UNIVERSITY OF Southampton
School of Engineering Sciences

# Aerodynamics & Flight Mechanics Research Group

## Simulated Flight of a Microlight Aircraft using Zero Order Hold Integration

S. J. Newman

Technical Report AFM-11/11

January 2011

UNIVERSITY OF SOUTHAMPTON

SCHOOL OF ENGINEERING SCIENCES

AERODYNAMICS AND FLIGHT MECHANICS RESEARCH GROUP

# Simulated Flight of a Microlight Aircraft using Zero Order Hold Integration

by

**S. J. Newman**

AFM Report No. AFM 11/11

January 2011

# Nomenclature

| Symbol | Description |
|---|---|
| $X$ | Horizontal Coordinate (+ve Fwd) |
| $Y$ | Vertical Coordinate (+ve Up) |
| $\Psi$ | Pitch Rotation (+ve Nose Up) |
| $L$ | Lift Force |
| $D$ | Drag Force |
| $M$ | Pitching Moment about CG (+ve Nose Up) |
| $U_P$ | Inflow Velocity Component Normal to Wing Chordline |
| $U_T$ | Inflow Velocity Component Parallel to Wing Chordline |
| $l_G$ | Horizontal Distance of CG behind Wing Leading Edge |
| $h_G$ | Vertical Distance of CG below Wing Leading Edge |
| $l_F$ | Horizontal Distance of Force Centre behind CG |
| $h_F$ | Vertical Distance of Force Centre above CG |
| $l_I$ | Horizontal Distance of Inflow Centre behind CG |
| $h_I$ | Vertical Distance of Inflow Centre above CG |
| $\theta$ | Inclination of Thrust Vector (at CG) relative to Wing Chordline |

| Symbol | Description |
| --- | --- |
| Φ | Inflow Angle (+ve from underneath Wing Chordline) |

| | |
| --- | --- |
| $C_L$ | Lift Coefficient |
| $C_{D0}$ | Drag Coefficient |
| $C_M$ | Pitching Moment Coefficient |
| T | Engine/Propeller Thrust Force |

# Tumble Equations of Motion

## *Introduction*

The forces and moments acting on the microlight aircraft are shown in Figure 1 – omitting gravity:

