1_f18full_DUopenloop Model Tutorial

directory:
C:\ …. \Documents\MATLAB\F18_model_Teach\2024Summer_R2015b\1_F18full_DUopenloop_R2015b

this folder includes:

1. f18full_DUopenloop_run1.m
2. f18full_DUopenloop.mdl
   f18full_DUopenloop.slx
3. F18full_DUopenloop_plot_run2.m
   or F18_Sim_plot_2by2_run2
   or F18_Sim_plot_5by2_run2
4. F18_Animation_run3.m
5. F18_Sim_animateFG4.mdl

The sequence to run the simulation is 1, which will call 2,
3, which will plot 4 or 10 figures,
4, which will call 5 to show FlightGear animation

FlightGear video animation Setup (skip this part if FlightGear is not used)

1. Install FlightGear software
2. Extract the f18.zip **file to a folder named f18 an=d move it to FlightGear \data\Aircraft folder**
3. Run runfgWindowsXP_F18.bat

C:
cd C:\Program Files\FlightGear_xp
SET FG_ROOT=C:\Program Files\FlightGear_xp\\data
.\\bin\\win32\\fgfs --aircraft=f18 --fdm=network,localhost,5501,5502,5503 --altitude=10 --enable-freeze --timeofday=noon --enable-horizon-effect

to pop up a FlightGear window on the screen.

4. Pull down View manual and click View Options to uncheck all except Chase View Without Yaw
   Press "v" to change View Options
   Press "x" or "Shift x" to zoom in or zoom out.
   Pull down View manual and click Adjust View Position.
   Choose Left/Right: 25.00m, Down/up: 2.00m, Fwd/Back: -25.00m
5. Set enableFG=1 and enableRT=1 to enable the FlightGear and Real-Time mode in the two m files:

   In case that the FlightGear is not used, reset them to zero.

```
There is one equilibrium at
V=436 ft/s,   beta=0,    alpha = 10 deg
P=0,   q = 0 deg/s,    r=0
Phi=-360 deg = 0 deg,   theta= 10 deg,
psi=107 deg
elevator = -1.26 deg,   thrust = 5471 lbf,   ailron = 0,   rudder = 0

Choose Initial Condition:  Init_Dramatic
and control inputs: Thrust=5470.5 lbf,  Ele=-1.26 deg,  Ail=0, Rud=0
```

```
Then run the following sequence of programs:
%   1. Use the dramatic initial condition, Init_Dramatic
%       when the initial states are dramatic.
       %{
        V       =  350;         % Airspeed , ft/s
        beta    =  20*d2r;       % Sideslip Angle, rad
        alpha   =  40*d2r;       % Angle-of-attack, rad

        p       =  10*d2r;       % Roll rate, rad/s
        q       =  0*d2r;       % Pitch rate, rad/s
        r       =  5*d2r;       % Yaw rate, rad/s

        phi     =  0*d2r;       % Roll Angle, rad
        theta   =  0*d2r;       % Pitch Angle, rad
        psi     =  0*d2r;       % Yaw Angle, rad

        pN      =  0;           % Position North,  ft
        pE      =  0;           % Position East,  ft
        h       =  h0;           % Altitude,  ft
       %}
%   2. Use the following control inputs
       %{
        Con.T=5470.5;
        Con.elev=-0.022; %in rad
        Con.ail=0;
        Con.rud=0;
       %}
%   3. Choose the simulation time to be
       % sim_time = 1000 % Simulink simulation time 1000s
%   4. Run the program f18full_DUopenloop_run1.m
%   5. Run the program F18full_DUopenloop_plot_run2.m to observe the
%       waveforms and the steady-state response
%   6. Rerun f18full_DUopenloop_run1.m with sim_time=30
%   7. Get FlightGear ready for animation
%   8. Run the animation program, F18_Animation_run3.m
%       and observe the flight of the aircraft in real time
```

