq$ 4 mm

g2:

$$\frac{F * L * OD * 64}{(OD^4 - ID^4) * 2} \leq 120\ N/mm^2$$

g3:

f1 ≥ 150 rps

g4:

f2 ≥ 150 rps

g5:

f3≥ 150 rps

g6:

f4 ≥ 150 rps

g7:

f5 ≥ 150 rps

Where,

$$f_n = a_n * \sqrt{\frac{E * \left(OD^4 - ID^4\right) * 4}{\left(OD^2 - ID^2\right) * 64 * \rho * L^4}}$$

For n = 1 to 5, a_1= 3.516,a_2=22.0345,a_3= 61.69,a_4=120.09,a_5= 199.86

g8:

OD ≥ 4 mm

g9:

ID ≥ 4 mm

Model analysis

Functional Dependency Table

	OD	ID	L	F	Motor Speed (rps)	Material Properties
f	x	x	x			
g1	x	x		x		
g2	x	x	x			x
g3	x	x	x		x	x
g4	x	x	x		x	x
g5	x	x	x		x	x
g6	x	x	x		x	x
g7	x	x	x		x	x
g8	x					
g9		x				

The functional dependency table shows the dependency of the objective and constraints on various variables and parameters.This table gives an initial idea about various interactions between the variables and the objective and constraints.

The table does not include all the parameters, except yield strength of rubber, because they appear only in the objective function and not in the constraints.

Monotonicity Analysis

Activity analysis of the constraints was carried using monotonicity analysis. The table below gives the summary about the same. The highlighted rows shows the constraints that are active.

	OD	ID
f	+	-
g1	-	+
g2	-	+
g3	+	+
g4	-	-
g5	-	-
g6	-	-
g7	-	-
g8	-	
g9		-

As it can be seen from the table that g1is active constraints. Also, later it can compared with the numerical results that are obtained.

The Lagrange's multiplier obtained from excel solver as follows:

Constraints

Cell	Name	Final Value	Lagrange Multiplier
$C$16	Sigma L	4.2313558	0
$C$8	L L	300	0
$E$11	mm od	3.000000002	0.071770392
$N$4	Frequency 1st Mode	341.9384265	0
$O$4:$R$4 >= $J$20			
$O$4	Frequency 2nd Mode	2142.901666	0
$P$4	Frequency 3rd Mode	6000.183017	0
$Q$4	Frequency 4th Mode	11679.0256	0
$R$4	Frequency 5th Mode	19436.80714	0

Figure 5 Excel output

Hence it can be seen that the analytical result matches with the Excel numerical result. Hence KKT the solution obtained is a KKT point. [5]

Numerical Results

Optimization study

The model for optimization was developed and later solved using GRG algorithm and the results obtained are as follows:

Variables	Start(1)	Optimum values	Start(2)	Optimum values	Start(3)	Optimum values
OD	10	29.7219198	30	29.7219198	100	29.7219198
ID	6	26.7219198	26	26.7219198	80	26.7219198
Function Value	4.22943852	1.030276804	1.41454638	1.030276804	5.6992633	1.030276804

Hence the optimum values of the variables is

OD – 30 mm

ID – 25 mm

L – 200mm

Parametric Study

The plots below shows the relation between the various parameters and the objective function.

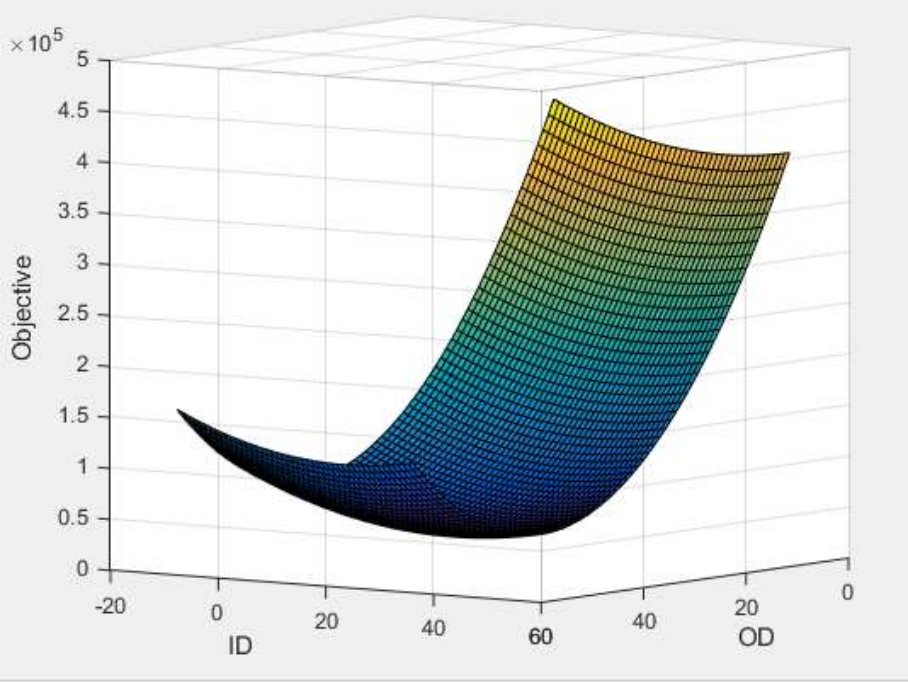


Figure 6 Objective v/s OD and ID

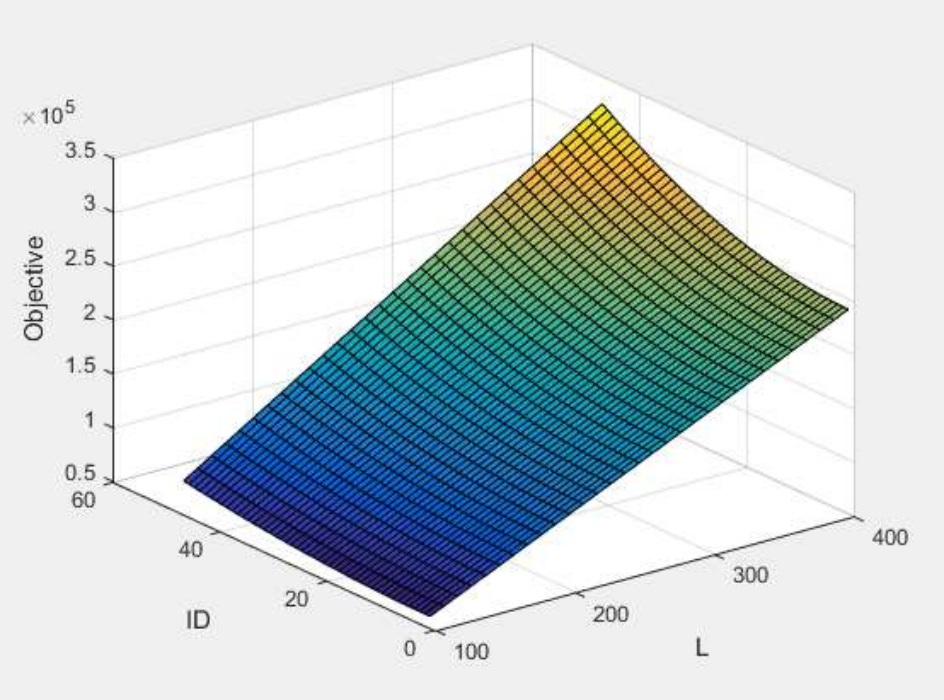


Figure 7 Objective v/s L and ID

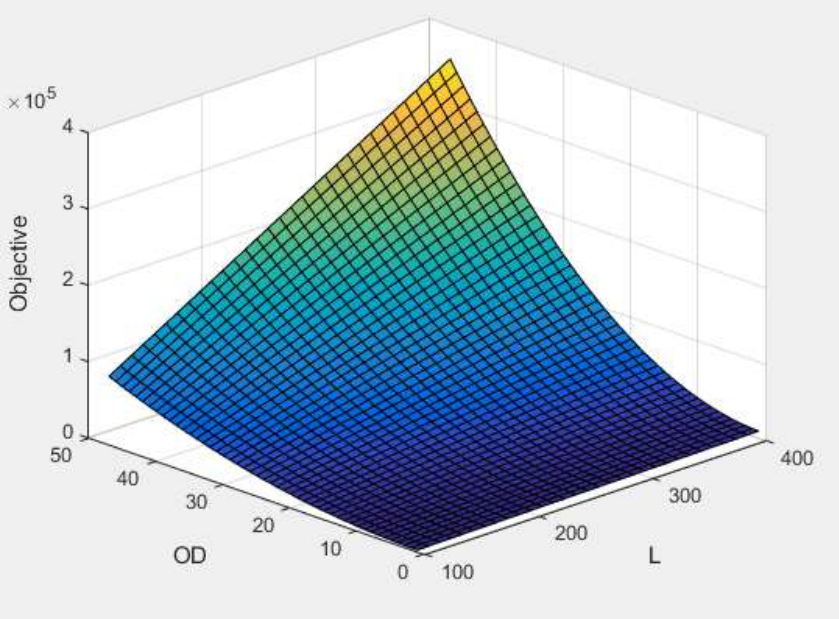


Figure 8 Objective v/s OD and L

As it can be seen from the above figures that the objective function varies non-linearly with respect to OD and ID and almost linearly with respect to L.

	Length (decreases)
f	decreases
g1	-
g2	decreases
g3	increases
g4	increases
g5	increases
g6	increases
g7	increases
g8	-
g9	-

Hence keeping L as constant depending upon the constraint and not variable. Because as L tends to zero stress tends to zero the objective function tends to zero and also the frequency tends to infinity. But L zero is and infeasible solution but it satisfies all the constraints. Hence keeping L constant we iterate for optimal values of OD and ID.

Discussion of Results

The results obtained depend upon the initial assumption for thrust force as 9 N and motor speed about 120 rps. But if these parameters change then it will affect this solution.

The thrust is dependent upon the motor, weight of the quadcopter etc. The speed of the motor also changes the forced vibration imposed upon it by the motor. If the length of the arm is kept as 200 mm it can lead to collision of the propeller with the components or other propellers. The length of the arm affects the components placement and propeller selection as well. For overall system level optimization out of these parameters it was decided to have length of arm s 300 mm and force as 15 N (1.5 kg) to be generated by one motor, moto speed as 11000 rpm (166.67 rps) causing the quadcopter to move at a velocity of about 7 m/s.

The transportation cost of the material required to manufacture the frame can be incorporated in the model to get an exact estimate of the cost and weight of the quadcopter and can lead to further optimization which can be used to optimize mass manufacturing of the quadcopter. The model can be made comprehensive by optimizing the material selection procedure and by considering a truss structure for arms or by using topology optimization software.

The results obtained for stress were compared to the ones obtained from soliworks hence the design is safe.

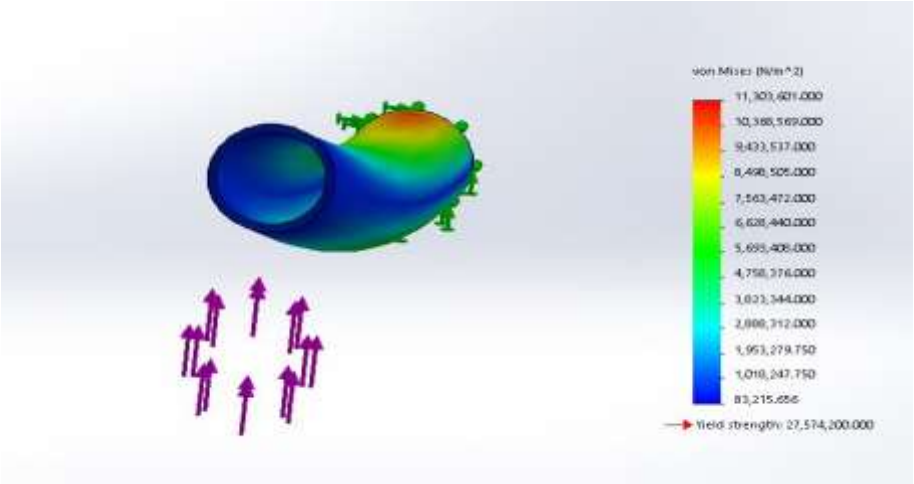


Figure 9 Solidworks Output

Subsystem 2: Optimization of propulsion system (Preeti Vaidya)

Mathematical model

Propulsion system provides the thrust force for quadcopters motion and as such, it is the main system whose performance indirectly influences the performance of the combined system. Each of the four propulsion systems is composed of a rotor or propeller, a motor and battery pack. The propulsion system interfaces with the main controller through electronic system controller (ESC).

Propeller design optimization:

Efficient propellers put less strain on motor and battery and thus improve quadcopter characteristics. Propeller performance is described by quantities like thrust T (N), torque M (Nm) and power P (W) needed from the motor. The figure of merit (FoM) relating the aerodynamic power generated and mechanical energy consumed by propeller is the efficiency of propeller. Therefore, the objective is selected to maximize the efficiency of propeller for given payload capacity and given aerofoil type.

Target requirements: The payload capacity of the quadcopter is expected to be 3 kg which divides up in to 1.5kg minimum thrust from each propeller. Assumptions that are considered while formulating the optimization problem are that the maximum velocity of the quadcopters should be limited to 7 m/sec; spatial dimensions of the propeller are obtained from structural frame studies and density of air is 1.225 kg/m^3. These assumptions are also accounted for in the constraints.

Theoretical studies show that the propeller thrust and efficiency depends on the diameter and angle of attack. Therefore, parametric study of efficiency as function of these two parameters can give us the best possible combination of diameter and angle of attack to select optimum propeller design for given payload capacity requirements.

Formulating the optimization problem in negative null form, the mathematical model based on Blade Element theory is as follows-

Objective function:

$$Min\ f(x) = -\left(\frac{Ct}{Cq}\right) * \left(\frac{J}{2 * pi}\right)$$

Constraints: $Ct = (\sum \Delta T)/(\rho * n^2 * D^4)$ $\qquad g1 = Ct > 0$

$Cq = (\sum \Delta M)/(\rho * n^2 * D^5)$

$g2 = \Delta T \geq 1.5\ kg$

$J = \frac{V}{n * D}$

$g3 = D < arm\ length\ (14\ inch)$ $\qquad g4 = D > 8\ inch$

Variables: The input variables for optimizing the rotors are rotor diameter D, angle of attack α.

Parameters: density of air ρ, aerofoil blade shape, payload to be carried, maximum forward velocity of the quadcopters.

Bounds on variables: $g5 = \alpha \geq 4°$ $g6 = \alpha \leq 20°$
$D \leq 14"$ $D \geq 8"$

Here the parameter α is varied between 4-20°. This range is selected from general studies of existing propeller theory which proves that angle of attack is usually never more than 16-18° as the drag forces are very high for high α.

For this model, all the constraints are inequality constraints as we define only starting and ending values of ranges. As the variables are two, this system has two degrees of freedom which are studied together.

	Variables		Parameters		
	D	α	Ct	Cq	J
f	X	X	X	X	X
g1	X	X	X		
g2	X	X			
g3	X				
g4	X				
g5		X			
g6		X			

The functional dependency table shows the appearance of different variables in the objective function and constraints. This table gives an initial understanding of various interactions between the variables. It shows that efficiency depends on Ct, Cq and J which in turn depend on D and α. Constraints given also put limits on D and α which automatically affects Ct, Cq, J and efficiency.

Modification in initial problem formulation:

The optimization problem developed in the initial period of the project focused only on the Ct and Cq parameters by neglecting the advance ratio. However, this assumption was found to be flawed, as J is not constant and varies as diameter and α vary. Without J efficiency is monotonically decreasing with respect to ratio of Ct and Cq as diameter is increased. However, efficiency should increase for increasing diameter and thus the problem formulation was

incorrect. This flaw was detected during first round of optimization trials and then the objective function was modified and J was included in the formulation.

Model Analysis

Theoretical background for optimization problem:

The thrust, torque and advance ratio cannot be explicitly found for a particular propeller blade using any single equation as the pitch, diameter and the chord thickness vary throughout the blade profile. As a result, the lift and drag forces generated in a blade vary throughout the blade. There are several theories quantifying the relations between these parameters. Blade Element theory is used for this optimization study. [5]

Design Model theory and assumptions:

Assuming Blade Element theory [6] for this study,the propeller is divided into a number of independent sections along the length. At each section a force balance equation is applied involving 2D section lift and drag with the thrust and torque produced by the section. At the same time a balance of axial and angular momentum is applied. This produces a set of non-linear equations that can be solved by iteration for each blade section. The resulting values of section thrust and torque can be summed to predict the overall performance of the propeller.

The theory does not include secondary effects such as 3-D flow velocities induced on the propeller by the shed tip vortex or radial components of flow induced by angular acceleration due to the rotation of the propeller.In spite of the above limitations, it is still the best tool available for getting good first order predictions of thrust, torque and efficiency for propellers.