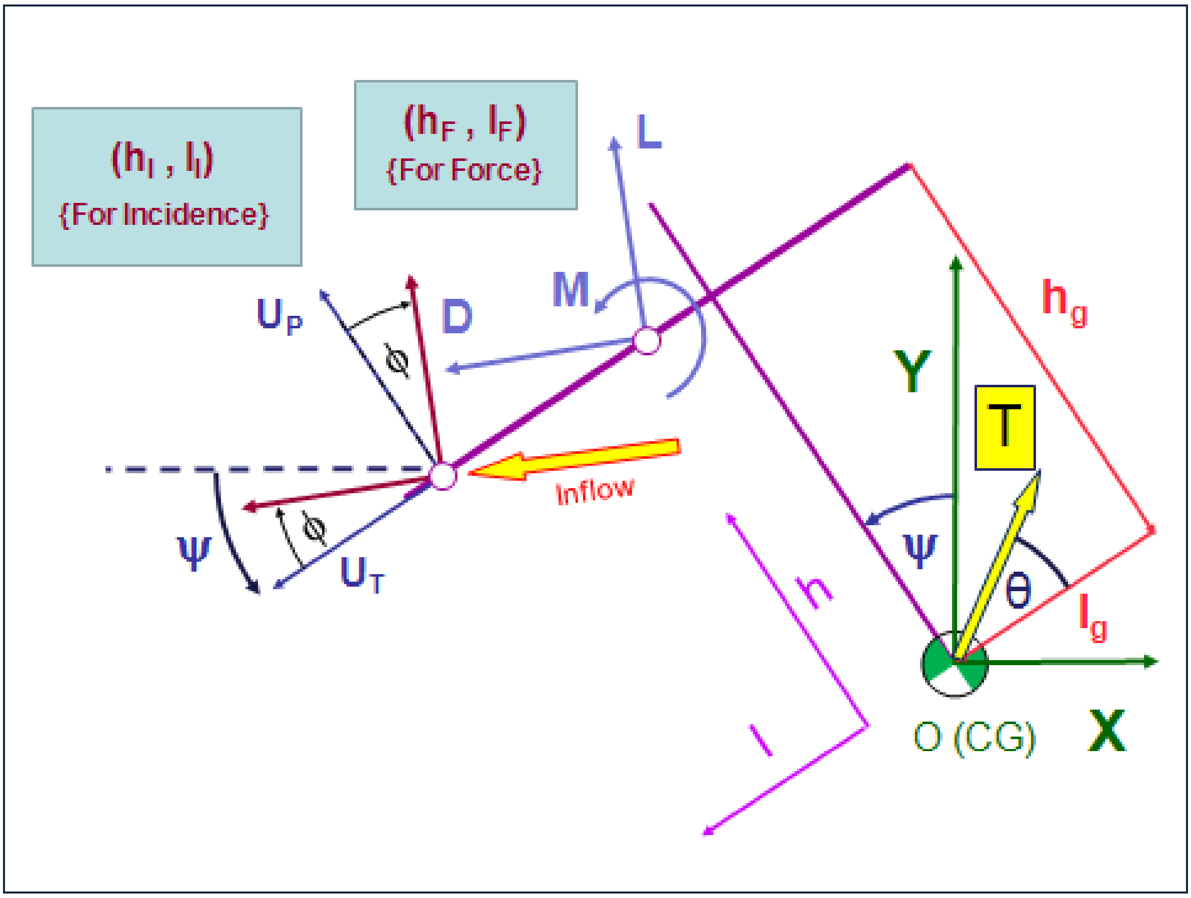


**Figure 1 - Coordinate System of Aircraft**

The aerodynamic forces and moments are calculated using point (I), which is usually taken to be the three-quarter chord. These forces and moments are then applied at point (F) which will normally be the quarter chord.

# Incident Velocities due to CG Translation& Rotation

The incident velocity components parallel to ($U_T$) and normal to ($U_P$) the wing chordline are given by:

$$U_T = \dot{X}\cos\psi + \dot{Y}\sin\psi - h_I\dot{\psi} \qquad (1.)$$

and

$$U_P = \dot{X}\sin\psi - \dot{Y}\cos\psi + l_I\dot{\psi} \qquad (2.)$$

The inflow angle $\phi$ is given by:

$$tan\,\phi = \frac{U_P}{U_T} \qquad (3.)$$

# Deriving the Equations of Motion

## Resolving Forces Horizontally

$$m\ddot{X} = T\cos(\theta+\psi) - L\sin(\psi-\phi) - D\cos(\psi-\phi) \quad (4.)$$

## Resolving Forces Vertically

$$m\ddot{Y} = -mg + T\sin(\theta+\psi) + L\cos(\psi-\phi) - D\sin(\psi-\phi) \quad (5.)$$

## Taking Clockwise Moments about the CG

$$\begin{aligned} mk^2\ddot{\psi} = M & \\ & - L\cdot\{l_F\cos\phi + h_F\sin\phi\} \\ & - D\cdot\{-h_F\cos\phi + l_F\sin\phi\} \end{aligned} \quad (6.)$$

# Calculation of Aerodynamic Forces and Moments

Combining velocity components gives:

$$\overline{V}^2 = U_T^2 + U_P^2 \tag{7.}$$

Hence, the Lift, Drag and Pitching Moment are given by:

$$\begin{bmatrix} L \\ D \\ M \end{bmatrix} = \frac{1}{2}\rho\overline{V}^2 \cdot S \cdot \begin{bmatrix} C_L \\ C_{D0} \\ c \cdot C_M \end{bmatrix} \tag{8.}$$

The pitch damping - generated by the rotation - is given by:

$$C_M = -\frac{c\pi}{8\overline{V}}\dot{\psi} \tag{9.}$$

## Data Input

To ease the data input, the wing coordinate system is shown in Figure 2. The coordinates used in equations of motion are derived in (10).

