Dynamic Soaring: Assignment 9 [Differential Flatness]

Izza (AE18B006) &
RV Ramkumar (AE20D025)

Problem

The problem is to find the minimum wind shear β required for supporting the dynamic soaring where the wind velocity at altitude h is given by $V_w = \beta h$ and the solution to be determined using Differential Flatness approach. The following are the parameters for the 3 DOF aircraft model (considering elliptic wing) to be assumed:

Table 1: Aircraft Parameters

AR	15
C_{D_0}	0.01
ρ	1.225 kg/m^3
$C_{L_{max}}$	1.2
g	9.81 m/s^2
m	8 kg
m/S	14 kg/m^2
μ_{max}	$\pi/3$

The guess for the model is given as follows

$$x(t) = 20(\cos(2\pi t/T_f) - 1) \tag{1}$$

$$y(t) = -20\sqrt{2}\sin(2\pi t/T_f) \tag{2}$$

$$z(t) = 20(\cos(2\pi t/T_f) - 1) \tag{3}$$

and the guess for Time of flight T_f is 20 seconds and β is 0.4 second^{-1}.

Formulation of Optimization Problem using Differential Flatness

Differential flatness is an approach for solving trajectory optimization problems wherein the states and the input are functions of some flat outputs with few dimensions. The equations of motion of the aircraft model flying in the steady wind shear W_x along x direction alone is given as follows:

$$\dot{x} = V\cos\gamma\cos\chi + W_x \tag{4}$$

$$\dot{y} = V\cos\gamma\sin\chi \tag{5}$$

$$\dot{z} = -V\sin\gamma \tag{6}$$

$$m\dot{V} = T - D - mg\sin\gamma - m\dot{W}_x\cos\gamma\cos\chi \tag{7}$$

$$mV\dot{\gamma} = L\cos\mu - mg\cos\gamma + m\dot{W}_x\sin\gamma\cos\chi \tag{8}$$

$$mV\cos\gamma\dot{\chi} = L\sin\mu + m\dot{W}_x\sin\chi \tag{9}$$

where V is the air speed, γ is the flight path angle, χ is the heading angle, μ is the bank angle - these all are in the relative wind frame.
The wind shear is given as :

$$W_x = \beta(-z) = \beta h \tag{10}$$

and the derivative of the wind shear is given as

$$\dot{W}_x = -\beta(\dot{z}) \tag{11}$$

The lift and drag (with parabolic drag polar) of the aircraft are given by:

$$L = C_L\frac{1}{2}\rho V^2 S \tag{12}$$

$$D = C_D\frac{1}{2}\rho V^2 S = (C_{D_0} + KC_L^2)\frac{1}{2}\rho V^2 S \tag{13}$$

The state space representation of the aircraf model is differentially flat with the parameters $\mathbf{z} = (x, y, z)$ being the flat output vector. The remaining states (V, χ, γ) and control inputs (C_L, μ) can be obtained from the flat outputs and their derivatives i.e., $\mathbf{z}$, $d\mathbf{z}/dt$ and $d^2\mathbf{z}/dt^2$:

$$V = \sqrt{(\dot{x} - W_x)^2 + \dot{y}^2 + \dot{z}^2} \tag{14}$$

$$\chi = \tan^{-1}\left(\frac{\dot{y}}{\dot{x} - W_x}\right) \tag{15}$$

$$\gamma = \sin^{-1}\left(\frac{-\dot{z}}{V}\right) \tag{16}$$

$$\dot{\gamma} = \frac{1}{\sqrt{1 - \sin^2\gamma}}\left(\frac{-V\ddot{z} + \dot{z}\dot{V}}{V^2}\right) \tag{17}$$

$$\dot{\chi} = \frac{1}{1 + \tan^2\chi}\left(\frac{(\dot{x} - W_x)\ddot{y} - (\ddot{x} - \dot{W}_x)\dot{y}}{(\dot{x} - W_x)^2}\right) \tag{18}$$