Force balance at each section is shown for one arbitrary section below-

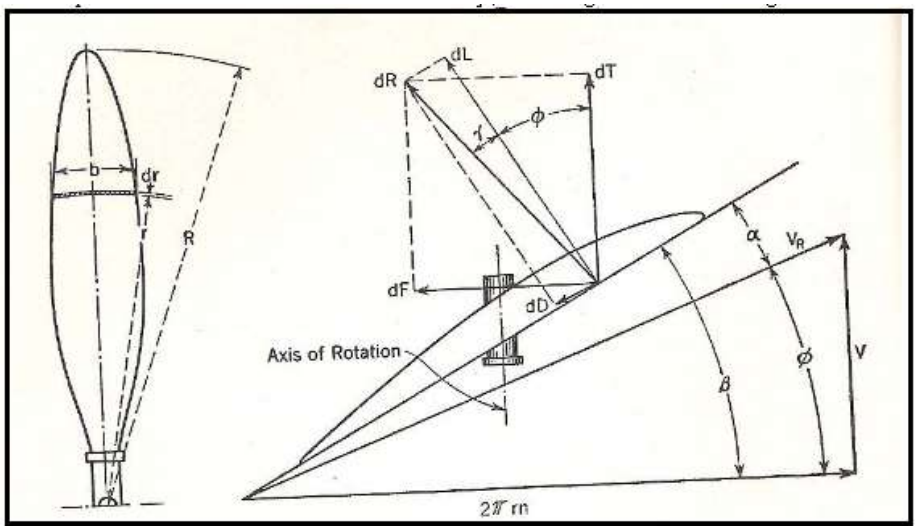


Figure 10 Aero foil blade profile and force balance

The principles of force balance, axial and angular flow conservation of momentum are applied and the thrust and torque over that section are calculated using these following set of main equations-

$$\Delta T = \rho * 4 * \pi * r * {V_{inf}}^2 * a * (1 + a) * dr$$

$$\Delta Q = \rho * 4 * \pi * r * {V_{inf}}^2 * b * (1 + a) * dr$$

$$V_1 = \sqrt{{V_0}^2 + {V_2}^2}$$

$$\alpha = \theta - \tan^{-1}(\frac{V_0}{V_2})$$

$$Cl \propto \alpha \qquad Cd \propto \alpha^2$$

$$\Delta T = \frac{1}{2} * \rho * {V_1}^2 * c * (C_l cos\varphi - C_d sin\varphi) * B * dr$$

$$\Delta Q = \frac{1}{2} * \rho * {V_1}^2 * c * (C_l cos\varphi + C_d sin\varphi) * B * r * dr$$

The detailed theory is attached as appendix at the end. A general flowchart depicting the sequence of calculations and inter-relations is as follows-

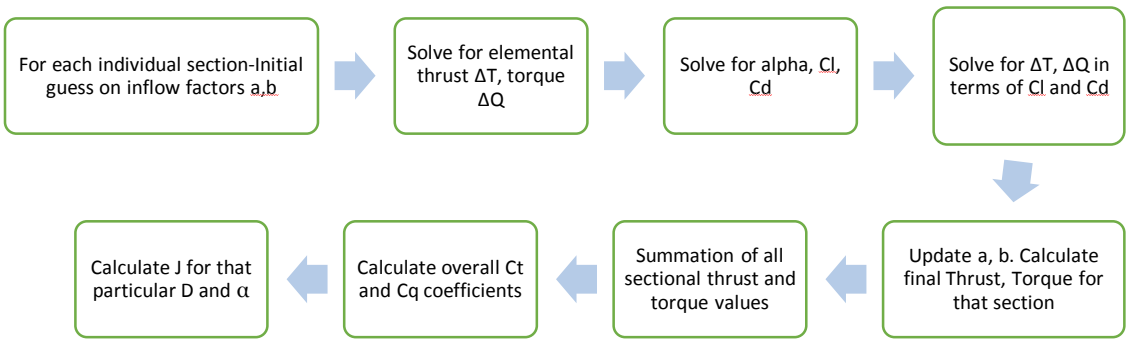


Figure 11Flow chart to show flow of logic in design model code

A separate function file is written in MATLAB for calculating the Ct, Cq and J values which could then be used to compute the efficiency in the objective function. The MATLAB code for the same is attached in the end in the appendix.

This design model is simplified by assuming a basic simplified theory for propeller design. This has been done to simplify the numerical calculations. In the code written, the equations for lift and drag coefficients (C_l and C_d) used to calculate lift and drag forces at each section are obtained by metamodeling. The propeller blade profile selected is scaled model of NACA 4415 and is assumed to have similar lift-drag characteristics. Experimentally measured values of C_l and C_d at different α for scaled model of NACA 4415 aero foil were considered from JAVAFOIL website [7] and NACA report 824. Basic curve fitting with good R^2 measure was performed in Excel and the same equation was used to compute different Cl and Cd for different α during the design model calculations in MATLAB.

The results of curve fitting are obtained as shown below-

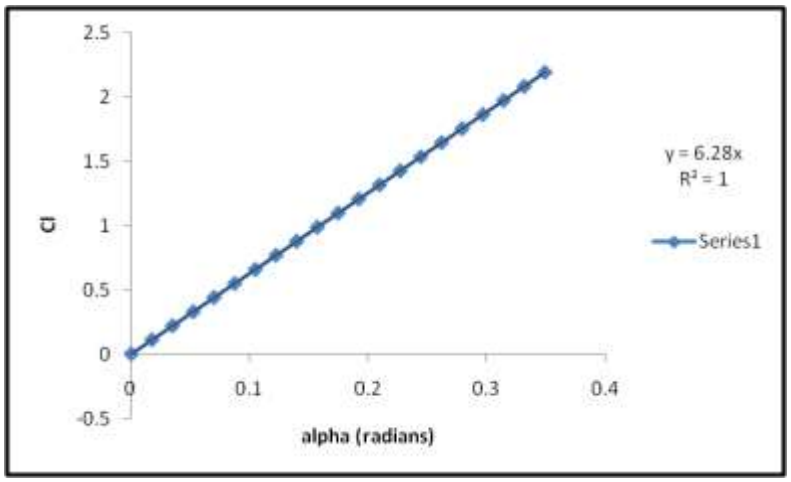


Figure 12 Curve fitting of Cl vs alpha data using metamodeling

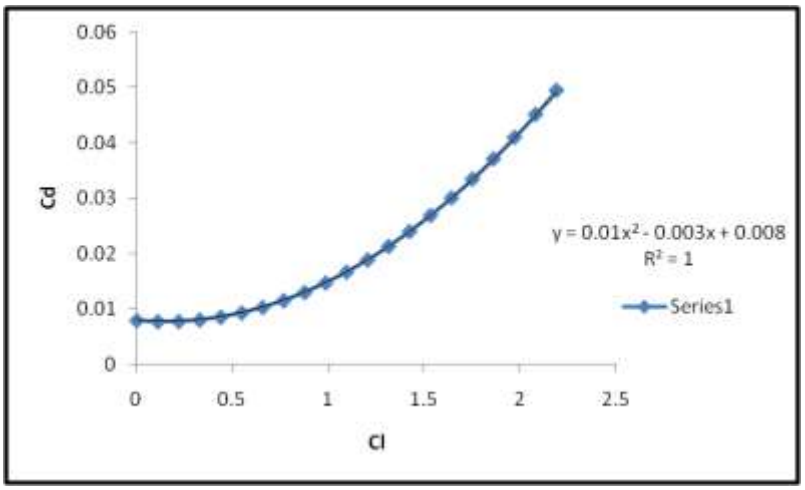


Figure 13 Curve fitting of Cd vs Cl data from metamodeling

Data used for meta-modeling is attached as follows:

alpha (deg)	alpha (radian)	cl	cd
0	0	0	0.008
1	0.017444	0.109551	0.007791
2	0.034889	0.219102	0.007823
3	0.052333	0.328653	0.008094
4	0.069778	0.438204	0.008606
5	0.087222	0.547756	0.009357
6	0.104667	0.657307	0.010349
7	0.122111	0.766858	0.01158
8	0.139556	0.876409	0.013052
9	0.157	0.98596	0.014763
10	0.174444	1.095511	0.016715
11	0.191889	1.205062	0.018907
12	0.209333	1.314613	0.021338
13	0.226778	1.424164	0.02401
14	0.244222	1.533716	0.026922
15	0.261667	1.643267	0.030073
16	0.279111	1.752818	0.033465
17	0.296556	1.862369	0.037097
18	0.314	1.97192	0.040969
19	0.331444	2.081471	0.045081
20	0.348889	2.191022	0.049433
21	0.296556	1.862369	0.037097
22	0.314	1.97192	0.040969
23	0.331444	2.081471	0.045081
24	0.348889	2.191022	0.049433
25	0.296556	1.862369	0.037097

As diameter is varied, it is revealed that the efficiency is monotonically increasing with ratio of (Ct/Cq), however it varies non-uniformly or non-monotonically with respect to advance ration J. This was the problem in the first optimization trials. As J was not considered during the optimization conducted using fmincon solver on MATLAB (only ratio of Ct/Cq was considered), the minimum objective value was always obtained on the lower boundary and it depended on the lower bound (which was p/D >0).

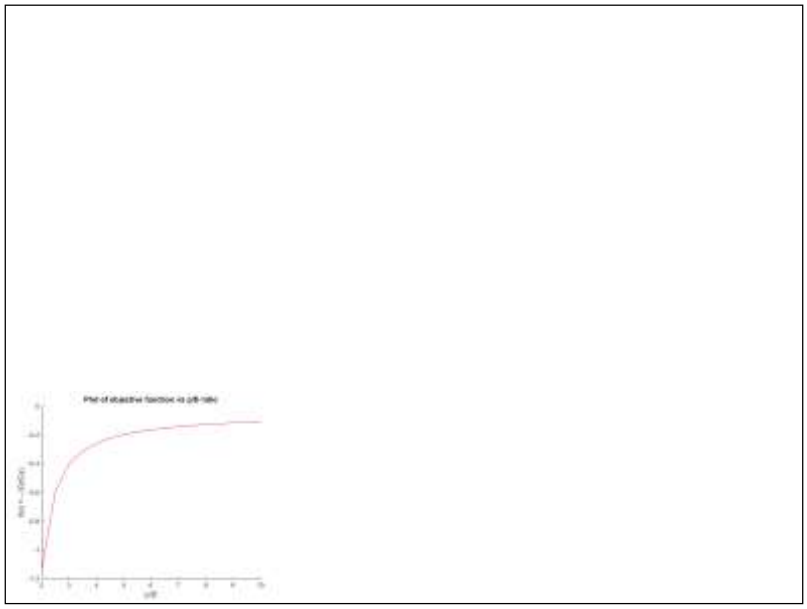


Figure 14 Monotonically increasing objective function wrt p/D ratio

During the analysis process this error was detected and later on corrected in the subsequent optimization trials.

Optimization study

Detailed calculations are shown for the final iteration which is the iteration where in the sub-system optimization is performed using the global targets for total quadcopter optimization. Optimization was performed using fmincon solver in MATLAB using the Sequential Quadratic Programming (SQP) algorithm. As per standard fmincon command format for unconstrained optimization problem with nonlinear constraints, separate codes for the objective function and nonlinear constraints were written. The upper and lower bounds of the parameters to be varied were taken as D=8" to 14" and alpha = 2° to 14°. The code was run at three different starting points and the results obtained were as follows:

Starting diameter (inch)	Starting alpha (degree)	Final diameter (inch)	Final alpha (deg)	Function value	Exit flag1
9.5	6	12.3717	13.6372	-0.5632	2
11.5	10.2	11.662	13.36	-0.5526	2
12.8	14	12.35	13.6	-0.5841	1

The three initial guesses were taken as one set closer to the lower bounds, one closer to the upper bounds and one in the local region where the minimum appears to lie. The screenshot of the first iteration output is given below-

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.