Manual Control Simulations

Run

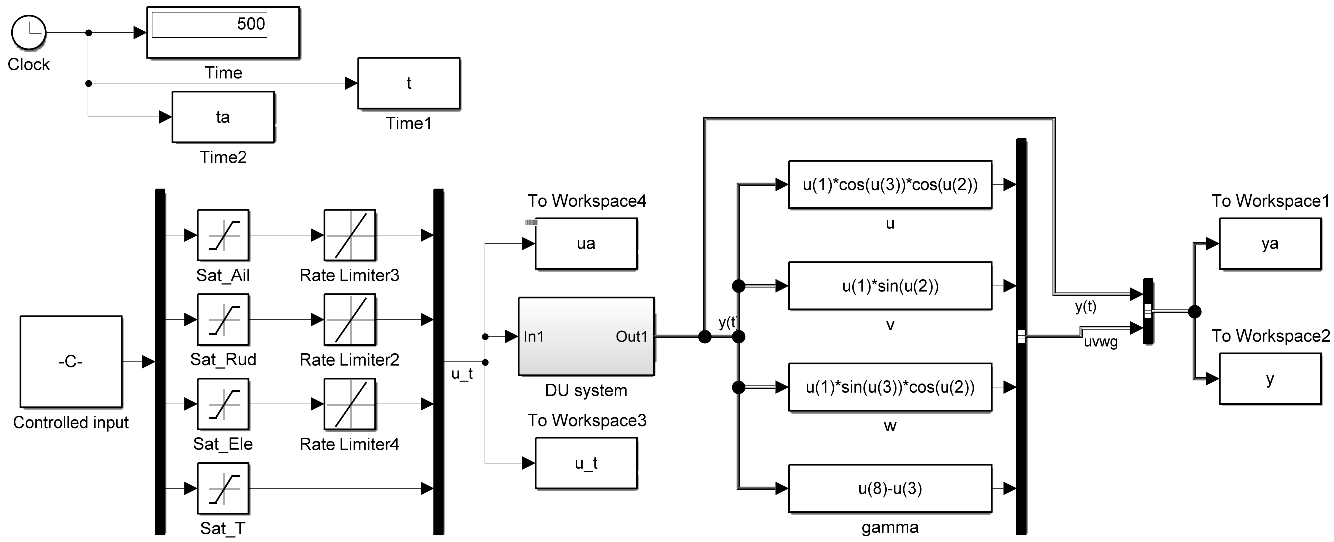
```
% Filename: f18full_DUopenloop_run1.m
% Objectives:
%   This program is to observe the open-loop behavior of the flight dynamics
%   model of f18full_DUopenloop.mdl.
% Simulation 1:
%   1. Use the dramatic iniial condition, Init_Dramatic
%      when the initial states are dramatic.
       %{
        V       =  350;         % Airspeed , ft/s
        beta    =  20*d2r;       % Sideslip Angle, rad
        alpha   =  40*d2r;       % Angle-of-attack, rad

        p       =  10*d2r;       % Roll rate, rad/s
        q       =  0*d2r;       % Pitch rate, rad/s
        r       =  5*d2r;       % Yaw rate, rad/s

        phi     =  0*d2r;       % Roll Angle, rad
        theta   =  0*d2r;       % Pitch Angle, rad
        psi     =  0*d2r;       % Yaw Angle, rad

        pN      =  0;           % Position North,  ft
        pE      =  0;           % Position East,  ft
        h       =  h0;           % Altitude,  ft
       %}
%   2. Use the following control inputs
       %{
        Con.T=5470.5;
        Con.elev=-0.022; %in rad
        Con.ail=0;
        Con.rud=0;
       %}
%   3. Choose the simulation time to be
       % sim_time = 1000 % Simulink simulation time 1000s
%   4. Run the program f18full_DUopenloop_run1.m
%   5. Run the program F18full_DUopenloop_plot_run2.m to observe the
%      waveforms and the steady-state response
%   6. Rerun f18full_DUopenloop_run1.m with sim_time=30
%   7. Get FlightGear ready for animation
%   8. Run the animation program, F18_Animation_run3.m
%      and observe the flight of the aircraft in real time


close all;clear all;clc;


% Reset enableFG and enableRT to 0 if FlightGear is not used
enableFG    = 1;
enableRT    = 1;
%enableFG    = 0;
%enableRT    = 0;


% Unit Conversion : Degree <--> Radian
d2r = pi/180;
r2d = 1/d2r;
```

```
h0=25000;
%h0=18000;% for flightgear only
%===========================================================================
% F18 data
%
% Aircraft Physical Paramters
% Reference: S. B. Buttrill and P. D. Arbuckle  and K. D. Hoffler
%            Simulation model of a twin-tail, high performance airplane
%            NASA 1992, NASA TM-107601

S = 400;                % ft^2
b =  37.4;                % ft
c =  11.52;             % ft
rho = 1.0660e-003;      % slugs/ft^3  --- 25C / 25000 ft
Ixx = 23000;            % slugs*ft^2
Iyy = 151293;           % slugs*ft^2
Izz = 169945;           % slugs*ft^2
Ixz = - 2971;           % slugs*ft^2
m = 1034.5;             % slugs
g = 32.2;               % ft/s^2
K=Ixx*Izz-Ixz^2;


%===========================================================================
% load aerodynamics coefficents
% the aerodynamic data is resulted from the Minnesota data rev2.
load aerodynamics_coefficients_rev2

%===========================================================================
% Initialization
% the initial states/conditions are dramatic (caused by turbulence).

V       =  350;         % Airspeed , ft/s
beta    =  20*d2r;       % Sideslip Angle, rad
alpha   =  40*d2r;       % Angle-of-attack, rad

p       =  10*d2r;       % Roll rate, rad/s
q       =  0*d2r;       % Pitch rate, rad/s
r       =  5*d2r;       % Yaw rate, rad/s

phi     =  0*d2r;       % Roll Angle, rad
theta   =  0*d2r;       % Pitch Angle, rad
psi     =  0*d2r;       % Yaw Angle, rad

pN      =  0;           % Position North,  ft
pE      =  0;           % Position East,  ft
h       =  h0;           % Altitude,  ft

% Stack Initial Condition for State
x_init = [V;beta;alpha;p;q;r;phi;theta;psi;pN;pE;h];
x = x_init;

%===========================================================================
% % Controlled input setup

Con.T=5470.5;
Con.elev=-0.022; %in rad
Con.ail=0;
Con.rud=0;
```