$$\mu = \tan^{-1}\left(\frac{V\dot{\gamma} + g\cos\gamma - \dot{W}_x\cos\chi\sin\chi}{V\cos\gamma\dot{\chi} - \dot{W}_x\sin\chi}\right) \tag{19}$$

$$C_L = \frac{mV\cos\gamma\dot{\chi} - m\dot{W}_x\sin\chi}{\frac{1}{2}\rho V^2 S\sin\mu} \tag{20}$$

$$T = m\dot{V} + D + mg\sin\gamma + m\dot{W}_x\cos\chi\cos\gamma \tag{21}$$

Considering sinusoidal basis functions for the flat outputs, we get the following:

$$x(t) = a_{x0} + \sum_{n=1}^{M} a_{xn}\sin\left(\frac{2\pi nt}{t_f} + \eta_{xn}\right) \tag{22}$$

$$y(t) = a_{y0} + \sum_{n=1}^{M} a_{yn} \sin\left(\frac{2\pi nt}{t_f} + \eta_{yn}\right) \tag{23}$$

$$z(t) = a_{z0} + \sum_{n=1}^{M} a_{zn} \sin\left(\frac{2\pi nt}{t_f} + \eta_{zn}\right) \tag{24}$$

In this differential flatness approach, the parameters to be found are the decision vector given as

$$\begin{aligned}\mathbf{X} = [&a_{x0}, ..., a_{xn}, a_{y0}, ..., a_{yn}, a_{z0}, ..., a_{zn}, \eta_{x1}, ..., \eta_{xn}, \\ &\eta_{y1}, ..., \eta_{yn}, \eta_{z1}, ..., \eta zn, t_f, \beta]\end{aligned} \tag{25}$$

for the minimization of β subject to the upper and lower bounds of the decision vector

$$\mathbf{Lb} \leq X \leq \mathbf{Ub}$$

and also the inequality constraints $\mathbf{C(x,t)}$ on the state vector which is obtained from these flat states as an output. i.e.,

$$\mathbf{C(X,t)} = \begin{pmatrix} C_L(t) - C_{Lmax} \\ V(t) - V_{max} \\ \mu_{min} - \mu(t) \\ \mu(t) - \mu_{max} \\ \gamma_{min} - \gamma(t) \\ \gamma(t) - \gamma_{max} \\ T_{min} - T(t) \\ T(t) - T_{max} \\ z + h_{min} \\ -z - h_{max} \end{pmatrix} \tag{26}$$

All the Fourier coefficients are restricted between -100 and 100, the phase angles are restricted between 0 and 2π, the t_f is between 0 and 100 secs and β between 0.05 and 0.5.

Results

The above formulation is implemented in Matlab to determine the minimum wind shear required for sustaining the dynamic soaring and for the minimization of the non-linear problem, 'fmincon' function is used. The control history and the trajectory are also determined. The following are the results for the initial guesses mentioned in the problem description for the Fourier Harmonics of M=10 and no. of nodes of N=50 (N = 5M).

