University of Colorado - Boulder

ASEN 3128 - Aircraft Dynamics

Section AM, Group 4

Lab 6: Fixed-Wing Aircraft Linear Equations

Author:
David Koester[a]

Author:
Tomaz Remec[b]

Author:
Zach Harris[c]

Author:
Connor O'Reilly[d]

Professor:
Eric Frew

[a] 108077881
[b] 107061117
[c] 108678256
[d] 107054811

11/23/2020

Ann and H.J. Smead
Aerospace Engineering Sciences
UNIVERSITY OF COLORADO **BOULDER**

I. Problem 1

A.

See code in Appendix A.

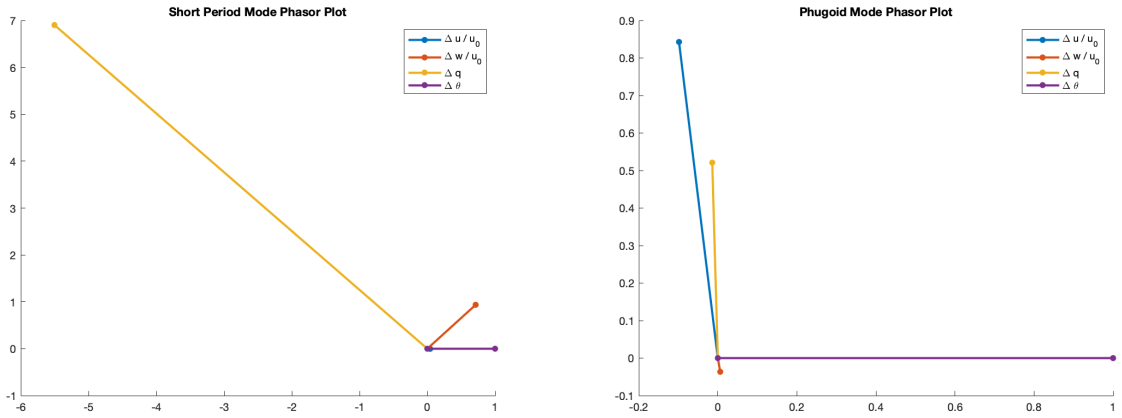
B.

Phugoid Mode	Short Period	Dutch Roll	Spiral
-0.01411 + 0.52111i	-5.50291 - 6.90310i	-0.59317 - 4.36035i	-17.94254
-0.01411 - 0.52111i	-5.50291 + 6.90310i	-0.59317 + 4.36035i	0.050396

The poles for the phugoid and short period modes were found using the Alon matrix. The poles can be separated into short period mode and phugoid mode poles by looking at the magnitude of the values. The short period mode has greater damping and a faster response so the real component of the pole is much larger than that of the phugoid mode. The dutch roll and spiral modes are the lateral modes so they were found using the Alat matrix. The dutch roll mode is characterized by poor damping but a fast response which means it still has an imaginary value while the spiral mode does not. In addition, the spiral mode has a smaller real value allowing for easy identification of the poles (Table 1).

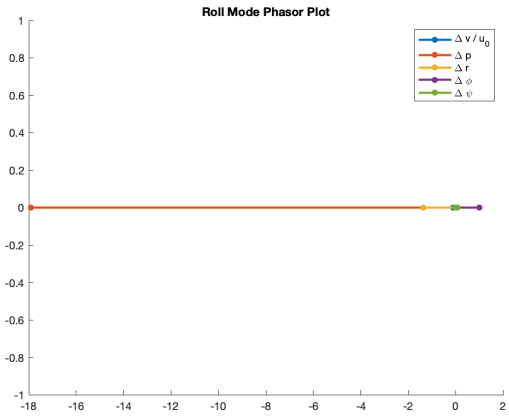
C.

The phasor plots for all five flight modes are shown below. The short period and phugoid modes correspond with the longitudinal flight dynamics of the UAS, and the roll, spiral, and dutch roll modes are related to the UAS' lateral flight dynamics.

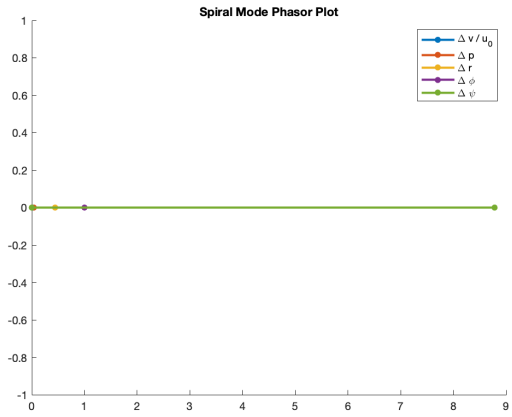


(a) Short Period Mode **(b) Phugoid Mode**

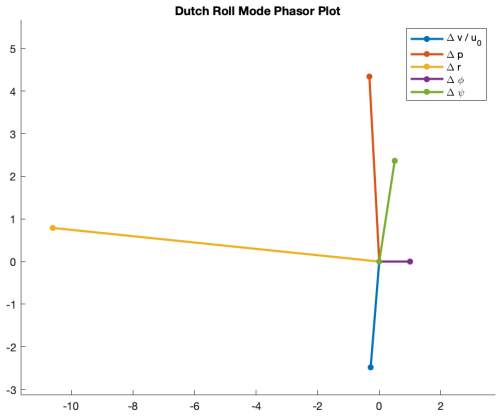
Fig. 1 Longitudinal Modes



(a) Roll Mode



(b) Spiral Mode



(c) Dutch Roll Mode

Fig. 2 Lateral Modes

II. Problem 2

A.

When the aircraft has a roll perturbation of 2°the aircraft only moves in the x and the z directions. Throughout the full 250 second simulation, the position changes linearly in the inertial x direction and oscillates in the inertial z direction (as seen in Appendix B). This is because the pitch angle perturbation creates a net change in the movement of the aircraft in which it gains altitude while in the phugoid mode. The change in velocities is easily seen in Appendix B figure 8. The linear velocity in in both the inertial x and z directions takes longer to lose its oscillatory behavior, however, the z-direction shows a much larger contrast with the nonlinear velocity even beginning with a smaller amplitude in it's oscillation.

B.

The short period response of an aircraft was simulated under an initial pitch perturbation of two degrees. The modal motion resulting from the simulation is in line with the short period response of an aircraft. Apparent in figure 11 the initial pitch rate is severely disturbed yet returns to a steady state rapidly. Furthermore, there is little to no deviation in the inertial z position of the aircraft over time. The aforementioned behavior is consistent with the short period response of a fixed wing aircraft.

C.

Running the simulation longer shows the nonlinear model starting in short period mode and then continuing a consistent oscillation in the z position while the linear model exhibits the same short period mode behavior in the beginning then continues linearly with a z position of around -2438.5 meters (figure 21). The pitch disturbance has a short period mode behavior with both the linear and nonlinear models initially reaching a pitch angle higher than the desired 2 degrees then returning to a pitch angle of around 1.8° for the duration of the simulation. The 3D path graph (figure 17) highlights the stark contrast between the movement of the linear and nonlinear models. The linear model has a more extreme initial reaction and then continues in an almost perfectly straight path in the x-z plane while the nonlinear model has a weak initial reaction and continues an almost constant oscillation for the remainder of the simulation.

D.

While simulating the Phugoid response of a fixed wing aircraft, it is observed that the linear response approximation diverges from the non-linear model as the initial pitch perturbation increases. Figures 23 and 24 illustrate that between an initial 5-10 degree pitch perturbation the linear model inaccurately describes the motion of the aircraft compared to smaller initial conditions.

III. Problem 3

A.

The dutch roll response for the aircraft with an initial pitch perturbation was simulated using both the *initial.m* built in MATLAB function to determine perturbation variables and also using the full nonlinear simulation. Both simulations were run for 10 seconds and the results are shown in figures 38 to 44. Referencing figure 42, the non linear simulation shows no change in the inertial z position and the inertial y position initially oscillates with a magnitude less than a tenth of a meter and appears to level out at zero meters after 7 seconds. The inertial x-position increases linearly throughout the simulation, combined with the inertial oscillation in the inertial y-direction this would produce a "tail-wagging" motion which is associated with the dutch roll mode. In addition, figures 44 and 41 show the side slip angle and the yaw angle oscillating with opposite sign and both eventually stabilizing around seven seconds. The linearized simulation shows oscillation in the inertial z-position between $2438.3[m]$ and $2438.6[m]$ throughout the 10 sec simulation. The inertial x-position increases linearly throughout the ten second interval similarly to the non-linear simulation, and the inertial y-position initially oscillates along with the non-linear inertial y-position but then around three seconds increases exponentially which is not a characteristic of dutch roll mode. Sideslip angle and yaw track closely to the non-linearized simulation but after an initial oscillation around 5 seconds the yaw angle appears to increase linearly and this is shown in the results array. furthermore the pitch angle does not remain constant as the non-linearized case, with the combination of the continuous increase in the yaw angle this could be the reason for both the exponential increase in inertial y-position and an oscillation in the inertial z-position. Although the inertial z-position oscillates and the inertial y-position increase exponentially towards the end, changes in both the y and z position are a magnitude smaller than changes in the x inertial position such that visually the aircraft appears to be in "tail-wagging" motion initially in the linearized simulation.

B.

The spiral response of the aircraft with a roll angle perturbation of about 2° has very similar linear and nonlinear case responses. As seen in figure 49 the two cases stay extremely close together especially within the first 20 seconds. The largest difference is in the inertial x position with the linear case ending the simulation about 100 meters further in the inertial x than the nonlinear case. The velocities fluctuate even less. The linear case stays at 21.99 m/s in the inertial x direction for the entirety of the simulation while the nonlinear case speeds up to about 22.15 m/s (figure 50). The y and z direction both stay within 0.01 m/s of each other. The euler angle plot (figure 48) shows both the roll and the yaw angles increasing with the linear and nonlinear models both changing at almost the exact same rates. The roll angle starts at 2° and increases to about 7° while the yaw angle increases to more than 60°. The pitch angle has the only real difference in body angles with the nonlinear angle decreasing to about 1.33° and the linear case staying completely stagnant at 1.48°.

C.

The spiral response simulation was run again for 100 seconds. Instead of the linear and nonlinear models being extremely similar, the simulation running longer reveals a stark contract between the models. The models are different because of the difference in controls. As seen in figures 55 to 57, the linear and nonlinear cases rarely change in the same directions. The position plot shows the linear case moving further in the x and y directions while the nonlinear case loses altitude while the linear stays stagnate. The nonlinear case also oscillates in the x direction especially after the 60 second mark. This could be due to the nonlinear model not having the correct controls to maintain unstable spiral flight as it begins to spiral and lose altitude. The second reason could be the linear model being unable to accurately depict such a complex movement because of its tendency towards linear and simple circular motion. The spiral incorporates all three directions at the same time but as seen in figure 57 the model never moves the aircraft in more than 2 directions at the same time.

D.

The aircraft's roll mode was simulated with an initial roll angle perturbation of 2°. Running both the full nonlinear simulation and the linearized simulation for one second and the resulting graphs are shown in the appendix. Within the first second of the simulation the simulated side slip angle, angle of attack, inertial x-position, roll angle, and the roll and pitch rate for each simulation are almost exactly the same for the duration of the 1 second simulation. The inertial y and z position, and the inertial velocity u^E for the linearized case do change linearly but the the values change by hundredths and thousandths of meters so the change is minuscule and hardly noticed visually. Initially there is a difference between the yaw angle, pitch rate , and inertial velocity components v^E and w^E for the non-linear and linearity models. Initially values for the linearized case oscillate and follow a different track than the non-linear model but by the time one second has passed the values appear to settle close to those computed with the non-linear model.

E.

The above simulation was then repeated with a time interval of 150 seconds. As can be seen in Figure 67, the angular velocities about the x and z axes increase quite dramatically as time goes on, this means that the aircraft is rolling at an increasing rate. This, of course, is an unstable behavior. However, when simulated using the linear mode, these rates remain relatively constant at a zero value, after the effects of the initial perturbations are overcome. Another interesting plot to observe the characteristics of the roll mode is the inertial position of the UAS as a function of time (Figure 70). Here it can be seen that the nonlinear model predicts a steady increase in the x position until the roll mode fully "kicks in" and then the aircraft is doing turns to the x and y positions oscillate sinusoidally. Because the aircraft is banking, and not applying any additional elevator deflection, as can be seen in Figure 68, the aircraft loses lift and descends at an increasing rate. The remaining plots of the Euler angles, wind angles, inertial velocities, and the 3D trajectory can be found in Figures 69,72,71, and 66 respectively in Appendix C.

IV. Team Participation Table

Table 1 Team Member Participation

Name	Plan	Model	Experiment	Results	Report	Code	ACK
David Koester	2	2	1	1	1	2	DK
Tomaz Remec	1	1	1	2	1	2	TR
Zach Harris	1	1	1	1	2	1	ZH
Connor O'Reilly	1	1	2	1	1	1	CO

References

[1] Frew, E., "ASEN 3128 Lab 6: Fixed-Wing Aircraft Linear Equations" Oct. 23, 2020.

[2] Etkin, B., Reid, L.D., "Dynamics of Flight: Stability and Control", 3rd ed., John Wiley & Sons, Inc., 1996.

Appendix

A. Code

Main Script

```
clear all;
close all;
clc;

recuv_tempest;

Va_trim = 22;
h_trim = 2438.5;

wind_inertial = [0;0;0];

trim_definition = [Va_trim; h_trim];

%% 1a

%%% Determine trim
[trim_variables, fval] = CalculateTrimVariables(trim_definition, aircraft_parameters);
[trim_state, trim_input]= TrimStateAndInput(trim_variables, trim_definition);


%%% Linear matrices
[Alon, Blon, Alat, Blat] = AircraftLinearModel(trim_definition, trim_variables, aircraft_parameters);

%% 1b Eigen Stuff

[evec_lat,eval_lat] = eig(Alat);

[evec_lon,eval_lon] = eig(Alon);


    %find natural freq
[wn_lon,zeta_lon,p_lon] = damp(eval_lon);

if wn_lon(1:2,1)>wn_lon(3:4,1)

    wn_shortP = wn_lon(1,1);
```

```
    wn_phugoid = wn_lon(3,1);
    zeta_shortP = zeta_lon(1,1);
    zeta_phugoid = zeta_lon(3,1);
else
    wn_phugoid = wn_lon(1,1);
    wn_shortP = wn_lon(3,1);
    zeta_shortP = zeta_lon(3,1);
    zeta_phugoid = zeta_lon(1,1);
end

%% 2
sys_lon = ss(Alon,Blon,eye(6,6),zeros(6,2));

%2a Phugoid sim
%mode_lon(sys_lon,Alon,'phugoid',deg2rad(2),trim_state,trim_input,wind_inertial,[0,250]);

%2b Short Period sim
%mode_lon(sys_lon,Alon,'shortP',deg2rad(2),trim_state,trim_input,wind_inertial,[0,2]);

%2c Short Period sim @ 25 sec
%mode_lon(sys_lon,Alon,'shortP',deg2rad(2),trim_state,trim_input,wind_inertial,[0,25]);

%2d Phugoid sim at pitch pert of 5 and 10 deg
    %Phugoid simulation with an initial pitch perturbation of 5 deg
%mode_lon(sys_lon,Alon,'phugoid',deg2rad(5),trim_state,trim_input,wind_inertial,[0,250]);
%mode_lon(sys_lon,Alon,'phugoid',deg2rad(10),trim_state,trim_input,wind_inertial,[0,250]);
%% 3
sys_lat = ss(Alat,Blat,eye(6,6),zeros(6,2));

%3a
%mode_lat(sys_lat,Alat,'dutch',deg2rad(2),trim_state,trim_input,wind_inertial,[0,10]);
%3b
%mode_lat(sys_lat,Alat,'spiral',deg2rad(2),trim_state,trim_input,wind_inertial,[0,25]);
%3c
%mode_lat(sys_lat,Alat,'spiral',deg2rad(2),trim_state,trim_input,wind_inertial,[0,100]);
%3d
%mode_lat(sys_lat,Alat,'roll',deg2rad(2),trim_state,trim_input,wind_inertial,[0,1]);
%3e
%mode_lat(sys_lat,Alat,'roll',deg2rad(2),trim_state,trim_input,wind_inertial,[0,150]);
```

Linear Model Function

```
function [Alon, Blon, Alat, Blat] = AircraftLinearModel(trim_definition, trim_variables, aircraft_parameters)
%
% STUDENT COMPLETE


u0 = trim_definition(1);
h0 = trim_definition(2);

alpha0 = trim_variables(1);
de0 = trim_variables(2);
dt0 = trim_variables(3);

theta0 = alpha0;

ap = aircraft_parameters;
rho = stdatmo(h0);



%%%%%%%%%%%%%%%%%%%%%%
%%% Longitudinal
%%%%%%%%%%%%%%%%%%%%%%

%%%% Trim values
CW0 = ap.W/((1/2)*rho*u0^2*ap.S);
```

```
CL0 = CW0*cos(theta0);
CD0 = ap.CDmin + ap.K*(CL0-ap.CLmin)^2;
CT0 = CD0 + CW0*sin(theta0);


%%%% Nondimensional stabiulity derivatives in body coordinates

%%%%% This is provided since we never discussed propulsion - Prof. Frew
dTdu = dt0*ap.Cprop*ap.Sprop*(ap.kmotor-2*u0+dt0*(-2*ap.kmotor+2*u0));
CXu = dTdu/(.5*rho*u0*ap.S)-2*CT0;

CDu = 0;%CDM*Ma;    % Compressibility only.  Ignore aeroelasticity and other effects (dynamic pressure and thrust
CLu = 0;%CLM*Ma;
Cmu = 0;%CmM*Ma;

CZu = 0;

CZalpha =-CD0-ap.CLalpha;
CXalpha = CL0*(1-2*ap.K*ap.CLalpha);

CZalphadot = -ap.CLalphadot;

CZq = -ap.CLq;


% Longitudinal dimensional stability derivatives (from Etkin and Reid)
Xu = rho*u0*ap.S*CW0*sin(theta0)+.5*rho*u0*ap.S*CXu;
Zu = -rho*u0*ap.S*CW0*cos(theta0)+.5*rho*u0*ap.S*CZu;
Mu = .5*rho*u0*ap.S*ap.c*Cmu;


Xw = [.5*rho*u0*ap.S*CXalpha];
Zw = [.5*rho*u0*ap.S*CZalpha];
Mw = [.5*rho*u0*ap.S*ap.c*ap.Cmalpha];

Xq = 0;
Zq = [.25*rho*u0*ap.c*ap.S*CZq];
Mq = [.25*rho*u0*ap.c^2*ap.S*ap.Cmq];

Xwdot = 0;
Zwdot = [.25*rho*ap.c*ap.S*CZalphadot];
Mwdot = [.25*rho*ap.c^2*ap.S*ap.Cmalphadot];


% Matrices
    %Define ap.m = m and ap.g = g for simplicity
m = ap.m;

g = ap.g;

Iy = ap.Iy;

Alon = [Xu/m,Xw/m,0,-g*cos(theta0),0,0;...
    Zu/(m-Zwdot),Zw/(m-Zwdot),(Zq+m*u0)/(m-Zwdot),-m*g*sin(theta0)/(m-Zwdot),0,0;...
    (1/Iy)*(Mu+Mwdot*Zu/(m-Zwdot)),(1/Iy)*(Mw+Mwdot*Zw/(m-Zwdot)),(1/Iy)*(Mq+Mwdot*(Zq+m*u0)/(m-Zwdot)),(1/Iy)*(-
    0,0,1,0,0,0;...
    1,0,0,0,0,0;...
    0,1,0,-u0,0,0];

Blon = zeros(6,2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Lateral
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lateral-directional dimensional stability derivatives
Yv = [.5*rho*u0*ap.S*ap.CYbeta];
```

```
Yp = [.25*rho*u0*ap.b*ap.S*ap.CYp];
Yr = [.25*rho*u0*ap.b*ap.S*ap.CYr];

Lv = [.5*rho*u0*ap.b*ap.S*ap.Clbeta];
Lp = [.25*rho*u0*ap.b^2*ap.S*ap.Clp];
Lr = [.25*rho*u0*ap.b^2*ap.S*ap.Clr];

Nv = [.5*rho*u0*ap.b*ap.S*ap.Cnbeta];
Np = [.25*rho*u0*ap.b^2*ap.S*ap.Cnp];
Nr = [.25*rho*u0*ap.b^2*ap.S*ap.Cnr];

G = ap.Ix*ap.Iz-ap.Ixz^2;

G3=ap.Iz/G;
G4=ap.Ixz/G;
G8=ap.Ix/G;


Alat = [Yv/m,Yp/m,(Yr/m-u0),g*cos(theta0),0,0;...
    G3*Lv+G4*Nv,G3*Lp+G4*Np,G3*Lr+G4*Nr,0,0,0;...
    G4*Lv+G8*Nv,G4*Lp+G8*Np,G4*Lr+G8*Nr,0,0,0;...
    0,1,tan(theta0),0,0,0;...
    0,0,sec(theta0),0,0,0;...
    1,0,0,0,u0*cos(theta0),0];

Blat = zeros(6,2);
end
```