xopt =

   12.3717
   13.6372


fval =

   -0.5632


exitflag1 =

     2
```

The solver message displays ‘local minimum possible’ as the solver might have reached a local minimum, but cannot be certain because the first-order optimality measure is not less than the TolFun tolerance. Some of the measures suggested in such cases are trying a different algorithm, checking nearby point in the region of interest or providing analytic gradients.

Different algorithm like Trust region algorithm cannot be used because it needs gradient of the objective function along with the objective function definition. In current case, the objective function gradient in terms of variables D and alpha cannot be explicitly computed. However, points in the area of interest can be sampled to check where the minimum function value is observed. The third initial guess indicates this approach and it is seen that the minimum function value is obtained there. The exit flag value at that has also improved from 2 to 1. Exit flag indicates satisfaction of first order optimality condition. On the scale of 1 to 5, exit flag value of 2 indicates a more satisfactory performance than when it is 5.

The minimum usually found by these methods (SQP, Trust region) are local minimum. They can also be the global minimum, but that has to be verified. Here, explicit gradient or Hessian of the objective function with respect to the variables cannot be confirmed whether the minimum is in fact global minimum.

Discussion of Results

The plot of objective function with variations in diameter and angle of attack was obtained and is displayed below-

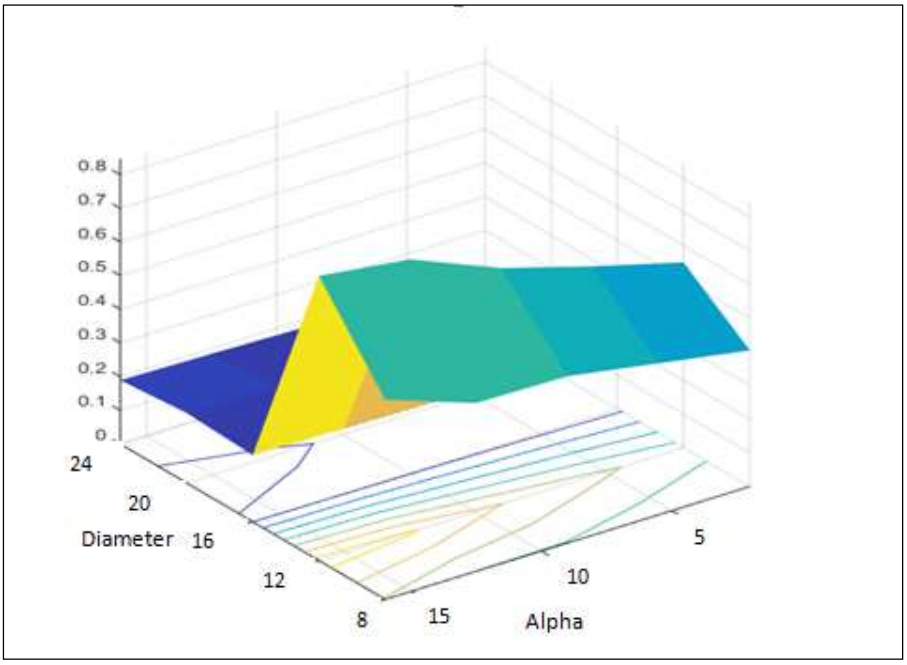


Figure 15 Plot of objective function (efficiency) wrt D and alpha

It is seen that efficiency is maximum in the region near diameter ~12" and alpha ~13-14°. Therefore, the results obtained from fmincon are valid. Validation can be also done by manually varying diameter, angle of attack in fixed steps (creating a matrix of different combinations) and finding efficiency matrix to define the region of interest where solution can be expected to lie.

From the plot the solution appears to be the global solution; at least in the defined boundaries of D and alpha, which are defined by spatial constraints. From experimental data and observations, the efficiency increases as the diameter is increased as more area is swept in every revolution. On the other hand, the efficiency increases with increase in alpha till a certain alpha (~14-16°) beyond which efficiency decreases again. The reason being that coefficient of drag depends exponentially on angle of attack. Therefore, at alpha beyond 16° the drag forces become very high as compared to the lift forces and hence, the overall efficiency of propeller reduces. Thus, the optimization solution appears to be in accordance to the observed results from experimental data.

Evaluating for KKT conditions for the trial, the values of Lagrangian multipliers from MATLAB are as follows:

gi	lambda multiplier
g1	0
g2	9.02E-10
g3	0
g4	5.55E-06
g5	0.00010053
g6	0.00E+00

These values of lambda (multipliers for inequality constraints) indicate that none of the inequality constraints are active. As a result, an interior solution can be expected for this case. Also, the point will be a KKT point as these multipliers are non-negative. However, this does not guarantee that solution so obtained is a global minima and this has to be verified using Hessian or analytic solution.

Parametric study was attempted by varying the thrust requirement to study if optimum (D, α) depends on the thrust constraint value which the user provides. In each case the diameter range is still 8-14" and angle of attack range is 2-18°. The results obtained were as given below-

Thrust (kg)	Final diameter of propeller (inch)	Final alpha of propeller (deg)
1	11.47	13.8
1.5	12.35	13.6
2	12.66	13.3
2.5	12.8	13.4
3	13.31	13.52

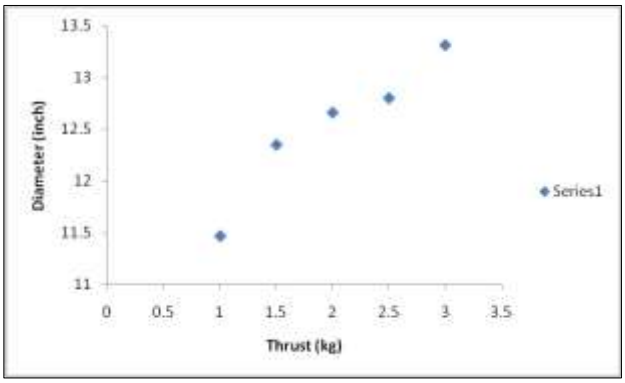


Figure 16 Plot of optimum D value at various thrusts

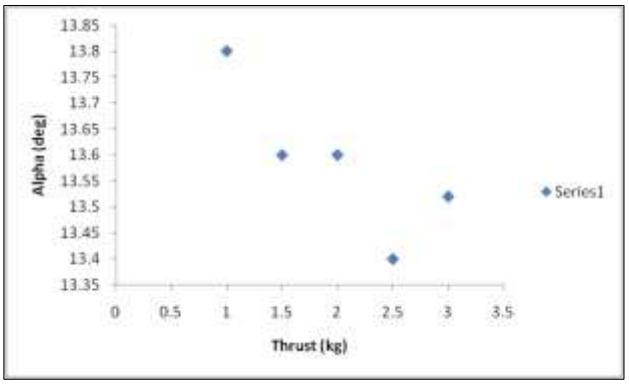


Figure 17 Plot of variation of optimum alpha wrt variation in thrust

From varying the thrust and checking for optimum (D, α) combination we do not clearly get any explicit design rule which explains how the optimal solution varies as thrust requirement varies. However, we can generally state that as the thrust needed increases the diameter of the propeller needed also increases and it decreases as thrust needed decreases. It is also seen that the alpha value always comes around 13-14° at any diameter and thrust. This is also as expected and the reason is as stated earlier (drag increases exponentially).

An optimized propeller for system requirement of thrust 1.5 kg is as given below:

Diameter D = 12.35”

Angle of attack α = 13.6°

Pitch = 5” (assumed to be constant)

The thrust, torque obtained for this solution is-

Thrust = 2.4921 kg

Torque = 0.5475 Nm

This case is considered here on for further calculations as it is the main objective of the overall system. Based on the above optimized results if a commercially available propeller has to be chosen, it can be-

1255 carbon fiber propeller

12” x 5”

Static thrust capacity: 3.52 kg

Perimeter speed: 175 m/sec

Required engine power: 1.09 Hp

The only limitation in this optimization solution is that the obtained efficiency values from the trial are comparatively smaller than those obtained for similar propellers experimentally. The possible reason for this shortcoming is the design theory which is considered here. Instead of Blade element theory if Vortex theory or combined blade element theory is used for modeling the design of propeller then a better estimate of efficiency may be achieved. These theories also account for wake field formation and other aerodynamic losses which are neglected in Blade element theory. Use of these will make the optimization problem more interesting.

Motor and Battery selection

In quadcopters usually Brushless DC motors (BLDC) are used due to good efficiency, quiet operation, reliability and repeatability. Motor selection depends on the torque, mechanical power demand by the propeller, the rpm per KV and maximum current it can deliver. The optimum motor selected (by observation and comparison) is such that it provides the required energy and has minimum weight out of all such possible motors. [8]

Torque required= 0.5475 Nm at 11V

Power = T x ω = 0.84 hp

Motor rpm/volt range: 11000 rpm at 11V. So, minimum 1000 KV motor needed.

Motor selected is: Outrunner 550 Plus 1470 KV BLDC motor

Specifications:

RPM/volt (KV): 1470

Maximum voltage: 14.8

Maximum Amps (A): 65

Motor Io@10V (A): 3.2

Max efficiency: 90%

Poles: 8

Weight: 205 gm

L, D dimensions: 50mm, 43 mm

Battery:

Out of available options like NiCad or NiMH or Li-Po batteries, Li-Po batteries are selected as they high energy storage to weight ratio and come in many different sizes and shapes. Another advantage offered by these batteries is that they have high discharge rates which are needed to power demanding BLDC motors.

Typical 1 Li-Po battery cell can give power @3.7 V. Therefore, considering motor voltage at 11V 3 battery cells will be needed in series. Considering capacity for providing enough flight time, 4 such battery packs are attached in parallel. Hence, final battery selection is 3S4P Li-Po battery.

Subsystem 3: Optimization of Placement of Components (Nikhil Sonawane)

Mathematical model

Various components placed on the frame contribute to the balancing of the quadcopter which is the most important factor to be worked upon. The components mounted will not only add to the weight of the system but also affect the center of gravity which has to coincide with frame center of gravity for maximum stability. Thus figuring out the distance at which the components are placed and their exact position would play a vital role in optimizing the design of quadcopter.

Components should be placed in such a way that the CG of the quadcopter remains at the frame origin and the weight is symmetrically placed about the origin and the arms. This will ensure that the quadcopter will not tilt on account of unbalanced weight. In order to achieve the given conditions we balance the moment due to weight of the components about CG and try to minimize it with respect to x y directions.

The various components which are to be placed on the quadcopter frame are as follows-

1. Camera
2. Battery
3. Antenna
4. Receiver
5. 4 × ESC’s (Electronic Speed Controller)
6. Flight control circuit

Assumptions-

1. The components are isotropic
2. The components are treated as cuboids
3. Aerodynamic drag is neglected.

Weights of each components and Dimensions of each components are as follows -

Component	Weight (gm)	Length (mm)	Breadth (mm)	Height (mm)
camera	180	41	59	30
battery	170	104	36	25
receiver	20	39	26	14
antenna	15	100	1	1
ESC	25	50	45	7
Flight Control Circuit	40	55	55	19

Objective function -

Mx =$\sum weight * Xcg$ My =$\sum weight * Ycg$

Minimize F=$Mx^2 + My^2$

Constraints -

All components are placed in x-z plane for stacking.

The components must not overlap with each other in x or z direction.

Position of components must be closer to CG of quadcopter.

The camera is placed at the bottom so that it does not have interference in visibility due to propeller blades.

The upper and lower bounds on the values are 58 mm and -58 mm, it signifies the radius of the circular disk to be mounted on the frame.

Inequality Constraints –

The distance between the CG of the components in the x or z direction must be greater than or equal to sum of half of the length and height of the component. The number of constraints are 36, which are the combinations of all components being compared for overlapping.

Equality Constraints –

The x coordinate of CG must be set to zero, as the closer the components will be placed to the CG, the lower will be the moment. Placing them on CG will give rise to zero moment for maximum stability. The number of constraints are 9, which correspond to placing the 9 components at Xcg =0

Variables -

X, Y coordinates of all components.

Degree of freedom –

The number of degrees of freedom are 18.

The x and y coordinates of CG of the 9 components.

Model analysis

The components are mounted on a circular disk, which in turn is fixed on the center of the frame of the quadcopter. The diameter of the disk limits the space over which the components can be placed. Thus defining the upper and lower bounds defines the disk diameter. We begin by setting a higher values for upper and lower bounds and check if we obtain a feasible solution or minimized solution.