Table 2: Result Summary

Time of flight	16.7 sec
Minimum Wind Shear	0.497 per sec
Maximum Height	49.0 m

Plots

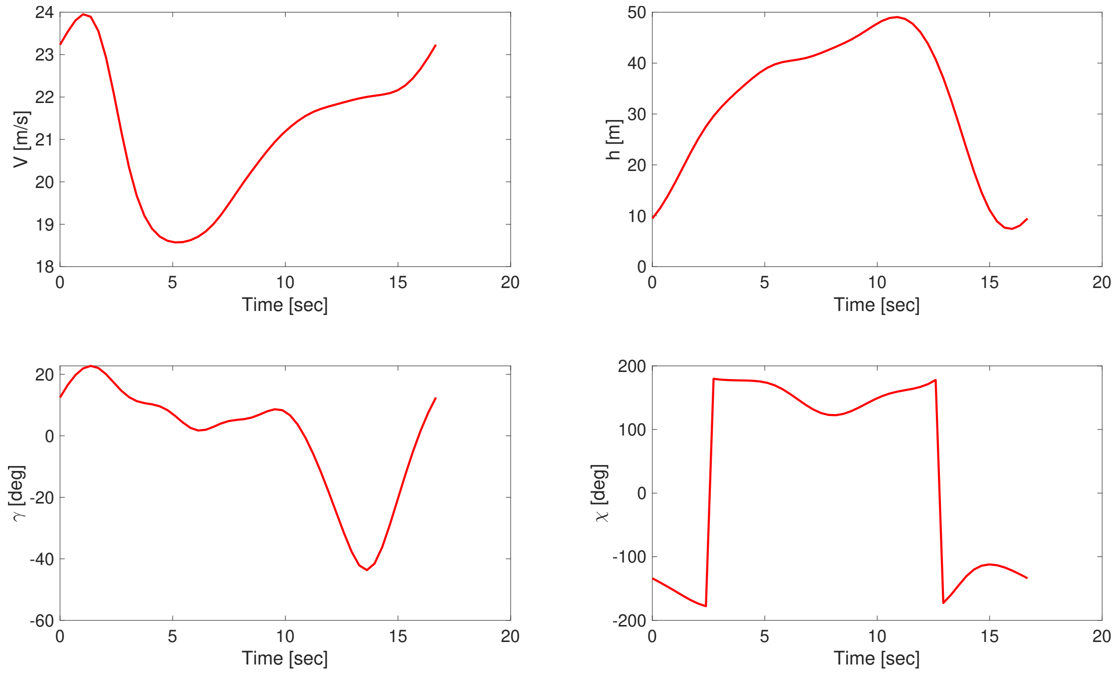

Figure 1: Time Evolution of States

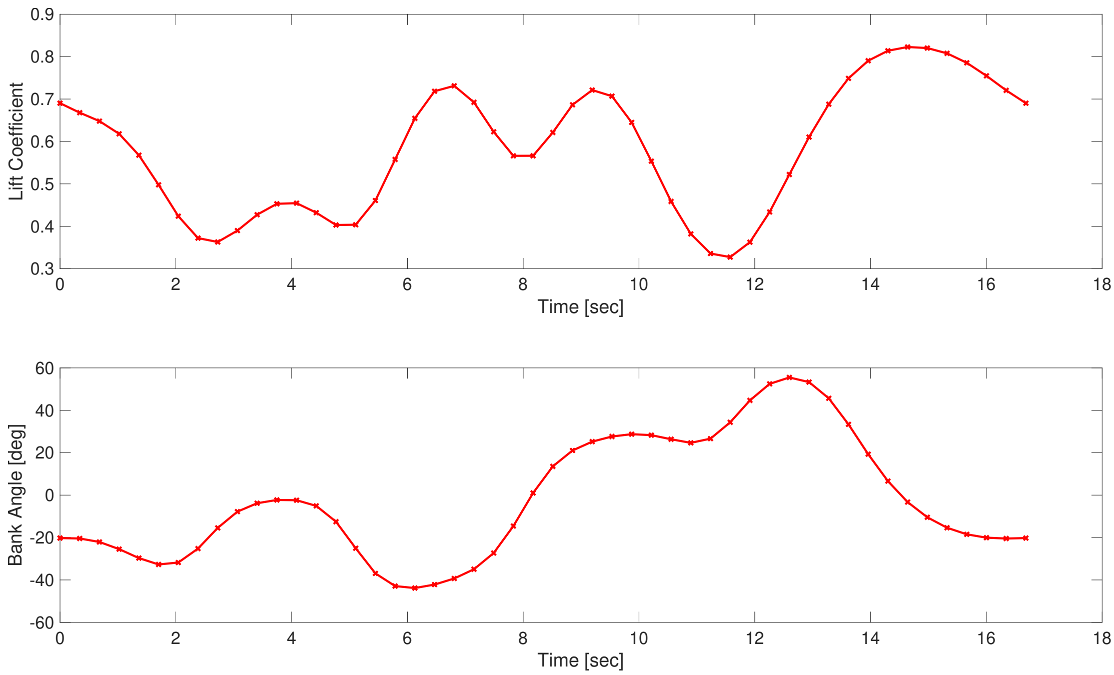

Figure 2: Time history of Controls

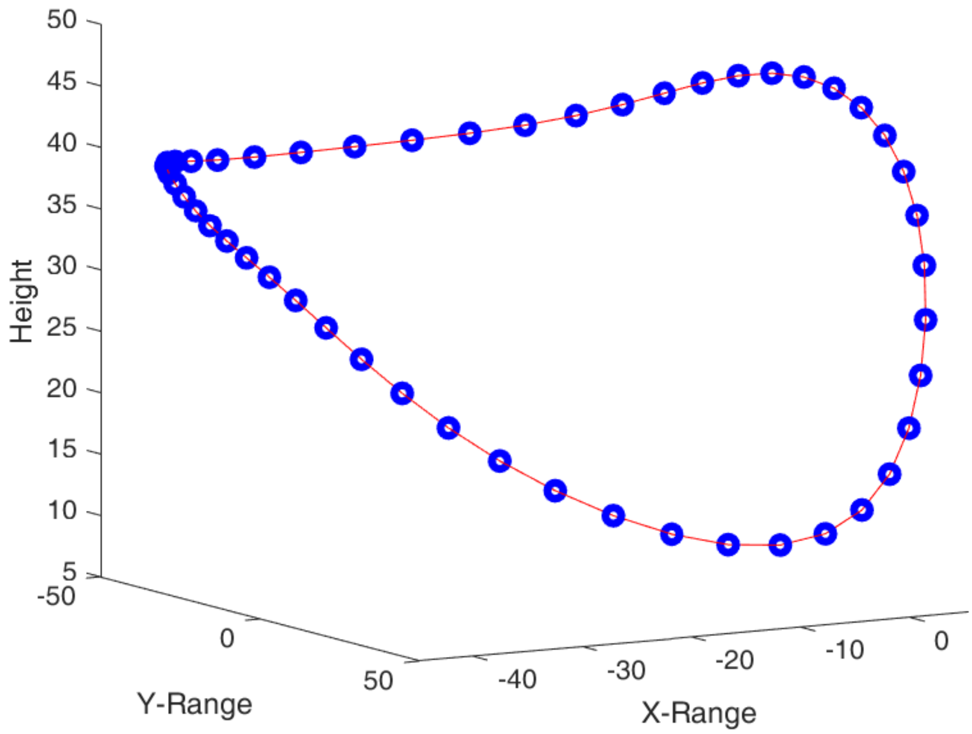

Figure 3: Trajectory

Code Listing

main.m

```
clear all; close all; clc;

global g rho m S CD0 K tau N M

g = 9.81;
rho = 1.225;
m   = 8;
mbyS = 14;
AR   = 15;
S    = m/mbyS;
K    = 1/(pi*AR);
CD0  = 0.01;
CLmax = 1.2;

% M = 6; % No. of Fourier Series Terms  % No of harmonics
% N = 30; % No. of nodal points   % No. of nodes;

M = 10; % No. of Fourier Series Terms  % No of harmonics
N = 50; % No. of nodal points   % No. of nodes;

tau = 2*pi/N*[1:N]';
```

```
%----------------------------------------------------------------------
% Defining the bounds for all the decision variables to be obtained
ax0min = -100;      ax0max = 100;
axmin = -100;       axmax = 100;
ay0min = -100;      ay0max = 100;
aymin = -100;       aymax = 100;
az0min = -100;      az0max = 100;
azmin = -100;       azmax = 100;

etaxmin = -1e-5;
etaxmax = 2*pi;
etaymin = -1e-5;
etaymax = 2*pi;
etazmin = -1e-5;
etazmax = 2*pi;

tfmin = 0;          tfmax = 100;
betamin = 0.05;     betamax = 0.5;

Lb = [ax0min*ones(1,1) axmin*ones(1,M) ay0min*ones(1,1) aymin*ones(1,M) az0min*ones(1,1)
azmin*ones(1,M)];
Lb = [Lb etaxmin*ones(1,M) etaymin*ones(1,M) etazmin*ones(1,M) tfmin betamin];