Longitudinal Mode Function

```
function mode_lon(sys,Alon,mode,pert,trim_state,trim_input,wind_inertial,tspan)

%Find eigenvals/vecs and natural freq
[evec,eval] = eig(Alon,'vector');

[w_n,~,~] = damp(sys);

%Since damp return the natural frequencies in ascending order, we need to
%match the eigenvalues/eigenvectors to the corresponding natural frequency. To do this
%we use the function "sort".

[eval,align_indy] = sort(eval);

evec = evec(:,align_indy);

% eliminate 0 evals
zero_indy = find(vecnorm([real(eval),imag(eval)],2,2) == 0);

w_n(zero_indy) = [];

evec(:,zero_indy) = [];

eval(zero_indy) = [];

% Compute mode eigenvectors for perturbations
lon_var_indy = [7,9,11,5,1,3];
if strcmp(mode,'phugoid')
[~,p_indy] = min(w_n);
evec = evec/evec(4)*pert;
mode_evec = real(evec(:,p_indy));
else if strcmp(mode,'shortP')
[~,s_indy] = max(w_n);
evec = evec/evec(4)*pert;
mode_evec = real(evec(:,s_indy));
    end
end
% Initial perturbations
```

```
pert0 = mode_evec;
% Create 12 element vector
pert0_12 = zeros(12,1);
pert0_12(lon_var_indy) = pert0;

% Run Simulation for both linear and nonlinear
ap = AircraftParameters();
[Tnonl,Ynonl] = ode45(@(t,state) AircraftEOM(t,state,trim_input,wind_inertial,ap),tspan,trim_state+pert0_12);
[~,Tl,Yl] = initial(sys,pert0,tspan(2));

%Resize Linear output
Yl_12 = repmat(trim_state',length(Tl),1);
Yl_12(:,lon_var_indy) = Yl_12(:,lon_var_indy) + Yl;
Yl_12(:,1) = Yl_12(:,1)+ap.u0*Tl;

for i=1:length(Tnonl)
    UOUTnl(i,:) = trim_input';
end
for i = 1:length(Tnonl)
    wind_body = TransformFromInertialToBody(wind_inertial, Ynonl(i,4:6));
    air_rel_vel_body = Ynonl(i,7:9) - wind_body;
    wind_anglesnl(i,1:3) = WindAnglesFromVelocityBody(air_rel_vel_body);
end
for i=1:length(Tl)
    UOUTl(i,:) = trim_input';
end
for i = 1:length(Tl)
    wind_body = TransformFromInertialToBody(wind_inertial, Yl_12(i,4:6));
    air_rel_vel_body = Yl_12(i,7:9) - wind_body;
    wind_anglesl(i,1:3) = WindAnglesFromVelocityBody(air_rel_vel_body);
end

PlotAircraftSim(Tnonl,Ynonl,UOUTnl,wind_anglesnl,'b')
PlotAircraftSim(Tl,Yl_12,UOUTl,wind_anglesl,'r')
end
```

Lateral Mode Function

```
function mode_lat(sys,Alat,mode,pert,trim_state,trim_input,wind_inertial,tspan)


%Find eigenvals/vecs and natural freq
[evec,eval] = eig(Alat,'vector');

[w_n,~,~] = damp(sys);
%Rearrange
[eval,align_indy] = sort(eval);

evec = evec(:,align_indy);

% eliminate 0 evals
zero_indy = find(vecnorm([real(eval),imag(eval)],2,2) == 0);

w_n(zero_indy) = [];

evec(:,zero_indy) = [];

eval(zero_indy) = [];

% Create indexing vectors:
lat_var_indy = [8,10,12,4,6,2];

real_indy = find(imag(eval) == 0);

imag_indy = find(imag(eval) ~= 0);

    if strcmp(mode,'dutch')
```

```
        evec = evec(:,imag_indy(1))/evec(4,imag_indy(1))*pert;

        mode_evec = real(evec);

    elseif strcmp(mode,'spiral')

       evec = evec(:,1)/evec(4,1)*pert;

        mode_evec = real(evec);

    elseif strcmp(mode,'roll')

        evec = evec(:,4)/evec(4,4)*pert;

        mode_evec = real(evec);
    end
% Initial perturbations
pert0 = mode_evec;

% Create 12 element vector
pert0_12 = zeros(12,1);

pert0_12(lat_var_indy) = pert0;

% Run Simulation for both linear and nonlinear
ap = AircraftParameters();
[Tnonl,Ynonl] = ode45(@(t,state) AircraftEOM(t,state,trim_input,wind_inertial,ap),tspan,trim_state+pert0_12);

[~,Tl,Yl] = initial(sys,pert0,tspan(2));

% Resize Linear output
Yl_12 = repmat(trim_state',length(Tl),1);

Yl_12(:,lat_var_indy) = Yl_12(:,lat_var_indy) + Yl;

Yl_12(:,1) = Yl_12(:,1)+ap.u0*Tl;

for i=1:length(Tnonl)
    UOUTnl(i,:) = trim_input';
end

for i = 1:length(Tnonl)

    wind_body = TransformFromInertialToBody(wind_inertial, Ynonl(i,4:6));

    air_rel_vel_body = Ynonl(i,7:9) - wind_body;

    wind_anglesnl(i,1:3) = WindAnglesFromVelocityBody(air_rel_vel_body);

end

for i=1:length(Tl)

    UOUTl(i,:) = trim_input';

end

for i = 1:length(Tl)

    wind_body = TransformFromInertialToBody(wind_inertial, Yl_12(i,4:6));

    air_rel_vel_body = Yl_12(i,7:9) - wind_body;

    wind_anglesl(i,1:3) = WindAnglesFromVelocityBody(air_rel_vel_body);

end
```

```
PlotAircraftSim(Tnonl,Ynonl,UOUTnl,wind_anglesnl,'b')

PlotAircraftSim(Tl,Yl_12,UOUTl,wind_anglesl,'r')
end
```

Phasor Mode Plotting Function

```
function [] = plotModes(eval_lon,evec_lon,eval_lat,evec_lat)
% Tomaz Remec
% 107061117
% ASEN 3128
% Lab 6 Phasor plotting function
% %%%%%%%
% % OLD AND NO BUENO CODE
% % MAKES COOL PLOTS THO
% %%%%%%%
% %     % Phasor plots
% %     for i = 3:length(eval_lat)
% %         figure(1)
% %             hold on
% %             plot([0,real(eval_lat(i,i))],[0,imag(eval_lat(i,i))],'LineWidth',2)
% %             title('Lateral Mode Phasor Plot')
% %         figure(2)
% %             hold on
% %             plot([0,real(eval_lon(i,i))],[0,imag(eval_lon(i,i))])
% %             title('Longitudinal Mode Phasor Plot')
% %     end
%
%     % Longitudinal Mode Plots
%     % SP 3-4, Phugiod 5-6
%
%     % SP Plot
%     tVecSP = linspace(0,2,1000);
%     spVec = evec_lon(3:end,3);
%     spVec = spVec./spVec(4);
%     spVec(1:2,:) = spVec(1:2,:)/22;
%     spMat = spVec*exp(eval_lon(3,3)*tVecSP);
%     figure(1)
%         plot(tVecSP,real(spMat))
%         title('Short Period Mode Perturbations')
%         xlabel('Time [s]')
%         ylabel('Perturbation')
%         legend('\Delta u / u_0','\Delta w / u_0','\Delta q','\Delta \theta')
%
%     % Phugiod Plot
%     tVecPH = linspace(0,200,1000);
%     phVec = evec_lon(3:end,5);
%     phVec = phVec./phVec(4);
%     %phVec(1:2,:) = phVec(1:2,:)/22;
%     phMat = phVec*exp(eval_lon(5,5)*tVecPH);
%     figure(2)
%         plot(tVecPH,real(phMat))
%         title('Phugoid Mode Perturbations')
%         xlabel('Time [s]')
%         ylabel('Perturbation')
%         legend('\Delta u','\Delta w','\Delta q','\Delta \theta')
%
%     % Lateral Mode Plots
%     % Roll 3, Dutch roll 4-5, Spiral 6
%
%     % Roll Plot
%     tVecR = linspace(0,1,1000);
%     rVec = evec_lat(1:5,3);
%     rVec = rVec./rVec(4);
%     rVec(1:2) = rVec(1:2)/22;
%     %rVec = rVec./rVec(4);
%     rMat = rVec*exp(eval_lat(3,3)*tVecR);
```

```
%     figure(3)
%         plot(tVecR,real(rMat))
%         title('Roll Mode Perturbations')
%         xlabel('Time [s]')
%         ylabel('Perturbation')
%         legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')
%
%     % Spiral Plot
%     tVecS = linspace(0,100,1000);
%     sVec = evec_lat(1:5,6);
%     sVec = sVec./sVec(4);
%     sVec(1) = sVec(1)/22;
%     %sVec = sVec./sVec(4);
%     sMat = sVec*exp(eval_lat(6,6)*tVecS);
%     figure(4)
%         plot(tVecS,real(sMat))
%         title('Spiral Mode Perturbations')
%         xlabel('Time [s]')
%         ylabel('Perturbation')
%         legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')
%
%     % Dutch Roll Plot
%     tVecD = linspace(0,10,1000);
%     dVec = evec_lat(1:5,4);
%     dVec = dVec./dVec(4);
%     dVec(1) = dVec(1)/22;
%     %dVec = dVec./dVec(4);
%     dMat = dVec*exp(eval_lat(4,4)*tVecD);
%     figure(5)
%         plot(tVecD,real(dMat))
%         title('Dutch Roll Mode Perturbations')
%         xlabel('Time [s]')
%         ylabel('Perturbation')
%         legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')
%
%     % Phasor Plots
%     % SP
%     figure(6)
%         hold on
%         plot([0,real(evec_lon(1,3))],[0,imag(evec_lon(1,3))])
%         plot([0,real(evec_lon(2,3))],[0,imag(evec_lon(2,3))])
%         plot([0,real(evec_lon(3,3))],[0,imag(evec_lon(3,3))])
%         plot([0,real(evec_lon(4,3))],[0,imag(evec_lon(4,3))])
%         title('Short Period Phasor Plot')
%         legend('\Delta u / u_0','\Delta w / u_0','\Delta q','\Delta \theta')
%     %%%%%%%
%     %%%%%%%

    % Longitudinal Modes
    % SP 3-4, Phugiod 5-6
    % Short Period
    spVec = evec_lon(1:4,3);
    spVec = spVec./spVec(4);
    spVec(1:2) = spVec(1:2)/22;
    figure(1)
        hold on
        plot([0,real(spVec(1))],[0,imag(spVec(1))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(spVec(2))],[0,imag(spVec(2))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(spVec(3))],[0,imag(spVec(3))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(spVec(4))],[0,imag(spVec(4))],'.-','LineWidth',2,'MarkerSize',15)
        title('Short Period Mode Phasor Plot')
        legend('\Delta u / u_0','\Delta w / u_0','\Delta q','\Delta \theta')
    % Phugoid
    phVec = evec_lon(1:4,5);
    phVec = phVec./phVec(4);
    phVec(1:2) = phVec(1:2)/22;
    figure(2)
        hold on
```

```
        plot([0,real(phVec(1))],[0,imag(phVec(1))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(phVec(2))],[0,imag(phVec(2))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(phVec(3))],[0,imag(phVec(3))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(phVec(4))],[0,imag(phVec(4))],'.-','LineWidth',2,'MarkerSize',15)
        title('Phugoid Mode Phasor Plot')
        legend('\Delta u / u_0','\Delta w / u_0','\Delta q','\Delta \theta')

    % Lateral Modes
    % Roll 3, Dutch roll 4-5, Spiral 6
    % Roll
    rVec = evec_lat(1:5,3);
    rVec = rVec./rVec(4);
    rVec(1) = rVec(1)/22;
    figure(3)
        hold on
        plot([0,real(rVec(1))],[0,imag(rVec(1))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(rVec(2))],[0,imag(rVec(2))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(rVec(3))],[0,imag(rVec(3))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(rVec(4))],[0,imag(rVec(4))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(rVec(5))],[0,imag(rVec(5))],'.-','LineWidth',2,'MarkerSize',15)
        title('Roll Mode Phasor Plot')
        legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')
    % Spiral
    sVec = evec_lat(1:5,6);
    sVec = sVec./sVec(4);
    sVec(1) = sVec(1)/22;
    figure(4)
        hold on
        plot([0,real(sVec(1))],[0,imag(sVec(1))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(sVec(2))],[0,imag(sVec(2))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(sVec(3))],[0,imag(sVec(3))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(sVec(4))],[0,imag(sVec(4))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(sVec(5))],[0,imag(sVec(5))],'.-','LineWidth',2,'MarkerSize',15)
        title('Spiral Mode Phasor Plot')
        legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')
    % Dutch Roll
    dVec = evec_lat(1:5,4);
    dVec = dVec./dVec(4);
    dVec(1) = dVec(1)/22;
    figure(5)
        hold on
        plot([0,real(dVec(1))],[0,imag(dVec(1))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(dVec(2))],[0,imag(dVec(2))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(dVec(3))],[0,imag(dVec(3))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(dVec(4))],[0,imag(dVec(4))],'.-','LineWidth',2,'MarkerSize',15)
        plot([0,real(dVec(5))],[0,imag(dVec(5))],'.-','LineWidth',2,'MarkerSize',15)
        title('Dutch Roll Mode Phasor Plot')
        legend('\Delta v / u_0','\Delta p','\Delta r','\Delta \phi','\Delta \psi')

    %clc
end
```

Aircraft Simulation Plotting Function

```
function PlotAircraftSim(TOUT, aircraft_state, control_surfaces, WindAngles, col)


%%%%%%%%%%%%%%%%%%%%%%%%
figure(1);

subplot(311);
h1= plot(TOUT, aircraft_state(:,1),col);hold on;
title('Position v Time');
ylabel('X [m]')

subplot(312);
plot(TOUT, aircraft_state(:,2),col);hold on;
```

```
 ylabel('Y [m]')

subplot(313);
plot(TOUT, aircraft_state(:,3),col);hold on;
ylabel('Z [m]')
xlabel('time [sec]');

%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);

subplot(311);
plot(TOUT, (180/pi)*aircraft_state(:,4),col);hold on;
title('Euler Angles v Time');
ylabel('Roll [deg]')

subplot(312);
plot(TOUT, (180/pi)*aircraft_state(:,5),col);hold on;
 ylabel('Pitch [deg]')

subplot(313);
plot(TOUT, (180/pi)*aircraft_state(:,6),col);hold on;
ylabel('Yaw [deg]')
xlabel('time [sec]');

%%%%%%%%%%%%%%%%%%%%%%%%%
figure(3);

subplot(311);
plot(TOUT, aircraft_state(:,7),col);hold on;
title('Velocity v Time');
ylabel('uE [m/s]')

subplot(312);
plot(TOUT, aircraft_state(:,8),col);hold on;
 ylabel('vE [m/s]')

subplot(313);
plot(TOUT, aircraft_state(:,9),col);hold on;
ylabel('wE [m/s]')
xlabel('time [sec]');

%%%%%%%%%%%%%%%%%%%%%%%%%
figure(4);

subplot(311);
plot(TOUT, (180/pi)*aircraft_state(:,10),col);hold on;
title('Angular Velocity v Time');
ylabel('p [deg/s]')

subplot(312);
plot(TOUT, (180/pi)*aircraft_state(:,11),col);hold on;
 ylabel('q [deg/s]')

subplot(313);
plot(TOUT, (180/pi)*aircraft_state(:,12),col);hold on;
ylabel('r [deg/s]')
xlabel('time [sec]');

%%%%%%%%%%%%%%%%%%%%%%%%%
figure(5);
plot3(aircraft_state(:,1),aircraft_state(:,2),-aircraft_state(:,3),col);hold on;
xlabel('x')
ylabel('y')
zlabel('z')
title('3-D Position Plot')


%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
if (~isempty(control_surfaces))
    figure(6);

    subplot(411);
    plot(TOUT, control_surfaces(:,1),col);hold on;
    title('Control Surfaces v Time');
    ylabel('\delta_e [rad]')

    subplot(412);
    plot(TOUT, control_surfaces(:,2),col);hold on;
    ylabel('\delta_a [rad]')

    subplot(413);
    plot(TOUT, control_surfaces(:,3),col);hold on;
    ylabel('\delta_r [rad]')

    subplot(414);
    plot(TOUT, control_surfaces(:,4),col);hold on;
    ylabel('\delta_t [frac]')
    xlabel('time [sec]');
end

figure(7);

subplot(311);
plot(TOUT,WindAngles(:,1),col);hold on;
title('Wind Angles v Time');
ylabel('V [m/s]')
subplot(312);
plot(TOUT, WindAngles(:,3),col);hold on;
 ylabel('\alpha [m/s]')

subplot(313);
plot(TOUT, WindAngles(:,2));hold on;
ylabel('\beta [m/s]')
xlabel('time [sec]');

end
```

B. Problem 2 Graphs

Note: In all graphs the red line is the linear case and the blue line is the nonlinear case.