The moment balance equation is in terms of x and y coordinates of the CG since the z location of the components does not affect the moment, whereas the constraints are with respect to x and z coordinates of the CG. Therefore when applying monotonicity principle the bounding parameter is controlled by x coordinates only, since the derivative of the constraints with respect to any y coordinate of the CG with be zero, or it won't affect the monotonicity. [5]

The monotonicity analysis for the above function gives the following results -

As can be seen from the monotonicity table the active and inactive constraints can be figured out. As can be seen from the table all the constraints with respect to x (1, 1) are unbounded. Whereas with respect to x (2, 1) g1 is active. As the x coordinates of the components are compared with each other, the number of constraints which are active keeps increasing. All are active at the same time, since this active status signifies the overlapping condition of the components with respect to the coordinates and their active constraints. The reoccurrence of the components while comparing leads to corresponding negative sign and thus is active.

There is no activity with respect to y coordinate since the constraints are a function of x and z coordinate of the cg. Therefore the derivative of the constraints with respect to y coordinate becomes zero.

	X(1,1)	X(2,1)	X(3,1)	X(4,1)	X(5,1)	X(6,1)	X(7,1)	X(8,1)	X(9,1)	X(1,2)	X(2,2)	X(3,2)	X(4,2)	X(5,2)	X(6,2)	X(7,2)	X(8,2)	X(9,2)
function	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
g1	+	-																
g2	+		-															
g3	+			-														
g4	+				-													
g5	+					-												
g6	+						-											
g7	+							-										
g8	+								-									
g9		+	-															
g10		+		-														
g11		+			-													
g12		+				-												
g13		+					-											
g14		+						-										
g15		+							-									
g16			+	-														
g17			+		-													
g18			+			-												
g19			+				-											
g20			+					-										
g21			+						-									
g22				+	-													
g23				+		-												
g24				+			-											
g25				+				-										
g26				+					-									
g27					+	-												
g28					+		-											
g29					+			-										
g30					+				-									
g31						+	-											
g32						+		-										
g33						+			-									
g34							+	-										
g35							+		-									
g36								+	-									
h1	+																	
h2		+																
h3			+															
h4				+														
h5					+													
h6						+												
h7							+											
h8								+										
h9									+									

From the nature of the function, global optimal solution for the placement of the components can be observed which is at 0, 0. The observed value for the x coordinate of the CG justifies the equality constraints being zero.

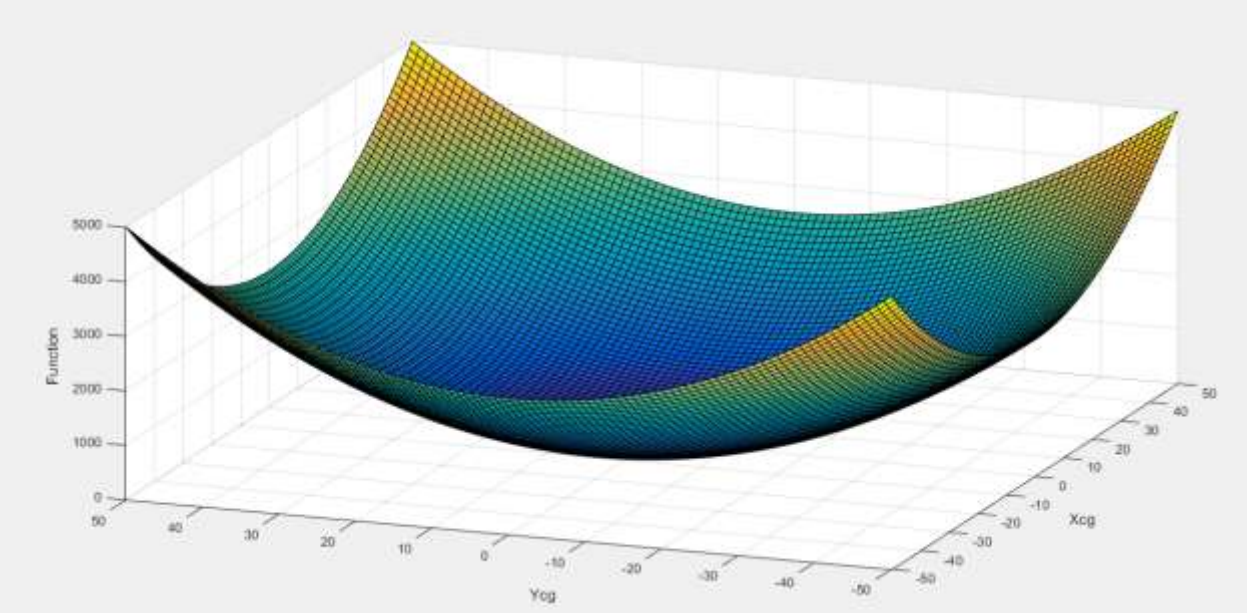


Figure 18 Objective function vs X,Y

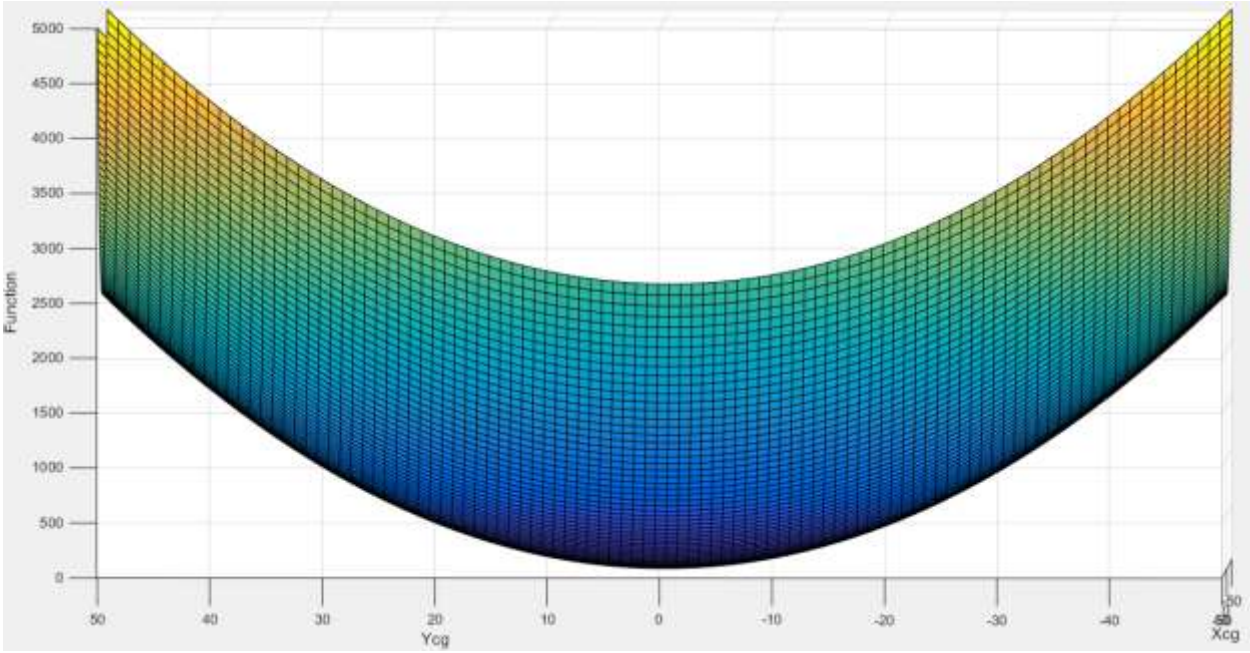


Figure 19 Objective fucntion vs X,Y

Optimization study

After the Mathematical model was formulated. The optimization problem was run successfully in Fmincon, MATLAB implementation of sequential quadratic programming. In our optimization study we had used three different start points and found that each time the optimization found a local minimum satisfying all the constraints.

The attempt aimed at stacking the components about the CG so as to obtain Zero moment. The X coordinate of the variables was fixed to origin while the Z coordinates were kept as variables so as to satisfy the non-overlapping conditions along the Z axis.

The optimized results with an initial point x0 are as follows –

x0 =

0	70.71	70.71
0	-74.9736	-74.9482
0	-74.9604	29.36963
0	-74.9703	38.44953
0	43.23979	-6.40E-07
0	-1.14E-07	5.238782
0	-44.7502	-6.40E-07
0	-1.14E-07	-60.7624
0	-7.09001	-74.9438

The results before and after optimization –

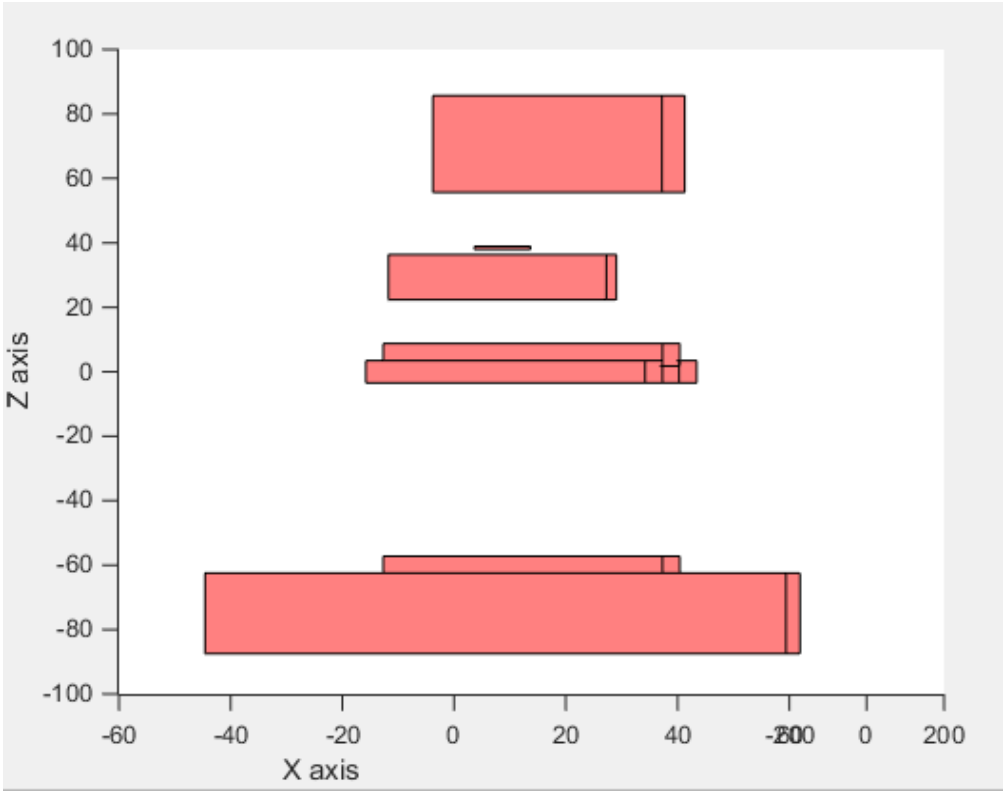


Figure 20 Initial placement of components

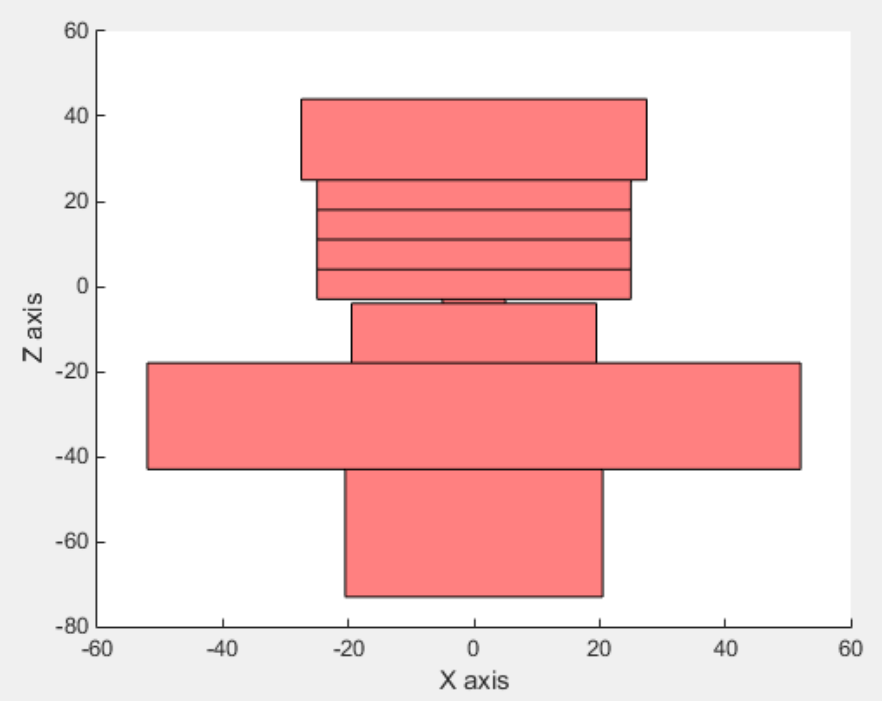


Figure 21 Optimised placment

Optimized Solution with Xopt =

0	-2.22451	-58
0	-28.8617	-30.5
0	16.30281	-11
0	17.80919	-3.5
0	44.42011	5.00E-01
0	3.11E+01	7.500494
0	17.36398	1.45E+01
0	3.11E+01	21.51737
0	25.93054	34.52021

The above obtained solution is an ideal solution, since it does not consider external forces like wind into consideration. If we consider the wind force, it will certainly create a pressure upon the stacked components. The more the surface area exposed to the wind, higher will be the instability caused due to increased pressure and will cause the battery to expend more power to the motors for the same performance. [9]

Thus placing the components on x-y plane with the same constraints with minor changes can give rise to more stable system. The updated constraints are as follows -

Constraints -

All components are placed in x-y plane.

The components must not overlap with each other in x or y direction.

Position of components must be closer to CG of quadcopter.

The camera is placed in the first quadrant so that it does not have interference in visibility due to propeller blades.

The upper and lower bounds on the values are 140 mm and --140 mm, it signifies the radius of the circular disk to be mounted on the frame.

Inequality Constraints –

The distance between the CG of the components in the x or y direction must be greater than or equal to sum of half of the length and breadth of the component. The number of constraints is 36, which are the combinations of all components being compared for overlapping.

The monotonicity analysis for the above function gives the following results -

As can be seen from the monotonicity table the active and inactive constraints can be figured out. As can be seen from the table all the constraints with respect to x (1, 1) are unbounded. Whereas with respect to x (2, 1) g1 is active. As the x coordinates of the components are compared with each other, the number of constraints which are active keeps increasing. All are active at the same time, since this active status signifies the overlapping condition of the components with respect to the coordinates and their active constraints. The reoccurrence of the components while comparing leads to corresponding negative sign and thus is active. The same applies for y coordinates too.

The equality constraints are with respect to z coordinate since the components are placed on the surface of x y plane, thus the derivative of the z coordinates with respect to x and y becomes zero and does not have any significant effect of the monotonicity analysis.

	X(1,1)	X(2,1)	X(3,1)	X(4,1)	X(5,1)	X(6,1)	X(7,1)	X(8,1)	X(9,1)	X(1,2)	X(2,2)	X(3,2)	X(4,2)	X(5,2)	X(6,2)	X(7,2)	X(8,2)	X(9,2)
function	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
g1	+	-								+	-							
g2	+		-							+		-						
g3	+			-						+			-					
g4	+				-					+				-				
g5	+					-				+					-			
g6	+						-			+						-		
g7	+							-		+							-	
g8	+								-	+								-
g9		+	-								+	-						
g10		+		-							+		-					
g11		+			-						+			-				
g12		+				-					+				-			
g13		+					-				+					-		
g14		+						-			+						-	
g15		+							-		+							-
g16			+	-								+	-					
g17			+		-							+		-				
g18			+			-						+			-			
g19			+				-					+				-		
g20			+					-				+					-	
g21			+						-			+						-
g22				+	-								+	-				
g23				+		-							+		-			
g24				+			-						+			-		
g25				+				-					+				-	
g26				+					-				+					-
g27					+	-								+				
g28					+		-							+				
g29					+			-						+				
g30					+				-					+				
g31						+	-								+			
g32						+		-							+			
g33						+			-						+			
g34							+	-										
g35							+		-									
g36								+	-									
h1																		
h2																		
h3																		
h4																		
h5																		
h6																		
h7																		
h8																		
h9																		

Trying to optimize the problem by placing the components as close as possible to the center.

Objective was to minimize the area within which the components can be placed.

Variables and constraints being the same. For different initial values each time the optimization found a local minimum satisfying all the constraints. The result for the given point gives the following result

For a sample point x0 =

70	-4.9896	-27.5
-61.409	-31.79	3
-3.3426	4.55661	-5.5
18.3662	-2.9099	1.5
-55.818	8.71029	5
0	-32.364	19
-12.366	12.6362	12
0	12.6362	19
-46.636	8.77773	32

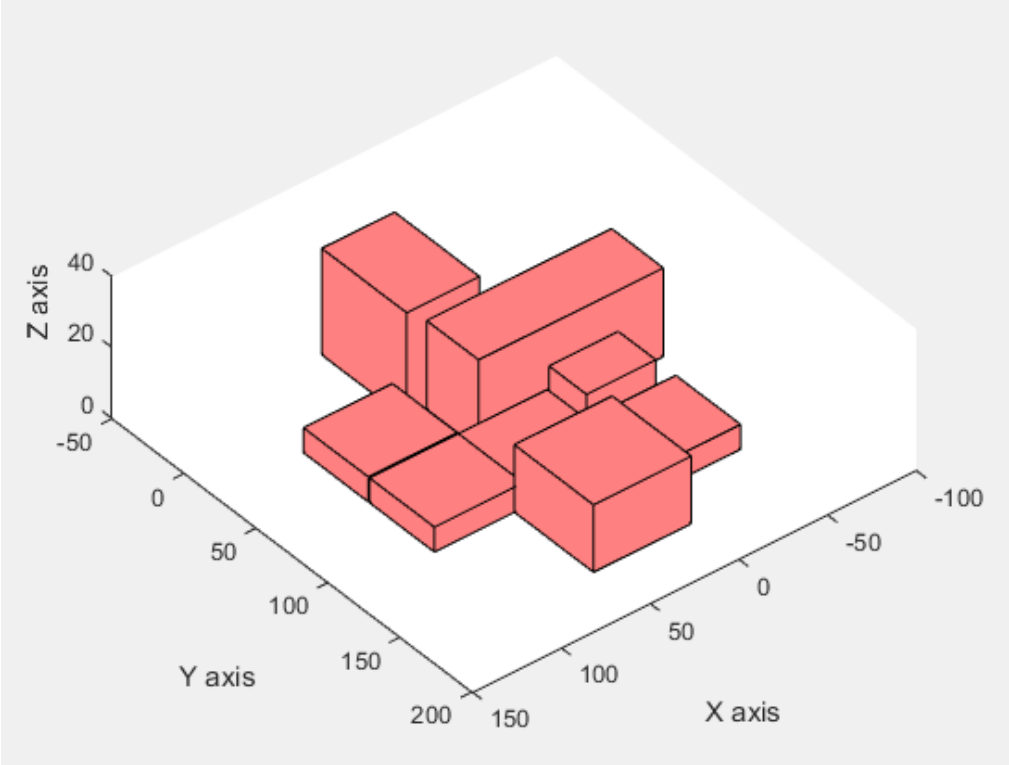


Figure 22 Closely packed components

This gives the feasible region for placing the components but gives highly unbalanced moment and thus such closed packing of components although being closest cannot satisfy moment minimization. The optimal solution of the above problem along with obtained upper and lower bounds is considered as an initial point to the moment balance problem as all the constraints are already satisfied and iterations were performed accordingly.

The optimal solution obtained is as follows –

Xopt = fval = 4.8979e-05

1.39E-17	0.01472	15
72.5	-94.387	12.5
-154.83	102.902	7
-129.62	90.3791	0.5
-155	147.134	3.5
59.3897	-46.014	3.5
-101.68	130.861	3.5
-49.505	76.782	3.5
2.99514	84.1548	9.5

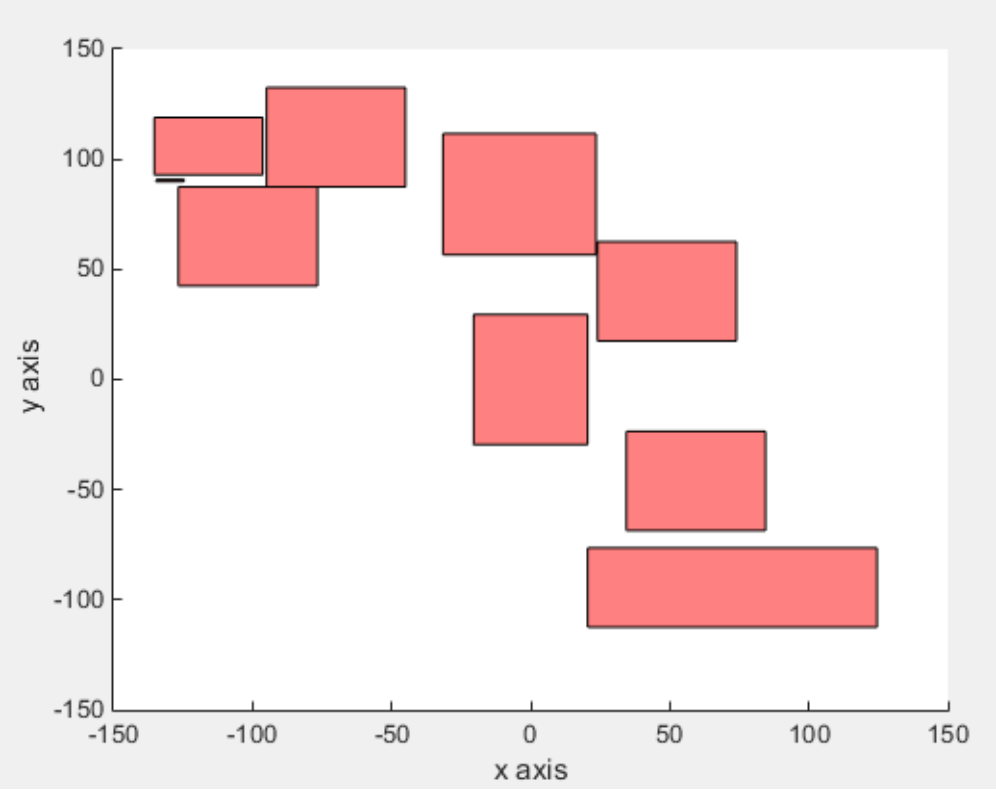


Figure 23 Optimal solution

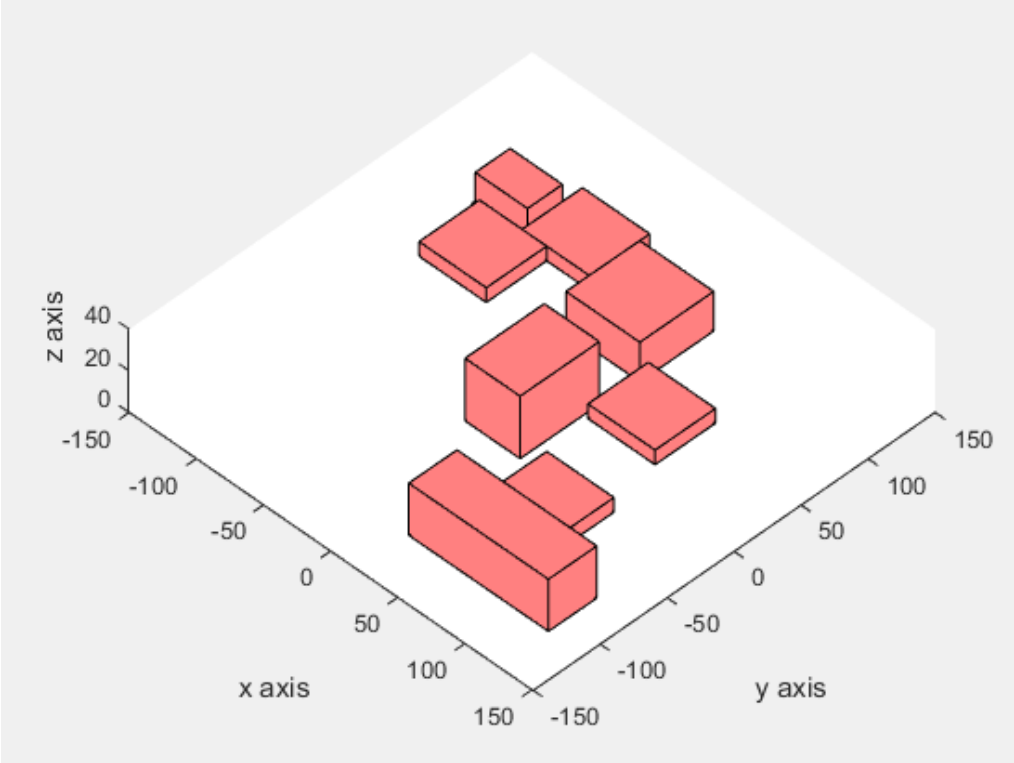


Figure 24 Optimal Solution

Considering the analysis of the Xopt obtained, the Lagrange's multiplier are as follows-

lambda.eqnonlin	lambda.ineqnonlin
-0.0013	0.000949
-0.0003	0
-0.0004	0
0.0036	0
0.00406	0
0.00047	0
0.00026	0
0.00047	0
0.00069	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0

	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0
	0.013804
	0
	0.017738

For the obtained solution to be a KKT point the values of lambda, i.e. multiplier for equality constraint must not be equal to zero. Also the value of mu must be greater than or equal to zero. As we can see from the obtained results since the values satisfy KKT conditions, the obtained solution is a KKT point.

Parametric Study

The solution to the function for different initial point gives the same optimal solution. The sample output is shown is the appendix. The function has 18 variables and the variation of the those variable in x and y direction can be shown as follows-

