# *** Q-2, DYNAMICS OF HANSA-3 AIRCRAFT ***

## # INITIAL CONDITIONS ASSUMED

TRIM CONDITIONS

>> TRIM

z0 = -2500

CL_trim = 0.3403

Thrust = 869.4045

alpha_trim = -0.0217

delE_trim = 0.2105

u_trim = 59.9859

v_trim = 0

w_trim = -1.3009

>> Assumptions

Beta = 0, i.e. No cross-wind
p0 = 0; q0 = 0.04; r0 = 0;
phi0 = 0; theta0 = 0.002; psi0 = 0;

# MATLAB code

## >> TRIM.m

```
% TRIM CONDITION OF AIRCRAFT (Assumed typical cruise velocity & cruise
% altitude from Wikipedia)

PARAMETERS;
beta_trim = 0;

h = 2500;
rho = DENSITY(h);
z0 = -h

Vinf = 60;

W = m*g;
Q = 1/2*rho*Vinf^2;

cLtrim = W / (Q*S)
cDtrim = CD_0 + k*cLtrim^2;
Thrust = Q*S*cDtrim

res = inv([CL_alpha CL_de; CM_alpha CM_de])*[cLtrim-CL_0; -CM_0];

alpha_trim = res(1)
delE_trim = res(2)

u_trim = Vinf*cos(alpha_trim)*cos(beta_trim)
v_trim = Vinf*sin(beta_trim)
w_trim = Vinf*sin(alpha_trim)*cos(beta_trim)
```

## >> MAIN_Q2.m

```
% MAIN

close all
clear all
clc

TRIM;

u0 = u_trim; v0 = v_trim; w0 = w_trim; % m/s
x0 = 0; y0 = 0; z0 = z0; % m
p0 = 0; q0 = 0.04; r0 = 0; % rad/s
phi0 = 0; theta0 = 0.002; psi0 = 0; % rad
```

```
% Initial states
a0 = [u0, v0, w0, x0, y0, z0, p0, q0, r0, phi0, theta0, psi0];

[t,y] = RK4(@EQUATIONS, 50, 0.01, a0);

% Cntrol inputs
c = zeros(numel(t), 4);
for i=1:numel(t)
    cc = CS_DEF(t(i), y(i));
    c(i,:) = cc;
end

PLOTTING;
```

## >> PARAMETERS.m

```
%% PARAMETERS

% Geometric & Inertial properties
m = 750;
g = 9.81;
Ixx = 873;
Iyy = 907;
Izz = 1680;
Ixz = 1144;
S = 12.47;
b = 10.47;
cbar = 1.211;
AR = 8.8;

% Aerodynamic parameters
% Longitudinal
CD_0 = 0.035;   k = 0.045;
CL_0 = 0.37;    CL_alpha = 5;       CL_q = 37.211;  CL_de = 0.374;
CM_0 = 0.091;   CM_alpha = -2.937;  CM_q = -8.719;  CM_de = -0.735;

%
CY_0 = 0;       CY_beta = -0.531;   CY_p = -0.0571;
CY_r = 0.4657;  CY_delr = 0.1502;

%
Cl_0 = 0;       Cl_beta = -0.031;   Cl_p = -0.262;  Cl_r = -0.0541;
Cl_delr = 0.005;  Cl_dela = -0.153;

%
Cn_0 = 0;       Cn_beta = 0.01;     Cn_p = -0.007;  Cn_r = -0.067;
Cn_delr = -0.047;
```