Ub = [ax0max*ones(1,1) axmax*ones(1,M) ay0max*ones(1,1) aymax*ones(1,M) az0max*ones(1,1)
azmax*ones(1,M)];
Ub = [Ub etaxmax*ones(1,M) etaymax*ones(1,M) etazmax*ones(1,M) tfmax betamax];

Lb = Lb';
Ub = Ub';

% Guess values at all the time instants

ax0g = [-20]; axg = [20 zeros(1,M-1)];            % x = -20 + 20*sin(2*pi*t/tfg + pi/2)
ay0g = [0];   ayg = [-20*sqrt(2) zeros(1,M-1)];     % y = -20*sqrt(2)*sin(2*pi*t/tfg)
az0g = [-20]; azg = [20 zeros(1,M-1)];            % z = -20 + 20*sin(2*pi*t/tfg + 3*pi/2)

etaxg = [pi/2 zeros(1,M-1)];          % phase values
etayg = [zeros(1,M)];
etazg = [pi/2 zeros(1,M-1)];
tfg = 20.0;
betag = 0.4;

tvec = linspace(0,tfg,N);

totalvars = 1 + M + M + 1 + M + M + 1 + M + M + 1 + 1;
X0 = [ax0g axg ay0g ayg az0g azg etaxg etayg etazg tfg betag];
X0 = X0';

A=[]; b=[]; Aeq=[]; beq=[];

opts = optimoptions('fmincon','Display','Iter','MaxFunEvals',80000,'MaxIter',8000);
[Z,costval,exitflag,output] = fmincon(@costfun,X0,A,b,Aeq,beq,Lb,Ub,@nlcon,opts);
```

```
%--------------------------------------------------------------------------
% Reassignment
%
ax0 = Z(1);         ax = Z(2:M+1);
ay0 = Z(M+2);       ay = Z(M+3:2*M+2);
az0 = Z(2*M+3);     az=Z(2*M+4:3*M+3);
etax = Z(3*M+4:4*M+3);
etay = Z(4*M+4:5*M+3);
etaz = Z(5*M+4:6*M+3);
tf = Z(end-1);
beta = Z(end);

% To compute x, y and h

t = linspace(0,tf,N)';
for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*sin(omegan*t(k) + etax(j));
        temp2 = temp2 + ay(j)*sin(omegan*t(k) + etay(j));