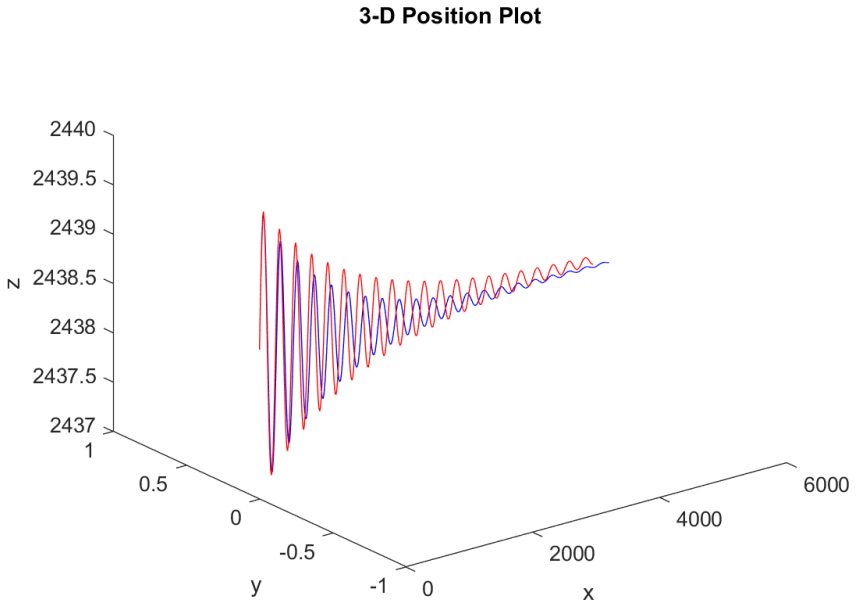


Fig. 3 2.A 3D Path

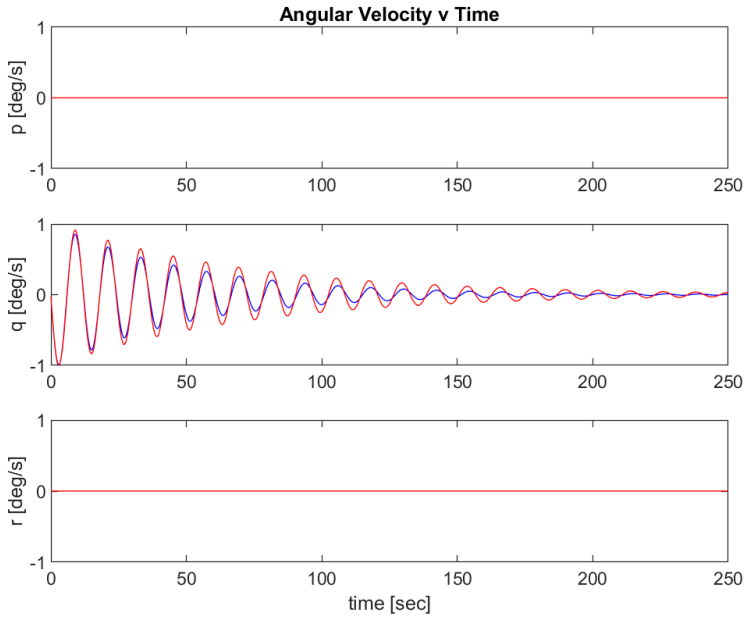


Fig. 4 2.A Angular Velocity

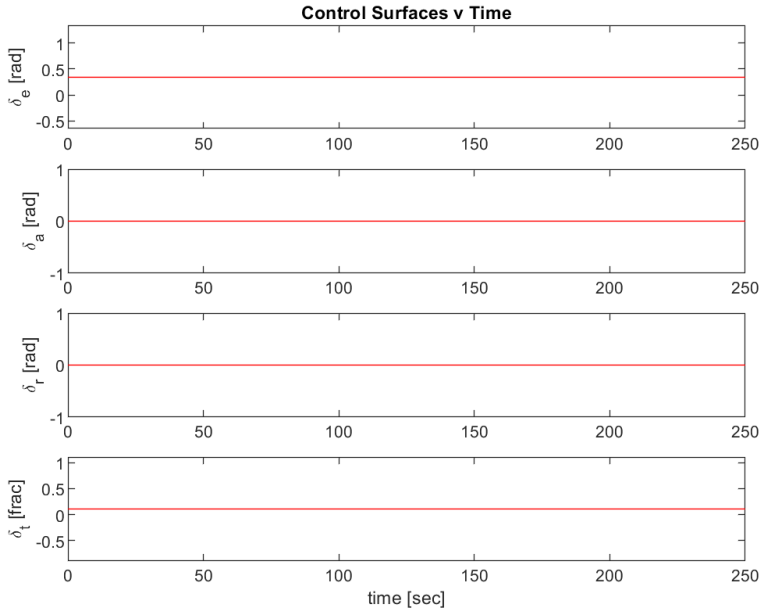


Fig. 5 2.A Control Surfaces

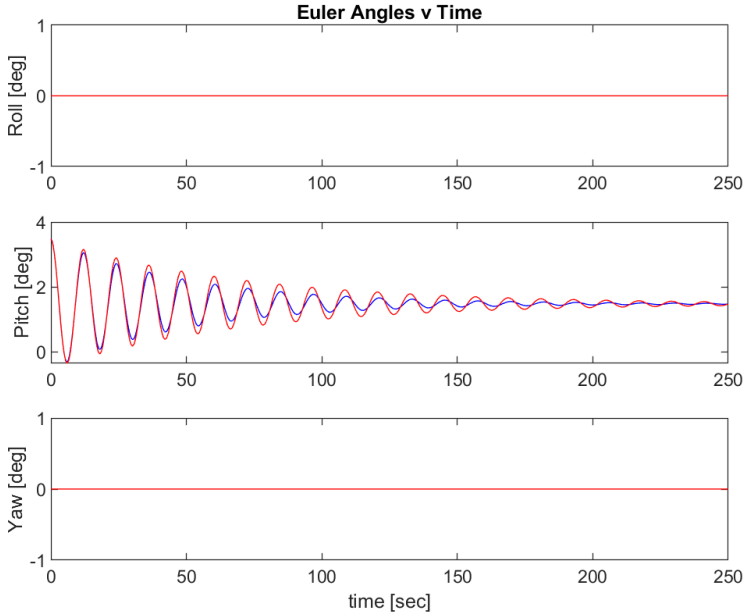


Fig. 6 2.A Euler Angles

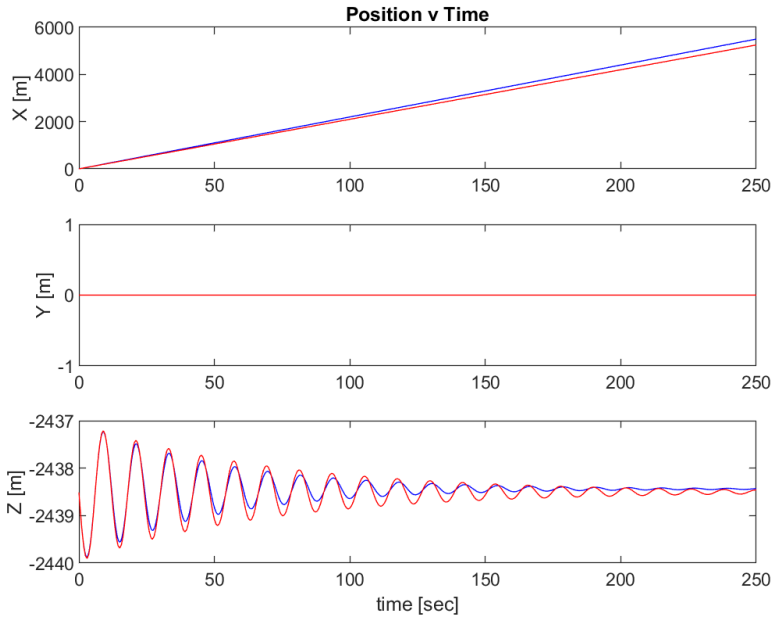


Fig. 7 2.A Position

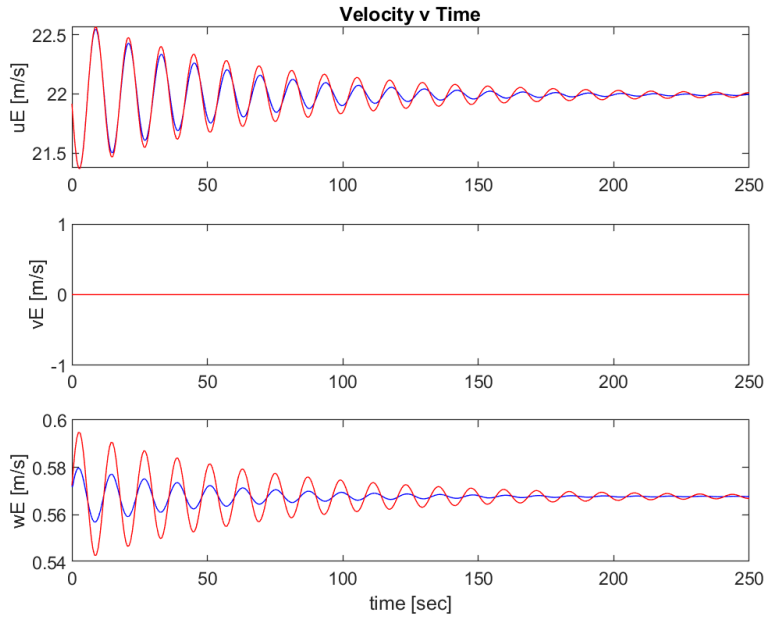


Fig. 8 2.A Velocity

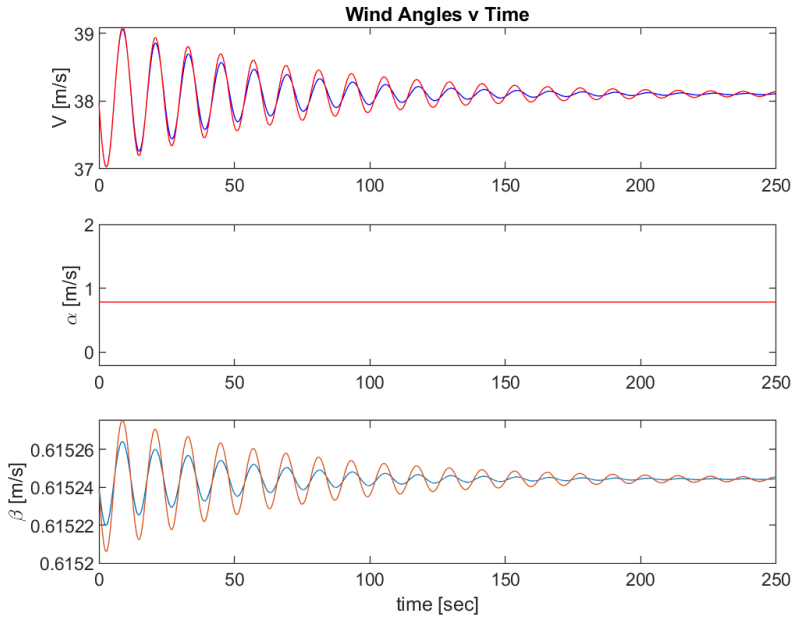


Fig. 9 2.A Wind Angles

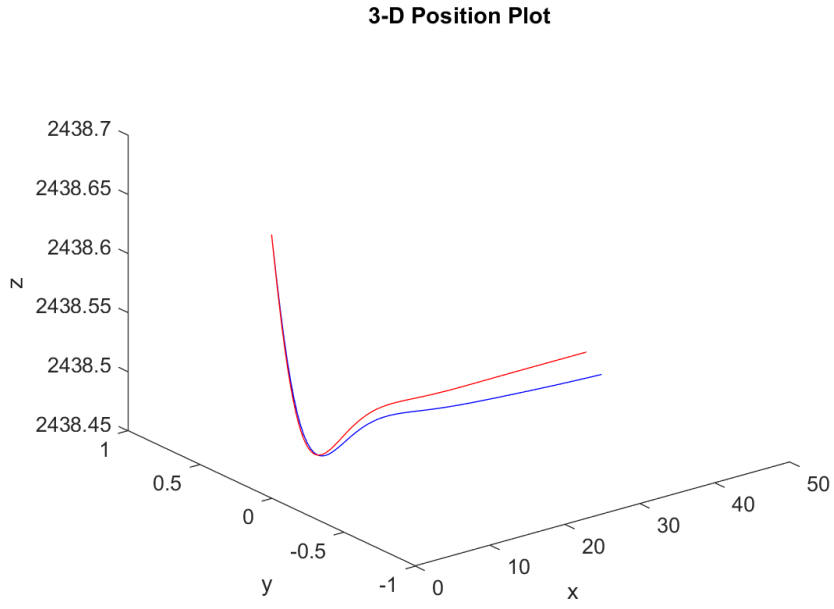


Fig. 10 2.B 3D Path

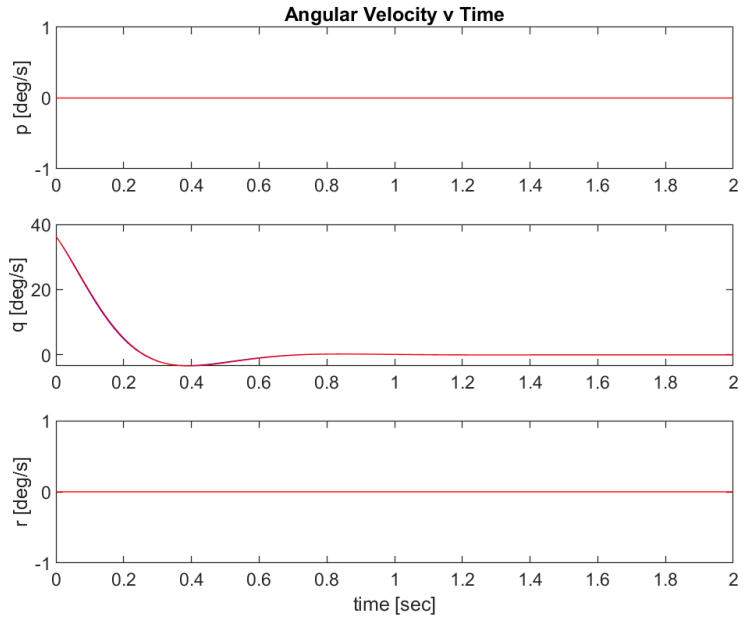


Fig. 11 2.B Angular Velocity

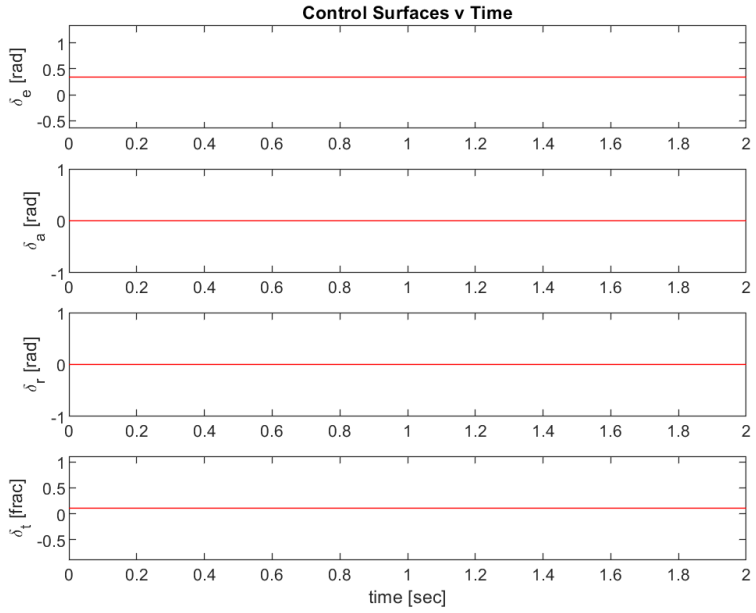


Fig. 12 2.B Control Surfaces

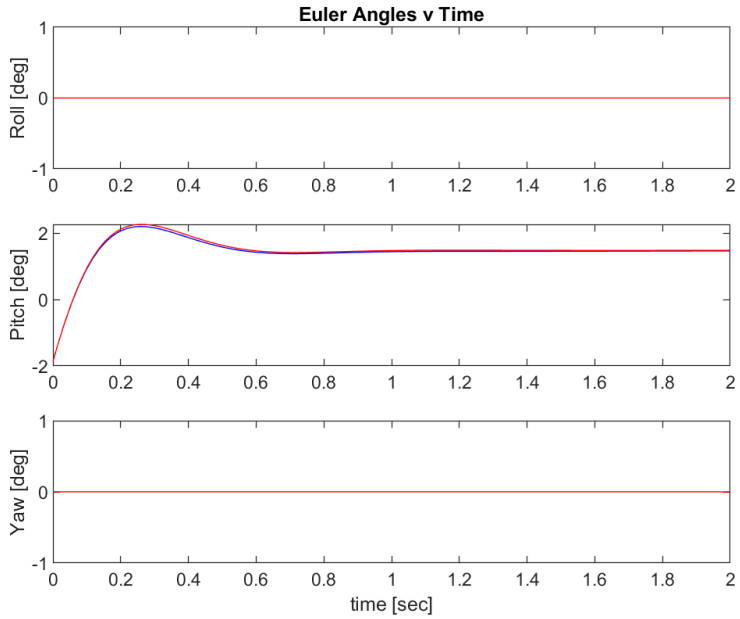


Fig. 13 2.B Euler Angles

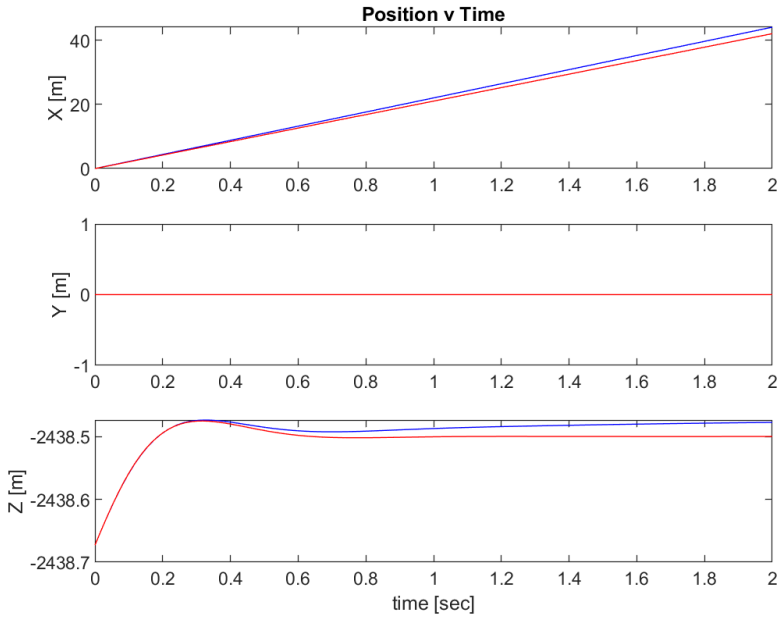


Fig. 14 2.B Position

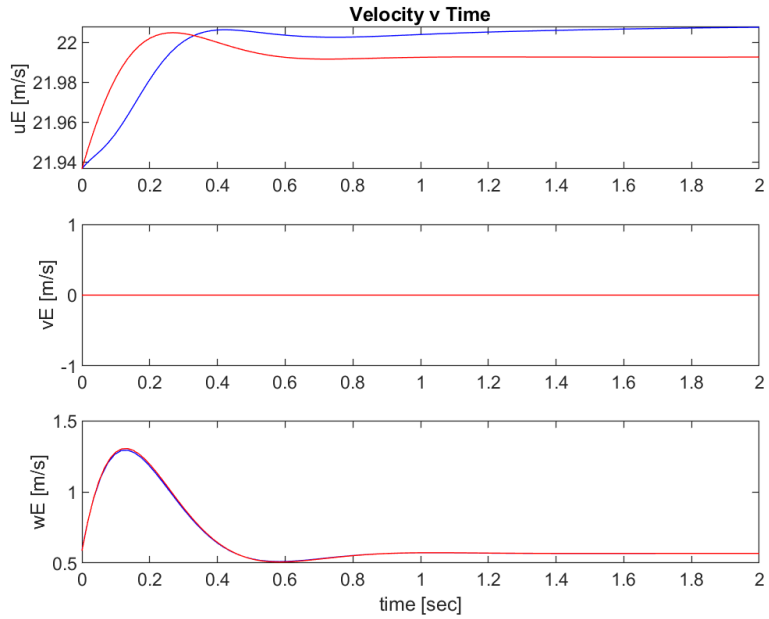


Fig. 15 2.B Velocity

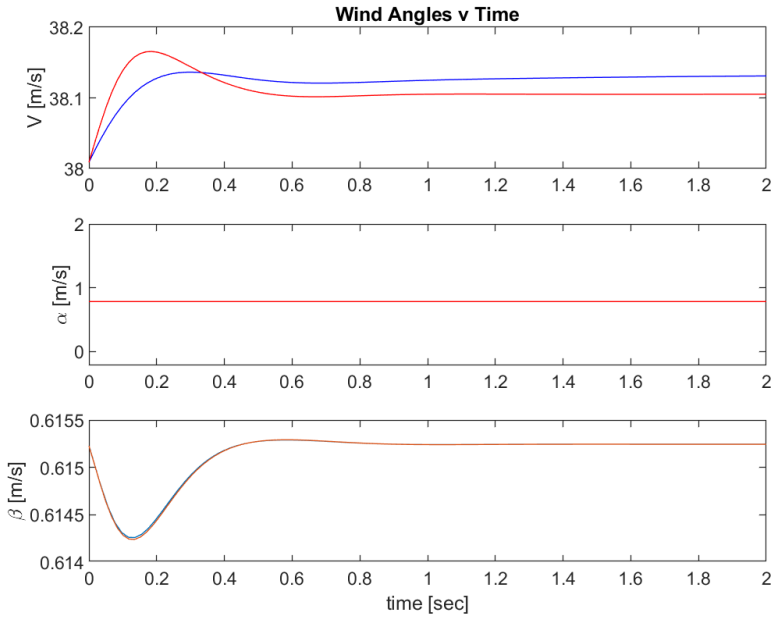


Fig. 16 2.B Wind Angles

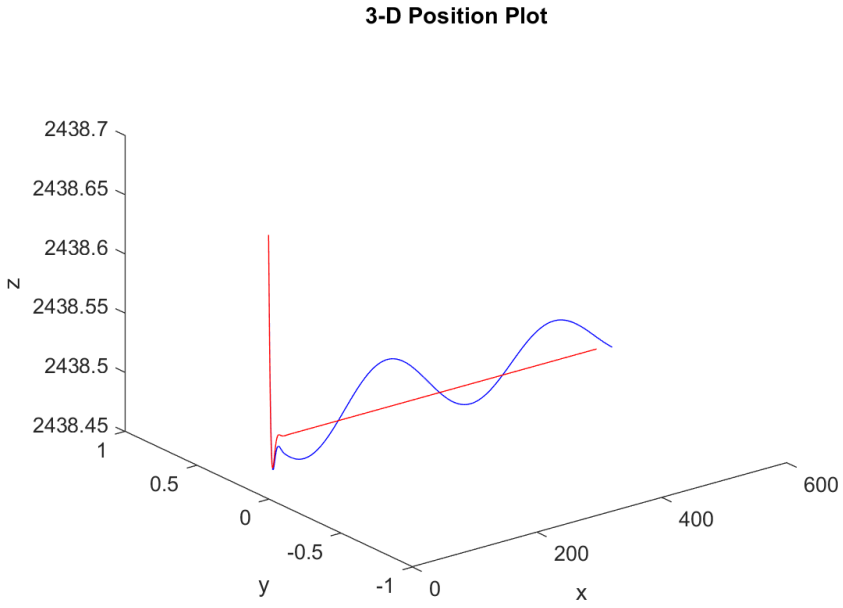