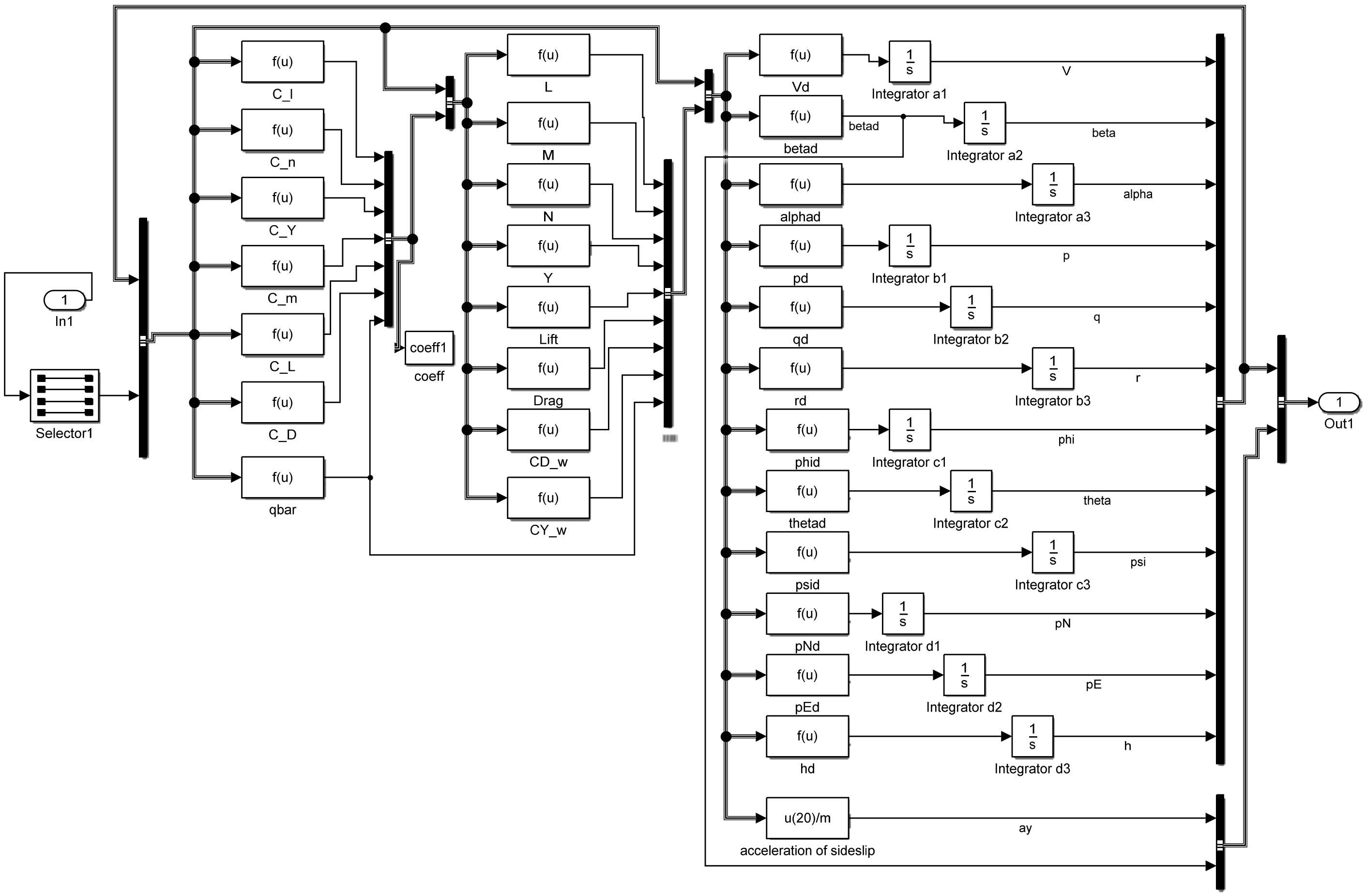
```
%=========================================================================
% Using the m-file to run the simulink

%sim_time = 100 % Simulink simulation time
sim_time = 500 % Simulink simulation time

sim_options = simset('SrcWorkspace ', 'current',...
    'DstWorkspace ', 'current');
open('f18full_DUopenloop')
sim('f18full_DUopenloop', [0, sim_time], sim_options);  %[t,x,y] =
sim(model,timespan,options,ut);
```