        temp3 = temp3 + az(j)*sin(omegan*t(k) + etaz(j));
    end
    x(k) = ax0 + temp1;
    y(k) = ay0 + temp2;
    z(k) = az0 + temp3;
end

for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*omegan*cos(omegan*t(k) + etax(j));
        temp2 = temp2 + ay(j)*omegan*cos(omegan*t(k) + etay(j));
        temp3 = temp3 + az(j)*omegan*cos(omegan*t(k) + etaz(j));
    end
    xdot(k) = temp1;
    ydot(k) = temp2;
    zdot(k) = temp3;
end

for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*(omegan).^2*(-sin(omegan*t(k) + etax(j)));
        temp2 = temp2 + ay(j)*(omegan).^2*(-sin(omegan*t(k) + etay(j)));
        temp3 = temp3 + az(j)*(omegan).^2*(-sin(omegan*t(k) + etaz(j)));
    end
    xddot(k) = temp1;
    yddot(k) = temp2;
    zddot(k) = temp3;
end
```

```
%% Wind Shear and other state variables
%% wind field

Wx=beta*(-z);
Wxdot = -beta*zdot;

%% other states

V = sqrt((xdot-Wx).^2 + ydot.^2 + zdot.^2);
Vdot = (2*(xdot-Wx).*(xddot-Wxdot)+2*ydot.*yddot + 2*zdot.*zddot)./(2*V);

gama = asin(-zdot./V);
gamadot = ((-V.*zddot + zdot.*Vdot)./(sqrt(1-(zdot./V).^2)))./(V.^2);

chi = atan2(ydot,(xdot-Wx));
chidot = (1./(1+tan(chi).^2)).*((xdot-Wx).*yddot - (xddot-Wxdot).*ydot)./((xdot-Wx).^2);

mu1 = atan2((V.*cos(gama).*chidot - Wxdot.*sin(chi)),(V.*gamadot + g*cos(gama)-
Wxdot.*cos(chi).*sin(gama)));
CL  =  (m*V.*cos(gama).*chidot - m*Wxdot.*sin(chi))./(0.5*rho*V.^2*S.*sin(mu1));
CD = CD0+K*CL.^2;

D = 0.5*rho*V.^2*S.*CD;
Th = m*Vdot + D + m*g*sin(gama) + m*Wxdot.*cos(chi).*cos(gama);

%%
disp([' ']);