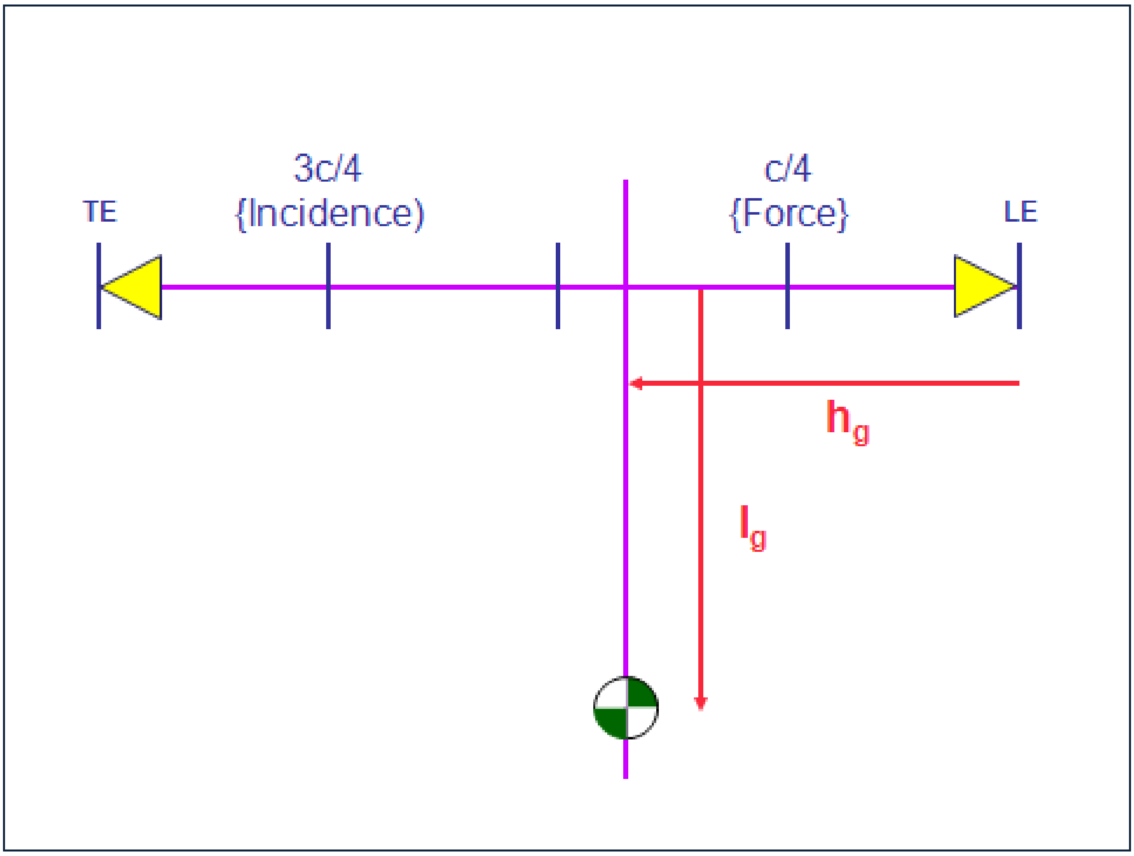


Figure 2 - Coordinate System in Wing

From these coordinates, we have the following results:

$$l_I = \frac{3c}{4} - l_G$$
$$h_I = h_G$$
$$l_F = \frac{c}{4} - l_G \qquad (10.)$$
$$h_F = h_G$$