The following mdl simulink program will automatically called and run until the simulation is done.

f18full_DUopenloop.mdl:



F18full_DUopenloop/DU system



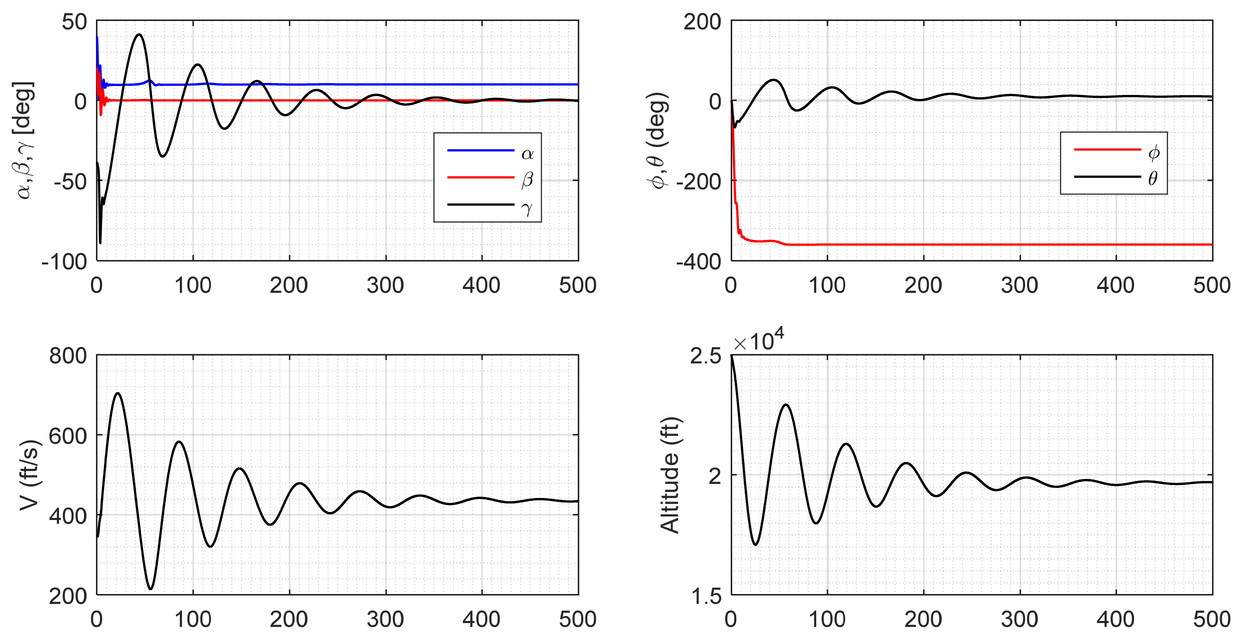
The simulation will generate simulation results in the work place. DO NOT delete them!

We can use the following program to print and plot the results.

```
% Filename: F18_Sim_plot_2by2.m
%
%==========================================================================
% Plot Results
d2r = pi/180;
r2d = 1/d2r;
disp('Final state values')
Final_State_Value=[ya(end,1) ;ya(end,2)*r2d;ya(end,3)*r2d;ya(end,4)*r2d;ya(end,5)*r2d;ya(
end,6)*r2d;ya(end,7)*r2d;ya(end,8)*r2d;ya(end,9)*r2d;ya(end,10);ya(end,11);ya(end,12)];
% %
% tsim=1821 ; % t=200s
%
Final_State_Value=[ya(tsim,1);ya(tsim,2)*r2d;ya(tsim,3)*r2d;ya(tsim,4)*r2d;ya(tsim,5)*r2d
;ya(tsim,6)*r2d;ya(tsim,7)*r2d;ya(tsim,8)*r2d;ya(tsim,9)*r2d;ya(tsim,10);ya(tsim,11);ya(t
sim,12)];

State_Unit=
{'ft/s';'deg';'deg';'deg/s';'deg/s';'deg/s';'deg';'deg';'deg';'ft';'ft';'ft'};
State_Names={'V';'beta';'alpha';'p';'q';'r';'phi';'theta';'psi';'pN';'pE';'Altitute'};
FinalState=table(Final_State_Value,State_Unit,'RowNames',State_Names)  % make table
```