Fig. 17 2.C 3D Path

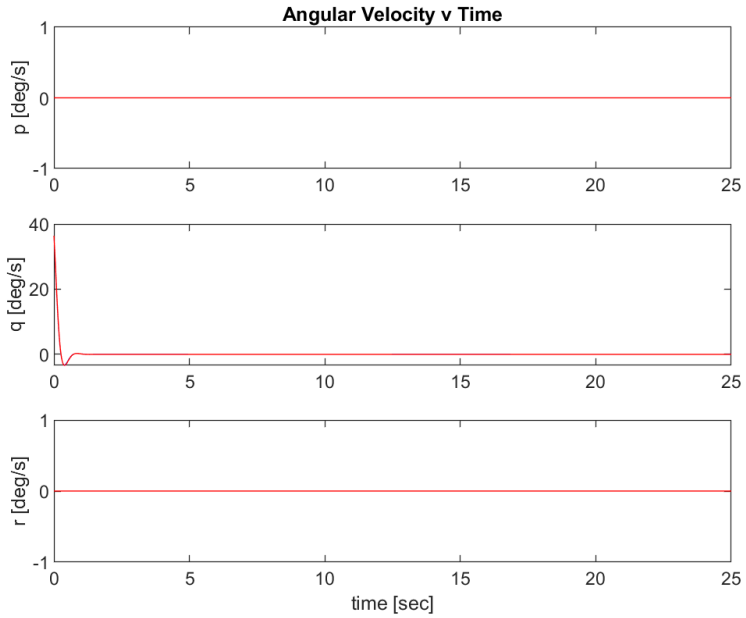


Fig. 18 2.C Angular Velocity

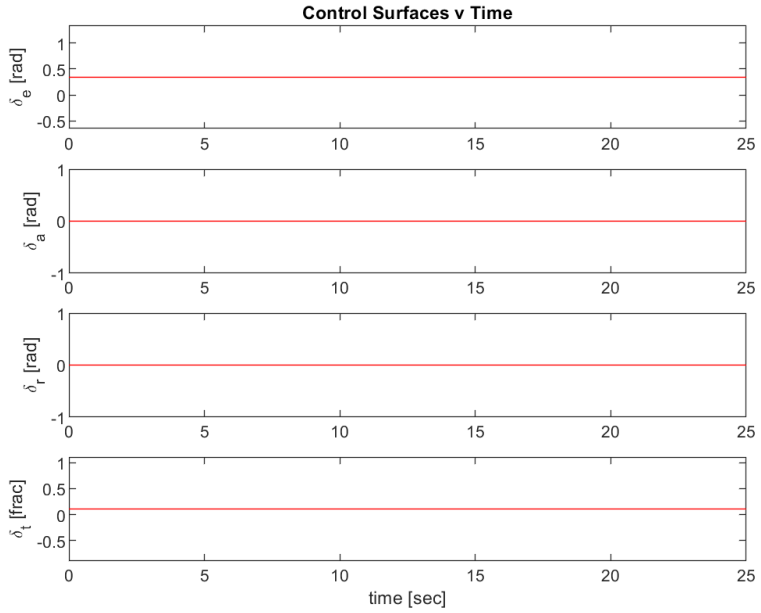


Fig. 19 2.C Control Surfaces

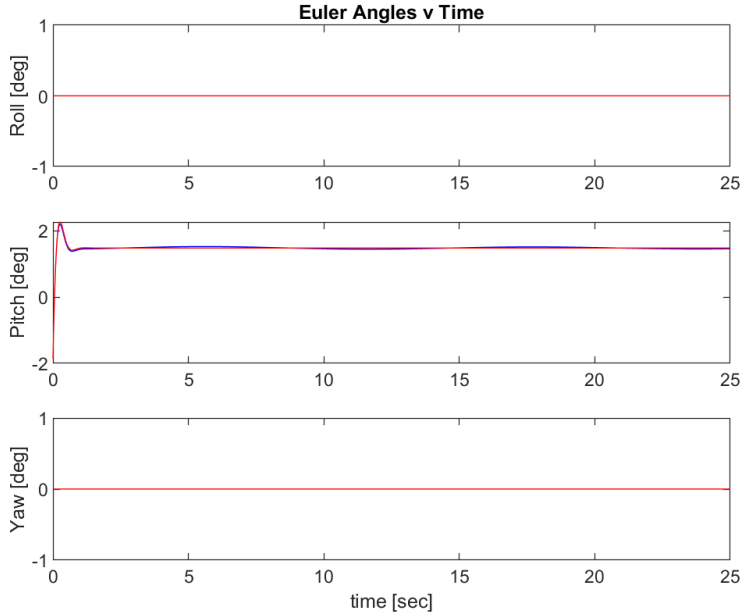


Fig. 20 2.C Euler Angles

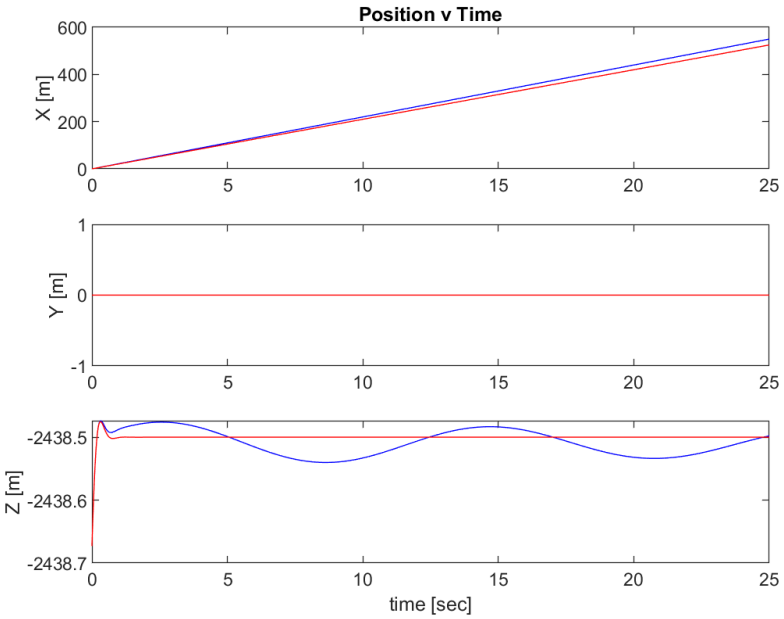


Fig. 21 2.C Position

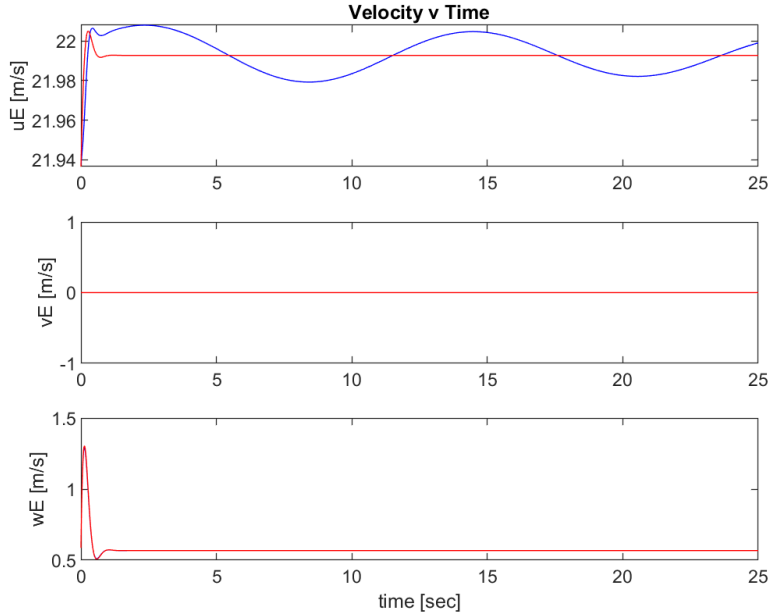


Fig. 22 2.C Velocity

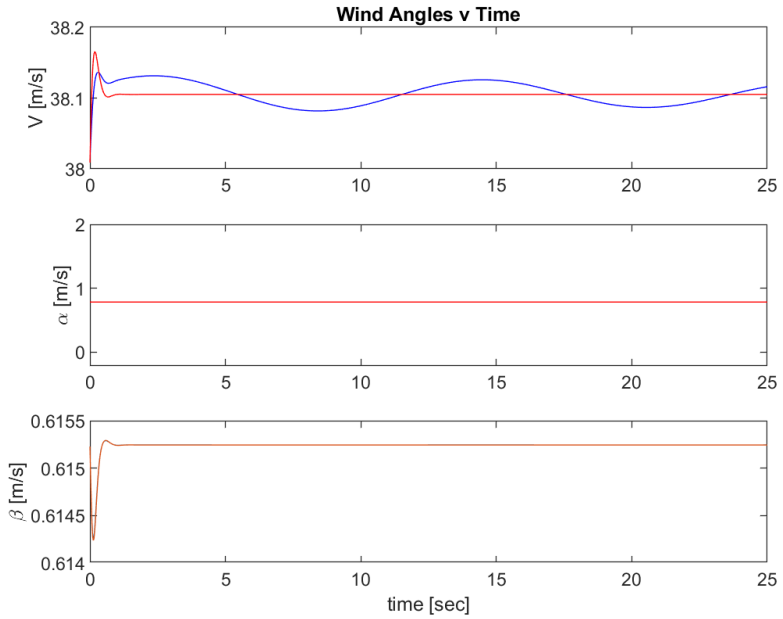


Fig. 23 2.C Wind Angles

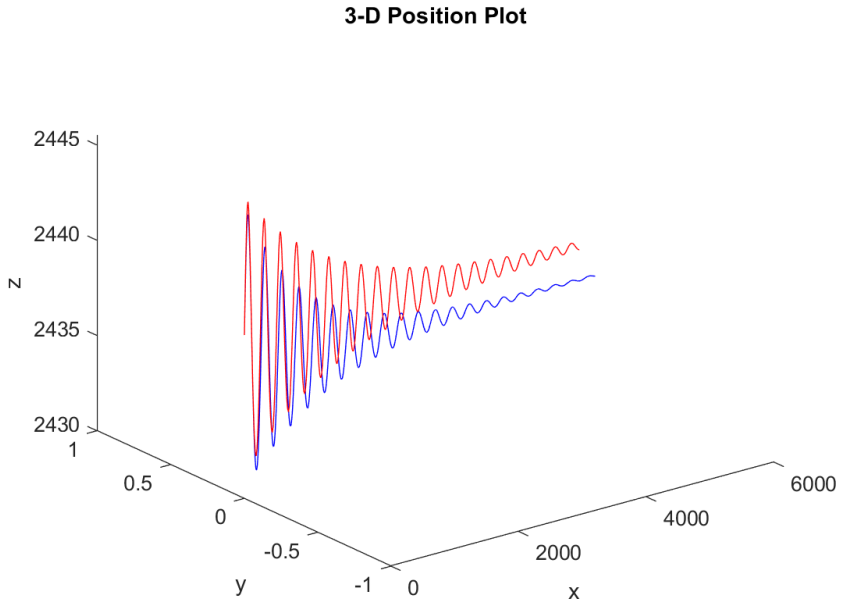


Fig. 24 2.D 3D Path with initial 10°pitch angle

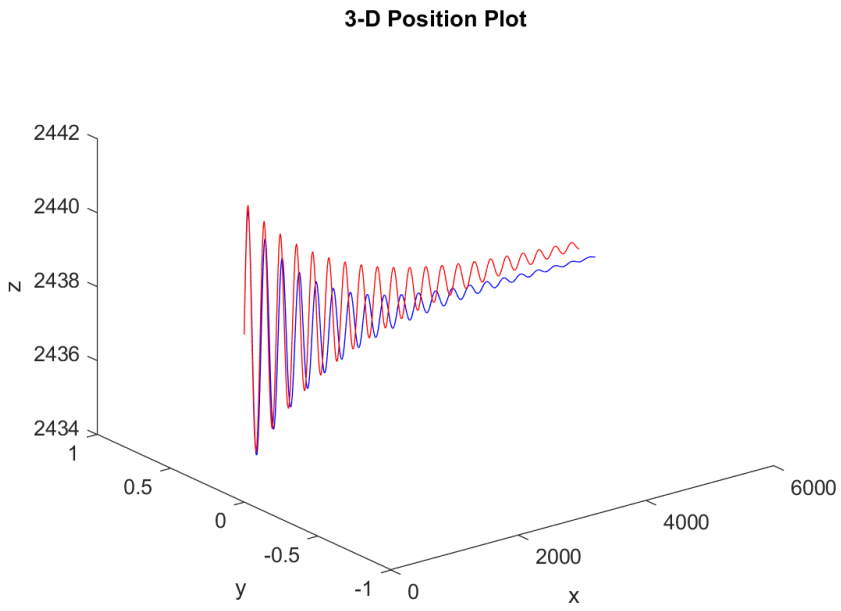


Fig. 25 2.D 3D Path with initial 5°pitch angle

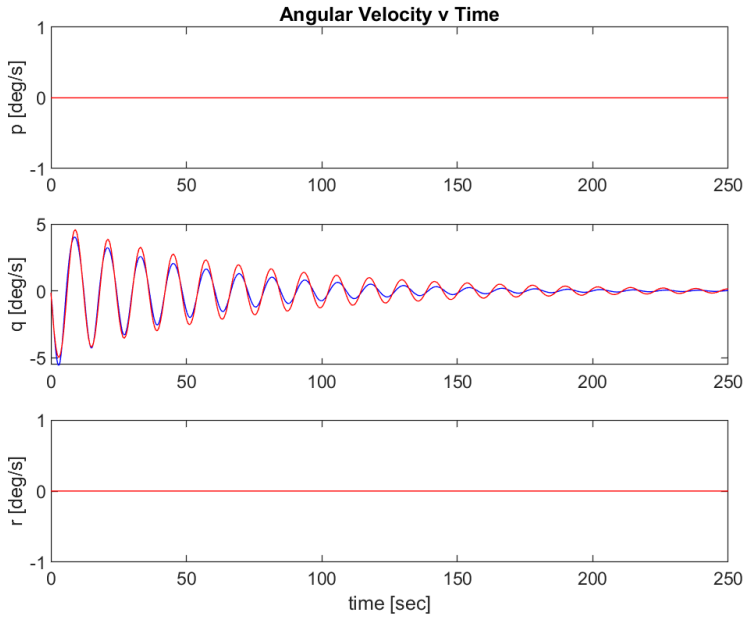


Fig. 26 2.D 10°pitch angle Angular Velocity

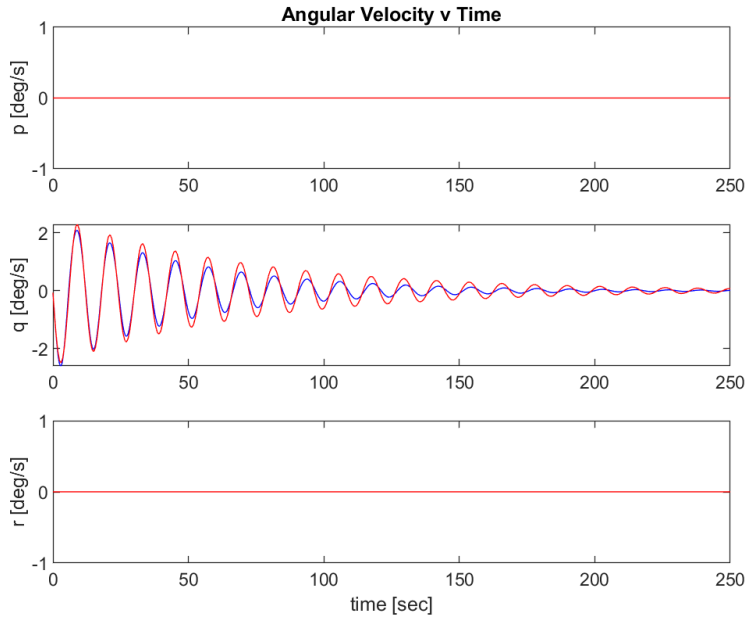


Fig. 27 2.D 5°pitch angle Angular Velocity

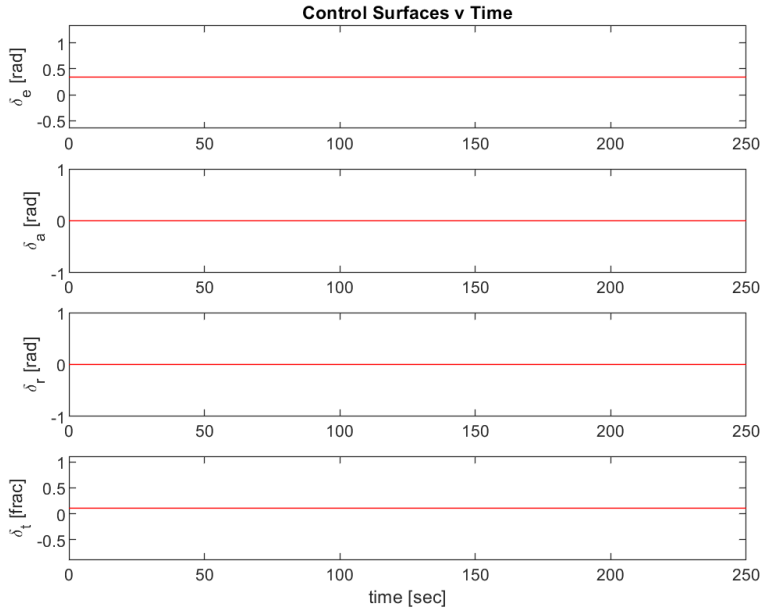


Fig. 28 2.D Control Surfaces with an initial 10°pitch angle

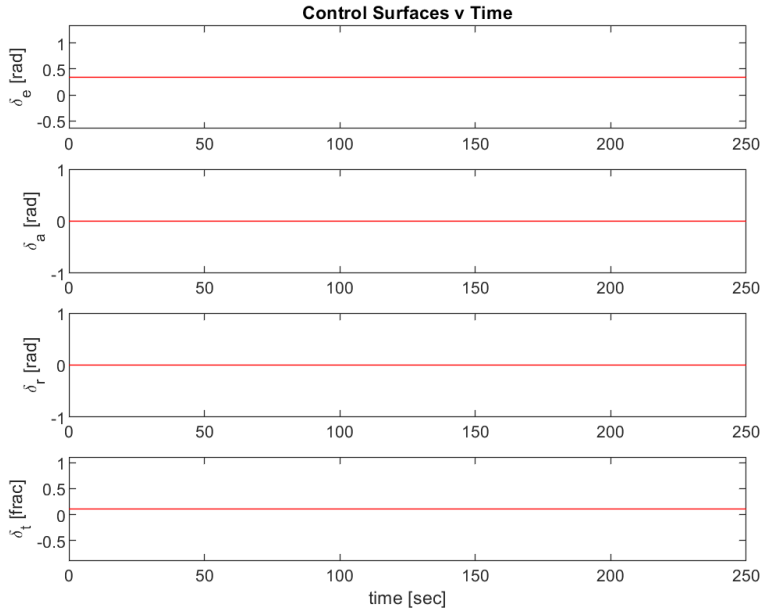