Specification of Post Angle

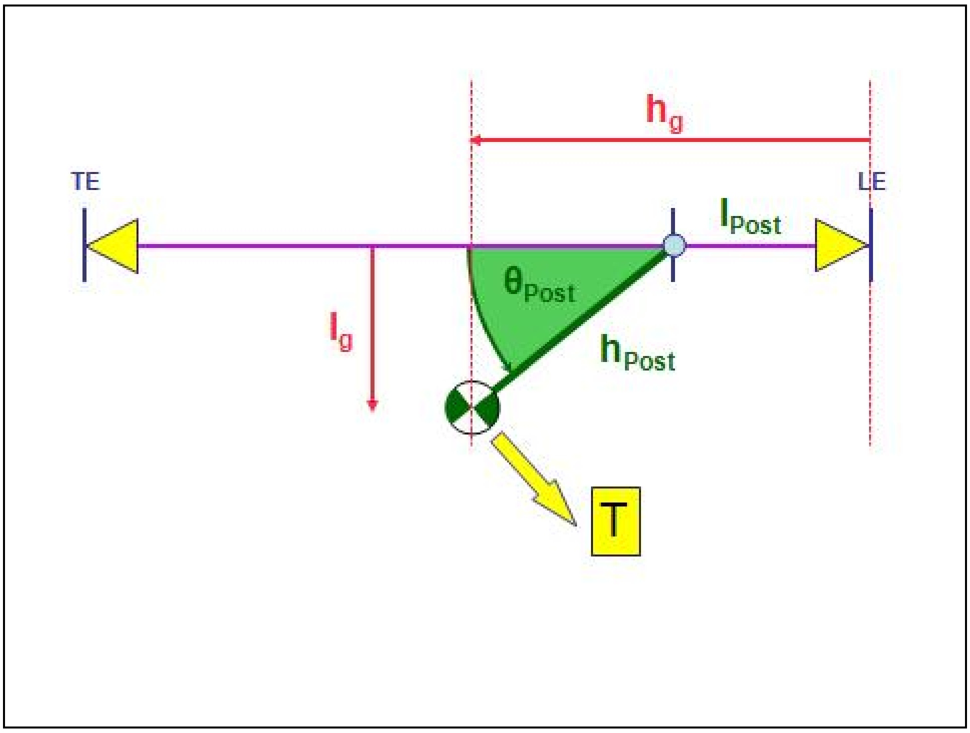


Figure 3 - Post Angle Specification

Using the post pivot location form the leading edge of the wing and the post angle we have for the CG location:

$$l_G = l_{POST} + h_{POST}\cos\left(\theta_{POST}\right)$$
$$h_G = h_{POST}\sin\left(\theta_{POST}\right) \quad (11.)$$

The thrust line is assumed to be perpendicular to the post axis.

From this we conclude that the thrust line inclination to the horizontal (θ) is given by:

$$\theta = \theta_{POST} - \frac{\pi}{2} \quad (12.)$$

## Integration of the Equations of Motion

The equations are integrated using zero-order hold:

$$\begin{aligned}
\Delta X &= \dot{X} \cdot \Delta t + \frac{1}{2}\ddot{X} \cdot \Delta t^2 \\
\Delta Y &= \dot{Y} \cdot \Delta t + \frac{1}{2}\ddot{Y} \cdot \Delta t^2 \\
\Delta \psi &= \dot{\psi} \cdot \Delta t + \frac{1}{2}\ddot{\psi} \cdot \Delta t^2 \\
\Delta \dot{X} &= \ddot{X} \cdot \Delta t \\
\Delta \dot{Y} &= \ddot{Y} \cdot \Delta t \\
\Delta \dot{\psi} &= \ddot{\psi} \cdot \Delta t
\end{aligned} \qquad (13.)$$

A Matlab file follows which gives an animation.

