Executive Summary

The task was to minimize the mass of a wing spar of an unmanned aerial vehicle without the structure reaching the ultimate strength of carbon fiber, the material. This was done by changing the inner and outer radii of the spar, and the number of elements along the length. The optimal value for the number of elements is 75, giving the spar a mass of 5.0170 kg.

Analysis

The spar was modelled using the Euler-Bernoulli Beam Theory. There were multiple assumptions made while using this theory. The cross section of the beam had planar symmetry, and the cross section varies smoothly. The internal strain energy accounts only for bending moment deformations. The material is assumed to be elastic and isotropic, and any deformations are negligible.

The displacement of the beam is given by Equation 1 below.

$$q = \frac{d^2}{dx^2}\left(EI_{yy}\frac{d^2w}{dx^2}\right) \qquad \textit{Equation 1}$$

q(x) is the applied load, E is Young's modulus with a constant value of 70 GPa, I_{yy} is the second moment of area with respect to the y axis, and w is the vertical displacement in the z direction. q(x) was calculated by Equations 2 and 3 below

$$\int_0^L q(x)dx = 2.5 * W/2\ , where\ q(L) = 0 \qquad \textit{Equation 2}$$

L is the length of the spar, 7.5 meters, and W is the total weight of the UAV, which is 500 kg times gravity, or 4905N. Equation 3 was derived from Equation 2 to calculate the force along each node along the length. As seen in this equation, the force distribution is a linear distribution with maximum load at the base and zero load at the tip.

$$q(x) = \int_0^L 1633\frac{1}{3} - 217\frac{7}{9}x \qquad \textit{Equation 3}$$

I_{yy} was calculated using Equation 4 and 5. Equation 4 is the general form of the second moment of area, and Equation 5 is the specific formula for the second moment of area for a circular annulus.

$$I_{yy} = \iint z^2 dzdy \qquad \textit{Equation 4}$$

$$I_{yy} = \frac{\pi}{4}(rout^4 - rin^4) \qquad \textit{Equation 5}$$

rout is the outer radius and rin is the inner radius.

The stress was calculated using Equation 6 below.

$$\sigma_{xx}(x) = -z_{max}E\frac{d^2w}{dx^2} \qquad \textit{Equation 6}$$

σ_{xx} is the stress at each node, and z_{max} is the maximum height of the beam at each node.

Finally the mass was calculated using the density and volume of the annulus, as shown in Equations 7 and 8.

$$m = \rho V \qquad \textit{Equation 7}$$

$$V = \int_0^L \pi(rout^2 - rin^2)dx \qquad \textit{Equation 8}$$

ρ is the density of carbon fiber with a value 1600kg/m^3 and V is the volume.

Design Variables

There were two different types of design variables in this task, the inner radius and the thickness of the annulus. This was organized in a single array of form n inner radius elements, then n thickness elements, where n is the number of nodes. In order to calculate the outer radius, the inner radius and the thickness at each individual node was summed up to create a new array of size n for the outer radius. The only thing that was changed manually was the number of elements, which is the number of nodes minus one.

Optimization

The objective was to design the spar, the main structural support for a wing, for a new aircraft. The goal was to minimize the mass of the spar. The spar would be a circular annulus. The inner radius must be at least 1 cm, and the other radius must be at most 5 cm. The thickness must be at least 2.5mm. The aircraft can undergo a maneuver where the total force on the spar is 2.5 times the weight of the entire aircraft, 500kg. The spar cannot reach the ultimate strength of the carbon fiber, 600 MPa, during this maneuver. In order to start within the constraints, the initial guess for the inner radius and the thickness were all 1 cm. The inner radii and the thickness of the annulus were changed by the MATLAB function fmincon to meet these constraints and to minimize mass.

Results

To calculate the optimal solution, the number of design variables were changed for different configurations. The first values calculated were the nominal values, with an inner radii of 4.635 cm and outer radii of 5 cm throughout the entire spar. This gave a nominal mass of 13.26 kg. In order to consider this problem optimized, the spar mass must be 60% lower than this nominal value.

In all the simulations, the radii followed the same trend. The outer radii starts at 5 cm, decreases linearly, and then flattens out and ends at 1.25 cm. The thickness is a little higher than 2.5 mm to start with, but is 2.5mm for most of the length of the spar. The stress also followed the same trend throughout all the simulations. It starts at 600 MPa, and then starts to decrease to 0.

The number of elements was changed from 10 to 100, as shown in Table 1 below.