disp([' ']);
disp(['******************************']);
disp(['*****Results******************']);

disp([' Min Wind shear = ', num2str(beta), '  sec^{-1}']);
disp([' Time of Flight = ', num2str(tf), '  sec']);
disp([' Max Height     = ', num2str(-min(z)),'  m']);
disp(['******************************']);
disp([' ']);
disp([' ']);

figure(1);
subplot (2,2,1); plot (t,flip(V),'r-','LineWidth',2.0);
                 xlabel ('Time [sec]'); ylabel ('V [m/s]'); set(gca,'Fontsize',16.0);
subplot (2,2,2); plot (t,-z,'r-','LineWidth',2.0);
 xlabel ('Time [sec]'); ylabel ('h [m]','Interpreter','Tex');
                 set(gca,'Fontsize',16.0);
subplot (2,2,3); plot (t,gama*180/pi,'r-','LineWidth',2.0);
                 xlabel ('Time [sec]'); ylabel ('\gamma [deg]','Interpreter','Tex');
                 set(gca,'Fontsize',16.0);
subplot (2,2,4); plot (t,chi*180/pi,'r-','LineWidth',2.0);
                 xlabel ('Time [sec]'); ylabel ('\chi [deg]','Interpreter','Tex');
                 set(gca,'Fontsize',16.0);

figure(2);
```

```
subplot(2,1,1);
plot (t,CL,'r-x','LineWidth',2.0); xlabel ('Time [sec]'); ylabel ('Lift Coefficient');
set(gca,'Fontsize',16.0);
subplot (2,1,2);
plot (t,mu1*180/pi,'r-x','LineWidth',2.0); xlabel ('Time [sec]'); ylabel ('Bank Angle [deg]');
set(gca,'Fontsize',16.0);


figure(3);
plot3 (x,y,-z,'r-o'); hold on;
xlabel ('X-Range'); ylabel ('Y-Range'); zlabel ('Height');
for j=1:N
    plot3 (x(j),y(j),-z(j),'b-o','LineWidth',3.0);
    pause(0.05); view(30,-10)
end
```