Fig. 29 2.D Control Surfaces with an initial 5°pitch angle

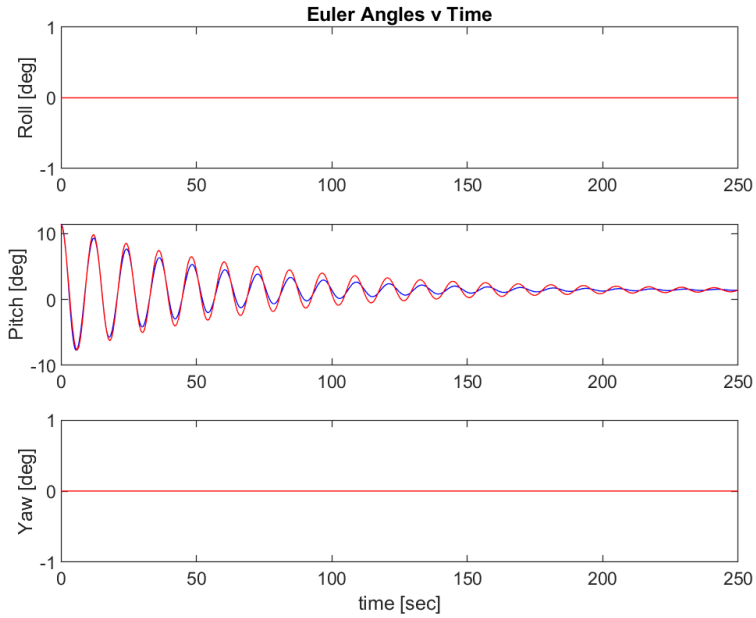


Fig. 30 2.D Euler Angles with an initial 10°pitch angle

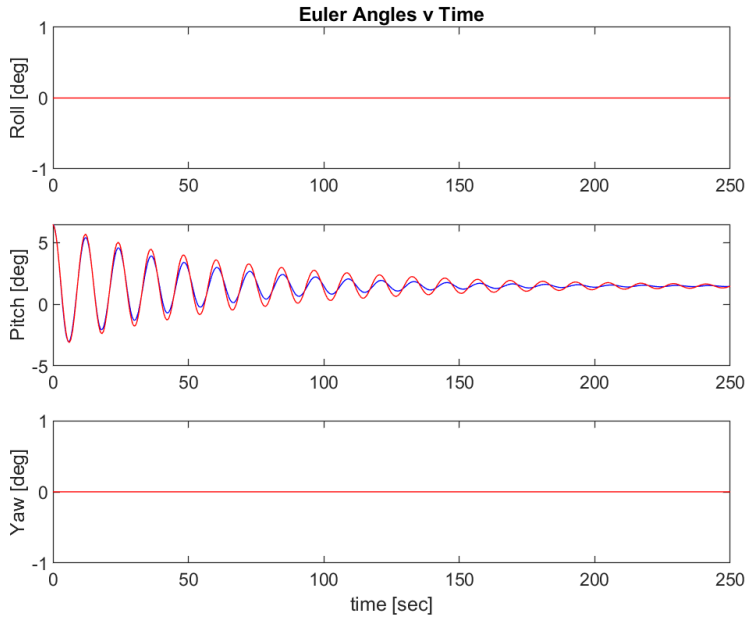


Fig. 31 2.D Euler Angles with an initial 5°pitch angle

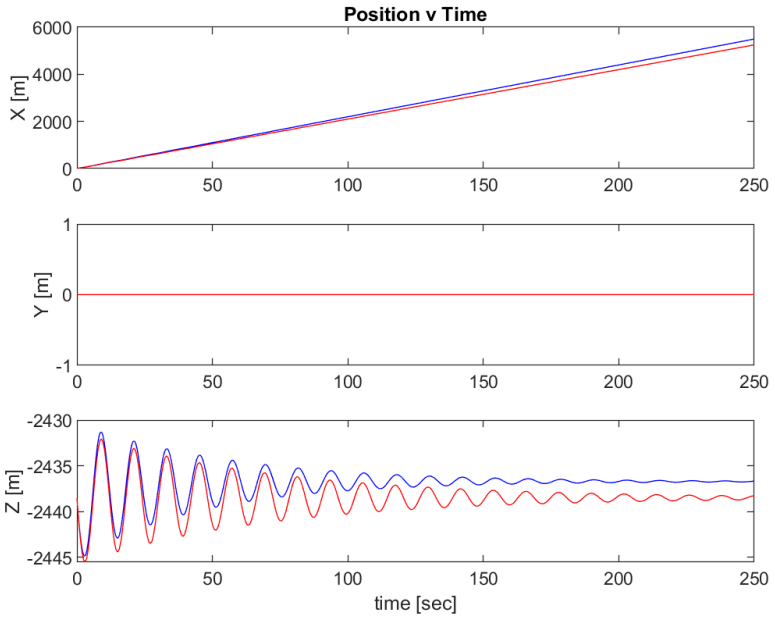


Fig. 32 2.D Position with an initial 10°pitch angle

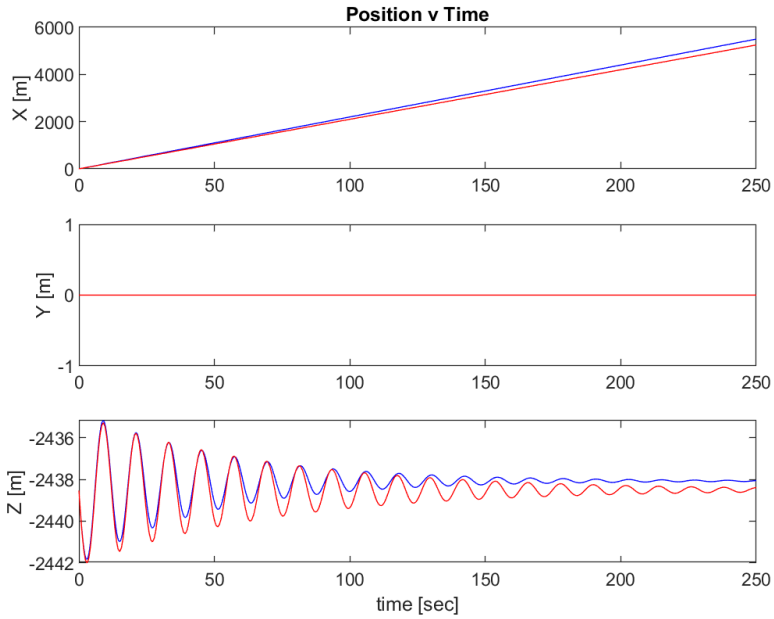


Fig. 33 2.D Position with an initial 5°pitch angle

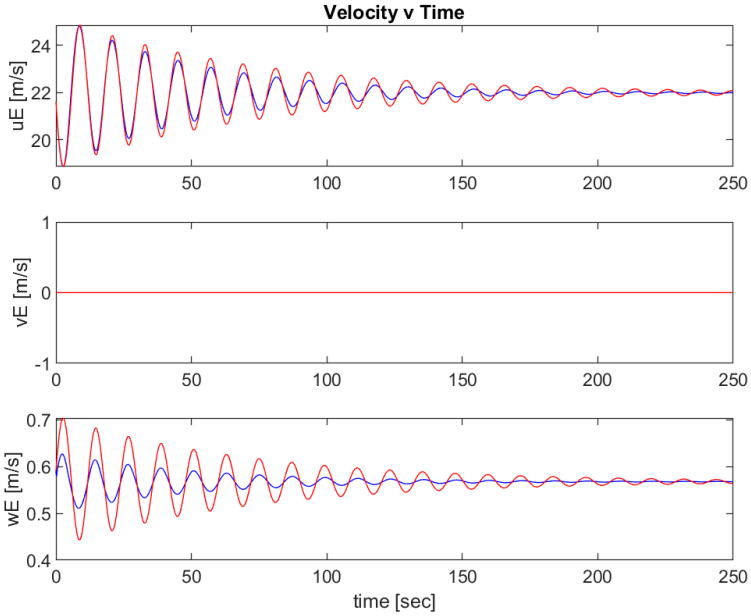


Fig. 34 2.D Velocity with an initial 10°change in pitch angle

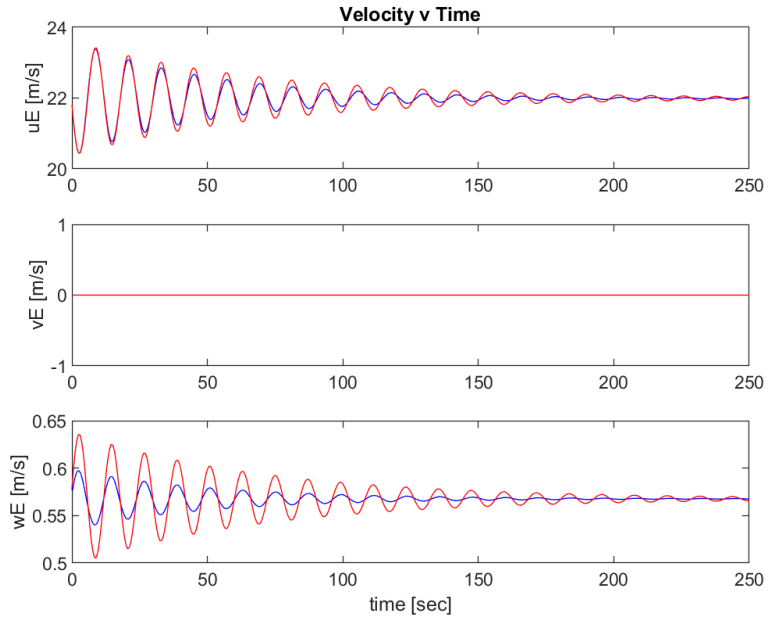


Fig. 35 2.D Velocity with an initial 5°change in pitch angle

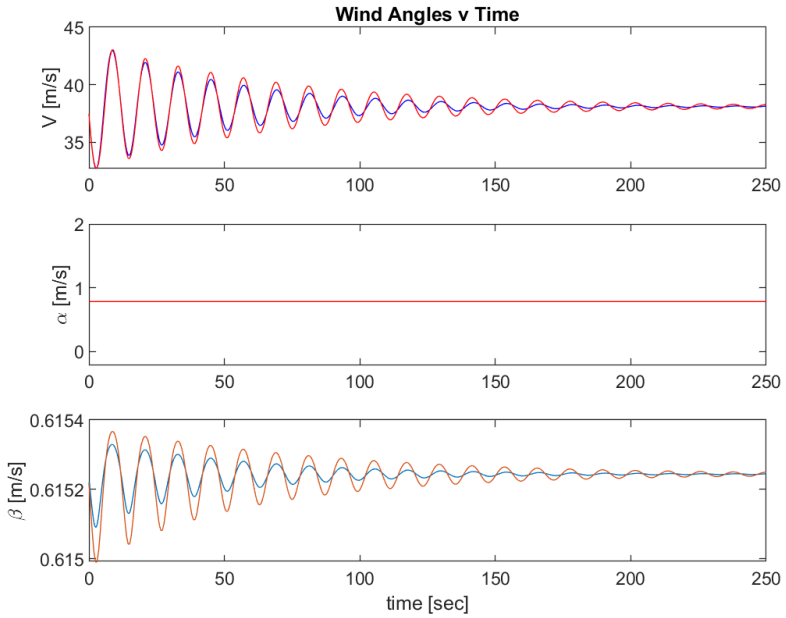


Fig. 36 2.D Wind Angles with an initial 10°pitch angle perturbation

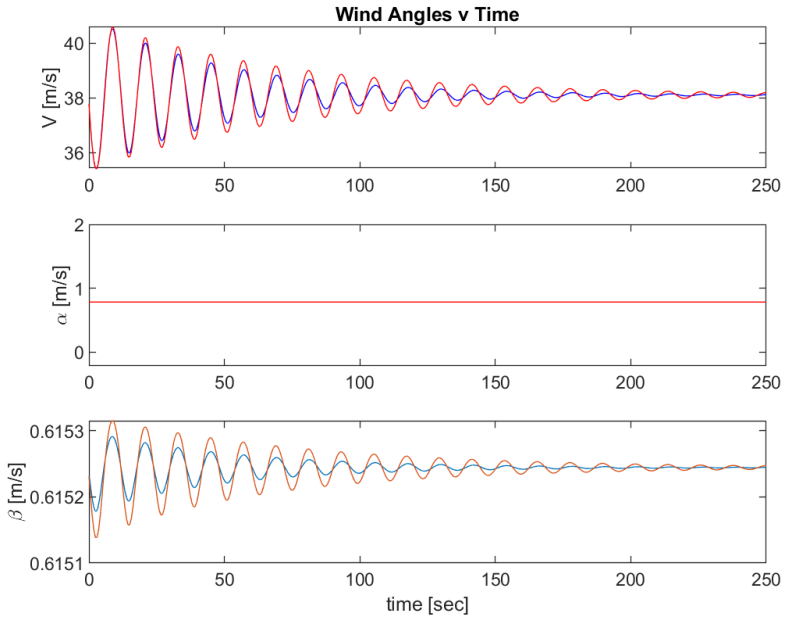


Fig. 37 2.D Wind Angles with an initial 5°pitch angle perturbation

C. Problem 3 Graphs

Note: In all graphs the red line is the linear case and the blue line is the nonlinear case.