## >> PLOTTING.m

```
%% PLOTTING

figure(1);
% title('3-2-1-1 input to elevator')

subplot(6,2,1);
plot(t,y(:,1),'r');
ylabel('u');

subplot(6,2,3);
plot(t,y(:,2),'r');
ylabel('v');

subplot(6,2,5);
plot(t,y(:,3),'r');
ylabel('w');

subplot(6,2,7);
plot(t,y(:,7),'r');
ylabel('p');

subplot(6,2,9);
plot(t,y(:,8),'r');
ylabel('q');

subplot(6,2,11);
plot(t,y(:,9),'r');
ylabel('r');
xlabel('t')

subplot(6,2,2);
plot(t,y(:,10),'r');
ylabel('\phi');

subplot(6,2,4);
plot(t,y(:,11),'r');
ylabel('\theta');

subplot(6,2,6);
plot(t,y(:,12),'r');
ylabel('\psi');

subplot(6,2,8);
plot(t,c(:,1),'r');
ylabel('\delta_e');

subplot(6,2,10);
plot(t,c(:,2),'r');
```

```
ylabel('\delta_a');

subplot(6,2,12);
plot(t,c(:,3),'r');
ylabel('\delta_r');
xlabel('t')
```

## >> DENSITY.m

```
function rho = DENSITY(hg)

hg1=0;           % Gradient layer 1-2 (0 to 11 km)
T1=288.16;
a1=-0.0065;

p1=101325;  % in pascal at MSL (Mean Sea Level)
d1=1.225;   % in kg/m3 at MSL
g0=9.81;    %gravitational acceleration at MSL
R=287;      %in J/kgK

r=6371000;                        % Radius of earth

h=(r*hg)/(r+hg);                  % Calculation for Geopotential altitude
h1=(r*hg1)/(r+hg1);               % Geo potential ALT CALC
T=T1+(a1*delh);                   % TEMP CALC
d=d1*((T/T1)^((-g0/(a1*R))-1));        % DENS CALC

rho = d;

end
```

## >> EQUATIONS.m

```
% EQUATIONS

function a_dot = EQUATIONS(t, a)

PARAMETERS;

u = a(1);      v = a(2);      w = a(3);
x = a(4);      y = a(5);      z = a(6);
p = a(7);      q = a(8);      r = a(9);
phi = a(10);   theta = a(11);  psi = a(12);

ctrl = CS_DEF(t,a);
```

```
[Fxaero, Fyaero, Fzaero, l, M, N] = FORCEMOMENT(a, ctrl);

T = ctrl(4);

u_dot = Fxaero/m + T/m - q*w + r*v - g*sin(theta);
v_dot = Fyaero/m - r*u + p*w + g*cos(theta)*sin(phi);
w_dot = Fzaero/m - p*v + q*u + g*cos(theta)*cos(phi);

p_dot = (l*Izz + N*Ixz - p*q*(Ixz*(Iyy-Ixx-Izz)) - q*r*(Izz*(Izz-Iyy)+Ixz^2))/(Ixx*Izz -
Ixz^2);
q_dot = (M + p*r*(Izz-Ixx) - Ixz*(p^2-r^2))/Iyy;
r_dot = (l*Ixz + N*Ixx + p*q*(Ixz^2+Ixx*(Ixx-Iyy)) + q*r*(Ixz*(Iyy-Ixx-Izz)))/(Ixx*Izz -
Ixz^2);

%*********************
phi_mat = [1    0           0;
           0    cos(phi)    sin(phi);
           0    -sin(phi)   cos(phi)];

theta_mat = [cos(theta)     0   -sin(theta);
             0              1   0;
             sin(theta)     0   cos(theta)];

psi_mat = [cos(psi)     sin(psi)    0;
           -sin(psi)    cos(psi)    0;
           0            0           1];

xyz_mat = psi_mat' * theta_mat' * phi_mat' * [u;  v;  w];
x_dot = xyz_mat(1);
y_dot = xyz_mat(2);
z_dot = xyz_mat(3);

% p,q,r to euler angles
pqr2eul = [1        sin(phi)*tan(theta)     cos(phi)*tan(theta);
           0        cos(phi)                -sin(phi);
           0        sin(phi)/cos(theta)     cos(phi)/cos(theta)];

phithetapsi_dot_mat = pqr2eul*[p;   q;  r];

phi_dot = phithetapsi_dot_mat(1);
theta_dot = phithetapsi_dot_mat(2);
psi_dot = phithetapsi_dot_mat(3);

%****************************

a_dot = [u_dot, v_dot, w_dot, x_dot, y_dot, z_dot, p_dot, q_dot, r_dot, phi_dot, theta_dot,
psi_dot]';

end
```

