Design of a Quadcopter for Payload Delivery .

By: Adam Haj.

Major : Mechanical Engineer.

University : UMES

Date: July 28th, 2018

Advisor: Dr Payam Matin.

Project Goals

- To design a Quadcopter capable of flying to a specific location, dropping off a package and returning to the original location autonomously.

Design and Prototype

- Must be able to transport a package of approximately 3 lb.
- Must be able to move autonomously from one place to another specified location.
- Lithium polymer battery powered.
- Flight time of at least 30 min.
- Package maximum size of 8 in. X 11 in. X 12 in.

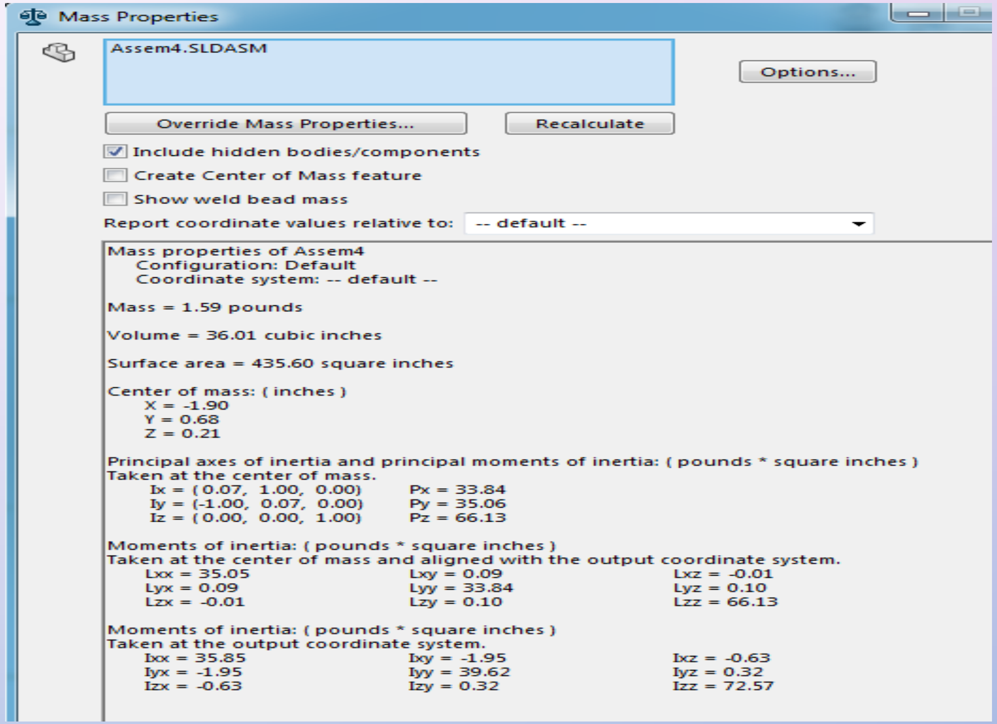
Design Constraints

- Weight cannot exceed: 55 lbs(25Kg) with a drone registration.
- FAA Part 107 (Unmanned Aerial Vehicle Operation) .
- Attitude cannot exceed: 400 ft AGL.
- Cost must be less than: $300.
- Must stay in the line of sight .

The Prototype

Cont.....

SolidWorks Modeling



Governing Equations

- The CLMEs .

$$m(\dot{U} + QW - RV) = -mg\sin\theta + (F_{A_x} + F_{T_x})$$

$$m(\dot{V} + UR - PW) = mg\cos\theta\sin\phi + (F_{A_y} + F_{T_y})$$

$$m(\dot{W} + PV - QU) = mg\cos\theta\cos\phi + (F_{A_z} + F_{T_z})$$

- The CAMEs.

$$\dot{P}I_{xx} - \dot{R}I_{xz} - PQI_{xz} + RQ(I_{zz} - I_{yy}) = L_A + L_T$$

$$\dot{Q}I_{yy} + PR(I_{xx} - I_{zz}) + (P^2 - R^2)I_{xz} = M_A + M_T$$

$$\dot{R}I_{zz} - \dot{P}I_{xz} + PQ(I_{yy} - I_{xx}) + QRI_{xz} = N_A + N_T$$

Cont….