# Matlab File

```
%
%   Microlight Tumble Modelling
%
%   SJN 20th July 2009
%
clear

colordef 'black'

mass=200; % Aircraft Mass
rgyr=1; % Aircraft Pitch Radius of Gyration

wingarea=15; % Wing Area
wingchord=1.5; % Wing Chord
wingpitch=0; % Wing Pitch Angle
dclda=5.8; % Lift Curve Slope
clmax=1.2; % CL Maximum
cd0=0.3; % Drag Coefficient
airden=1.2256; %Air Density
grav=9.81; % Gravitational Acceleration

engt=0; % Propeller Thrust in g

ifmtdamp=1; % Whether to Apply Pitch Rate Damping

lpoc=.25; % Hinge Post Location from LE in wing chords
postloc=.5; % Post Length in wing chords
postangdeg=45; % Post/Wing Angle in degrees

%>>> Plotting Parameters >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

cgrad=.1; % CG Disc Radius

fscale=10; % Dividing Factor for Aero Force Plot
mscale=10; % Dividing Factor for Moment Circle Plot

ntime=10001; % No. of Time Steps
dtime=.0005; % Time Increment
nplotout=25; % Count Interval for Aircraft Position Plots

%>>> Starting Values >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
time=0;

x=0; % Horizontal Coordinate (+ve Fwd)
xdot=50; % Horizontal Velocity (+ve Fwd)
y=0; % Vertical Coordinate (+ve Up)
ydot=0; % Vertical Velocity (+ve Up)
psi=.5; % Pitch Angle (+ve Nose Up)
psidot=0; % Pitch Rate (+ve Nose Up)
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
    auw=mass*grav; % Aircraft Weight
    engthrust=engt*auw; % Engine Thrust

    theta=postangdeg*pi/180;
    lg=(lpoc+postloc*cos(theta))*wingchord; % CG Location from LE Parallel
to Wing
    hg=postloc*sin(theta)*wingchord; % CG Location from LE Normal to Wing

    lp=lpoc*wingchord; % Hinge Post Location from LE along Wing
    hp=postloc*wingchord; % Post Length
%>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

    theta1=theta-pi/2; % Resetting the Engine Thrust Angle for the
Calculation

    li=.75*wingchord-lg;
    hi=hg;
    lf=.25*wingchord-lg;
    hf=hg;
    ipitch=mass*rgyr^2;

    ang=linspace(0,2*pi,101);
    cang=cos(ang);
    sang=sin(ang);
    cgcircx=cgrad*cang;
    cgcircy=cgrad*sang;

    timeout=[time];
    xout=[x];
    yout=[y];
    psiout=[psi];
    xdotout=[xdot];
    ydotout=[ydot];
    psidotout=[psidot];
    gliftout=[0];
    gdragout=[0];
    phiout=[0];
    pmomentout=[0];

    %fid = fopen('hvcg.txt', 'wt');

    % Time Loop Start
======================================================================
    for itime=1:ntime

        cpsi=cos(psi);
        spsi=sin(psi);
        ut=xdot*cpsi+ydot*spsi-hi*psidot;
        up=xdot*spsi-ydot*cpsi+li*psidot;
        vtot=sqrt(ut^2+up^2);
        phi=atan2(up,ut);
        alf=wingpitch+phi;
        %>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        alfmod=abs(alf);
        clarg=dclda*alfmod;
        if clarg>clmax
            clarg=clmax;
        end
```

```
        cl=clarg*sign(alf);
        %>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        lift=.5*airden*vtot^2*wingarea*cl;
        gliftout=[gliftout lift/mass];
        drag=.5*airden*vtot^2*wingarea*cd0;
        gdragout=[gdragout drag/mass];
        cm=-psidot*wingchord*pi/(8*vtot);
        pmoment1=.5*airden*vtot^2*wingarea*wingchord*cm;
        pmoment=lift*(-lf*cos(phi)-hf*sin(phi))+drag*(-
lf*sin(phi)+hf*cos(phi));
        if ifmtdamp~=0
            pmoment=pmoment+pmoment1;
        end
        pmomentout=[pmomentout pmoment/(auw*postloc*wingchord)];
        %>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        xddot=( engthrust*cos(theta1+psi)-lift*sin(psi-phi)-drag*sin(psi-
phi) )/mass;
        yddot=-grav + ( engthrust*sin(theta1+psi)+lift*cos(psi-phi)-
drag*sin(psi-phi) )/mass;
        psiddot=pmoment/ipitch;

        %>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        %fprintf(fid, '%10.6f %10.6f %10.6f %10.6f\n',
time,xddot,yddot,psiddot);
        %>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

        time=time+dtime;
        dx=xdot*dtime+.5*xddot*dtime^2;
        dy=ydot*dtime+.5*yddot*dtime^2;
        dpsi=psidot*dtime+.5*psiddot*dtime^2;
        dxdot=xddot*dtime;
        dydot=yddot*dtime;
        dpsidot=psiddot*dtime;
        x=x+dx;
        y=y+dy;
        psi=psi+dpsi;
        xdot=xdot+dxdot;
        ydot=ydot+dydot;
        psidot=psidot+dpsidot;

        timeout=[timeout time];