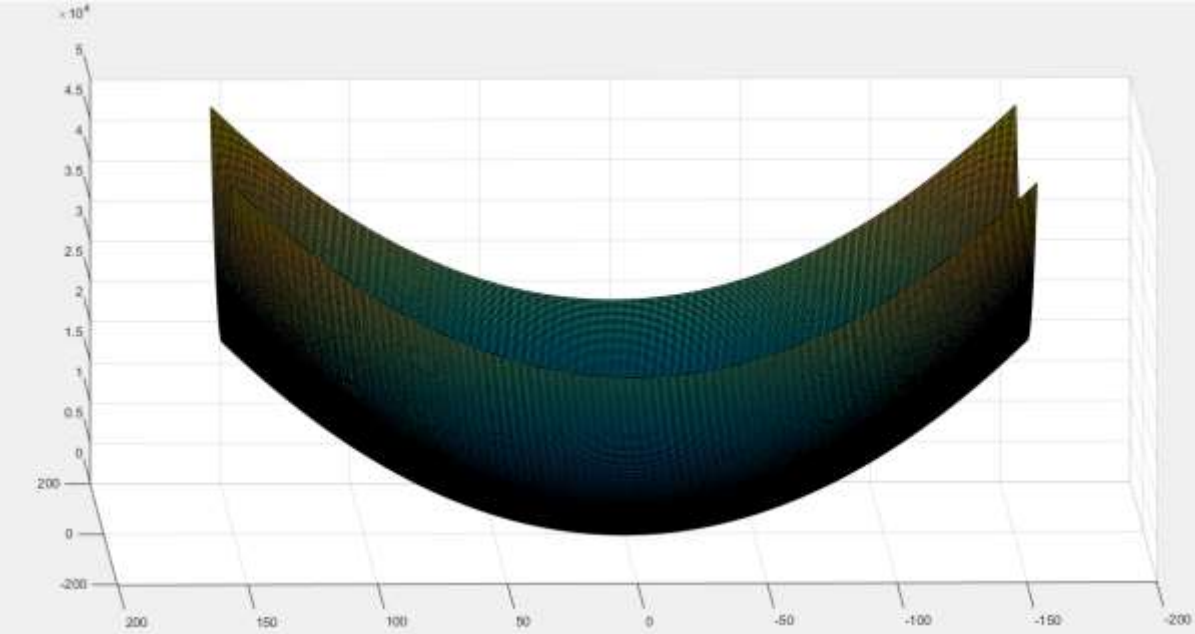


Figure 25 Parametric Variation of function

With a given set of X cg and Y cg coordinates ranging from -155 mm to 155 mm, since those are the minimum bounds obtained by iteration process for the optimal solution, we can observed that if the x and y coordinates of the components are at zero the function will be minimum, but since it does not give a feasible solution in terms of practicality, we place them close to center such that the moment obtained considering clockwise ad anti-clockwise moment can fetch us proper results.

Keeping one of the parameter constant will result in overlapping constraints being calculated just with respect to the other parameter. For example if x coordinates for the components are kept constant, then the components will adjust themselves with respect to y axis and vice versa which gives the following result –

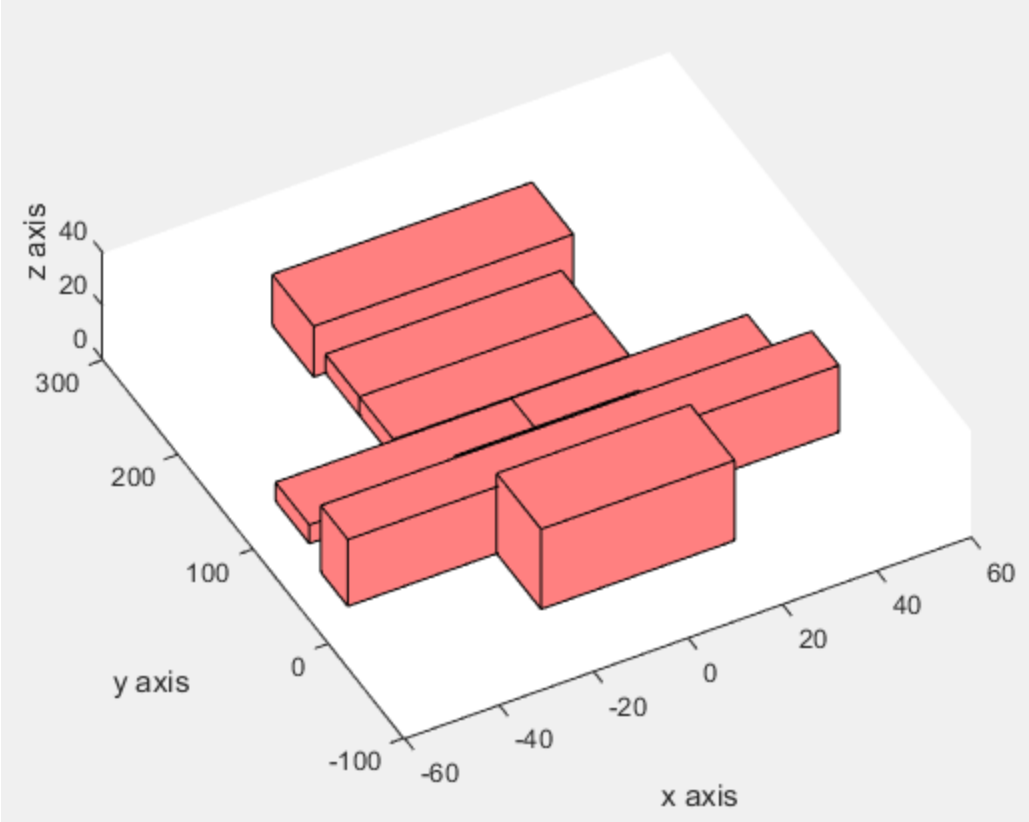


Figure 26 Components adjusted in Y direction

The components got rearranged with respect to same x but the solution cannot be obtained even for upper and lower bound as large as 1000 mm. same is the case with x coordinates. Thus the variation of both the coordinates is necessary to balance the moments and obtain an optimal solution.

Discussion of Results

The results obtained for the optimal solution give rise to minimum function value, or moment of the order 1e-5, which is almost equal to zero. The aim was to place the components as close to center for stability purposes and also the components must not overlap. All the conditions are satisfied from the results obtained. The positions as obtained by the solver are feasible and possible when compared to any model we get in the actual world.

Optimization of component placement is a bit neglected area of research but as observed from the solver solutions and various iterations it plays a major part in keeping the system stable during take-off and the flight. The design constraints can be changed depending on the area of application and this solution can be obtained by applying the same codes with a slight difference in the constraints and varying the upper and lower bound.

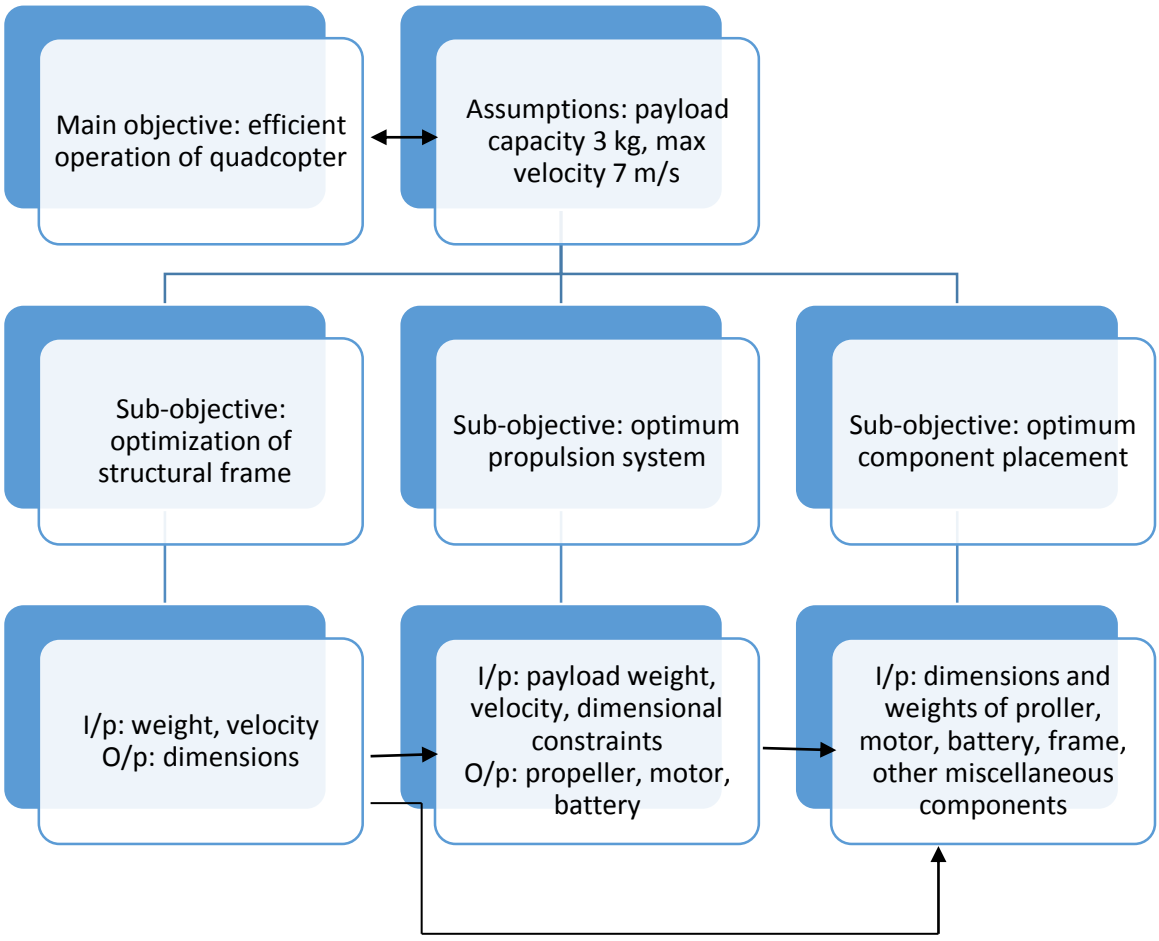
There is no general rule which can be set using this optimization as the constraints can be varied as per the manufacturers demand but ultimate balance of components can be achieved using the

model example developed here. The model was designed for 3 kg payload, having thrust capacity of 1.5 kgs and length of the arm being 300 mm. All these parameters indirectly affect the components placement. The limitations of the considered model lies in the payload and thrust capacity of the model. Also the external factors like rain, wind and other climatic conditions affect the performance of the quadcopter significantly. Thus, considering all these additional factors may make the system problem to be more realistic though complicated. Designing the quadcopters for specific applications like surveillance, or delivery of goods etc. can add to challenges and further work on the quadcopter design can be done keeping in mind one of the applications.

The results obtained as the optimal solution for this subsystem is dependent on the diameter of the arm which comes from the first subsystem to set an initial guess for upper and lower bound of the plate. Also the initial assumptions made for the battery, esc and other components are affected by the results obtained from the second subsystem.

System Integration

While looking at combined optimization of all three systems of quadcopters the main objective needs to be satisfied along with system level function optimization. A flowchart depicting the overall problem, requirements and flow of data between various systems is shown below-



The overall targets need to be satisfied when considering combined optimization. Based on these targets, the inputs and output variables to each system and inter-system dependency, flow of information to various sub-systems is influenced. The structural optimization gives the optimum dimensions which are taken as inputs by sub-systems 2 and 3. Optimization of propulsion system provides input to sub-system 3 for optimum placement of components and balancing the moment. Sub-system 3 also needs data on other miscellaneous components like ESCs, camera, sensors.

All the three sub-systems need to re-iterate again as per the overall system parameters. Calculations for this are shown in brief below- (detailed calculations for sub-system show this case itself, where in the payload capacity is 3 kg overall)

	Arm Design	**Propulsion System**	**Component Placement**
First Iteration	L ~ 200 mm	Diameter 8 to 14 inches	Battery weight assumed
Assumptions	Payload capacity assumed 3 kg	First design model iteration Thrust 2.5 kg, v=7 m/s	Battery Dimensions
		alpha 4 to 20 deg	Max limits -96 to 96
First Iteration	OD- 33	Dopt = 12.8"	Obj = 2.47*10^8
Results	ID- 30	Pitch = 5"	Constraints satisfied
	Weight ~ 550 gm	Alphaopt= 13.4 deg	Max limits -120 to 120
	Constraints satisfied		Weight ~ 600 gm
	Obj = 1.08898		
Second Iteration	New constraint L >= 250		Max limits -100 to 100
:	:	:	:
:	:	:	:
:	:	:	:
:	:	:	:
After n iterations	L = 300	Dopt =12.3" for thrust 1.5 kg, v=7m/s	Max limits -155to 155
Results	OD- 32.23	Pitch = 5 inch	No overlapping
	ID- 29.23	Alphaopt = 13.6 deg	
	Obj = 2.03	Obj= 58.2%	Obj = 0

System trade-offs and discussion on results:

As there is no direct explicit relation between the subsystems it was decided to solve each subsystem separately and simultaneously. The results of one subsystem are linked to each other as per the flow chart given above.

The table above shows assumptions with which the subsystems design was started and how the output of one subsystem affected the output of the other subsystem and which resulted into overall optimization of the problem.

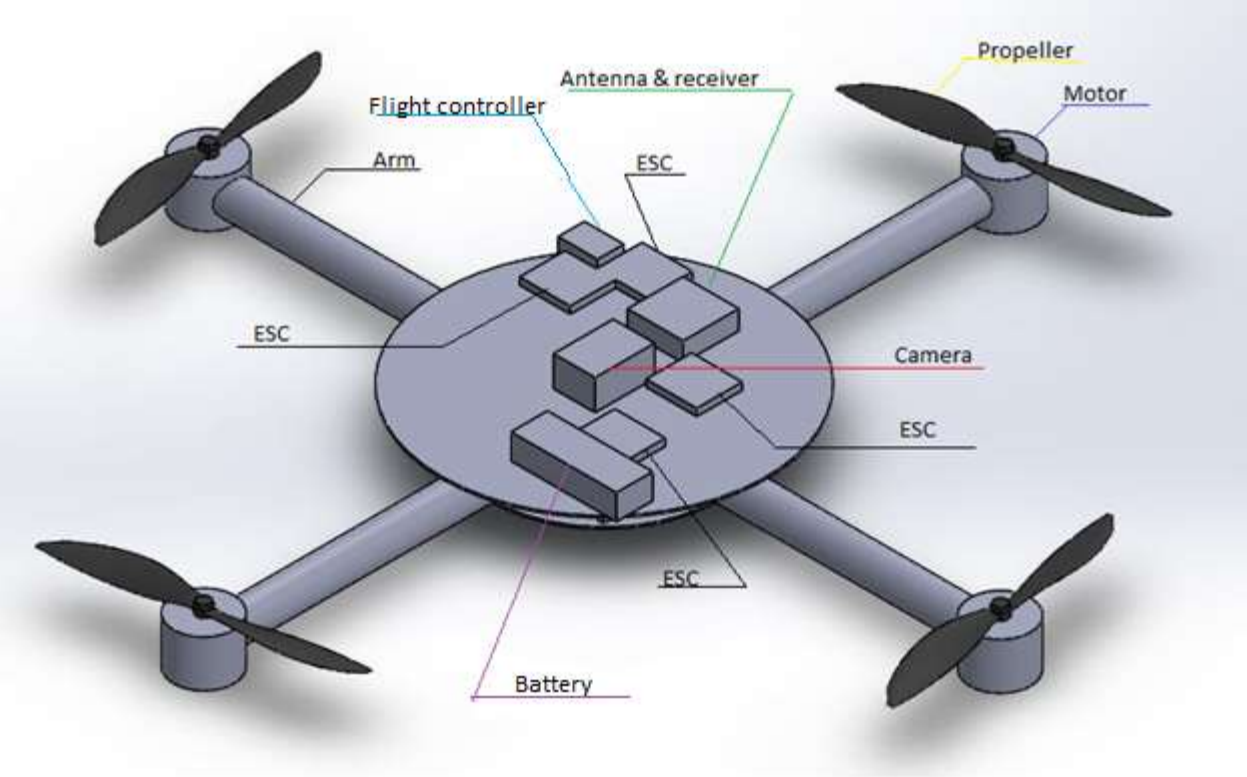


Figure 27 CAD model of optimised quadcopter

Acknowledgment

We would like to thank Professor Max Yi Ren. Without his help and guidance the project would not have been successfully completed.

References

[1] "Beam Formulas with Shear and Moment Diagrams," American Wood Council, 2007.

[2] IIT GUWAHATI Virtual Lab, "Sakshat Virtual Labs : Free Vibration of a Cantilever Beam (Continuous System)," [Online]. Available: http://iitg.vlab.co.in/?sub=62&brch=175&sim=1080&cnt=1.

[3] D. S. Talukdar, "Vibration of Continuous Systems," Guwahati.

[4] "ASM Aerospace Specification Metals Inc.," ASM Aerospace Specification Metals Inc., [Online]. Available: http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA6061t6.

[5] P. Y. PAPALAMBROS and D. J. WILDE, "Principles of Optimal Design: Modeling and Computation".

[6] "Aerodynamics for students, Aerospace, Mechanical & Mechatronics engineering," [Online]. Available: mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/contents.html.

[7] "Aerodynamics of model aircrafts website," [Online]. Available: http://www.mh-aerotools.de/airfoils/javafoil.html.

[8] "DCRC club online library resources," [Online]. Available: http://www.dc-rc.org/index.php/information/library-resources.

[9] "National Certified Testing Laboratories," [Online]. Available: http://www.nctlinc.com/velocity-chart/.

[10] P. E. I. Pounds, "Design, Construction and Control of large quadrotor micro vehicle," Australian National University, 2007.

[11] R. M. Moses Bangura, "Nonlinear dynamic modeling for high performance control of quadrotor," in *Proceedings of Australasian Conference on Robotics and Automation,*, Victoria University of Wellington, New Zealand, 3-5 Dec 2012,.

[12] S. Boubdallah, "Design and Control of Quadcopters with application to autonomous flying," Ecole Polytechnique Federale Lausanne, 2007.

[13] "Principles of Optimal Design," [Online]. Available:

http://www.optimaldesign.org/archive.html.

[14] "Office Excel functions," Microsoft, [Online]. Available: http://office.microsoft.com/en-us/excel-help/excel-functions-by-category-HP010342656.aspx.

[15] "MathWorks Documentation," MathWorks, [Online]. Available: http://www.mathworks.com/help/index.html.

[16] S. K. Phang, K. Li, K. H. Yu, B. M. Chen and T. H. Lee, "Systematic Design and Implementation of a Micro Unmanned Quadrotor System".

[17] B. A. Zai, "Structural optimization of cantilever beam in conjunction," Academia.edu.

[18] "Matlab," [Online]. Available: http://www.mathworks.com/help/optim/ug/fmincon.html.

[19] "Quadcopter Dynamics, Simulation, and Control".

Appendix

Subsystem 1: Optimization of structural frame design (Gaurav Pokharkar)

Solver Engine

Engine: GRG Nonlinear

Solution Time: 0.172 Seconds.

Iterations: 7 Subproblems: 0

Solver Options

Max Time Unlimited, Iterations Unlimited, Precision 0.000001, Use Automatic Scaling

Convergence 0.0001, Population Size 100, Random Seed 0, Derivatives Forward, Require Bounds

Max Subproblems Unlimited, Max Integer Sols Unlimited, Integer Tolerance 1%, Assume NonNegative

Objective Cell (Min)

Cell	Name	Original Value	Final Value
$H$28	Objective Shear Strength	1.035462524	2.031841014

Variable Cells

Cell	Name	Original Value	Final Value	Integer
$C$10	OD L	12	32.23315096	Contin
$C$11	ID L	29.2331508	29.23315096	Contin

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$C$16	Sigma L	4.2313558	$C$16<=120	Not Binding	115.7686442
$C$8	L L	300	$C$8>=200	Not Binding	100
$E$11	mm od	3.000000002	$E$11>=3	Binding	0
$N$4	Frequency 1st Mode	341.9384265	$N$4>=$J$20	Not Binding	145.2684265
$O$4:$R$4 >= $J$20					
$O$4	Frequency 2nd Mode	2142.901666	$O$4>=$J$20	Not Binding	1946.231666
$P$4	Frequency 3rd Mode	6000.183017	$P$4>=$J$20	Not Binding	5803.513017
$Q$4	Frequency 4th Mode	11679.0256	$Q$4>=$J$20	Not Binding	11482.3556
$R$4	Frequency 5th Mode	19436.80714	$R$4>=$J$20	Not Binding	19240.13714
$C$10	OD L	32.23315096	$C$10>=4	Not Binding	28.23315096
$C$11	ID L	29.23315096	$C$11>=4	Not Binding	25.23315096

Variable Cells

Cell	Name	Final Value	Reduced Gradient
$C$10	OD L	32.23315096	0
$C$11	ID L	29.23315096	0

Constraints

Cell	Name	Final Value	Lagrange Multiplier
$C$16	Sigma L	4.2313558	0
$C$8	L L	300	0
$E$11	mm od	3.000000002	0.071770392
$N$4	Frequency 1st Mode	341.9384265	0
$O$4:$R$4 >= $J$20			
$O$4	Frequency 2nd Mode	2142.901666	0
$P$4	Frequency 3rd Mode	6000.183017	0
$Q$4	Frequency 4th Mode	11679.0256	0
$R$4	Frequency 5th Mode	19436.80714	0

Cell	Objective Name	Value
$H$28	Objective Shear Strength	2.031841014

Cell	Variable Name	Value