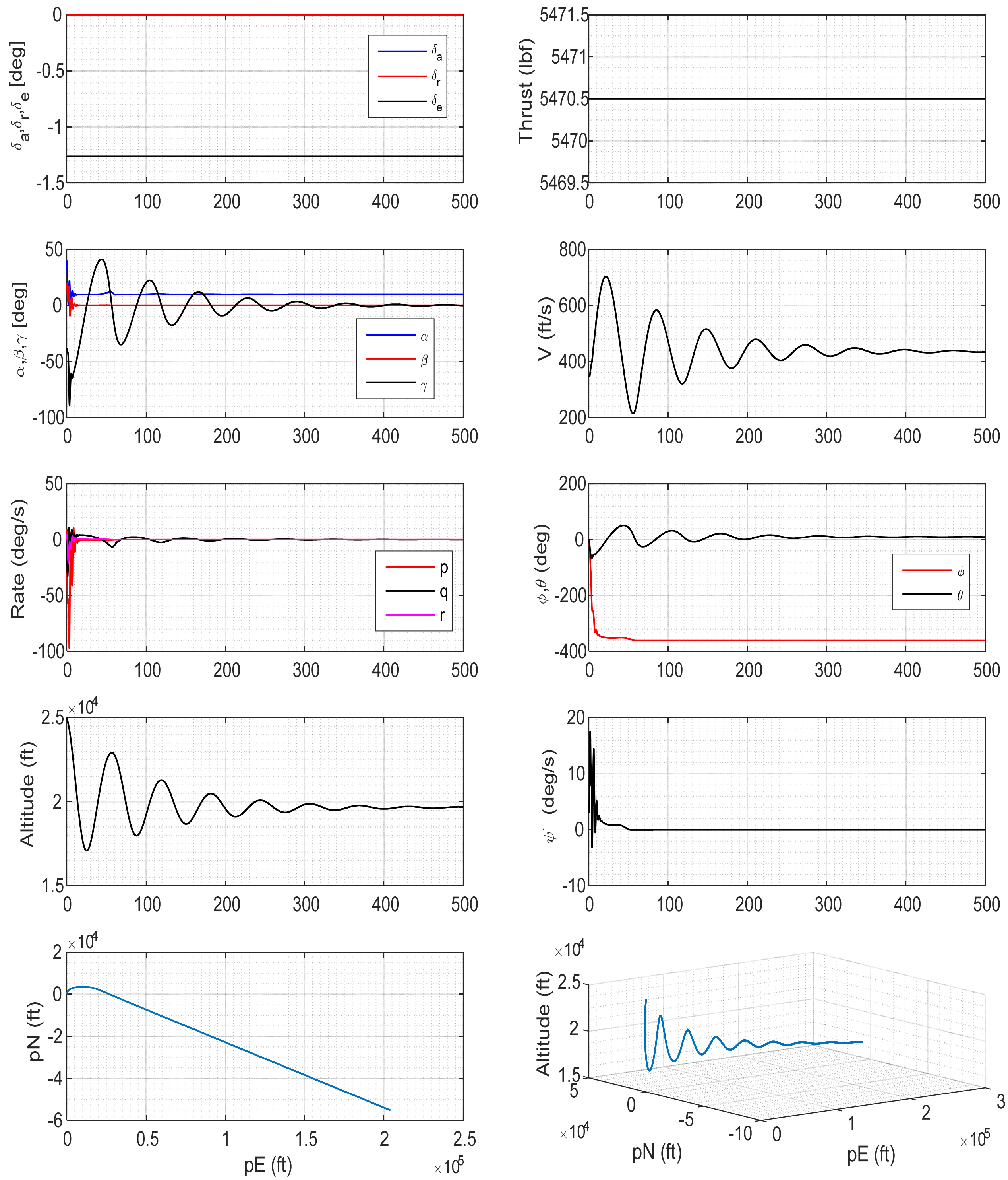
```
disp('Final input values')
Final_Input_Value=[ua(end,1)*r2d;ua(end,2)*r2d;ua(end,3)*r2d;ua(end,4)];
State_Unit1=  {'deg';'deg';'deg';'lbf'};
State_Names1={'Aileron';'Rudder';'Elevator'; 'Thrust' };
FinalInput=table(Final_Input_Value,State_Unit1,'RowNames',State_Names1)  % make table

% disp('Final input values')
%
Final_Input_Value=[ua_command(tsim,1)*r2d;ua_command(tsim,2)*r2d;ua_command(tsim,3)*r2d;u
a_command(tsim,4)];
% State_Unit1=  {'deg';'deg';'deg';'lbf'};
% State_Names1={'Aileron';'Rudder';'Elevator'; 'Thrust' };
% FinalInput=table(Final_Input_Value,State_Unit1,'RowNames',State_Names1)  % make table

%%
pr=1:size(ta)/1;  %print size

figure (2)

subplot(2,2,1)
plot (ta(pr),(180/pi)*ya(pr,3),'b-',ta(pr),(180/pi)*ya(pr,2),'r-
',ta(pr),ya(pr,18)*r2d,'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\alpha,\beta,\gamma [deg]')
legend ('\alpha','\beta','\gamma')
hold on

subplot(2,2,2)
plot (ta(pr),(180/pi)*ya(pr,7),'r-',ta(pr),(180/pi)*ya(pr,8),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\phi,\theta (deg)')
legend ('\phi','\theta')
hold on

subplot(2,2,3)
plot (ta(pr), ya(pr,1),'k','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('V (ft/s)')
hold on

subplot(2,2,4)
plot (ta(pr), ya(pr,12),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('Altitude (ft)')
hold on
```

```
sim_time =
   500
>> F18_Sim_plot_2by2_run2
Final state values
FinalState =
                Final_State_Value    State_Unit
                _________________    __________

    V                434.27          'ft/s'
    beta        -8.5401e-13          'deg'
    alpha            10.001          'deg'
    p            4.2238e-12          'deg/s'
    q             -0.031256          'deg/s'
    r           -4.2607e-12          'deg/s'
    phi                -360          'deg'
    theta            9.7462          'deg'
    psi              107.27          'deg'
    pN               -55217          'ft'
    pE           2.0409e+05          'ft'
    Altitute          19690          'ft'
Final input values
FinalInput =
                Final_Input_Value    State_Unit1
                _________________    ___________

    Aileron            0             'deg'
    Rudder             0             'deg'
    Elevator     -1.2605             'deg'
    Thrust        5470.5             'lbf'
>>
```



```
% Filename: F18_Sim_plot_5by2_run2.m
%
%=============================================================================
% Plot Results
d2r = pi/180;
r2d = 1/d2r;
disp('Final state values')
Final_State_Value=[ya(end,1) ;ya(end,2)*r2d;ya(end,3)*r2d;ya(end,4)*r2d;ya(end,5)*r2d;ya(
end,6)*r2d;ya(end,7)*r2d;ya(end,8)*r2d;ya(end,9)*r2d;ya(end,10);ya(end,11);ya(end,12)];
% %
% tsim=1821 ; % t=200s
%
Final_State_Value=[ya(tsim,1);ya(tsim,2)*r2d;ya(tsim,3)*r2d;ya(tsim,4)*r2d;ya(tsim,5)*r2d
;ya(tsim,6)*r2d;ya(tsim,7)*r2d;ya(tsim,8)*r2d;ya(tsim,9)*r2d;ya(tsim,10);ya(tsim,11);ya(t