- Kinematic equations.

$$\begin{Bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{Bmatrix} = \begin{bmatrix} 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{Bmatrix} P \\ Q \\ R \end{Bmatrix}$$

- Flight Path equations.

$$\begin{Bmatrix} \dot{X}' \\ \dot{Y}' \\ \dot{Z}' \end{Bmatrix} = \begin{bmatrix} \cos\Psi\cos\Theta & -\sin\Psi\cos\Phi + \cos\Psi\sin\Theta\sin\Phi & \sin\Psi\sin\Phi + \cos\Psi\sin\Theta\cos\Phi \\ \sin\Psi\cos\Theta & \cos\Psi\cos\Phi + \sin\Psi\sin\Theta\sin\Phi & -\sin\Phi\cos\Psi + \sin\Psi\sin\Theta\cos\Phi \\ -\sin\Theta & \cos\Theta\sin\Phi & \cos\Theta\cos\Phi \end{bmatrix} \begin{Bmatrix} U \\ V \\ W \end{Bmatrix}$$

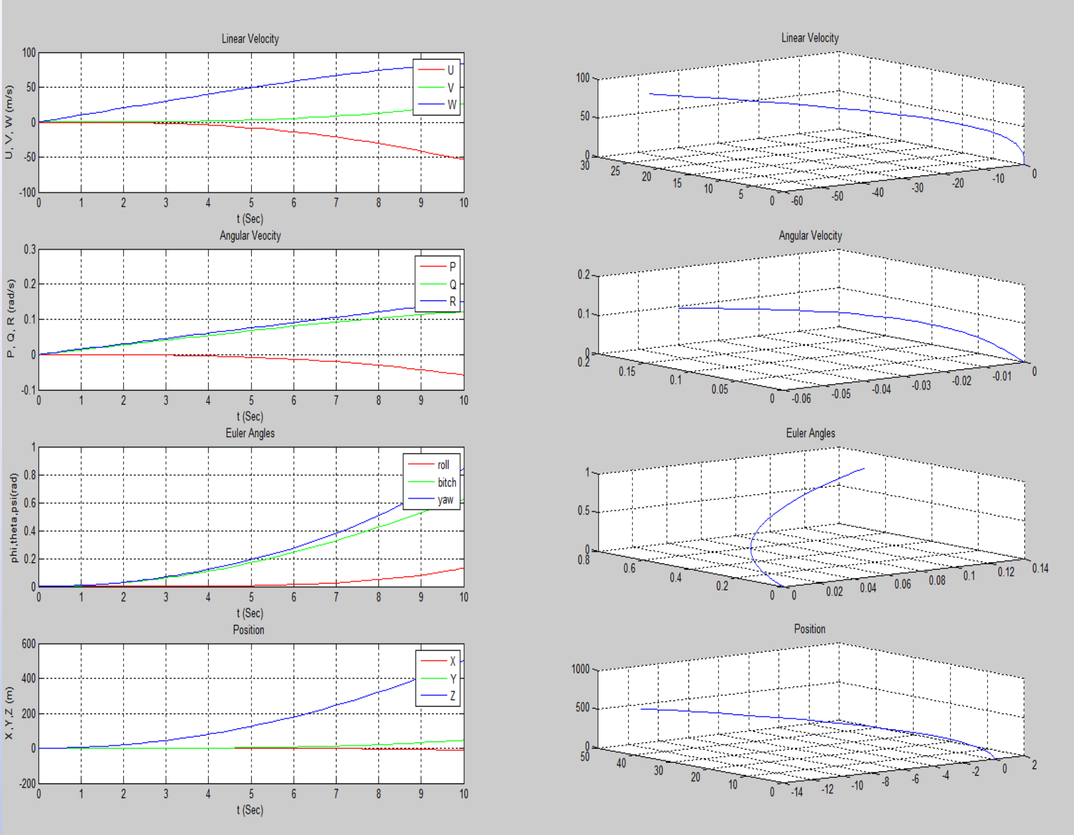
Matlab Code & Simulation

```
clc;
%This is the code meant to perform flight simulations for a Quadcopter
%for the University of Maryland Engineering 2018 Senior Design
%Class. The purpose of this code is to return the variables
%U,V,W,P,Q,R,phi,theta,psi,X,Y,and Z. This will show the behavior of the
%system
%defined by the Equations of Motion described in the Design Calculations
%section. (Equations of motion were created in relation to body frame.)
%Developed by Adam Haj for the senior design 2018.

% constant
if (0<t<10);
    Ftz = 1;
else
    Ftz = 0
end
g = 9.81; % gravity
%Ftz= 1; this is the increase in thrust of two adjacent motors
Izz=66.13 ;
Ixx= 35.05;
Mt=  .48  ;% Ftz *r
Nt=1 ;% = a(1)*b*w(1).^2-a(2)*b*w(2).^2+ a(3)*b*w(3).^2- a(4)*b*w(4).^2 ; % the first propeller cause bitching moment ( for this run only)
m = 5 ; % Mass

%u = [U V W P Q R phi theta psi]
%u(1)= U
%u(2)= V
%u(3)= W
%u(4)= P
%u(5)= Q
%u(6)= R
%u(7)= phi
%u(8)= theta
%u(9)= psi
%u(10)= Xdot
%u(11)= Ydot
%u(12)=Zdot
```

```
F=@(t,u)[-g*sin(u(8))-u(5)*u(3)+u(6)*u(2);                 %Udot
        g*sin(u(8))*cos(u(8))-u(1)*u(6)+u(4)*u(3);          %Vdot    %% CLME %%
        g*cos(u(8))*cos(u(7))+Ftz/m-u(4)*u(2)+u(5)*u(1);    %Wdot

          ((-u(6)*u(5)*Izz)/Ixx)+u(6)*u(5);                 %Pdot
          Mt/Ixx-u(4)*u(6)+(u(4)*u(6)*Izz)/Ixx;             %Qdot    %% CAME %%
          Nt/Izz;                                           %Rdot



                u(4)+sin(u(7))*tan(u(8))*u(5)+cos(u(7))*tan(u(8))*u(6);  %phidot
                cos(u(7))*u(5)-sin(u(7))*u(6);                           %thetadot  %% Kinematic Equations %%
                sin(u(7))*sec(u(8))*u(5)+cos(u(7))*sec(u(8))*u(6);       %psidot

%Flight Fath equations
(cos(u(9))*cos(u(8)))*u(1)+(-sin(u(9))*cos(u(7))+cos(u(9))*sin(u(8))*sin(u(7)))*u(2)+(sin(u(9))*sin(u(7))+cos(u(9))*sin(u(8))*cos(u(7)))*u(3); %Xdot
(sin(u(9))*cos(u(8)))*u(1)+(cos(u(9))*cos(u(7))+sin(u(9))*sin(u(8))*sin(u(7)))*u(2)+(-sin(u(7))*cos(u(9))+sin(u(9))*sin(u(8))*cos(u(7)))*u(3); %Ydot    %% Flight Path Equations
-sin(u(8))*u(1)+cos(u(8))*sin(u(7))*u(2)+cos(u(8))*cos(u(7))*u(3)];%Zdot

[t,u]= ode45(F,[0 10],[0 0 0 0 0 0 0 0 0 0 0 0]);

subplot(4,2,1),plot(t, u(:,1),'-r',t,u(:,2),'-g',t,u(:,3),'-b');
grid on
title('Linear Velocity');
xlabel('t (Sec)')
ylabel(' U, V, W (m/s)');
legend('U','V','W');

subplot(4,2,2),plot3( u(:,1),u(:,2),u(:,3));
grid on
title('Linear Velocity');
```