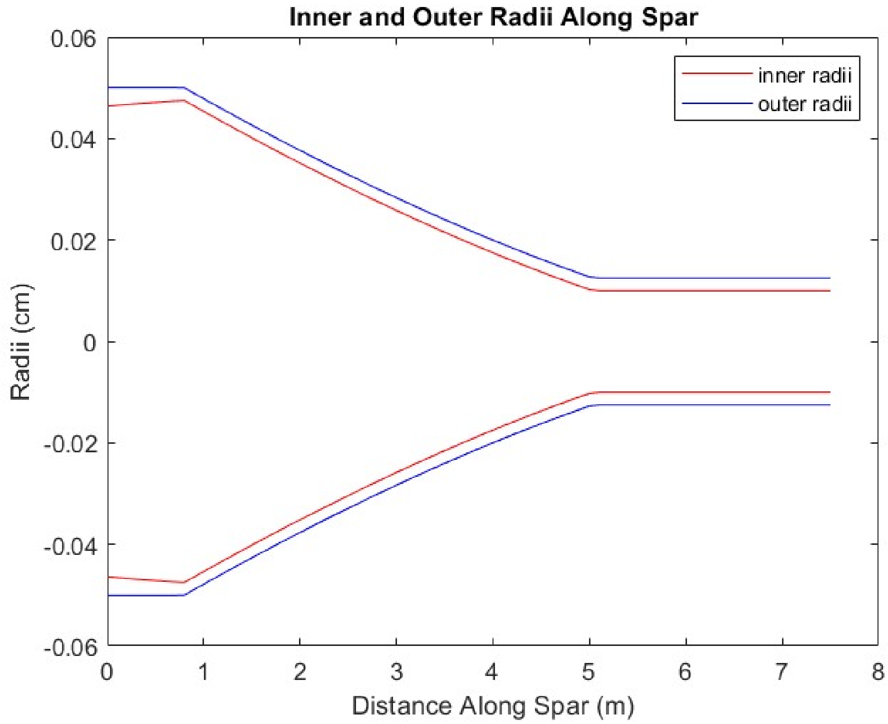
Table 1: Number of Elements vs Time

Number of Elements	10	20	50	75	100
Mass (kg)	5.6441	5.2901	5.0677	5.0170	4.9919
Time (s)	2.760	5.086	16.704	36.972	119.462

As seen in Table 1, as the number of elements increase, the mass decreases. However, the time it takes to run increases dramatically. The optimal solution is having 75 elements. This is because the cost of time isn't very long, but it has the second lowest mass. Having 100 elements took almost a minute and a half longer for the design optimization for a decrease of only .0251 kg. It was determined such a small difference was not worth the extra cost. With a mass of 5.0170, this design is 62.16% lower than the nominal design.

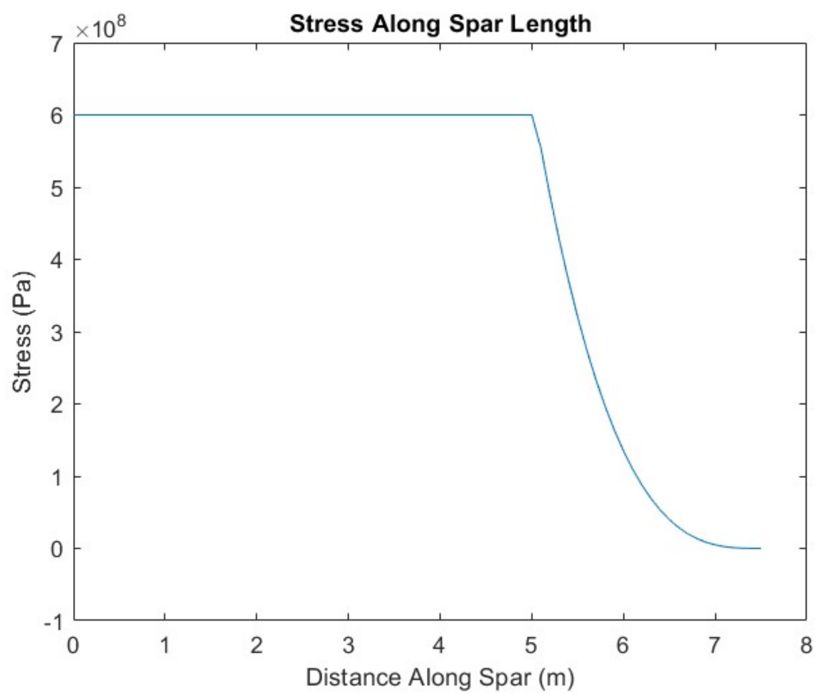
The radii of the spar with 75 elements is shown in Figure 1 below.