sim,12)];

State_Unit=
{'ft/s';'deg';'deg';'deg/s';'deg/s';'deg/s';'deg';'deg';'deg';'ft';'ft';'ft'};
State_Names={'V';'beta';'alpha';'p';'q';'r';'phi';'theta';'psi';'pN';'pE';'Altitute'};
FinalState=table(Final_State_Value,State_Unit,'RowNames',State_Names)  % make table

disp('Final input values')
Final_Input_Value=[ua(end,1)*r2d;ua(end,2)*r2d;ua(end,3)*r2d;ua(end,4)];
State_Unit1=  {'deg';'deg';'deg';'lbf'};
State_Names1={'Aileron';'Rudder';'Elevator'; 'Thrust' };
FinalInput=table(Final_Input_Value,State_Unit1,'RowNames',State_Names1)  % make table

% disp('Final input values')
%
Final_Input_Value=[ua_command(tsim,1)*r2d;ua_command(tsim,2)*r2d;ua_command(tsim,3)*r2d;u
a_command(tsim,4)];
% State_Unit1=  {'deg';'deg';'deg';'lbf'};
% State_Names1={'Aileron';'Rudder';'Elevator'; 'Thrust' };
% FinalInput=table(Final_Input_Value,State_Unit1,'RowNames',State_Names1)  % make table

%%
pr=1:size(ta)/1;  %print size

figure (3)
subplot(5,2,1)
%plot (ta,(180/pi)*ua(:,1),'b-',ta,(180/pi)*ua(:,2),'r-',ta,(180/pi)*ua(:,3),'k-
','LineWidth',1)
plot (ta(pr),(180/pi)*ua(pr,1),'b-',ta(pr),(180/pi)*ua(pr,2),'r-
',ta(pr),(180/pi)*ua(pr,3),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\delta_a,\delta_r,\delta_e [deg]')
legend ('\delta_a','\delta_r','\delta_e')
% axis([0 500 -11 1])
% axis([0 2000 -1.5 3.5])

subplot(5,2,2)
plot (ta(pr),ua(pr,4),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('Thrust (lbf)')
% axis([0 2000 1000 6000])

subplot(5,2,3)
plot (ta(pr),(180/pi)*ya(pr,3),'b-',ta(pr),(180/pi)*ya(pr,2),'r-
',ta(pr),ya(pr,18)*r2d,'k-','LineWidth',1)
```

```
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\alpha,\beta,\gamma [deg]')
legend ('\alpha','\beta','\gamma')
% axis([0 2000 -25 25])

subplot(5,2,4)
plot (ta(pr), ya(pr,1),'k','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('V (ft/s)')
% axis([0 2000 400 850])

subplot(5,2,5)
plot(ta(pr),ya(pr,4)*r2d,'-r', ta(pr),ya(pr,5)*r2d,'-k', ta(pr),ya(pr,6)*r2d,'-
m','LineWidth',1)
grid on, grid minor
%xlabel('time, sec')
ylabel('Rate (deg/s)')
legend('p','q','r')
% axis([0 2000 -8 2])

subplot(5,2,6)
plot (ta(pr),(180/pi)*ya(pr,7),'r-',ta(pr),(180/pi)*ya(pr,8),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\phi,\theta (deg)')
legend ('\phi','\theta')
% axis([0 2000 -20 30])

subplot(5,2,7)
plot (ta(pr), ya(pr,12),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('Altitude (ft)')
% axis([0 2000 15000 35000])

subplot(5,2,8)
yaw_angle_rate=(([ya(:,5)].*[sin(ya(:,7))]+[ya(:,6)].*[cos(ya(:,7))]).*[sec(ya(:,8))])*r2
d;
plot(ta(pr),yaw_angle_rate(pr),'k-','LineWidth',1)
grid on, grid minor
%xlabel ('Time (sec)')
ylabel ('\psi^\cdot  (deg/s)')
% axis([0 2000 -0.1 0.55])

subplot(5,2,9)
plot(ya(pr,11),ya(pr,10),'LineWidth',1)
grid on, grid minor
xlabel('pE (ft)')
ylabel('pN (ft)')

subplot(5,2,10)
plot3(ya(pr,11),ya(pr,10),ya(pr,12),'LineWidth',1)
xlabel('pE (ft)')
ylabel('pN (ft)')
zlabel('Altitude (ft)')
grid on, grid minor
```