## >> FORCEMOMENT.m

```
% Forces & Moments

function [Fx, Fy, Fz, l, M, N] = FORCEMOMENT(a, CS_DEF)

PARAMETERS;

u = a(1);     v = a(2);     w = a(3);
x = a(4);     y = a(5);     z = a(6);
p = a(7);     q = a(8);     r = a(9);
phi = a(10);  theta = a(11); psi = a(12);

dele = CS_DEF(1);
dela = CS_DEF(2);
delr = CS_DEF(3);

rho = DENSITY(-z);

Vinf = sqrt(u*u + v*v + w*w);
alpha = atan2(w,u);
beta = asin(v/Vinf);

p_hat = p*b/(2*Vinf);
q_hat = q*cbar/(2*Vinf);
r_hat = r*b/(2*Vinf);

CL = CL_0 + CL_alpha*alpha + CL_q*q_hat + CL_de*dele;
CD = CD_0 + k*CL^2;
Cm = CM_0 + CM_alpha*alpha + CM_q*q_hat + CM_de*dele;

Cy = CY_0 + CY_beta*beta + CY_p*p_hat + CY_r*r_hat + CY_delr*delr;
Cl = Cl_0 + Cl_beta*beta + Cl_p*p_hat + Cl_r*r_hat + Cl_delr*delr + Cl_dela*dela;
Cn = Cn_0 + Cn_beta*beta + Cn_p*p_hat + Cn_r*r_hat + Cn_delr*delr;

Cx = CL*sin(alpha) - CD*cos(alpha);
Cz = -CL*cos(alpha) + CD*sin(alpha);

Fx = 0.5*rho*(Vinf^2)*S*Cx;
Fy = 0.5*rho*(Vinf^2)*S*Cy;
Fz = 0.5*rho*(Vinf^2)*S*Cz;

l = 0.5*rho*(Vinf^2)*S*Cl*b;
M = 0.5*rho*(Vinf^2)*S*Cm*cbar;
N = 0.5*rho*(Vinf^2)*S*Cn*b;

end
```

**>> RK4.m**

```
%% RK4

function [t,y] = RK4(dydt, tf, h, y0)     % dydt means derivative states

% Converted to column vector
y0 = reshape(y0, [], 1);

n = tf/h;
d = size(y0,1);

t = zeros(n,1);
y = zeros(n,d);

tn = 0;
yn = y0;

t(1,:) = tn;
y(1,:) = yn';

for i = 2:n+1

k1 = dydt(tn, yn);
k2 = dydt(tn + h/2, yn + h*k1/2);
k3 = dydt(tn + h/2, yn + h*k2/2);
k4 = dydt(tn + h, yn + h*k3);

yn = yn + (k1 + 2*k2 + 2*k3 + k4)*h/6;
tn = tn + h;

t(i) = tn;
y(i,:) = yn';

end

end
```