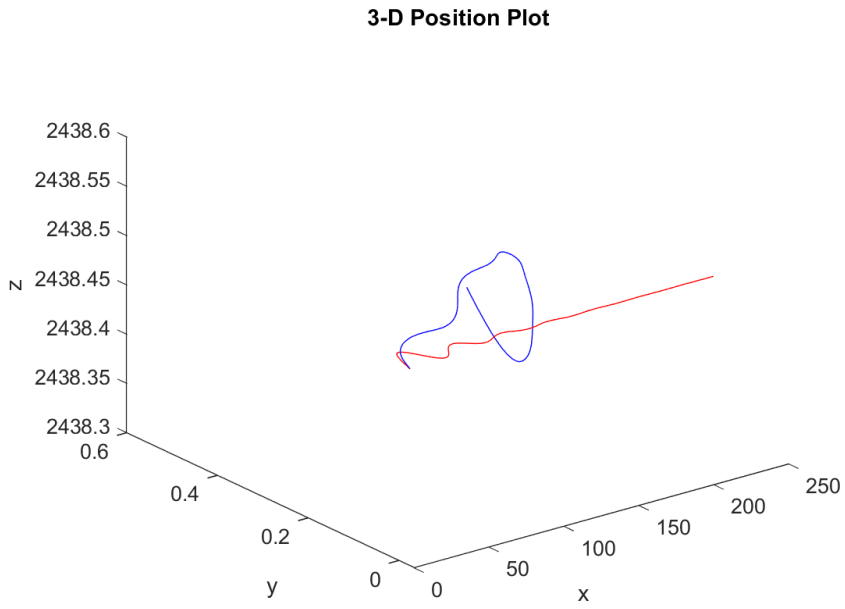


Fig. 38 3.A 3D Path

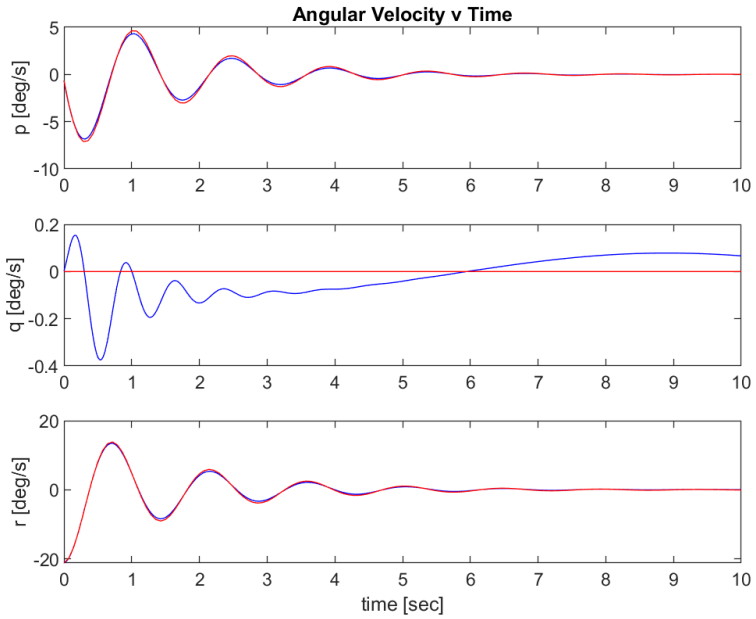


Fig. 39 3.A Angular Velocity

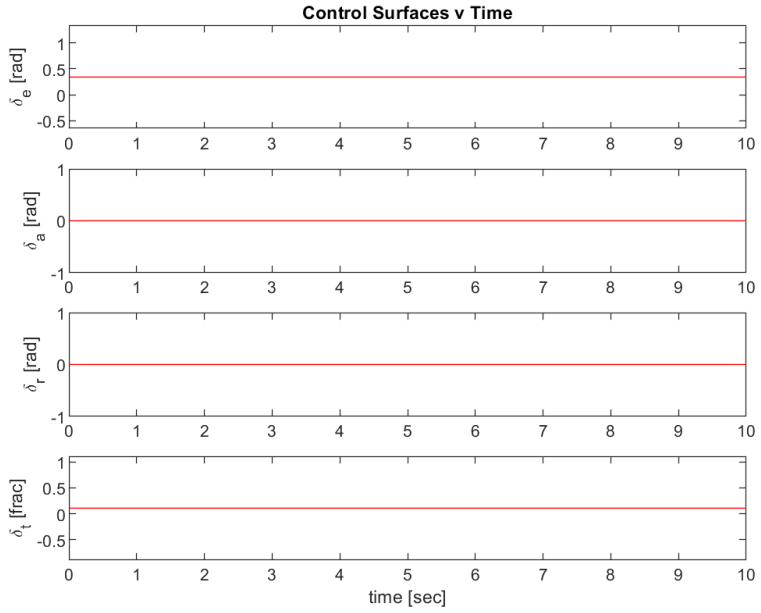


Fig. 40 3.A Control Surfaces

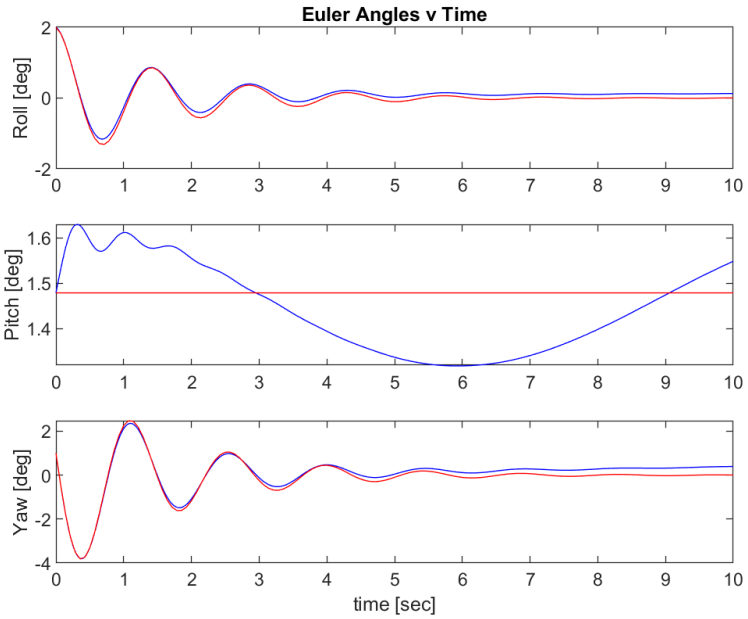


Fig. 41 3.A Euler Angles

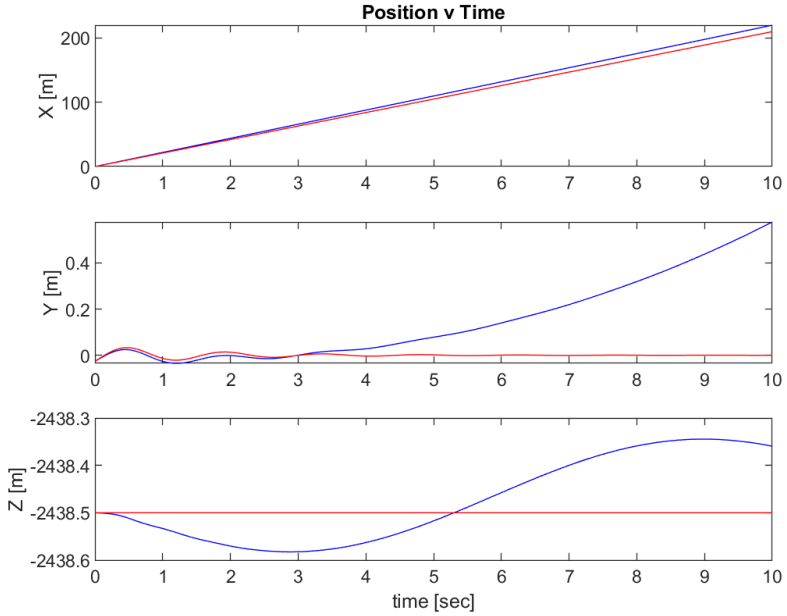


Fig. 42 3.A Position

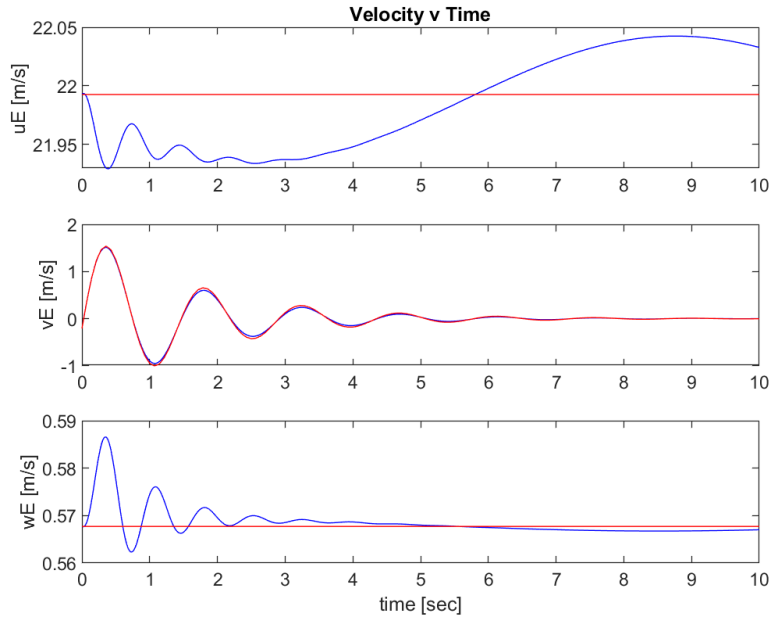


Fig. 43 3.A Velocity

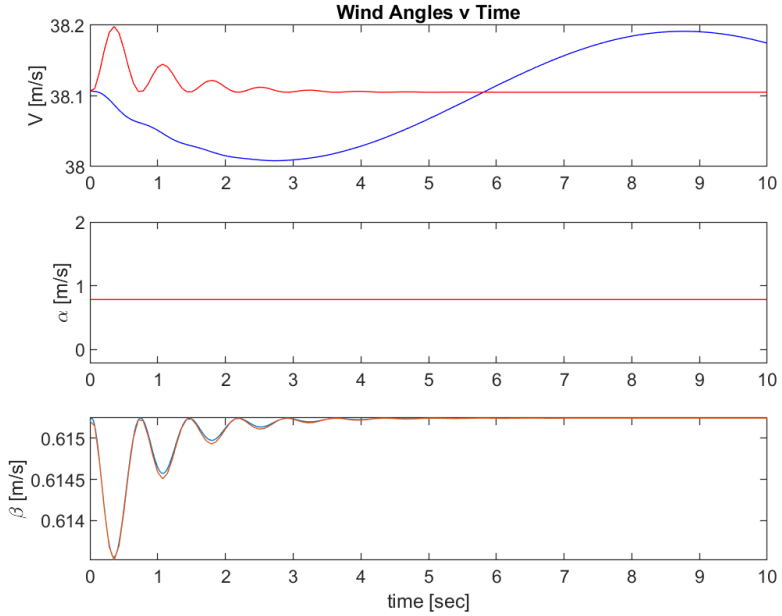


Fig. 44 3.A Wind Angles

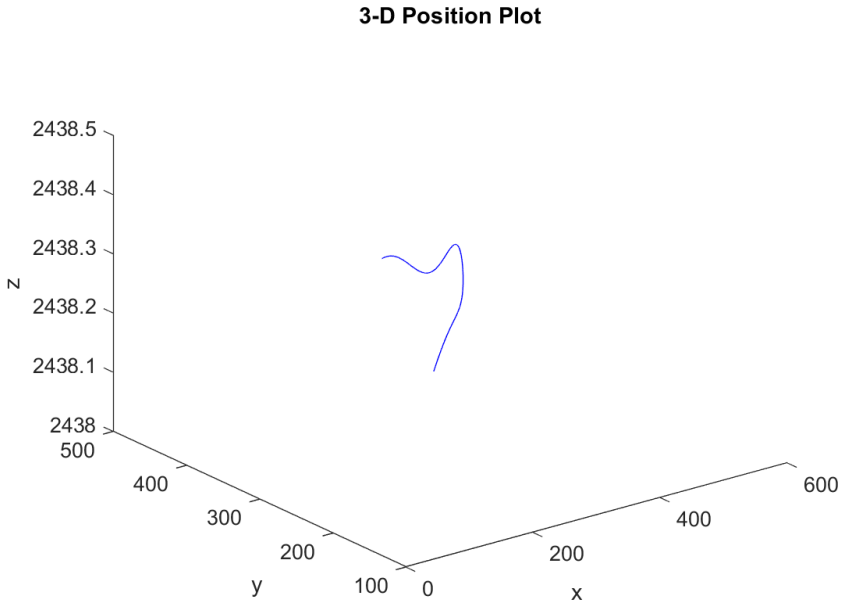


Fig. 45 3.B 3D Path

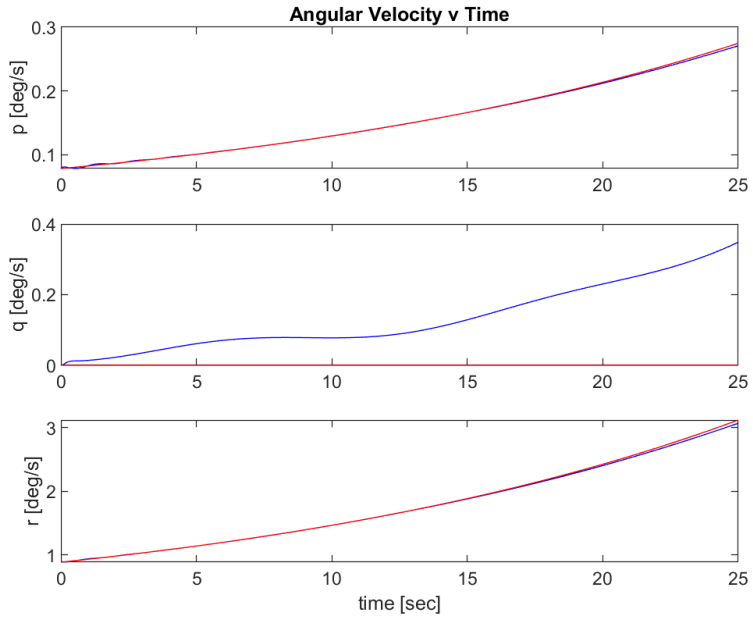


Fig. 46 3.B Angular Velocity

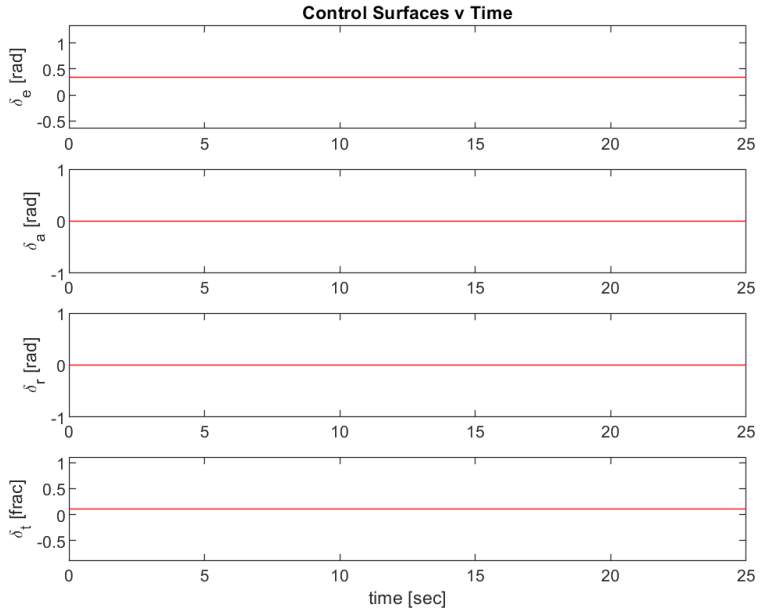


Fig. 47 3.B Control Surfaces

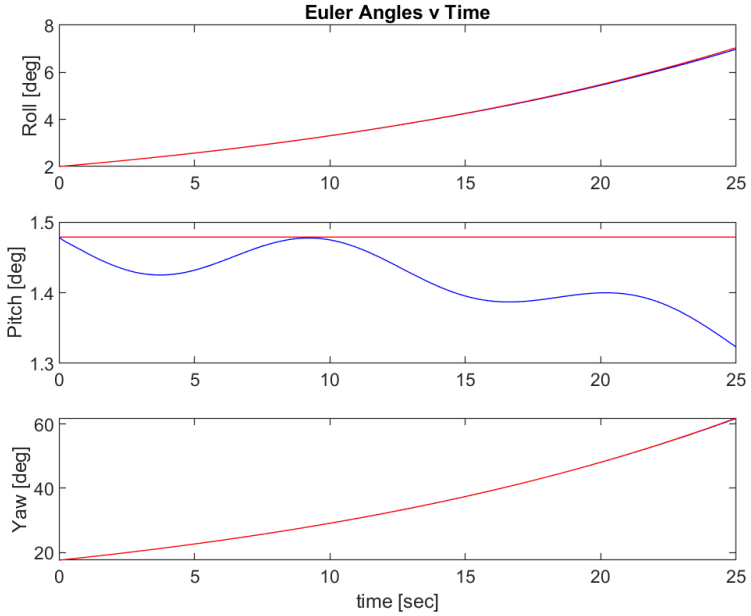


Fig. 48 3.B Euler Angles

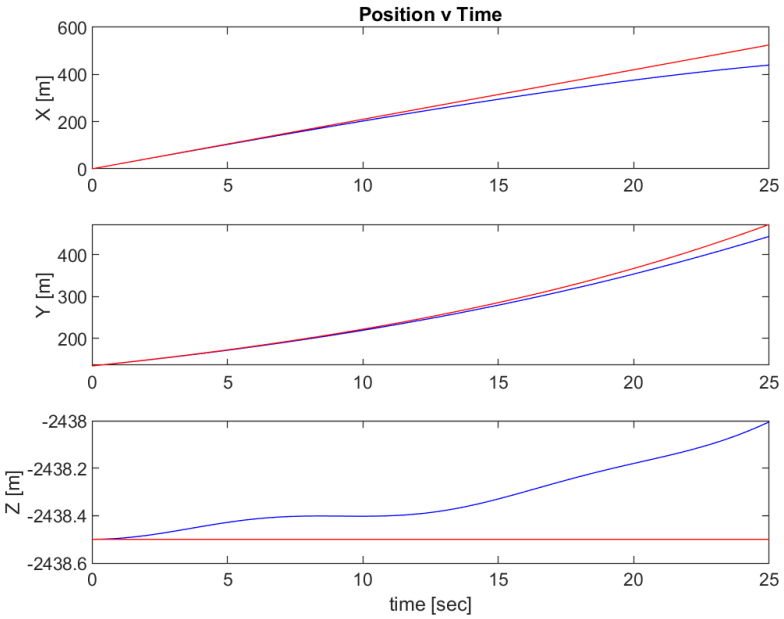