$C$10	OD L	32.23315096
$C$11	ID L	29.23315096

Lower Limit	Objective Result
32.23315096	2.031841014
4	10.52406179

Upper Limit	Objective Result
#N/A	#N/A
29.23315096	2.031841014

F	15	N
L	300	mm
OD	32.2332	mm
ID	29.2332	mm
A	144.827	mm^2
I	17139.8	
Sigma	4.23136	N/mm^2

Cm*Vr	0.00040271				
Cp*L/ID	5.13116086				
Cr*Wr	2.64387194		Weight	0.11731	gm
Manufacturing Cost	7.77543551	7.77544			
	Objective	2.03184			

	10000	rpm
Motor speed	166.67	Hz

Modes	1st Mode	2nd Mode	3rd Mode	4th Mode	5th Mode
Frequency	341.938	2142.902	6000.18	11679	19436.8

Subsystem 2: Optimization of propulsion system (Preeti Vaidya)

A) Detailed explanation of blade element theory

Analysis of Propellers

Glauert Blade Element Theory

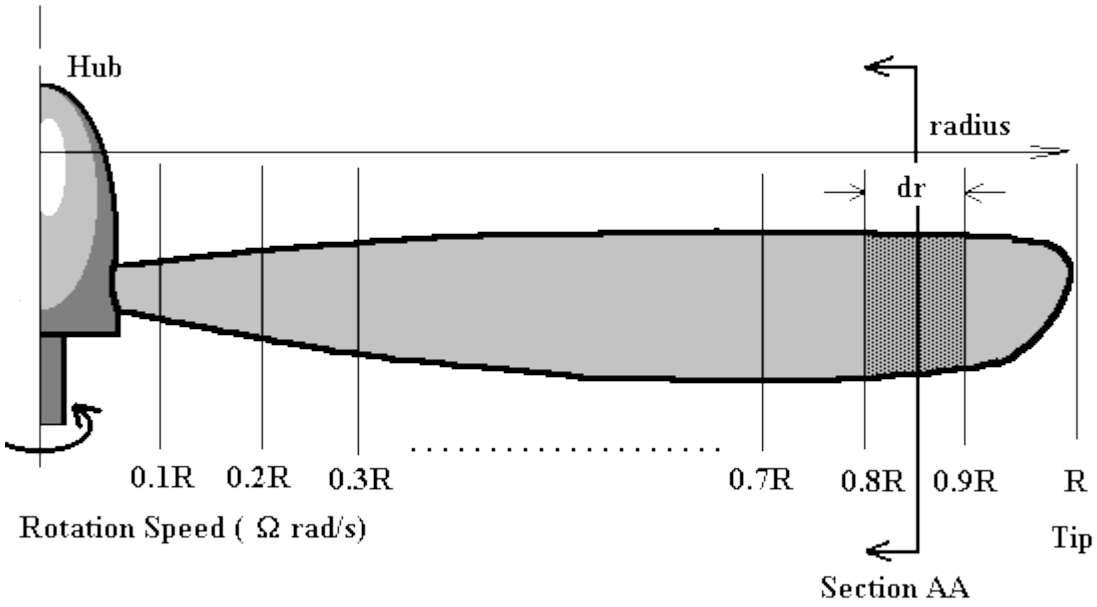
A relatively simple method of predicting the performance of a propeller (as well as fans or windmills) is the use of Blade Element Theory. In this method the propeller is divided into a number of independent sections along the length. At each section a force balance is applied involving 2D section lift and drag with the thrust and torque produced by the section. At the same time a balance of axial and angular momentum is applied. This produces a set of non-linear equations that can be solved by iteration for each blade section. The resulting values of section thrust and torque can be summed to predict the overall performance of the propeller.

The theory does not include secondary effects such as 3-D flow velocities induced on the propeller by the shed tip vortex or radial components of flow induced by angular acceleration due to the rotation of the propeller. In comparison with real propeller results this theory will over-predict thrust and under-predict torque with a resulting increase in theoretical efficiency of 5% to 10% over measured performance. Some of the flow assumptions made also breakdown for extreme conditions when the flow on the blade becomes stalled or there is a significant proportion of the propeller blade in wind milling configuration while other parts are still thrust producing.

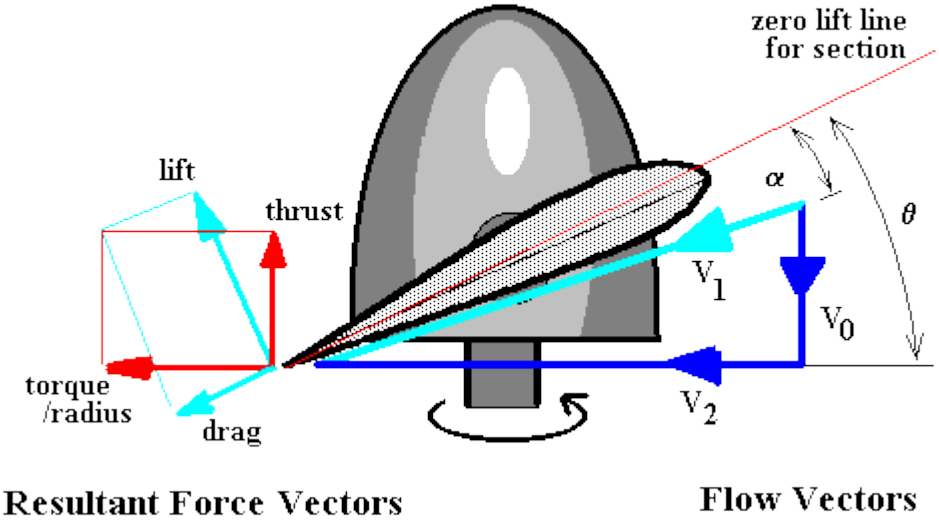
The theory has been found very useful for comparative studies such as optimizing blade pitch setting for a given cruise speed or in determining the optimum blade solidity for a propeller. Given the above limitations it is still the best tool available for getting good first order predictions of thrust, torque and efficiency for propellers under a large range of operating conditions.

Blade Element Subdivision

A propeller blade can be subdivided as shown into a discrete number of sections.



For each section the flow can be analyzed independently if the assumption is made that for each there are only axial and angular velocity components and that the induced flow input from other sections is negligible. Thus at section AA (radius = r) shown above, the flow on the blade would consist of the following components.



V_0 -- axial flow at propeller disk, V_2 -- Angular flow velocity vector

V_1 -- section local flow velocity vector, summation of vectors V_0 and V_2

Since the propeller blade will be set at a given geometric pitch angle (θ) the local velocity vector will create a flow angle of attack on the section. Lift and drag of the section can be calculated using standard 2-D aero foil properties. (Note: change of reference line from chord to zero lift line). The lift and drag

components normal to and parallel to the propeller disk can be calculated so that the contribution to thrust and torque of the complete propeller from this single element can be found.

The difference in angle between thrust and lift directions is defined as

$$\phi = \theta - \alpha$$

The elemental thrust and torque of this blade element can thus be written as

$$\Delta T = \Delta L\cos(\phi) - \Delta D\sin(\phi) \quad , \quad \frac{\Delta Q}{r} = \Delta D\cos(\phi) + \Delta L\sin(\phi)$$

Substituting section data (C_L and C_D for the given α) leads to the following equations.

$$\Delta L = C_L \tfrac{1}{2}\rho V_1^2 c.dr \quad , \quad \Delta D = C_D \tfrac{1}{2}\rho V_1^2 c.dr \text{ per blade}$$

where ρ is the air density, c is the blade chord so that the lift producing area of the blade element is c.dr.

If the number of propeller blades is (B) then,

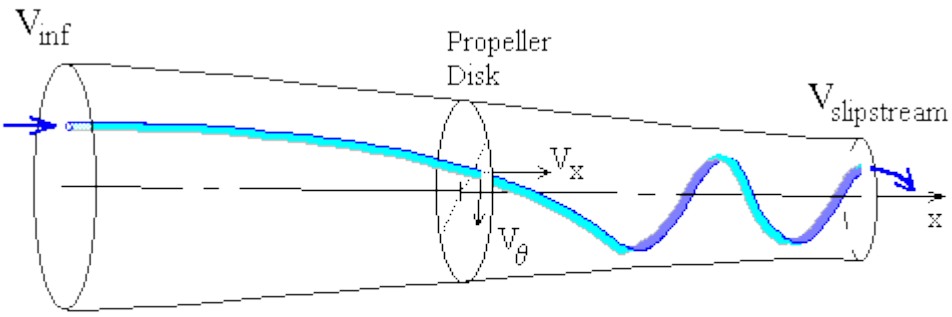
$$\Delta T = \tfrac{1}{2}\rho V_1^2 c(C_L\cos(\phi) - C_D\sin(\phi)).B.dr \quad \ldots\ldots\ldots (1)$$

$$\frac{\Delta Q}{r} = \tfrac{1}{2}\rho V_1^2 c(C_L\sin(\phi) + C_D\cos(\phi)).B.dr$$

$$\Delta Q = \tfrac{1}{2}\rho V_1^2 c(C_L\sin(\phi) + C_D\cos(\phi)).B.r.dr \quad \ldots\ldots\ldots (2)$$

2. Inflow Factors

A major complexity in applying this theory arises when trying to determine the magnitude of the two flow components V_0 and V_2. V_0 is roughly equal to the aircraft's forward velocity (V_{inf}) but is increased by the propeller's own induced axial flow into a slipstream. V_2 is roughly equal to the blade section's angular speed (Ωr) but is reduced slightly due to the swirling nature of the flow induced by the propeller. To calculate V_0 and V_2 accurately both axial and angular momentum balances must be applied to predict the induced flow effects on a given blade element. As shown in the following diagram the induced flow components can be defined as factors increasing or decreasing the major flow components.



Typical Streamtube of flow passing through Section AA

$V_\theta = b\Omega r \qquad V_x = V_{inf}(1+a)$

So for the velocities V0 and V2 as shown in the previous section flow diagram,

$V_o = V_{inf} + a.V_{inf}$ where a -- axial inflow factor

$V_2 = \Omega r - b.\Omega r$ where b -- angular inflow factor (swirl factor)

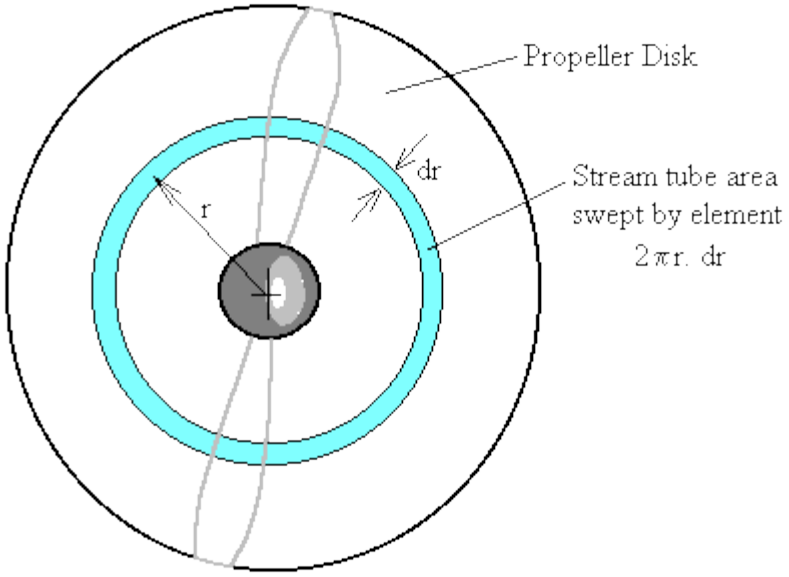
The local flow velocity and the angle of attack for the blade section is thus

$$V_1 = \sqrt{(V_o^2 + V_2^2)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$\alpha = \theta - \tan^{-1}(V_o/V_2) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

3. Axial and Angular Flow Conservation of Momentum

The governing principle of conservation of flow momentum can be applied for both axial and circumferential directions.



For the axial direction, the change in flow momentum along a stream-tube starting upstream, passing through the propeller at section AA and then moving off into the slipstream, must equal the thrust produced by this element of the blade.

To remove the unsteady effects due to the propeller's rotation, the stream-tube used is one covering the

complete area of the propeller disk swept out by the blade element and all variables are assumed to be time averaged values.

ΔT = change in momentum flow rate

= mass flow rate in tube x change in velocity

$$= \rho 2\,\pi r\,dr\,V_o(V_{slipstream} - V_{inf})$$

By applying Bernoulli's equation and conservation of momentum, for the three separate components of the tube, from freestream to face of disk, from rear of disk to slipstream far downstream and balancing pressure and area versus thrust, it can be shown that the axial velocity at the disk will be the average of the freestream and slipstream velocities.

$V_0 = (V_{inf} + V_{slipstream})/2$, that means $V_{slipstream} = V_{inf}(1 + 2a)$

Thus

$$\Delta T = \rho 2\,\pi r V_{inf}(1+a).\left(V_{inf}(1+2a) - V_{inf}\right).dr$$

$$\Delta T = \rho 2\,\pi r V_{inf}^{2.}(1+a).2\,a.dr$$

$$\Delta T = \rho 4\,\pi r V_{inf}^{2}(1+a).a.dr \quad \ldots\ldots\ldots (5)$$

For angular momentum

ΔQ = change in angular momentum rate for flow x radius

= mass flow rate in tube x change in circumferential velocity x radius

$$\Delta Q = \rho 2\,\pi r\,dr V_o.(V_\theta(slipstream) - 0(freestream)).r$$

By considering conservation of angular momentum in conjunction with the axial velocity change, it can be shown that the angular velocity in the slipstream will be twice the value at the propeller disk.

$$V_\theta(slipstream) = 2\,b\,\Omega\,r$$

Thus

$$\Delta Q = \rho 2\,\pi r V_{inf}(1+a).(2\,b\,\Omega\,r).r.dr$$

$$\Delta Q = \rho 4\,\pi r^3 V_{inf}^{2}(1+a).b\,\Omega.dr \quad \ldots\ldots\ldots\ldots (6)$$

Because these final forms of the momentum equation balance still contain the variables for element thrust and torque, they cannot be used directly to solve for inflow factors.

However there now exists a nonlinear system of equations (1),(2),(3),(4),(5) and (6) containing the four primary unknown variables ΔT, ΔQ, a, b. So an iterative solution to this system is possible.

4. Iterative Solution procedure for Blade Element Theory.

The method of solution for the blade element flow will be to start with some initial guess of inflow factors (a) and (b). Use these to find the flow angle on the blade (equations (3),(4)), then use blade section properties to estimate the element thrust and torque (equations (1),(2)). With these approximate values of thrust and torque equations (5) and (6) can be used to give improved estimates of the inflow factors (a) and (b). This process can be repeated until values for (a) and (b) have converged to within a specified tolerance.

It should be noted that convergence for this nonlinear system of equations is not guaranteed. It is usually a simple matter of applying some convergence enhancing techniques (ie Crank-Nicholson under-relaxation) to get a result when linear aerofoil section properties are used. When non-linear properties are used, ie including stall effects, then obtaining convergence will be significantly more difficult.

For the final values of inflow factor (a) and (b) an accurate prediction of element thrust and torque will be obtained from equations (1) and (2).

5. Propeller Thrust and Torque Coefficients and Efficiency.

The overall propeller thrust and torque will be obtained by summing the results of all the radial blade element values.

$$T = \Sigma \Delta T \text{ (for all elements), and } Q = \Sigma \Delta Q \text{ (for all elements)}$$

The non-dimensional thrust and torque coefficients can then be calculated along with the advance ratio at which they have been calculated.

$$C_T = T/(\rho n^2 D^4) \text{ and } C_Q = Q/(\rho n^2 D^5) \text{ for } J = V_{inf}/(nD)$$

Where n is the rotation speed of propeller in revs per second and D is the propeller diameter.

The efficiency of the propeller under these flight conditions will then be

$$\eta(\text{propeller}) = J/(2\pi).(C_T/C_Q).$$

B) MATLAB codes used in optimization of Propeller

1) Code written to execute the fmincon command for the given objective function and constraints.

