```
%% Phugoid flight simulation (2DoF), Simulación de vuelo fugoide (2GL).
% (c) 2023, Antonio Sala Piqueras. All rights reserved.

%% Model
%parameters below are arbitrarily invented, not thinking in any actual aircraft.
Drag_c=0.05;  Lift_c=1; g=9.8; m=1; %numeric values for constant params.
Thrust=0.05*m*g;  %Thrust may depend on airspeed (propeller engines, particularly)
StateEq=@(v,theta) [-g*sin(theta)-Drag_c*v^2*sign(v)/m+Thrust/m; %dv/dt
                    -g*cos(theta)/v+Lift_c*v/m; %dtheta/dt
                    v*cos(theta); %dx/dt
                    v*sin(theta)]; %dy/dt

%% Simulation
v0=5;  %initial airspeed
Tout=0:0.05:20; NT=length(Tout);
ODEfun=@(t,x) StateEq(x(1),x(2)) ; %x, y do not influence state derivatives.
x0=[v0; 6*pi/180; 0; 10]; %initial [speed, pitch, x, altitude] vector.
[Tout,StatesODE]=ode45(ODEfun,Tout,x0); %simulation with default tolerances.

%% Plotting results
figure(1)
yyaxis left
plot(Tout,StatesODE(:,1),LineWidth=2), grid on,
yyaxis right
plot(Tout,StatesODE(:,2)*180/pi,LineWidth=2)
legend("v (airspeed)","\theta (def)",Location="best")
ax = gca; ax.FontSize = 24;
xlabel("Time")
acceleratio=zeros(NT,1);
accelerationN=zeros(NT,1);
liftforce=zeros(NT,1);
for k=1:NT
   tmp=StateEq(StatesODE(k,1),StatesODE(k,2));
   acceleratio(k)=norm([tmp(1); StatesODE(k,1)*tmp(2)]); % [dv/dt; v*dtheta/dt]
   accelerationN(k)=StatesODE(k,1)*tmp(2); %Normal accel. only
   liftforce(k)=Lift_c*StatesODE(k,1)^2; %Lift exerted by wings
end

figure(3)
plot(Tout,[acceleratio/g],LineWidth=4), grid on,
hold on
plot(Tout,[accelerationN/g liftforce/m/g],LineWidth=2), hold off
yline(1,':r'), legend("Acceleration (g) [modulus]","Normal accel (g) [signed]","Lift/
Weight", Location="best")
ax = gca; ax.FontSize = 24; xlabel("Time")

%% Animation
Outputs=StatesODE(:,[3 4 2]); %x y theta selected (output equation)
Minxy=min(Outputs(:,[1 2])); %for axis scaling
Maxxy=max(Outputs(:,[1 2])); %for axis scaling
RotM=@(theta) [cos(theta) -sin(theta);sin(theta) cos(theta)]; %Rotation matrix
plane=[-0.5608   -0.5608   -0.5281   -0.4234   -0.3891   -0.3842   -0.2795   -0.1569
-0.0784    0.1522    0.4090    0.7671    0.8063    0.7049    0.4335    0.1522 -0.0947
-0.2844   -0.5475   -0.5591   -0.6237;
   -0.0021    0.0666    0.2154    0.1958    0.0597    0.0240    0.0094    0.0105
0.0241    0.0454    0.0437   -0.0233   -0.0544   -0.0969   -0.1231   -0.1313    -0.1149
-0.0806   -0.0659   -0.0430   -0.0119];
 wing=[0.0205    0.2551    0.4492    0.2129;
   -0.0356   -0.0137   -0.0204   -0.037];

for i=1:length(Tout)
    x_cog=Outputs(i,1);
    y_cog=Outputs(i,2);
    th=Outputs(i,3);
    planeRT=RotM(th)*plane+[x_cog;y_cog]; %rotate+translate plane outline
    wingRT=RotM(th)*wing+[x_cog;y_cog-.04];%rotate+translate wing outline
    figure(2)
    clf
```

Video presentation [Spanish]:
http://personales.upv.es/asala/YT/V/fugsimcod.html http://personales.upv.es/asala/YT/V/fugsim.html
Video presentation [English]:
http://personales.upv.es/asala/YT/V/fugsimcodEN.html http://personales.upv.es/asala/YT/V/fugsimEN.html

```
    plot(Outputs(1:i,1),Outputs(1:i,2),'-.',LineWidth=2); %trajectory plot
    patch(planeRT(1,:),planeRT(2,:),'r') %plane plot
    patch(wingRT(1,:),wingRT(2,:),[0.8 0.75 0.65]) %wing outline plot
    xline(x_cog,':'), yline(y_cog,':') %highlight (x,y)
    axis equal
    axis([Minxy(1)-1 Maxxy(1)+1 Minxy(2)-1 Maxxy(2)+1]), grid on
    ax = gca;  ax.FontSize = 28;
    drawnow
end

%% Actual Glider loopings
% https://www.youtube.com/watch?v=aE2yOgMWdak from minute 1:00
% https://www.youtube.com/watch?v=CYziDr0wPOc minute 2:14
```