nlcon.m

```
function [c,ceq] = nlcon(Z)

global g rho m S CD0 K N M

% Maximum values of the parameters

CLmax   = 1.2;
Vmax    = 100;
mumin   = -pi/3;        mumax   = pi/3;
Thmin   = -0.001;       Thmax   = 0.001;
gamamin = -pi/4;        gamamax = pi/4;

ax0 = Z(1);         ax = Z(2:M+1);
ay0 = Z(M+2);       ay = Z(M+3:2*M+2);
az0 = Z(2*M+3);     az=Z(2*M+4:3*M+3);
etax = Z(3*M+4:4*M+3);
etay = Z(4*M+4:5*M+3);
etaz = Z(5*M+4:6*M+3);
tf = Z(end-1);
beta = Z(end);

% To compute x, y and h

t = linspace(0,tf,N)';
for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*sin(omegan*t(k) + etax(j));
        temp2 = temp2 + ay(j)*sin(omegan*t(k) + etay(j));
        temp3 = temp3 + az(j)*sin(omegan*t(k) + etaz(j));
    end
    x(k) = ax0 + temp1;
    y(k) = ay0 + temp2;
```

```
    z(k) = az0 + temp3;
end

for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*omegan*cos(omegan*t(k) + etax(j));
        temp2 = temp2 + ay(j)*omegan*cos(omegan*t(k) + etay(j));
        temp3 = temp3 + az(j)*omegan*cos(omegan*t(k) + etaz(j));
    end
    xdot(k) = temp1;
    ydot(k) = temp2;
    zdot(k) = temp3;
end

for k=1:length(t)
    temp1 = 0.0; temp2 = 0.0; temp3 = 0.0;
    for j=1:M
        omegan = (2*pi/tf)*j;
        temp1 = temp1 + ax(j)*(omegan).^2*(-sin(omegan*t(k) + etax(j)));
        temp2 = temp2 + ay(j)*(omegan).^2*(-sin(omegan*t(k) + etay(j)));
        temp3 = temp3 + az(j)*(omegan).^2*(-sin(omegan*t(k) + etaz(j)));
    end
    xddot(k) = temp1;
    yddot(k) = temp2;
    zddot(k) = temp3;
end

%% Wind Shear and other state variables
%% wind field

Wx=beta*(-z);
Wxdot = -beta*zdot;

%%

V = sqrt((xdot-Wx).^2 + ydot.^2 + zdot.^2);
Vdot = (2*(xdot-Wx).*(xddot-Wxdot)+2*ydot.*yddot + 2*zdot.*zddot)./(2*V);

gama = asin(-zdot./V);
gamadot = ((-V.*zddot + zdot.*Vdot)./(sqrt(1-(zdot./V).^2)))./(V.^2);

chi = atan2(ydot,(xdot-Wx));
chidot = (1./(1+tan(chi).^2)).*((xdot-Wx).*yddot - (xddot-Wxdot).*ydot)./((xdot-Wx).^2);

mu1 = atan2((V.*cos(gama).*chidot - Wxdot.*sin(chi)),(V.*gamadot + g*cos(gama)-
Wxdot.*cos(chi).*sin(gama)));
CL  =  (m*V.*cos(gama).*chidot - m*Wxdot.*sin(chi))./(0.5*rho*V.^2*S.*sin(mu1));
CD = CD0+K*CL.^2;

D = 0.5*rho*V.^2*S.*CD;
Th = m*Vdot + D + m*g*sin(gama) + m*Wxdot.*cos(chi).*cos(gama);
```

```
%%
ceq =[];

c(1:N,1)        = CL-CLmax;
c(N+1:2*N,1)    = V-Vmax;
c(2*N+1:3*N,1)  = mumin-mu1;
c(3*N+1:4*N,1)  = mu1-mumax;
c(4*N+1:5*N,1)  = gamamin-gama;
c(5*N+1:6*N,1)  = gama-gamamax;
c(6*N+1:7*N,1)  = Thmin-Th;
c(7*N+1:8*N,1)  = Th-Thmax;
c(8*N+1:9*N,1)  = z+2;
c(9*N+1:10*N,1) = -z-100;
```

costfun.m

```
function cost = costfun(Z)
cost = Z(end);
end
```