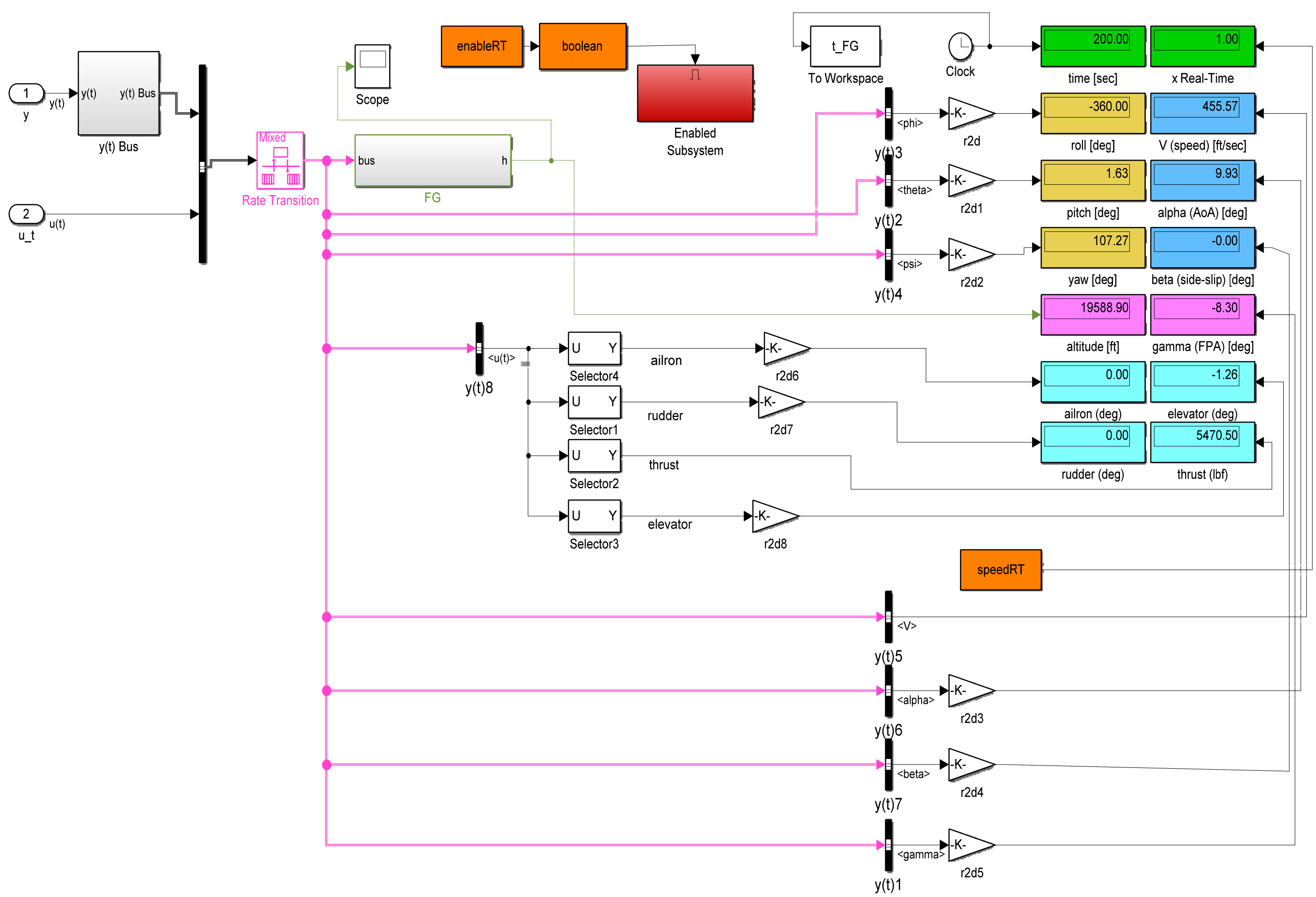
$\delta_a, \delta_r, \delta_e$ [deg]
δ_a
δ_r
δ_e
Thrust (lbf)
α, β, γ [deg]
α
β
γ
V (ft/s)
Rate (deg/s)
p
q
r
ϕ, θ (deg)
ϕ
θ
Altitude (ft)
$\dot{\psi}$ (deg/s)
pN (ft)
pE (ft)