>> CS_DEF.m

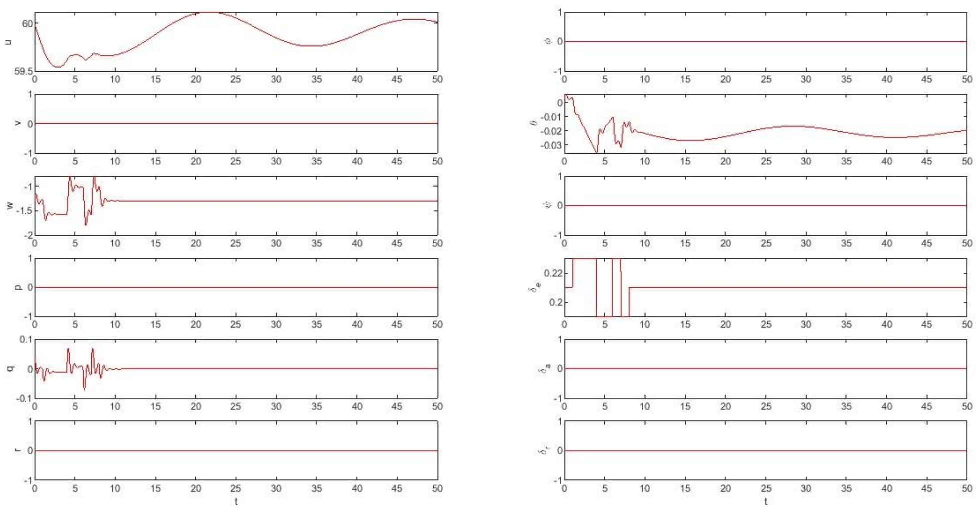
```
function control_inputs = CS_DEF(t,~)

thrust = 869.4045;
dele_trim = 0.2105;   % rad
dela_trim = 0;          % rad
delr_trim = 0;          % rad

% 3-2-1-1 inputs
if t < 1
    offset = 0; % rad
elseif t < 4
    offset = 0.02; % rad
elseif t < 6
    offset = -0.02; % rad
elseif t < 7
    offset = 0.02; % rad
elseif t < 8
    offset = -0.02; % rad
else
    offset = 0; % rad
end

% doublet inputs
if t < 1
    offset = 0; % rad
elseif t < 3
    offset = 0.02; % rad
elseif t < 5
    offset = -0.02; % rad
else
    offset = 0; % rad
end

% sinusoidal inputs
offset = 0.02*sin(pi/2*t); % rad

% dele = dele_trim;
dele = dele_trim + offset;
% dela = dela_trim;
dela = dela_trim + offset;
% delr = delr_trim;
delr = delr_trim + offset;

control_inputs = [dele, dela, delr, thrust];

end
```

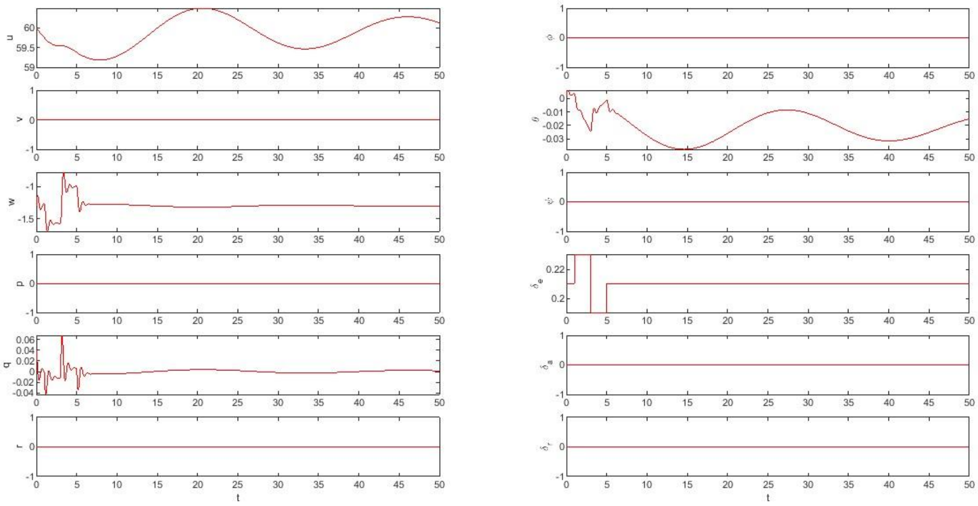
# NOTE (To get all the plots)