MATLAB Code & Simulation

```
subplot(4,2,3),plot(t, u(:,4),'-r',t,u(:,5),'-g',t,u(:,6),'-b');
grid on
title('Angular Veocity');
xlabel('t (Sec)')
ylabel(' P, Q, R (rad/s)');
legend('P','Q','R');

subplot(4,2,4),plot3( u(:,4),u(:,5),u(:,6));
grid on
title('Angular Velocity');


subplot(4,2,5),plot(t, u(:,7),'-r',t,u(:,8),'-g',t,u(:,9),'-b');
grid on
title('Euler Angles');
xlabel('t (Sec)');
ylabel('phi,theta,psi(rad)');
legend('roll','bitch','yaw');

subplot(4,2,6),plot3( u(:,7),u(:,8),u(:,9));
grid on
title('Euler Angles');

subplot(4,2,7),plot(t, u(:,10),'-r',t,u(:,11),'-g',t,u(:,12),'-b');
grid on
title('Position ');
xlabel('t (Sec)');
ylabel('X,Y,Z (m)');
legend('X','Y','Z');

subplot(4,2,8),plot3( u(:,10),u(:,11),u(:,12));
grid on
title('Position');
```



The accomplishment

- MATLAB code and simulation on progress .
- camera, servos and actuators being add it to the system .
- Designing and building the prototype.
- Analytical design slide.

Acknowledgment

- ❖ Maryland Space Grant Consortium.
- ❖ University of Maryland Eastern Shore- Engineering Department.

Questions????