Rerun f18full_DUopenloop_run1.m with sim_time=200

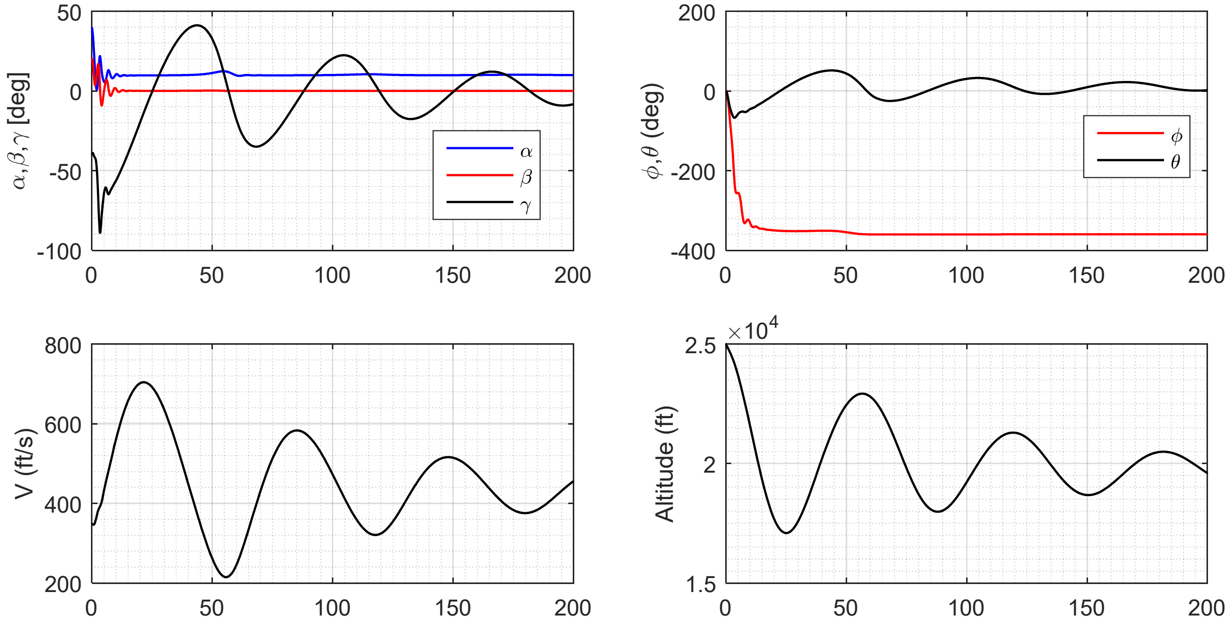
```
>> sim_time =
     200
>>
```

Run F18_Animation_run3.m
Which will open

F18_Sim_animateFG4.mdl




```
>> F18_Sim_plot_2by2_run2
Final state values
FinalState =
                Final_State_Value    State_Unit
                _________________    __________

    V                 455.57         'ft/s'
    beta         -3.8818e-06         'deg'
    alpha             9.9281         'deg'
    p             1.7805e-05         'deg/s'
    q                0.37368         'deg/s'
    r            -2.2428e-05         'deg/s'
    phi                 -360         'deg'
    theta             1.6293         'deg'
    psi               107.27         'deg'
    pN                -16282         'ft'
    pE                 78885         'ft'
    Altitute           19589         'ft'
Final input values
FinalInput =
                Final_Input_Value    State_Unit1
                _________________    ___________

    Aileron             0            'deg'
    Rudder              0            'deg'
    Elevator      -1.2605            'deg'
    Thrust         5470.5            'lbf'
```



F18_Sim_animateFG4/y(t) Bus