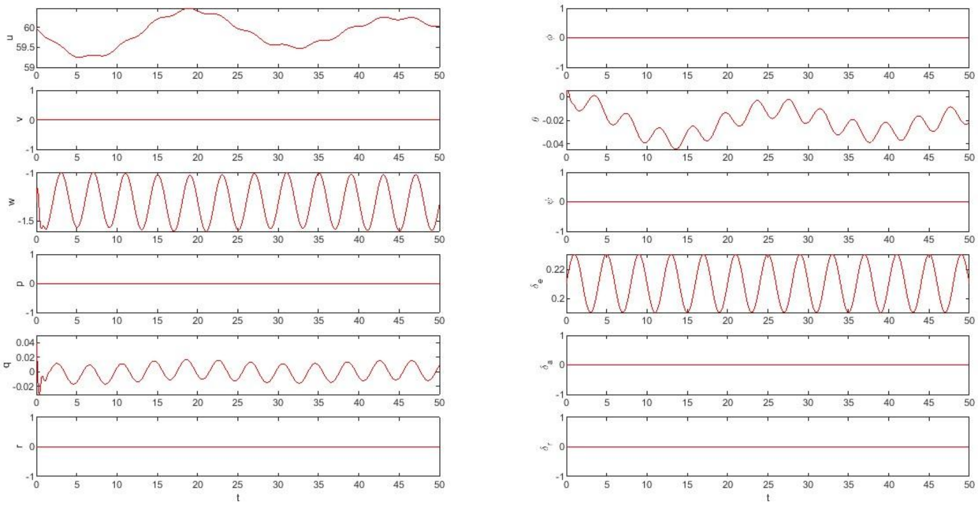
1. To get plot of (a) use control of 3-2-1-1, comment lines of other two inputs. Also run “dele=dele_trim + offset”, “dela = dela_trim” & “delr = delr_trim”. Then repeat for separate inputs.
2. By commenting & Uncommenting of MATLAB code lines we get can all plots.

# Output (PLOTS)

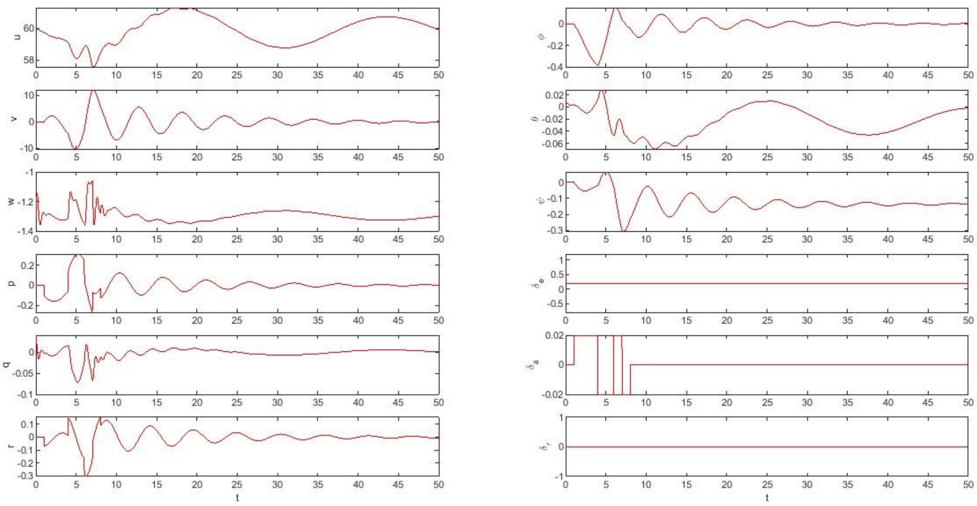
**(a) Aircraft dynamics by giving 3-2-1-1, doublet and sinusoidal inputs separately to elevator.**