Fig. 49 3.B Position

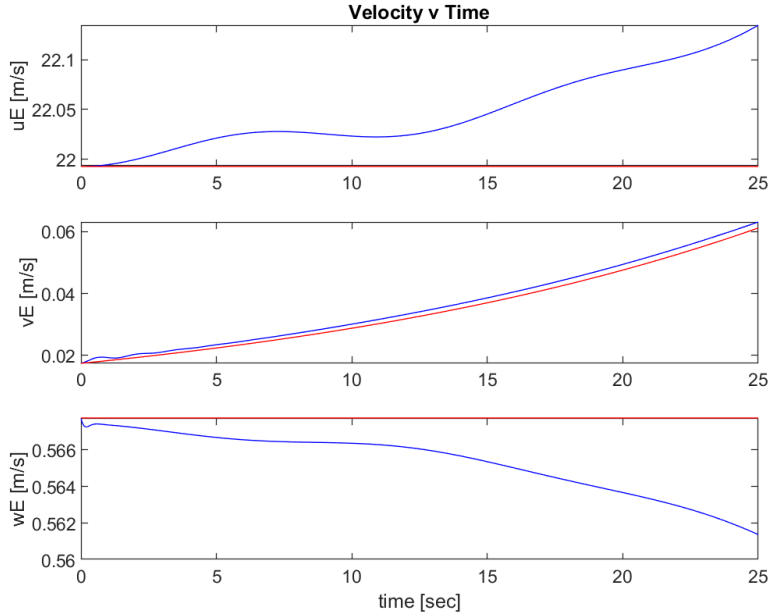


Fig. 50 3.B Velocity

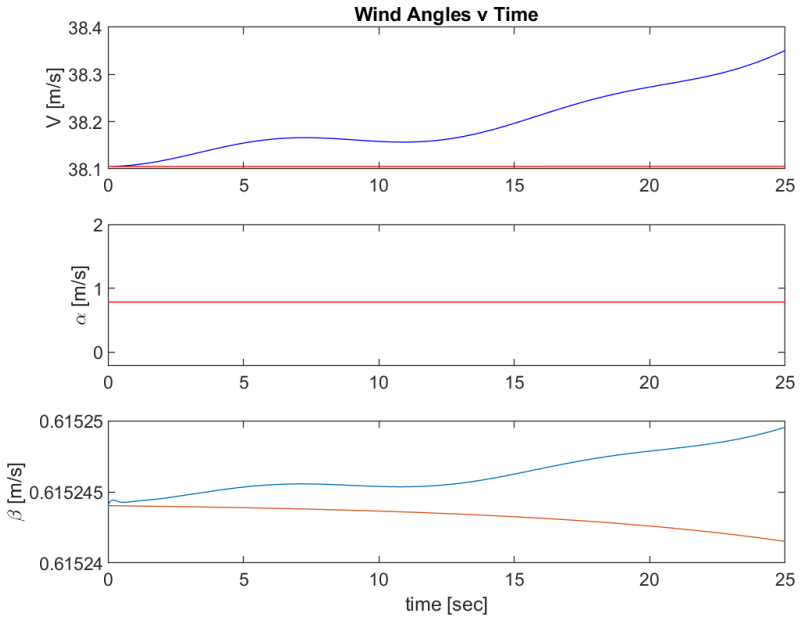


Fig. 51 3.B Wind Angles

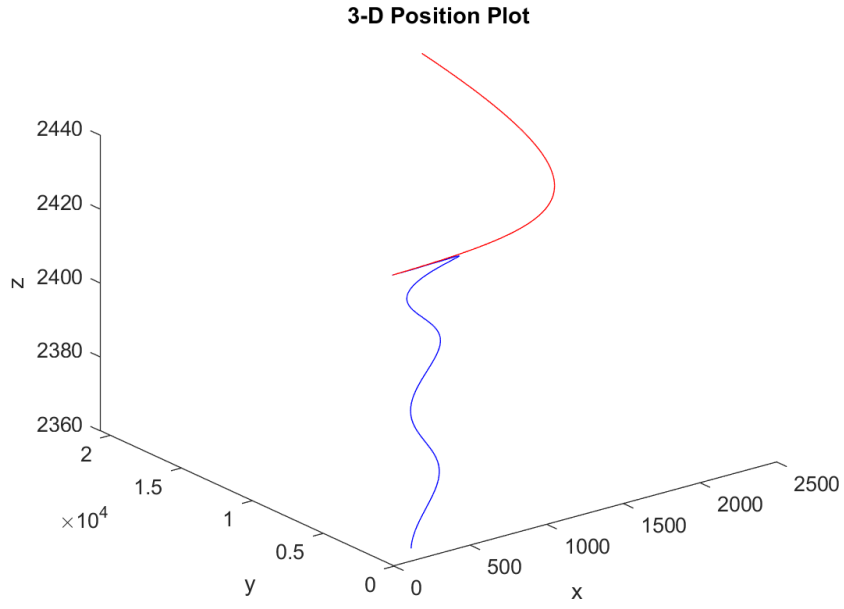


Fig. 52 3.C 3D Path

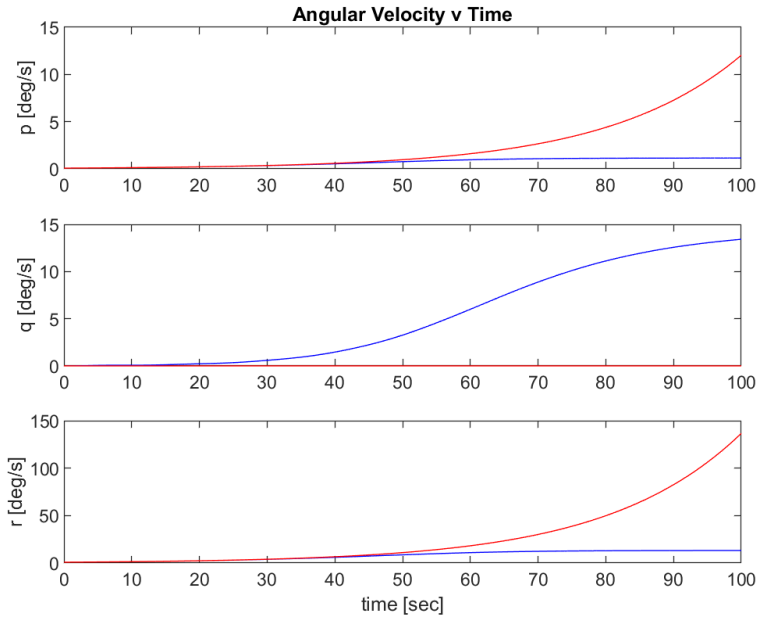


Fig. 53 3.C Angular Velocity

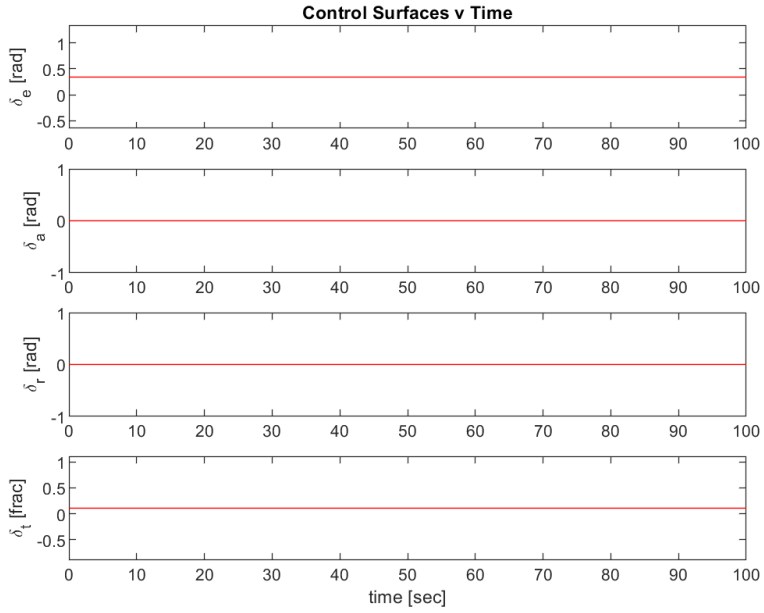


Fig. 54 3.C Control Surfaces

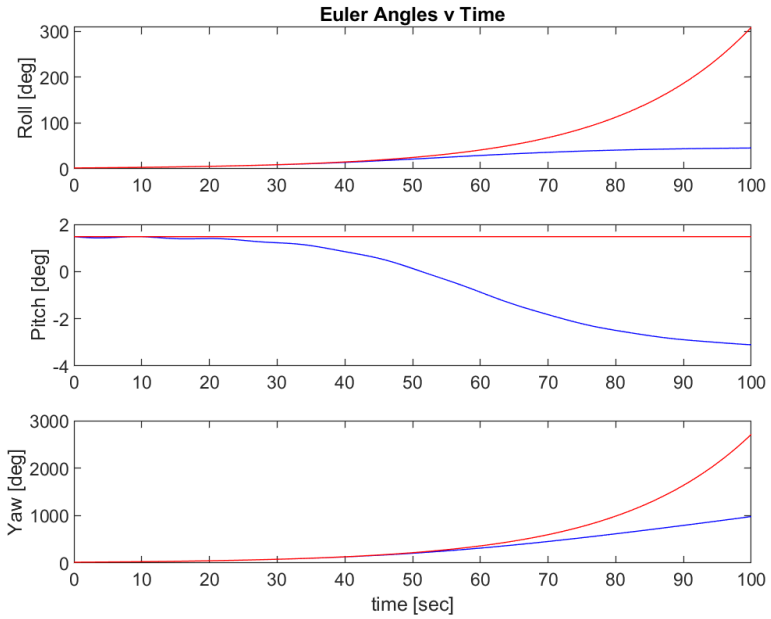


Fig. 55 3.C Euler Angles

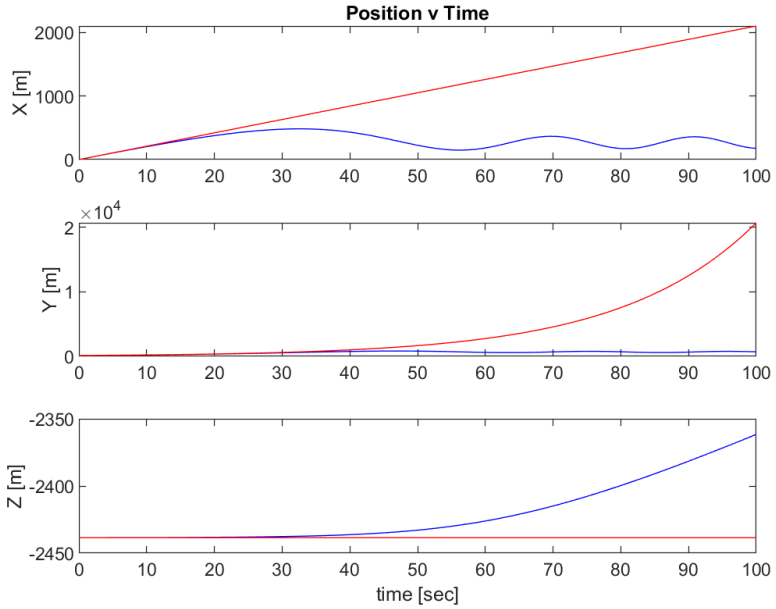


Fig. 56 3.C Position

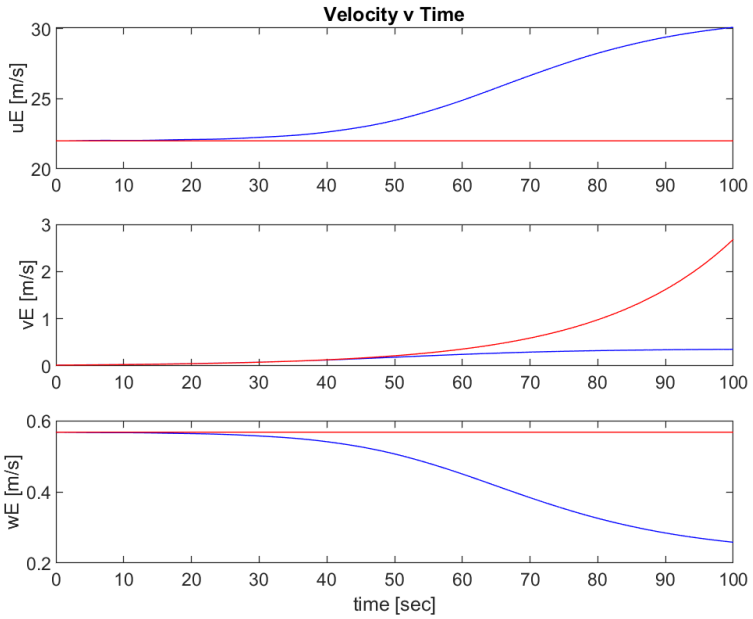


Fig. 57 3.C Velocity

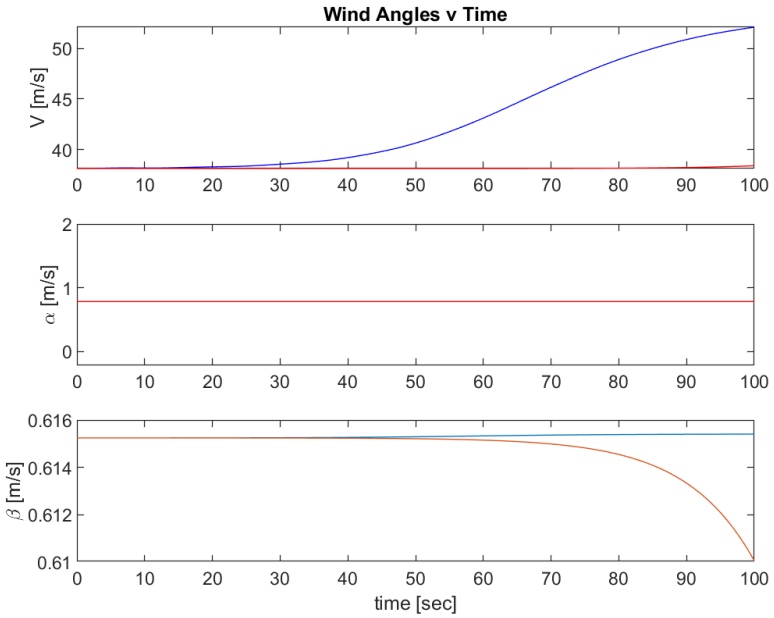


Fig. 58 3.C Wind Angles

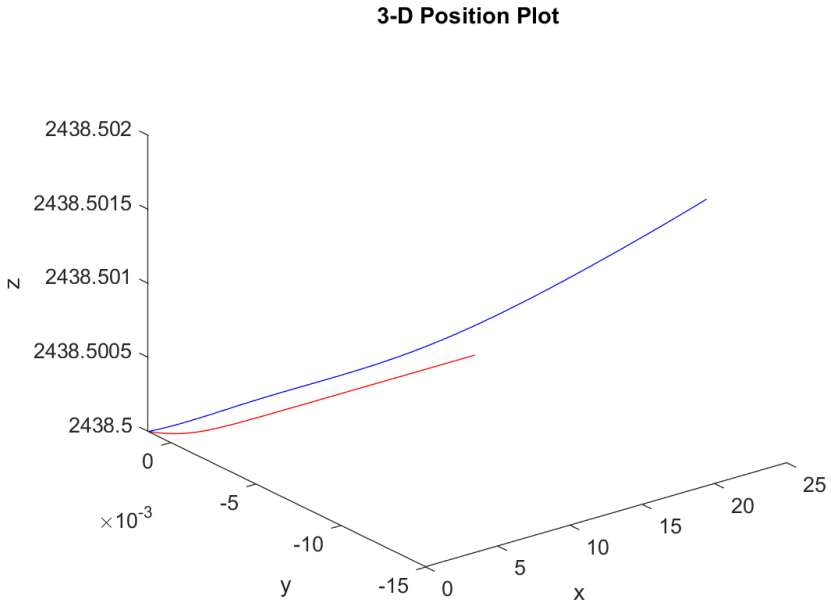


Fig. 59 3.D 3D Path

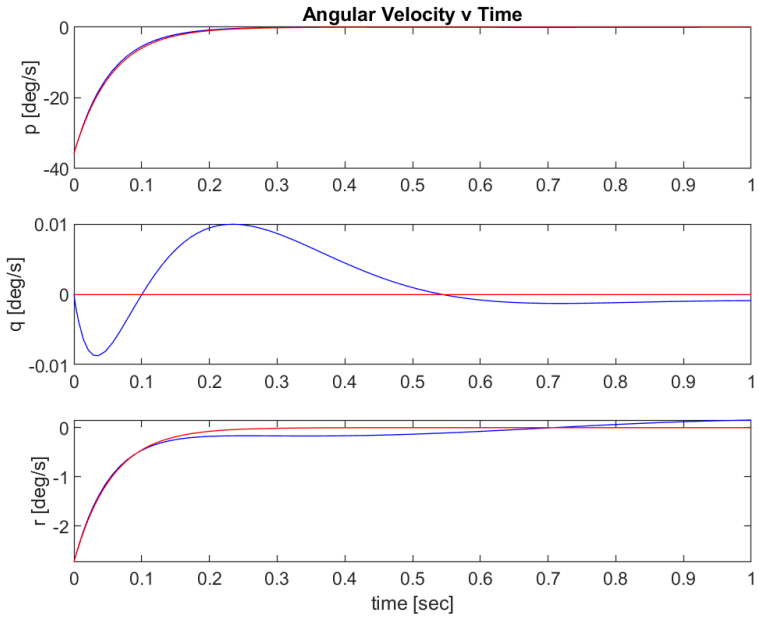


Fig. 60 3.D Angular Velocity