        xout=[xout x];
        yout=[yout y];
        psiout=[psiout psi];
        xdotout=[xdotout xdot];
        ydotout=[ydotout ydot];
        psidotout=[psidotout psidot];
        phiout=[phiout phi];

    end
    % Time Loop End =====================================================

    % Plot CG Displacement ==============================================
    XCG=xout;
    YCG=yout;
    plot(XCG,YCG,'y');
    grid on

    figure
```

```
    % Plot CG Velocity ==================================================
    XVEL=xdotout;
    YVEL=ydotout;
    plot(XVEL,YVEL,'r');
    grid on

    figure

    % Plot Pitch Angle in Revs
================================================

    plot(timeout,psiout/(2*pi),'c');
    grid on

    % Prepare for Plot Animation ==========================================

    % Wing & Post Coordinates
    cpsiout=cos(psiout);
    spsiout=sin(psiout);
    XLE=(lg*cpsiout-hg*spsiout)+XCG;
    YLE=(lg*spsiout+hg*cpsiout)+YCG;
    XTE=((lg-wingchord)*cpsiout-hg*spsiout)+XCG;
    YTE=((lg-wingchord)*spsiout+hg*cpsiout)+YCG;
    XPOST=((lg-lp)*cpsiout-hg*spsiout)+XCG;
    YPOST=((lg-lp)*spsiout+hg*cpsiout)+YCG;
    % Aero Force Vector Coordinates
    COFX=((lg-.25*wingchord)*cpsiout-hg*spsiout)+XCG;
    COFY=((lg-.25*wingchord)*spsiout+hg*cpsiout)+YCG;
    FX=COFX-gliftout.*sin(psiout-phiout)/fscale-gdragout.*cos(psiout-
phiout)/fscale;
    FY=COFY+gliftout.*cos(psiout-phiout)/fscale-gdragout.*sin(psiout-
phiout)/fscale;

    % Engine Thrust Force Vector Coordinates
    COETX=XCG;
    COETY=YCG;
    ETFX=COETX+engt.*cos(theta1+psiout);
    ETFY=COETY+engt.*sin(theta1+psiout);

    % Inflow Vector Coordinates
    COIX=XCG-hi*sin(psiout)-li*cos(psiout);
    COIY=YCG+hi*cos(psiout)-li*sin(psiout);
    PHIX=COIX-.5*cos(psiout-phiout);
    PHIY=COIY-.5*sin(psiout-phiout);

    % Plot Animation ======================================================

    axis equal

    figure
    for itime2=1:nplotout:ntime
        clf
        % Plot Post -------------------------------------------------------
        plot([XCG(itime2) XPOST(itime2)],[YCG(itime2) YPOST(itime2)],'r');
        hold on
        % Plot Wing -------------------------------------------------------
        plot([XLE(itime2) XTE(itime2)],[YLE(itime2) YTE(itime2)],'g');
```

```
        % Plot CG -----------------------------------------------------------
        fill(XCG(itime2)+cgcircx,YCG(itime2)+cgcircy,'b');
        % Plot Force Vector -----------------------------------------------
-
        plot([COFX(itime2) FX(itime2)],[COFY(itime2) FY(itime2)],'m');
        % Plot Force Vector Arrowhead -------------------------------------
-
        fill(FX(itime2)+cgcircx/5,FY(itime2)+cgcircy/5,'m');
        % Plot Pitching Moment Circle -------------------------------------
-
        if pmomentout(itime2)>0

fill(COFX(itime2)+pmomentout(itime2).*cgcircx/mscale,COFY(itime2)+pmomentou
t(itime2).*cgcircy/mscale,'y')
        else

fill(COFX(itime2)+pmomentout(itime2).*cgcircx/mscale,COFY(itime2)+pmomentou
t(itime2).*cgcircy/mscale,'m')
        end
        % Plot Engine Thrust Force Vector ---------------------------------
---------------
        plot([COETX(itime2) ETFX(itime2)],[COETY(itime2)
ETFY(itime2)],'y');
        % Plot Engine Thrust Force Vector Arrowhead -----------------------
---------------
        fill(ETFX(itime2)+cgcircx/5,ETFY(itime2)+cgcircy/5,'y');
        % Plot Gravitational Force Vector ---------------------------------
---------------
        plot([XCG(itime2) XCG(itime2)],[YCG(itime2) YCG(itime2)-1],'m');
        % Plot Gravitational Vector Arrowhead -----------------------------
---------
        fill(XCG(itime2)+cgcircx/5,YCG(itime2)-1+cgcircy/5,'m');
        % Plot Inflow Vector ----------------------------------------------
--
        plot([COIX(itime2) PHIX(itime2)],[COIY(itime2) PHIY(itime2)],'c');
        % Plot Inflow Vector Arrowhead ------------------------------------
--
        fill(PHIX(itime2)+cgcircx/5,PHIY(itime2)+cgcircy/5,'c')

        axis(2*[-1 1 -1 1]+XCG(itime2)*[1 1 0 0]+YCG(itime2)*[0 0 1 1]);
        axis square
        grid on
        title(['Time = ',num2str(timeout(itime2),'%10.2f')]);
        M(itime)=getframe(gcf);
    end

    %fclose(fid)
```