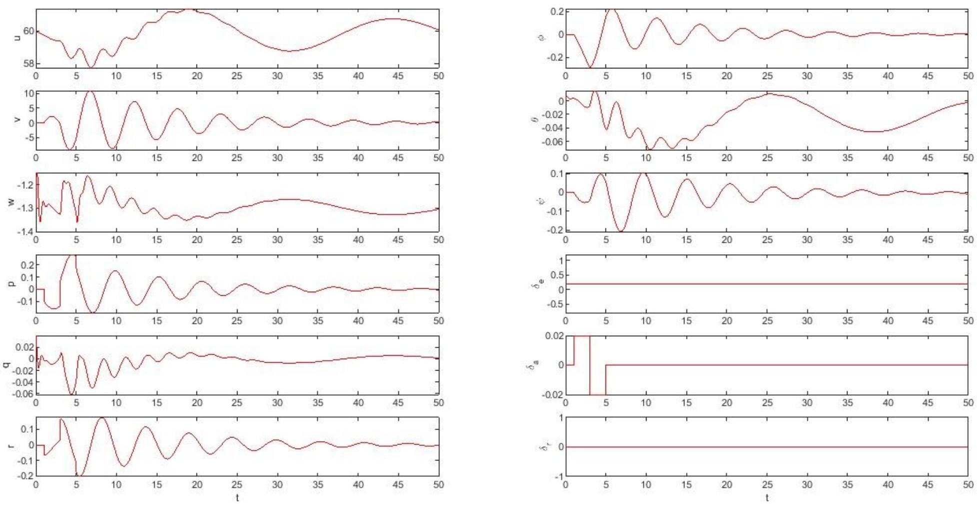


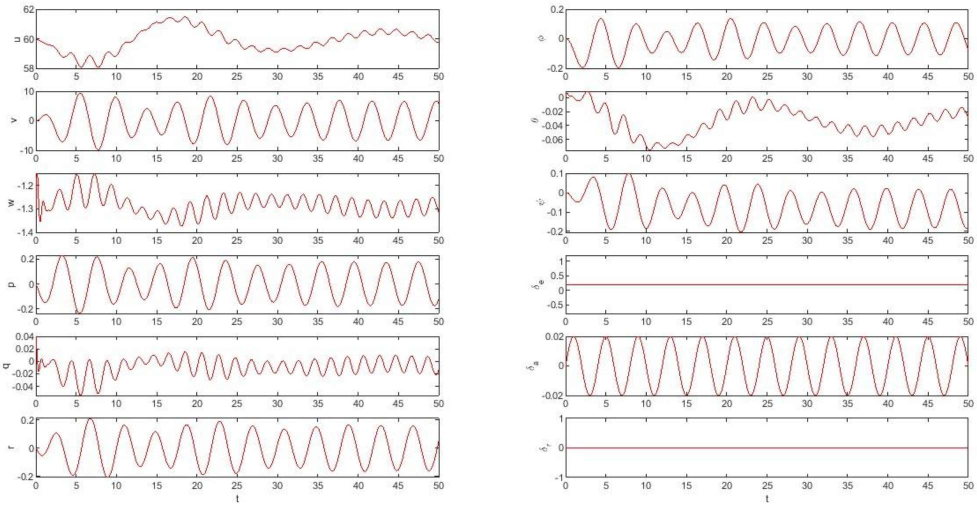

u
v
w
p
q
r
t
θ
t


u
v
w
p
q
r
t
θ
t

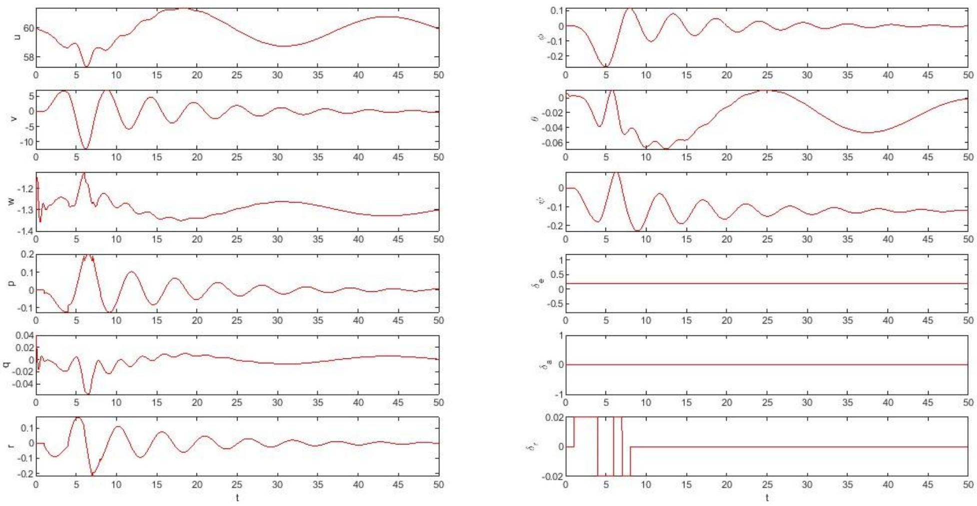
(b) Aircraft dynamics by giving 3-2-1-1, doublet and sinusoidal inputs separately to aileron.