Figure 1



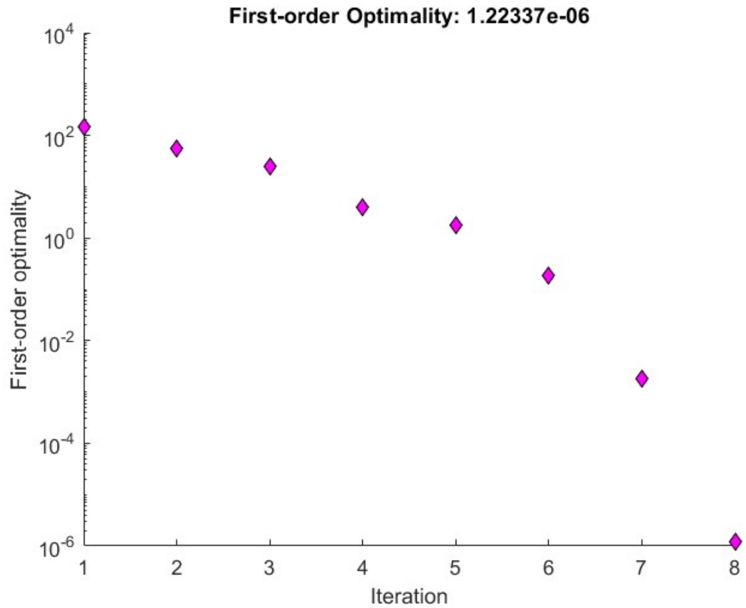
The stress is shown in Figure 2 below.

Figure 2



These values are optimal, as it has a first-order optimality of 1.22337e-6, as shown in Figure 3 below.

Figure 3



Conclusion

In conclusion, having the spar being split into 75 elements will allow for an optimized mass of 5.017 kg. The radii starts at its upper bound, decreases and flattens out at its lower bound. The stress is near the ultimate strength for over half the spar, and the decreases dramatically to 0 at the tip of the spar.

Appendix

A1: findradii.m

```
function [rin, rout] = findradii(Nelem,r)
%function to seperate the design variables
%Inputs
%   Nelem = number of finite elements to use
%   r = an 1D array consisting of the inner radii and thickness, with
inner
%   radii being the first half and the second half being thickness
%Outputs
%   rin = inner radius
%   rout = outer radius

rin = zeros(Nelem+1,1);
rout = zeros(Nelem+1,1);
for i = 1:Nelem+1
    rin(i) = r(i);
    rout(i) = r(i)+r(Nelem+1+i);
end

end
```

A2: calcmass.m

```
function m = calcmass(Nelem,D,x, r)
%function to compute the mass of the spar
%Inputs
%   Nelem = number of finite elements to use
%   D = Density of carbon fiber composite
%   rin = inner radius
%   rout = outer radius
%Outputs
%   m = mass of the spar
[rin, rout] = findradii(Nelem,r);
volume = 0;
dx = x(2) - x(1);
for i = 1:Nelem+1
    volume =volume + pi*(rout(i)^2-rin(i)^2)*dx;
end
m = D*volume;

end
```

A3: obj.m

```
function [f, g] = obj(L,D,E,U,Nelem,x,r)
%objective function for fmincon
%Inputs
%   L = length of the beam
%   D = Density of carbon fiber composite
%   E = longitudinal elastic modulus
%   U = ultimate tensile/compressive strength
%   Nelem = number of finite elements to use
%   x = evenly spaced nodes along the length
%   r = an 1D array consisting of the inner radii and thickness, with
inner
%   radii being the first half and the second half being thickness
%Outputs
%   f = objective value at design variables

f = calcmass(Nelem, D, x, r);
g = zeros(2*(Nelem+1),1);
h = 1e-60;
for i = 1:2*(Nelem+1)
    rc = r;
    rc(i) = rc(i)+complex(0,h);
    g(i) = imag(calcmass(Nelem, D, x, rc))/h;
end

end
```

A4: calcmoa.m

```
function [Iyy] = calcmoa(Nelem,x,rin,rout,mass)
%function to calculate the moment of inertia Iyy
%Inputs
%   Nelem = number of finite elements to use
%   x = evenly spaced nodes along the length
%   rin = inner radius
%   rout = outer radius
%   mass = mass of the spar
%Outputs
%   Iyy = moment of inertia with respect to the y axis, as function of
x
Iyy = zeros(Nelem+1, 1);
for i = 1:Nelem+1
    %equation of a circular annulus
    Iyy(i) = pi/4*(rout(i)^4-rin(i)^4);
    if Iyy(i) < 1e-12
        Iyy(i) = 1e-12;
    end
end

end
```

A5: calcforce.m

```
function [force] = calcforce(Nelem, x)
%function to calculate the force at each node
%Inputs
%   Nelem = number of finite elements to use
%   x = evenly spaced nodes along the length
%Outputs
%   force = force along the length
force = zeros(Nelem+1,1);
for i = 1:Nelem+1
    %calculated from integral of 2.5W/2
    force(i) = 1633+1/3-(217+7/9)*x(i);
end

end
```