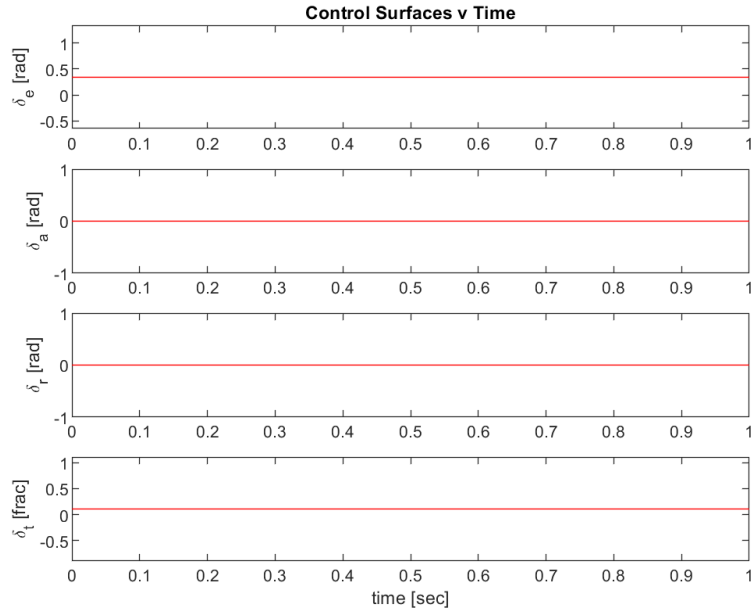


Fig. 61 3.D Control Surfaces

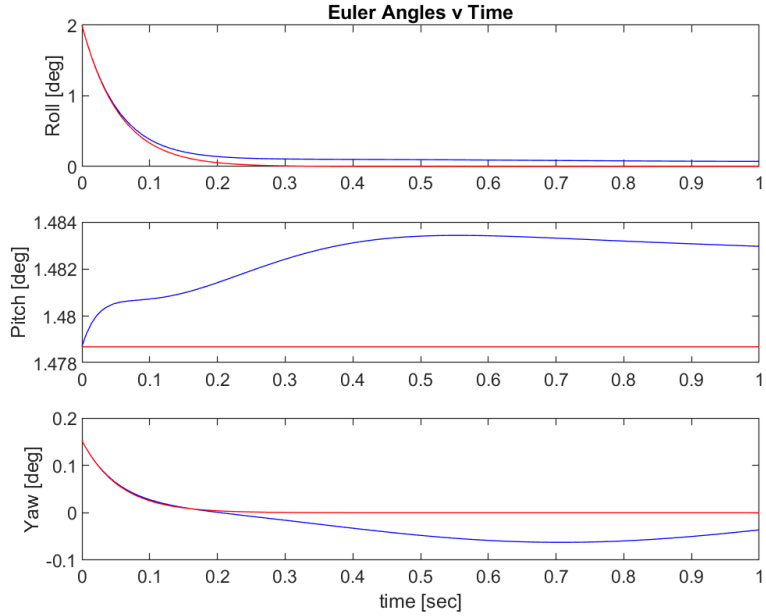


Fig. 62 3.D Euler Angles

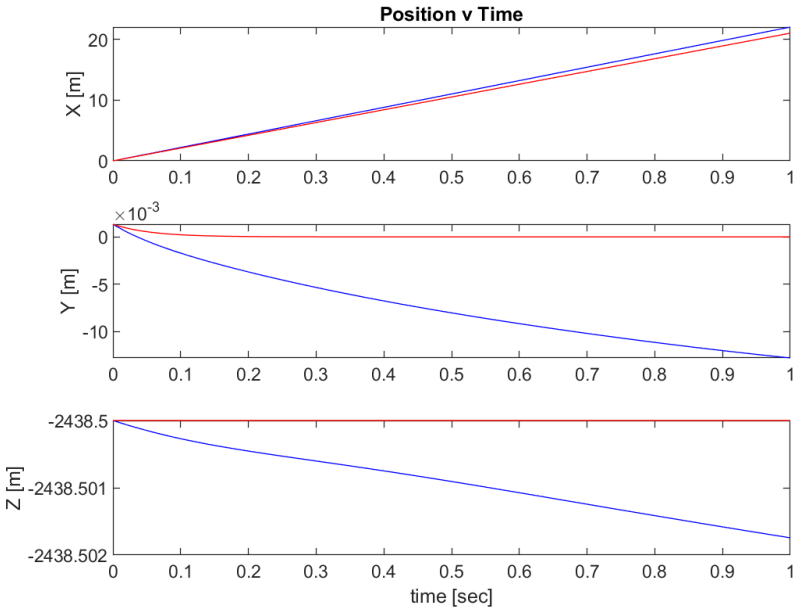


Fig. 63 3.D Position

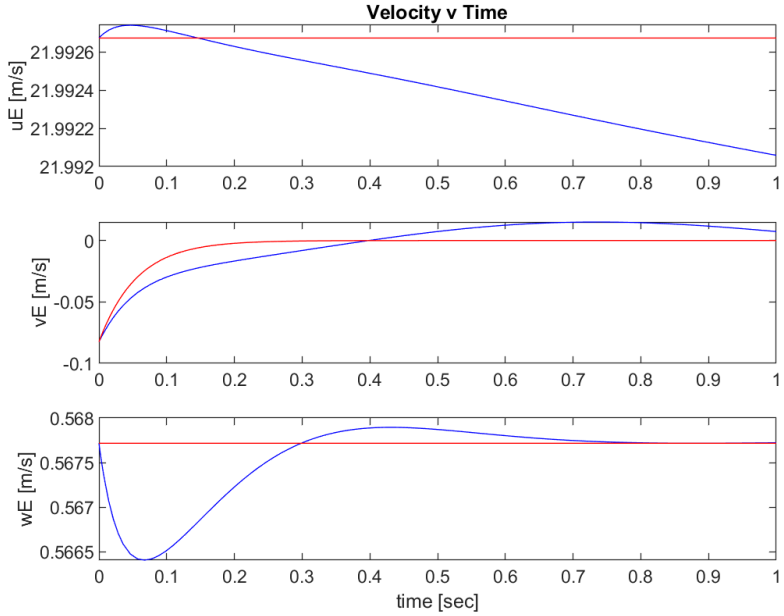


Fig. 64 3.D Velocity

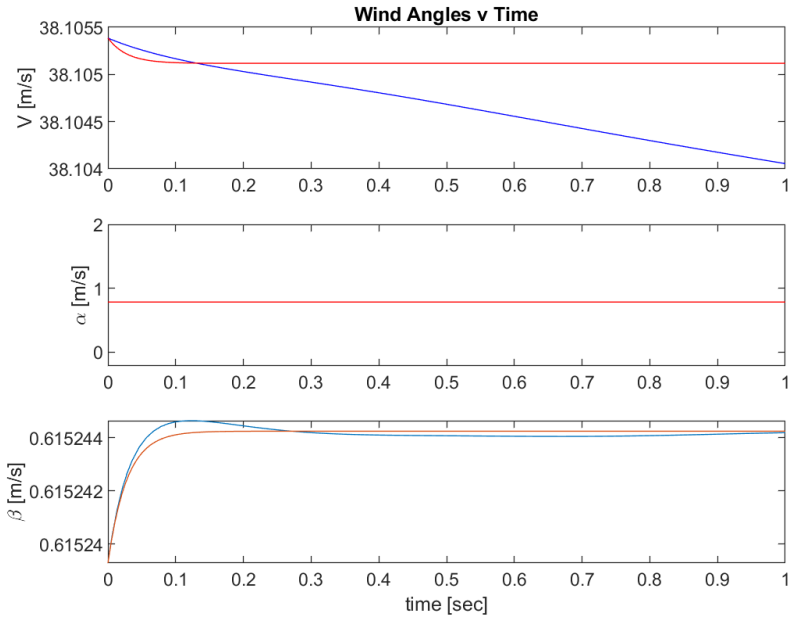


Fig. 65 3.D Wind Angles

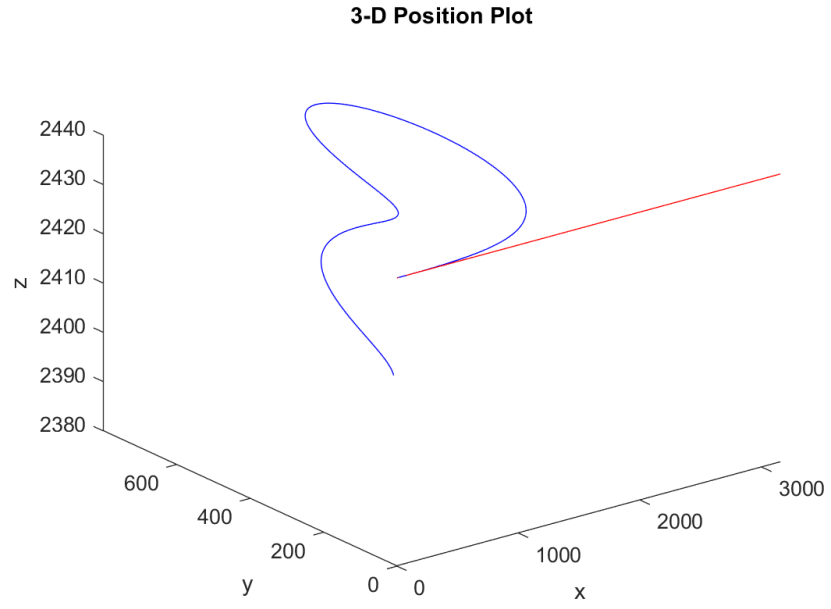


Fig. 66 3.E 3D Path

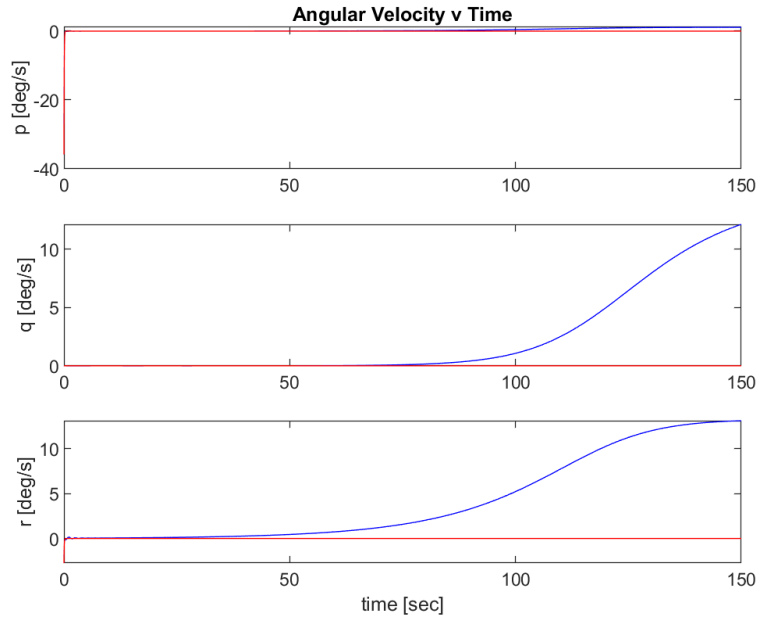


Fig. 67 3.E Angular Velocity

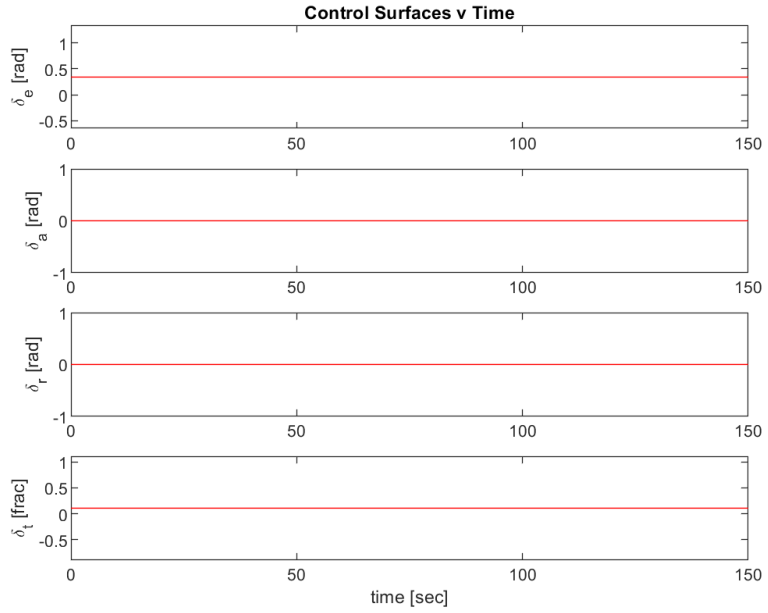


Fig. 68 3.E Control Surfaces

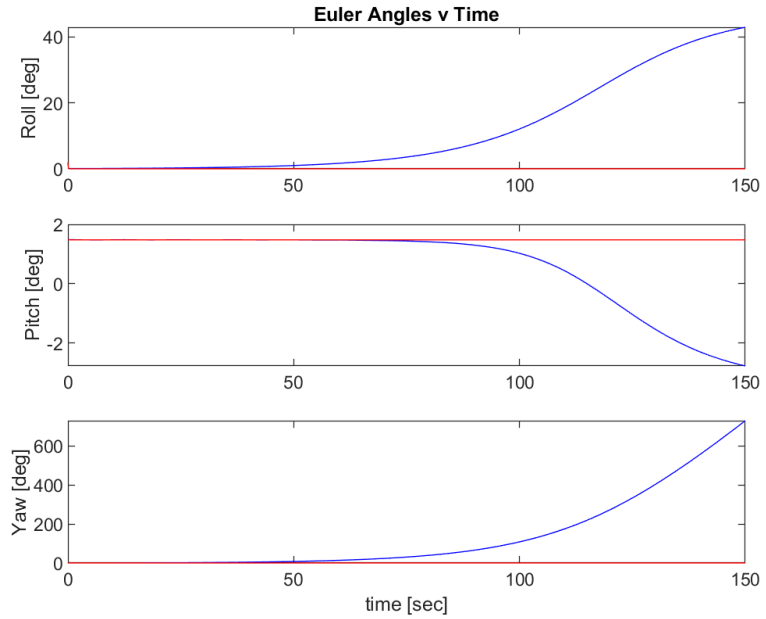


Fig. 69 3.E Euler Angles

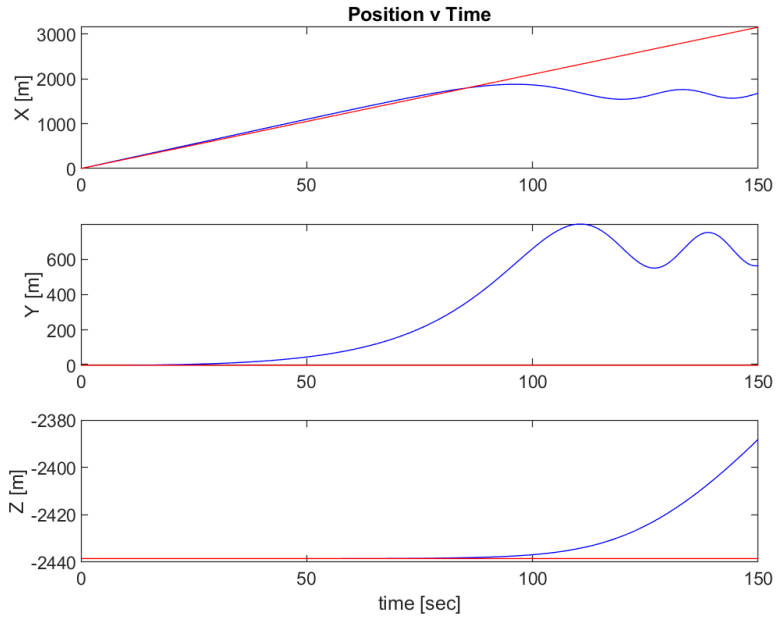


Fig. 70 3.E Position

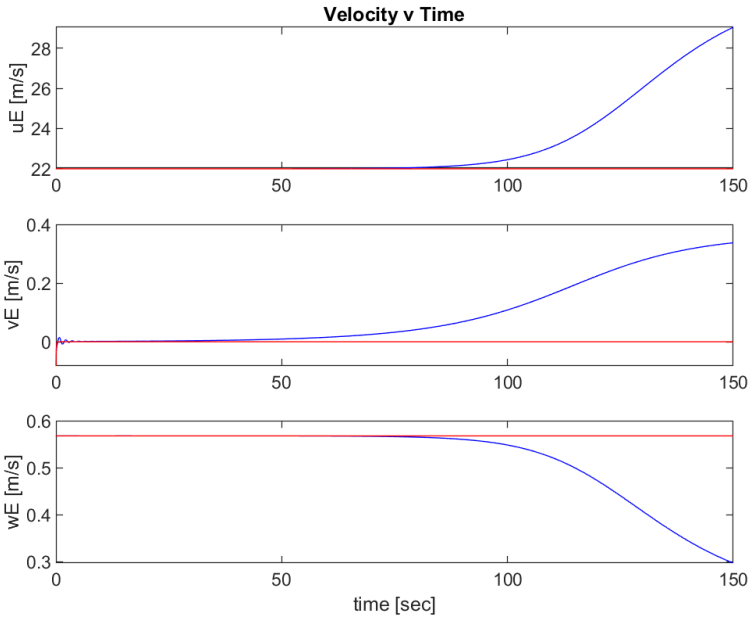


Fig. 71 3.E Velocity

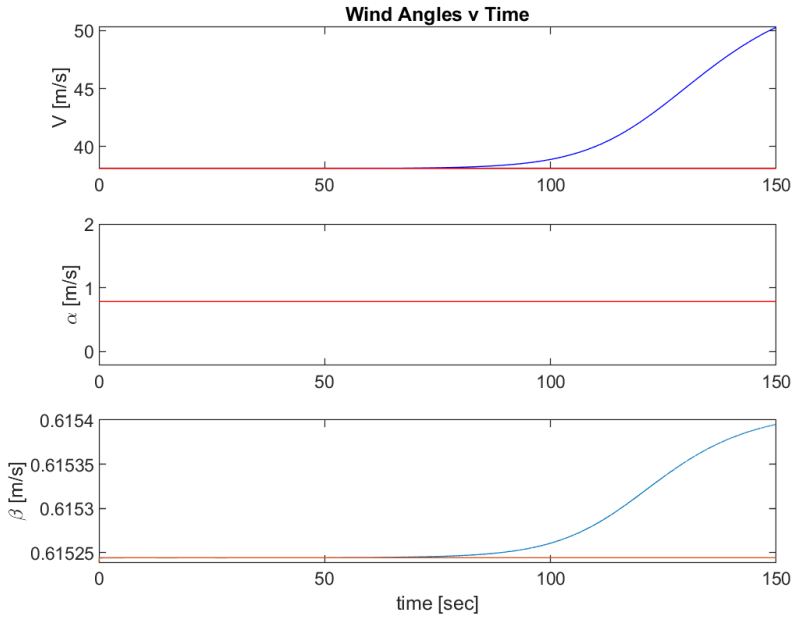


Fig. 72 3.E Wind Angles