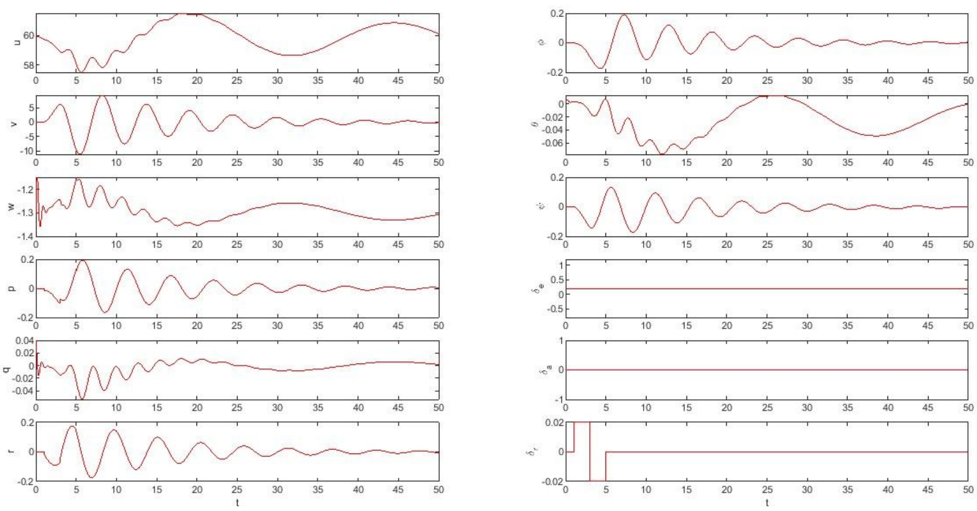


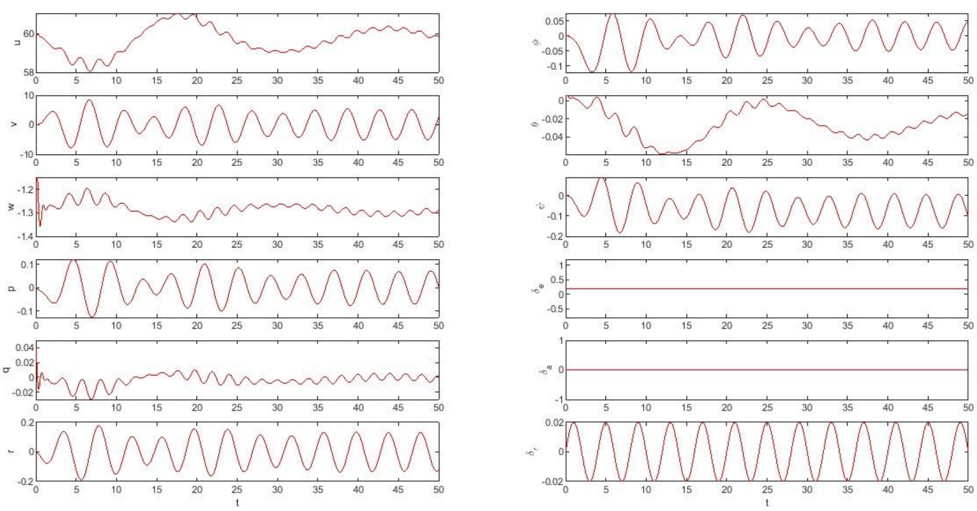




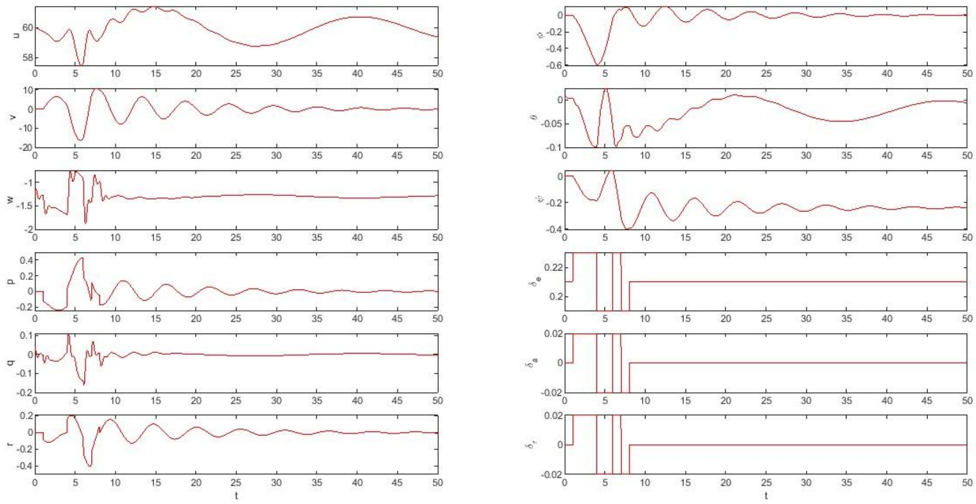
(c) Aircraft dynamics by giving 3-2-1-1, doublet and sinusoidal inputs separately to rudder.



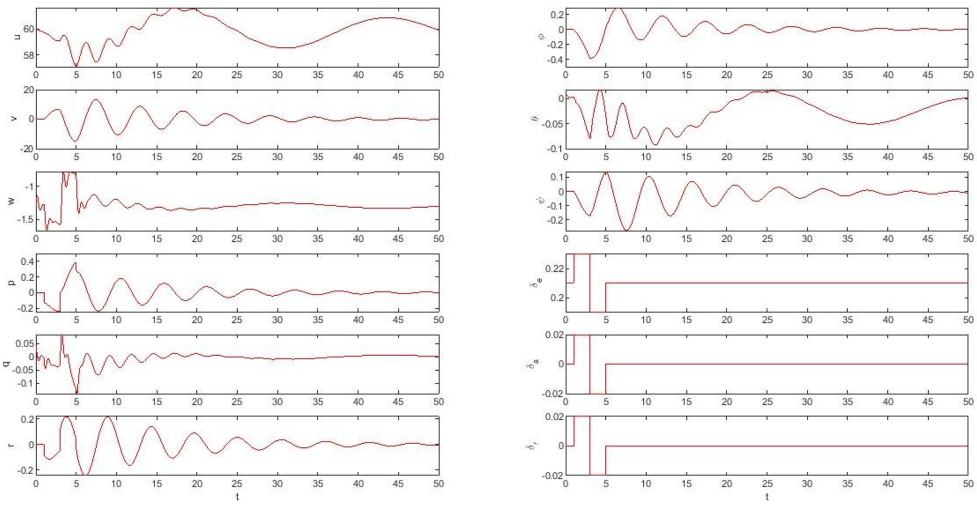



(e) Aircraft dynamics by giving 3-2-1-1 type of input to elevator, aileron, and rudder simultaneously.



(f) Aircraft dynamics by giving doublet type of input to elevator, aileron, and rudder simultaneously.



(g) Aircraft dynamics by giving sinusoidal type of input to elevator, aileron, and rudder simultaneously.