```
%Code to use fmincon for objective function minimize (-Ct/Cp)
xo=[9.5;6];

A=[]; b=[]; Aeq=[]; beq=[];
lb=[8;2];
ub=[14;18];
[xopt,fval,exitflag, output, lambda]=fmincon(@func,xo,A,b,Aeq,beq,lb,ub,@nonlcon)
```

```
Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.



xopt =

   12.3717
   13.6372


fval =

   -0.5632


exitflag1 =

     2
```

Published with MATLAB® R2014b

2) Objective function code

```
function [f]= func(x)
% Objective function to be minimized is (-Ct/Cp)*J/2*pi
  [Ct,Cq,J]= parameter1(x);
   f = -((Ct/Cq)*J/2/pi);
end
```

```
Error using func (line 3)
Not enough input arguments.
```

Published with MATLAB® R2014b

3) Code to specify nonlinear constraints

```
function [c,ceq]= nonlcon(x)

    n = 11000/60;
    rho = 1.225;
    [Ct,Cq] = parameter1(x);

    c =[-(Ct*rho*n^2*x(1,1)^4-1.5);-(Cq*rho*n^2*x(1,1)^5);-Ct];
    ceq=[];

end
```

```
Error using nonlcon (line 5)
Not enough input arguments.
```

Published with MATLAB® R2014b

4) Code to generate Ct, Cq and J (design model code)


```
function [Ct,Cq,J,thrust,torque] = parameter1(x)
dia= x(1,1);
alphastep= x(2,1);
```

Geometry inputs

```
pitch=5*0.0254;                    % pitch
dia=dia*0.0254;                    % diameter
R=dia/2.0;                         % maximum radius
% Equation of chord at different cross-section radius by
% metamodeling
chord = @(r)-0.165*r^2 + 0.897*r -0.078;
%Equation for theta at different cross-section radius by meta modeling
theta= @(r) -0.055*r^2 - 1.819*r +27.82;
V=7;                                    % Forward velocity of quadcopter
RPM=11000;                              % Rotational speed of propeller
n=RPM/60.0;                             % rev per sec
omega=n*2.0*pi;                         % angular velocity
```

```
xt=R;
xs=0.1*R;                               % radius at 0.1R
rho=1.225;                              % density of air kg/m^3
rstep=(xt-xs)/10;                       % dr
r1=[xs:rstep:xt];                       % r

k=0;
thrust=0.0;
thr=0;
torque=0.0;
tor=0;
```

Thrust torque calculations

run loop for number of sections of the propeller aerofoil

```
for j=1:size(r1,2)
    rad=r1(j);                          % radius at that section
    t2(j)=theta(rad);                   % theta at that radius value
    th=t2(j)/180*pi;                    % theta in radians
    a=0.1;                              % initial guess a ,b
    b=0.01;
    finished=0;
    sum=1;
while (finished==0)
        V0=V*(1+a);                                 % V0 from forward velo
        V2=omega*rad*(1-b);                         % V2 from angular velo
        phi=atan2(V0,V2);                           % inflow angle
        alpha=th-phi+(alphastep*pi/180);            % angle of attack
        cl=2*alpha*pi*pi/180;                       % lift co-efficient
        cd = 0.0254*(0.0081 -0.014*cl + 0.4*cl*cl); % drag co-efficient
        Vlocal=sqrt(V0.*V0+V2.*V2);                 % resultant of Vo and V2

        DtDr=0.5*rho*Vlocal*Vlocal*2.0*chord(rad)*(cl*cos(phi)-cd*sin(phi));
        DqDr=0.5*rho*Vlocal*Vlocal*2.0*chord(rad)*rad*(cd*cos(phi)+cl*sin(phi));
% ratio of delta T from both formulae
        tem1=DtDr/(4.0*pi*rad*rho*V*V*(1+a));
% ratio of delta Q from both formulae
        tem2=DqDr/(4.0*pi*rad*rad*rad*rho*V*(1+a)*omega);
        anew=0.5*(a+tem1);
        bnew=0.5*(b+tem2);
% tolerance on a and b values
if (abs(anew-a)<1.0e-5)
if (abs(bnew-b)<1.0e-5)
                finished=1;
end;
end;
        a=anew;
        b=bnew;
        sum=sum+1;
if (sum>500),
            finished=1;
```

```
end;
end;
    a2(j)=a;
    b2(j)=b;
    thr=thr+DtDr*rstep;                 % total thrust
    tor=tor+DqDr*rstep;                 % total torque
end;
torque=tor*10;
```

Calculate Ct and Cq and J

```
Ct=(thr)/(rho*n*n*dia*dia*dia*dia);
Cq=(torque)/(rho*n*n*dia*dia*dia*dia*dia);
J=V/(n*dia);
thrust= thr*0.5;
end
```

```
Ct =

    0.0125


Cq =

    0.0044


J =

    0.1217


thrust =

    2.4921


torque =

    0.5475
```

Published with MATLAB® R2014b

5) Code to plot the objective function

```
function createfigure

% Create figure
```

```
figure1 = figure('Position',[1 400 1200 600]);
%colormap('gray');
axis square;
R=8:2:20;
TH=2:3.5:16;
%\ X=R'*cos(TH);
% Y=R'*sin(TH);
[X,Y]= meshgrid(R,TH);
for i=1:length(X)
for j=1:length(X)
        x=[X(i,j);Y(i,j)];
        Z(i,j)=abs(func(x));
end
end


% Create subplot
subplot1 = subplot(1,2,1,'Parent',figure1);
view([124 34]);
grid('on');
hold('all');

% Create surface
surf(X,Y,Z,'Parent',subplot1,'LineStyle','none');

% Create contour
contour(X,Y,Z,'Parent',subplot1);

% Create subplot
subplot2 = subplot(1,2,2,'Parent',figure1);
view([234 34]);
grid('on');
hold('all');

% Create surface
surf(X,Y,Z,'Parent',subplot2,'LineStyle','none');

% Create contour
contour(X,Y,Z,'Parent',subplot2);
```

Published with MATLAB® R2014b

Subsystem 3: Optimization of Placement of Components (Nikhil Sonawane)

Overlap Check

```
  function[g,h]=overlap(x)


lbh=[41 59      30
    104 36      25
    39  26      14
    10  01      01
    50  45      7
    50  45      7
    50  45      7
    50  45      7
    55  55      19];



g=[(lbh(1,1)+lbh(2,1)-2*(x(2,1)-x(1,1)))*(lbh(1,2)+lbh(2,2)-2*(x(2,2)-x(1,2)));
(lbh(1,1)+lbh(3,1)-2*(x(3,1)-x(1,1)))*(lbh(1,2)+lbh(3,2)-2*(x(3,2)-x(1,2)));
(lbh(1,1)+lbh(4,1)-2*(x(4,1)-x(1,1)))*(lbh(1,2)+lbh(4,2)-2*(x(4,2)-x(1,2)));
(lbh(1,1)+lbh(5,1)-2*(x(5,1)-x(1,1)))*(lbh(1,2)+lbh(5,2)-2*(x(5,2)-x(1,2)));
(lbh(1,1)+lbh(6,1)-2*(x(6,1)-x(1,1)))*(lbh(1,2)+lbh(6,2)-2*(x(6,2)-x(1,2)));
(lbh(1,1)+lbh(7,1)-2*(x(7,1)-x(1,1)))*(lbh(1,2)+lbh(7,2)-2*(x(7,2)-x(1,2)));
(lbh(1,1)+lbh(8,1)-2*(x(8,1)-x(1,1)))*(lbh(1,2)+lbh(8,2)-2*(x(8,2)-x(1,2)));
(lbh(1,1)+lbh(9,1)-2*(x(9,1)-x(1,1)))*(lbh(1,2)+lbh(9,2)-2*(x(9,2)-x(1,2)));
(lbh(2,1)+lbh(3,1)-2*(x(3,1)-x(2,1)))*(lbh(2,2)+lbh(3,2)-2*(x(3,2)-x(2,2)));
(lbh(2,1)+lbh(4,1)-2*(x(4,1)-x(2,1)))*(lbh(2,2)+lbh(4,2)-2*(x(4,2)-x(2,2)));
(lbh(2,1)+lbh(5,1)-2*(x(5,1)-x(2,1)))*(lbh(2,2)+lbh(5,2)-2*(x(5,2)-x(2,2)));
(lbh(2,1)+lbh(6,1)-2*(x(6,1)-x(2,1)))*(lbh(2,2)+lbh(6,2)-2*(x(6,2)-x(2,2)));
(lbh(2,1)+lbh(7,1)-2*(x(7,1)-x(2,1)))*(lbh(2,2)+lbh(7,2)-2*(x(7,2)-x(2,2)));
(lbh(2,1)+lbh(8,1)-2*(x(8,1)-x(2,1)))*(lbh(2,2)+lbh(8,2)-2*(x(8,2)-x(2,2)));
(lbh(2,1)+lbh(9,1)-2*(x(9,1)-x(2,1)))*(lbh(2,2)+lbh(9,2)-2*(x(9,2)-x(2,2)));
(lbh(3,1)+lbh(4,1)-2*(x(4,1)-x(3,1)))*(lbh(3,2)+lbh(4,2)-2*(x(4,2)-x(3,2)));
(lbh(3,1)+lbh(5,1)-2*(x(5,1)-x(3,1)))*(lbh(3,2)+lbh(5,2)-2*(x(5,2)-x(3,2)));
(lbh(3,1)+lbh(6,1)-2*(x(6,1)-x(3,1)))*(lbh(3,2)+lbh(6,2)-2*(x(6,2)-x(3,2)));
(lbh(3,1)+lbh(7,1)-2*(x(7,1)-x(3,1)))*(lbh(3,2)+lbh(7,2)-2*(x(7,2)-x(3,2)));
(lbh(3,1)+lbh(8,1)-2*(x(8,1)-x(3,1)))*(lbh(3,2)+lbh(8,2)-2*(x(8,2)-x(3,2)));
(lbh(3,1)+lbh(9,1)-2*(x(9,1)-x(3,1)))*(lbh(3,2)+lbh(9,2)-2*(x(9,2)-x(3,2)));
(lbh(4,1)+lbh(5,1)-2*(x(5,1)-x(4,1)))*(lbh(4,2)+lbh(5,2)-2*(x(5,2)-x(4,2)));
(lbh(4,1)+lbh(6,1)-2*(x(6,1)-x(4,1)))*(lbh(4,2)+lbh(6,2)-2*(x(6,2)-x(4,2)));
(lbh(4,1)+lbh(7,1)-2*(x(7,1)-x(4,1)))*(lbh(4,2)+lbh(7,2)-2*(x(7,2)-x(4,2)));
(lbh(4,1)+lbh(8,1)-2*(x(8,1)-x(4,1)))*(lbh(4,2)+lbh(8,2)-2*(x(8,2)-x(4,2)));
(lbh(4,1)+lbh(9,1)-2*(x(9,1)-x(4,1)))*(lbh(4,2)+lbh(9,2)-2*(x(9,2)-x(4,2)));
(lbh(5,1)+lbh(6,1)-2*(x(6,1)-x(5,1)))*(lbh(5,2)+lbh(6,2)-2*(x(6,2)-x(5,2)));
(lbh(5,1)+lbh(7,1)-2*(x(7,1)-x(5,1)))*(lbh(5,2)+lbh(7,2)-2*(x(7,2)-x(5,2)));
(lbh(5,1)+lbh(8,1)-2*(x(8,1)-x(5,1)))*(lbh(5,2)+lbh(8,2)-2*(x(8,2)-x(5,2)));
(lbh(5,1)+lbh(9,1)-2*(x(9,1)-x(5,1)))*(lbh(5,2)+lbh(9,2)-2*(x(9,2)-x(5,2)));
(lbh(6,1)+lbh(7,1)-2*(x(7,1)-x(6,1)))*(lbh(6,2)+lbh(7,2)-2*(x(7,2)-x(6,2)));
(lbh(6,1)+lbh(8,1)-2*(x(8,1)-x(6,1)))*(lbh(6,2)+lbh(8,2)-2*(x(8,2)-x(6,2)));
```