A6: cnstr_func.m

```
function [sigma] = cnstr_func(L,D,E,Nelem,x,r)
%function to call other functions to put into constraints function
%Inputs
%   L = length of the beam
%   D = Density of carbon fiber composite
%   E = longitudinal elastic modulus
%   Nelem = number of finite elements to use
%   x = evenly spaced nodes along the length
%   r = an 1D array consisting of the inner radii and thickness, with
inner
%   radii being the first half and the second half being thickness
%Outputs
%   sigma = stress at each node in the beam
    [rin, rout] = findradii(Nelem,r);
    mass = calcmass(Nelem, D, x, r);
    Iyy = calcmoa(Nelem,x,rin,rout,mass);
    force = calcforce(Nelem, x);
    u = CalcBeamDisplacement(L, E, Iyy, force, Nelem);
    sigma = CalcBeamStress(L, E, rout, u, Nelem);
end
```

A7: constraints.m

```
function [cineq, c, Jineq, J] = constraints(L,D,E,U,Nelem,x,r)
%function to calculate the nonlinear stress constraint
%Inputs
%   L = length of the beam
%   D = Density of carbon fiber composite
%   E = longitudinal elastic modulus
%   U = ultimate tensile/compressive strength
%   W = weight of the entire UAV
%   Nelem = number of finite elements to use
%   x = evenly spaced nodes along the length
%   r = an 1D array consisting of the inner radii and thickness, with
inner
%   radii being the first half and the second half being thickness
%Outputs
%   cineq = minimized stress constraint along each node of along the
length
%   c = empty array
%   Jineq = jacobian of constraint gradients
%   J = empty array

sigma = cnstr_func(L,D,E,Nelem,x,r);
cineq = zeros(Nelem+1,1);
%puting cineq < 0
for i = 1:Nelem+1
    cineq(i) = sigma(i)/U-1;
end
```

```
Jineq = zeros(2*(Nelem+1), Nelem+1);
h = 1e-60;
%complex step
for i = 1:2*(Nelem+1)
    rc = r;
    rc(i) = rc(i) +complex(0,h);
    cineqc = cnstr_func(L,D,E,U,Nelem,x,rc);
    Jineq(i,:) = imag(cineqc/U-1)/h;
end
c=[];
J=[];

end
```

A8: run_obj.m

```
%% trying to minimize weight
%two design varaibles: either inner radii and thickness or inner radii
and
%outer radii
close all
clear all
%% parameters
L = 7.5;
D = 1600;
E = 70*10^9;
U = 600*10^6;
W = 500;
Nelem = 75;
x = [0:L/Nelem:L];
r = .01*ones(2*(Nelem+1),1);

%% defining obj function
fun = @(r) obj(L,D,E,U,Nelem,x,r);
cnstr = @(r) constraints(L,D,E,U,Nelem,x,r);
%% defining Aineq and Bineq
nvar = size(r,1);
Aineq = zeros((Nelem+1),nvar);
bineq = zeros((Nelem+1), 1);
for i = 1:Nelem+1
    %constraint of outer radius < .05
    ptr = Nelem+1;
    %Aineq(i,1) = 1;
    for k = 1:nvar
        if i == k
            Aineq(i,k) = 1;%r(i) + r(i+ptr);
        end
        if i == k-ptr
            Aineq(i,k) = 1;
        end
```

```
    end
    bineq(i) = .05;
end
lb = zeros(2*(Nelem+1),1);
ub = zeros(2*(Nelem+1),1);
for i = 1:Nelem+1
    lb(i) = .01;
    ub(i) = .0475;
end
for i = Nelem+2:2*(Nelem+1)
    lb(i) = .0025;
    ub(i) = .04;
end
%% using fmincon
%creating the options for fmincon
options = optimoptions('fmincon', 'Algorithm', 'sqp','Display',
'iter', 'GradConstr', 'on', 'GradObj','on', 'CheckGradients',
false,"FiniteDifferenceType",
"central",'Plotfcn',@optimplotfirstorderopt );
[rt,fval] = fmincon(fun, r, Aineq, bineq, [], [], lb, ub, cnstr,
options);
%% results
%plotting the radii of the annulus at each node
rt;
[rin, rout] = findradii(Nelem, rt);
plot(x,rin, 'r', x, rout, 'b')
hold on
plot(x,-rin, 'r', x, -rout, 'b')
hold off
xlabel('Distance Along Spar (m)')
ylabel('Radii (cm)')
title('Inner and Outer Radii Along Spar')
legend('inner radii','outer radii')

%plotting sigma
mass = calcmass(Nelem, D, x, rt)
Iyy = calcmoa(Nelem,x,rin,rout,mass);
force = calcforce(Nelem, x);
u = CalcBeamDisplacement(L, E, Iyy, force, Nelem);
sigma = CalcBeamStress(L, E, rout, u, Nelem);
figure
plot(x,sigma)
xlabel('Distance Along Spar (m)')
ylabel('Stress (Pa)')
title('Stress Along Spar Length')
```