```
(lbh(6,1)+lbh(9,1)-2*(x(9,1)-x(6,1)))*(lbh(6,2)+lbh(9,2)-2*(x(9,2)-x(6,2)));
(lbh(7,1)+lbh(8,1)-2*(x(8,1)-x(7,1)))*(lbh(7,2)+lbh(8,2)-2*(x(8,2)-x(7,2)));
(lbh(7,1)+lbh(9,1)-2*(x(9,1)-x(7,1)))*(lbh(7,2)+lbh(9,2)-2*(x(9,2)-x(7,2)));
(lbh(8,1)+lbh(9,1)-2*(x(9,1)-x(8,1)))*(lbh(8,2)+lbh(9,2)-2*(x(9,2)-x(8,2)));
-x(1,1);
-x(1,2)];
h=[x(1,3)-lbh(1,3)/2;
x(2,3)-lbh(2,3)/2;
x(3,3)-lbh(3,3)/2;
x(4,3)-lbh(4,3)/2;
x(5,3)-lbh(5,3)/2;
x(6,3)-lbh(6,3)/2;
x(7,3)-lbh(7,3)/2;
x(8,3)-lbh(8,3)/2;
x(9,3)-lbh(9,3)/2];
% g=[(lbh(1,1)+lbh(2,1)-2*(x(2,1)-x(1,1)))*(lbh(1,3)+lbh(2,3)-2*(x(2,3)-x(1,3)));
% (lbh(1,1)+lbh(3,1)-2*(x(3,1)-x(1,1)))*(lbh(1,3)+lbh(3,3)-2*(x(3,3)-x(1,3)));
% (lbh(1,1)+lbh(4,1)-2*(x(4,1)-x(1,1)))*(lbh(1,3)+lbh(4,3)-2*(x(4,3)-x(1,3)));
% (lbh(1,1)+lbh(5,1)-2*(x(5,1)-x(1,1)))*(lbh(1,3)+lbh(5,3)-2*(x(5,3)-x(1,3)));
% (lbh(1,1)+lbh(6,1)-2*(x(6,1)-x(1,1)))*(lbh(1,3)+lbh(6,3)-2*(x(6,3)-x(1,3)));
% (lbh(1,1)+lbh(7,1)-2*(x(7,1)-x(1,1)))*(lbh(1,3)+lbh(7,3)-2*(x(7,3)-x(1,3)));
% (lbh(1,1)+lbh(8,1)-2*(x(8,1)-x(1,1)))*(lbh(1,3)+lbh(8,3)-2*(x(8,3)-x(1,3)));
% (lbh(1,1)+lbh(9,1)-2*(x(9,1)-x(1,1)))*(lbh(1,3)+lbh(9,3)-2*(x(9,3)-x(1,3)));
% (lbh(2,1)+lbh(3,1)-2*(x(3,1)-x(2,1)))*(lbh(2,3)+lbh(3,3)-2*(x(3,3)-x(2,3)));
% (lbh(2,1)+lbh(4,1)-2*(x(4,1)-x(2,1)))*(lbh(2,3)+lbh(4,3)-2*(x(4,3)-x(2,3)));
% (lbh(2,1)+lbh(5,1)-2*(x(5,1)-x(2,1)))*(lbh(2,3)+lbh(5,3)-2*(x(5,3)-x(2,3)));
% (lbh(2,1)+lbh(6,1)-2*(x(6,1)-x(2,1)))*(lbh(2,3)+lbh(6,3)-2*(x(6,3)-x(2,3)));
% (lbh(2,1)+lbh(7,1)-2*(x(7,1)-x(2,1)))*(lbh(2,3)+lbh(7,3)-2*(x(7,3)-x(2,3)));
% (lbh(2,1)+lbh(8,1)-2*(x(8,1)-x(2,1)))*(lbh(2,3)+lbh(8,3)-2*(x(8,3)-x(2,3)));
% (lbh(2,1)+lbh(9,1)-2*(x(9,1)-x(2,1)))*(lbh(2,3)+lbh(9,3)-2*(x(9,3)-x(2,3)));
% (lbh(3,1)+lbh(4,1)-2*(x(4,1)-x(3,1)))*(lbh(3,3)+lbh(4,3)-2*(x(4,3)-x(3,3)));
% (lbh(3,1)+lbh(5,1)-2*(x(5,1)-x(3,1)))*(lbh(3,3)+lbh(5,3)-2*(x(5,3)-x(3,3)));
% (lbh(3,1)+lbh(6,1)-2*(x(6,1)-x(3,1)))*(lbh(3,3)+lbh(6,3)-2*(x(6,3)-x(3,3)));
% (lbh(3,1)+lbh(7,1)-2*(x(7,1)-x(3,1)))*(lbh(3,3)+lbh(7,3)-2*(x(7,3)-x(3,3)));
% (lbh(3,1)+lbh(8,1)-2*(x(8,1)-x(3,1)))*(lbh(3,3)+lbh(8,3)-2*(x(8,3)-x(3,3)));
% (lbh(3,1)+lbh(9,1)-2*(x(9,1)-x(3,1)))*(lbh(3,3)+lbh(9,3)-2*(x(9,3)-x(3,3)));
% (lbh(4,1)+lbh(5,1)-2*(x(5,1)-x(4,1)))*(lbh(4,3)+lbh(5,3)-2*(x(5,3)-x(4,3)));
% (lbh(4,1)+lbh(6,1)-2*(x(6,1)-x(4,1)))*(lbh(4,3)+lbh(6,3)-2*(x(6,3)-x(4,3)));
% (lbh(4,1)+lbh(7,1)-2*(x(7,1)-x(4,1)))*(lbh(4,3)+lbh(7,3)-2*(x(7,3)-x(4,3)));
% (lbh(4,1)+lbh(8,1)-2*(x(8,1)-x(4,1)))*(lbh(4,3)+lbh(8,3)-2*(x(8,3)-x(4,3)));
% (lbh(4,1)+lbh(9,1)-2*(x(9,1)-x(4,1)))*(lbh(4,3)+lbh(9,3)-2*(x(9,3)-x(4,3)));
% (lbh(5,1)+lbh(6,1)-2*(x(6,1)-x(5,1)))*(lbh(5,3)+lbh(6,3)-2*(x(6,3)-x(5,3)));
% (lbh(5,1)+lbh(7,1)-2*(x(7,1)-x(5,1)))*(lbh(5,3)+lbh(7,3)-2*(x(7,3)-x(5,3)));
% (lbh(5,1)+lbh(8,1)-2*(x(8,1)-x(5,1)))*(lbh(5,3)+lbh(8,3)-2*(x(8,3)-x(5,3)));
% (lbh(5,1)+lbh(9,1)-2*(x(9,1)-x(5,1)))*(lbh(5,3)+lbh(9,3)-2*(x(9,3)-x(5,3)));
% (lbh(6,1)+lbh(7,1)-2*(x(7,1)-x(6,1)))*(lbh(6,3)+lbh(7,3)-2*(x(7,3)-x(6,3)));
% (lbh(6,1)+lbh(8,1)-2*(x(8,1)-x(6,1)))*(lbh(6,3)+lbh(8,3)-2*(x(8,3)-x(6,3)));
% (lbh(6,1)+lbh(9,1)-2*(x(9,1)-x(6,1)))*(lbh(6,3)+lbh(9,3)-2*(x(9,3)-x(6,3)));
% (lbh(7,1)+lbh(8,1)-2*(x(8,1)-x(7,1)))*(lbh(7,3)+lbh(8,3)-2*(x(8,3)-x(7,3)));
% (lbh(7,1)+lbh(9,1)-2*(x(9,1)-x(7,1)))*(lbh(7,3)+lbh(9,3)-2*(x(9,3)-x(7,3)));
% (lbh(8,1)+lbh(9,1)-2*(x(9,1)-x(8,1)))*(lbh(8,3)+lbh(9,3)-2*(x(9,3)-x(8,3)))];
% h=[x(1,1);
%     x(2,1);
```

```
%     x(3,1);
%     x(4,1);
%     x(5,1);
%     x(6,1);
%     x(7,1);
%     x(8,1);
%     x(9,1)];
```

Published with MATLAB® R2014b

Main Fucntion File

```
clear
clc
close all
A=[]; b=[];  Aeq=[]; beq=[]; % matrix/vectors for defining linear constraints (not used)
lb = -155*ones(9,3); % lower bounds on the problem
ub = 155*ones(9,3); % upper bounds on the problem (not used)

lbh=[41 59      30
    104 36      25
    39  26      14
    10  01      01
    50  45      7
    50  45      7
    50  45      7
    50  45      7
    55  55      19];



% x0=[0 70.71  70.71
%  0 -74.9736309       -74.94815692
% 0  -74.9604302       29.36963137
% 0  -74.97032212      38.44952616
% 0   43.2397855       -6.40497E-07
% 0  -1.14017E-07      5.238782418
% 0  -44.7502298       -6.40497E-07
% 0  -1.14017E-07      -60.76237042
% 0  -7.090006119      -74.94384906];



x0=[69.99999616        -4.989582001   -27.49999866
-61.40892586   -31.78971042   3.000001343
-3.342618498   4.556611221    -5.499998657
18.36624045    -2.909927013   1.500001343
-55.81838707   8.710289578    5.000001343
0       -32.36382387   19.00000134
-12.36620809   12.63617613    12.00000134
7.28E-13       12.63617613    19.00000134
```

```
-46.63555538   8.777733711    32.00000134];

% x0=[29.68489167      1.827086848    15
% -13.08640507 49.32708685    12.5
% -19.31510833 81.5281823     7
% 45.18489167  54.73336018    0.5
% -24.81510833 117.3567865    3.50E+00
% 75.18489167  4.39E+01       3.5
% 25.18489167  89.82708685    3.50E+00
% 75.18489167  8.98E+01       3.5
% 27.68489167  139.8270868    9.5];



opts = optimset('Display','iter','Algorithm','sqp',...
                  'MaxFunEval',Inf,'MaxIter',Inf);

[xopt,fval,exitflag,output,lambda] = fmincon(@do_project,x0,A,b,Aeq,beq,lb,ub,@overlap,opts);



subsetD=0.5*ones(9,3);
% subsetA= x0;
subsetA = [xopt];
subsetB = lbh;
subsetC = [0; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0 ; 0];
DrawCuboid(subsetB(1,:)', subsetA(1,:)', [subsetC(1);0;0]);
hold on
DrawCuboid(subsetB(2,:)', subsetA(2,:)', [subsetC(2);0;0])
hold on
DrawCuboid(subsetB(3,:)', subsetA(3,:)', [subsetC(3);0;0])
hold on
DrawCuboid(subsetB(4,:)', subsetA(4,:)', [subsetC(4);0;0])
hold on
DrawCuboid(subsetB(5,:)', subsetA(5,:)', [subsetC(5);0;0])
hold on
DrawCuboid(subsetB(6,:)', subsetA(6,:)', [subsetC(6);0;0])
hold on
DrawCuboid(subsetB(7,:)', subsetA(7,:)', [subsetC(7);0;0])
hold on
DrawCuboid(subsetB(8,:)', subsetA(8,:)', [subsetC(8);0;0])
hold on
DrawCuboid(subsetB(9,:)', subsetA(9,:)', [subsetC(9);0;0])
hold on
```

Objectivee Function

```
function[f]=do_project(x)
weight=[200 170 20 15 30 30 30 30 40]; % camera battery reciever antenna esc1 esc2 esc3 esc4
flt_cntrol_circuit

fun1=x(1,1)*weight(1)+x(2,1)*weight(2)+x(3,1)*weight(3)+x(4,1)*weight(4)+x(5,1)*weight(5)+x(6,1)*
weight(6)+x(7,1)*weight(7)+x(8,1)*weight(8)+x(9,1)*weight(9);
fun2=x(1,2)*weight(1)+x(2,2)*weight(2)+x(3,2)*weight(3)+x(4,2)*weight(4)+x(5,2)*weight(5)+x(6,2)*
weight(6)+x(7,2)*weight(7)+x(8,2)*weight(8)+x(9,2)*weight(9);

f=(fun1^2+fun2^2);
display(f);
end
```

```
Error using do_project (line 4)
Not enough input arguments.
```

Published with MATLAB® R2014b

Stacking

```
function [f]= soln_stack_close(x)

f=(max(x(:,1)) - min(x(:,1))) * (max(x(:,2)) - min(x(:,2))) - 19203;
%  f=max(x(:,3)) - min(x(:,3));
```

```
Error using soln_stack_close (line 3)
Not enough input arguments.
```

Published with MATLAB® R2014b

5th Code


```
function [mesh] = DrawCuboid1(varargin)
```

```
% Draw Cuboid
% Draw a Cuboid using 8 rectangular faces. Places Cuboid Into Current Figure
%
% Inputs (SL, CV, EA, colr, alph)
```

```
% ----------------
% SL    - [X;Y;Z] Length of Cuboid Side (SL - SideLength)
% CV    - [X;Y;Z] Center of volume
% EA    - [Yaw(Z-axis);Pitch(y-axis);Roll(x-axis)] Euler/Rotation angles [radians]
% colr  - Color of cuboid; string (ex. 'r','b','g') or vector [R G B]
% alph  - Alpha transparency value of cuboid
%
% Outputs (CuboidHandle, verts, facs)
% ----------------
% CuboidHandle  = Handle for Patch Object
% verts         = 3x8 XYZ Vertices
% facs          = 6x4 Order of faces
%
```

Number of inputs and assign defaults if not specified

Define Default Values

```
colr = [0.5 0.5 0.5];   alph = 0.01;

switch nargin
    case 0
        % All Inputs Empty Using Default Values
    case 1
        SL = varargin{1};
    case 2
        SL = varargin{1};   CV = varargin{2};
    case 3
        SL = varargin{1};   CV = varargin{2};   EA = varargin{3};
    case 4
        SL = varargin{1};   CV = varargin{2};   EA = varargin{3};   colr = varargin{4};
    case 5
        SL = varargin{1};   CV = varargin{2};   EA = varargin{3};   colr = varargin{4};     alph
= varargin{5};
    otherwise
        error('Invalid number of inputs')
end
```

Form ZYX Rotation Matrix

```
% Calculate Sines and Cosines
c1 = cos(EA(1));       s1 = sin(EA(1));
c2 = cos(EA(2));       s2 = sin(EA(2));
c3 = cos(EA(3));       s3 = sin(EA(3));

% Calculate Matrix
R = [c1*c2           -c2*s1          s2
     c3*s1+c1*s2*s3  c1*c3-s1*s2*s3  -c2*s3
     s1*s3-c1*c3*s2  c3*s1*s2+c1*s3  c2*c3]';
```

```
Undefined function or variable "EA".
```

```
Error in DrawCuboid1 (line 46)
c1 = cos(EA(1));       s1 = sin(EA(1));
```

Create Vertices

```
x = 0.5*SL(1)*[-1 1 1 -1 -1 1 1 -1]';
y = 0.5*SL(2)*[1 1 1 1 -1 -1 -1 -1]';
z = 0.5*SL(3)*[-1 -1 1 1 1 1 -1 -1]';
```

Create Faces

```
facs = [1 2 3 4
        5 6 7 8
        4 3 6 5
        3 2 7 6
        2 1 8 7
        1 4 5 8];
```

Rotate and Translate Vertices

```
verts = zeros(3,8);
for i = 1:8
    verts(1:3,i) = R*[x(i);y(i);z(i)]+CV;
end
mesh.x=verts;
```

Draw Patch Object

CuboidHandle = patch('Faces',facs,'Vertices',verts','FaceColor',colr,'FaceAlpha',alph);

```
end
```

Published with MATLAB® R2014b

Norm of First-order

Iter	F-count	f(x)	Feasibility	Steplength	step	optimality
0	28	5.314798e+07	2.941e+04			2.402e+06
1	62	4.483138e+07	2.603e+04	1.176e-01	3.852e+01	2.073e+06
2	90	2.599893e+06	3.708e+04	1.000e+00	2.258e+02	3.325e+06
3	130	2.613520e+06	3.657e+04	1.384e-02	3.555e+00	3.304e+06
4	164	8.021499e+05	3.233e+04	1.176e-01	2.352e+01	3.181e+06
5	192	3.473763e+05	9.463e+03	1.000e+00	1.661e+02	5.212e+05

6	228	3.279172e+05	8.290e+03	5.765e-02	1.140e+01	3.923e+05
7	257	4.524239e+05	2.389e+03	7.000e-01	2.661e+01	2.634e+05
8	285	4.262990e+05	1.926e+02	1.000e+00	5.188e+01	9.046e+04
9	316	4.318100e+05	1.315e+02	3.430e-01	3.232e+01	2.138e+04
10	345	4.425477e+05	5.208e+01	7.000e-01	4.648e+01	3.380e+04
11	373	4.485856e+05	7.672e+00	1.000e+00	3.472e+01	4.960e+04
12	401	4.455930e+05	6.916e-02	1.000e+00	1.134e+01	6.496e+04
13	429	4.349366e+05	2.206e-05	1.000e+00	1.861e+00	6.743e+04
14	457	3.839836e+05	8.786e-05	1.000e+00	9.430e+00	7.991e+04
15	485	1.983903e+05	6.371e-05	1.000e+00	4.227e+01	6.084e+04
16	515	7.052409e+04	5.303e-05	7.000e-01	7.637e+01	1.128e+05
17	545	5.971113e+04	7.373e-07	4.900e-01	1.935e+01	1.048e+05
18	573	2.206043e+02	4.395e+01	1.000e+00	1.602e+01	2.459e+04
19	607	1.062749e+03	3.870e+01	1.176e-01	3.444e+00	3.290e+04
20	635	1.151411e+03	1.751e+01	1.000e+00	3.326e+01	1.089e+04
21	663	2.877511e+02	1.817e+00	1.000e+00	2.732e+01	5.131e+03
22	691	1.081395e+02	1.910e+00	1.000e+00	4.413e+00	4.079e+03
23	721	2.926709e+02	3.488e-02	4.900e-01	2.359e+00	7.188e+03
24	750	2.294622e+02	4.374e-02	1.000e+00	2.392e+01	5.830e+03
25	784	2.130768e+02	3.860e-02	1.176e-01	2.138e+00	3.993e+03
26	812	1.171310e+02	1.674e-01	1.000e+00	6.186e+00	4.050e+03
27	846	7.219166e+01	1.477e-01	1.176e-01	2.072e+00	2.967e+03
28	875	4.454031e+01	4.815e-02	7.000e-01	1.404e+01	2.398e+03
29	904	1.666152e+01	5.088e-04	1.000e+00	1.320e+01	1.259e+03
30	934	2.942492e+00	0.000e+00	7.000e-01	8.955e+00	5.097e+02

Iter	F-count	f(x)	Feasibility	Steplength	step	Norm of First-order optimality
31	964	8.269628e-02	0.000e+00	7.000e-01	5.788e+00	1.117e+02
32	994	1.819269e-03	0.000e+00	7.000e-01	1.937e+00	1.601e+01
33	1022	7.219641e-04	1.110e-16	1.000e+00	2.264e+00	9.063e+00
34	1051	7.219641e-04	1.110e-16	3.220e-05	3.097e-05	9.063e+00

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.