Ingeniería de Control
Práctica 2

Aircraft vertical takeoff and landing

Consider the simplified planar model of the system for vertical takeoff and landing of an aircraft represented in Figure 1, in which the aircraft is represented by a bar. The position of the center of mass of the aircraft, $\mathbf{c} = (x, y)^T$, the roll angle of the aircraft, θ, and their time derivatives are the state variables of the system. The thrust force S, applied to the center of mass of the aircraft, and the forces F, applied to the wing tips, are the control inputs u_1 and u_2 of the system, respectively. The thrust force S keeps the aircraft flying. The forces F, which always act in opposite directions, control the roll of the aircraft.

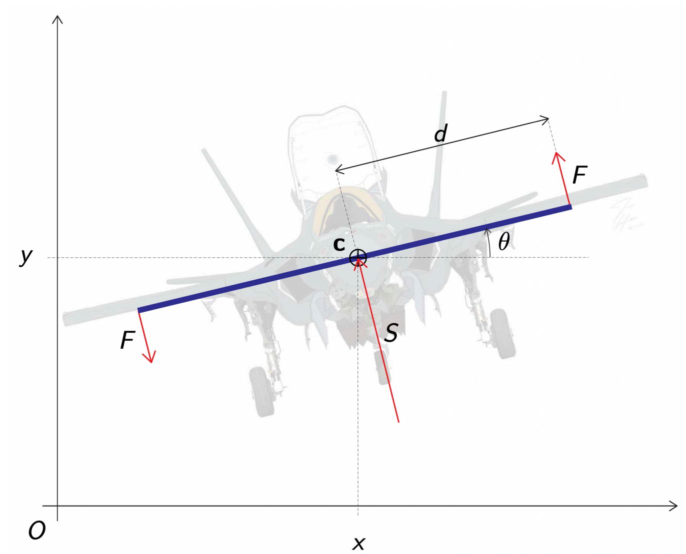

Figure 1: Sketch of the system for aircraft vertical takeoff and landing.

The dynamic model of this system is

$$
\begin{aligned}
\ddot{x} &= -\frac{1}{m}\sin(\theta)S, \\
\ddot{y} &= -g + \frac{1}{m}\cos(\theta)S, \\
\ddot{\theta} &= \frac{2d}{J}F,
\end{aligned}
$$

with the following parameters

- barycentric moment of inertia of the aircraft: $J = 10000$ [kg m^2],
- mass of the aircraft: $m = 30000$ [kg],
- $d = 5.5$ [m],
- gravity acceleration: $g = 9.81$ [m/s^2].

1) Demonstrate the equations of the dynamic model using the Lagrange method. (Copy the solution from Problem 1)

2) Calculate the state space representation of the system, assuming that $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (x_1, x_2, x_3, x_4, x_5, x_6)^T$, where distances are measured in [m], angles in [rad], linear velocities in [m/s], and angular velocities in [rad/s]. (Copy the solution from Problem 1)

3) Calculate all the operating points of the system and explain the obtained result.

4) Find the operating point that corresponds to $\overline{u}_1 = mg, \overline{u}_2 = 0$. Linearize the system around this operating point.

5) Is the linearized system controllable using both control inputs u_1 and u_2? Is the linearized system controllable using only the control input u_1?

6) Using the pole placement method, design a state feedback controller to control the landing of the aircraft. We want to steer the aircraft from the state $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (1, 5, 0.0174533, -0.1, -0.2, 0.00174533)^T$ to the state $\mathbf{x} = (0, 2.4, 0, 0, 0, 0)^T$. Give the eigenvalues that have been assigned to the controlled system and illustrate the behaviour of the controller by plotting the relevant state and control variables and by a graphical animation.

7) Using the pole placement method, design a state feedback controller to control a lateral dispacement of the aircraft. We want to steer the aircraft from the state $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (0, 5, 0.0174533, -0.1, -0.2, 0.00174533)^T$ to the state $\mathbf{x} = (10, 5, 0, 0, 0, 0)^T$. Give the eigenvalues that have been assigned to the controlled system and illustrate the behaviour of the controller by plotting the relevant state and control variables and by a graphical animation.

Write a detailed report answering each question in a different section.

Originality and completeness of the answers will be the aspects that will be taken into account in the grading of the report. Include the Matlab code in the report.

Additionally, upload the Matlab code to Aula Virtual. Upload the code of each answer in a separate folder.

Solution of Práctica 2 Daniel Parra Segovia

1) Demonstrate the equations of the dynamic model using the Lagrange method. (Copy the solution from Problem 1)

To demonstrate the equations of the dynamic model using the Lagrange, we follow these steps:

1. Find the kinetic energy equations of the system
   $K = mv^2/2$
   $K_x = m\dot{x}^2/2$
   $K_y = m\dot{y}^2/2$
   $K_\theta = J\dot{\theta}^2/2$
   $K_{total} = (m\dot{x}^2 + m\dot{y}^2 + J\dot{\theta}^2)/2$
2. Find the potential energy equations
   $P = mgh$
   $P = mgy$
3. Calculate the Lagrangian
   $\mathcal{L} = K - P = (m\dot{x}^2 + m\dot{y}^2 + J\dot{\theta}^2)/2 - mgy$
4. We generate the 3 Lagrange Equations of the system
   $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = -Ssin(\theta)$
   $m\ddot{x} = -Ssin(\theta)$
   $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{y}} - \frac{\partial \mathcal{L}}{\partial y} = Scos(\theta)$
   $m\ddot{y} + mg = Scos(\theta)$
   $\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\theta}} - \frac{\partial \theta}{\partial y} = 2dF$
   $J\ddot{\theta} = -2dF$
5. Simplify the equations
   $\ddot{x} = -Ssin(\theta)/m$
   $\ddot{y} = -g + Scos(\theta)/m$
   $\ddot{\theta} = -2dF/J$

2) Calculate the state space representation of the system, assuming that $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (x_1, x_2, x_3, x_4, x_5, x_6)^T$, where distances are measured in [m], angles in [rad], linear velocities in [m/s], and angular velocities in [rad/s]. (Copy the solution from Problem 1)

From the dynamic model, we can find the state space representation of the system

$\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (x_1, x_2, x_3, x_4, x_5, x_6)^T$
$\dot{\mathbf{x}} = (\dot{x}, \dot{y}, \dot{\theta}, \ddot{x}, \ddot{y}, \ddot{\theta})^T$

$$\dot{\mathbf{x}} = (x_4, x_5, x_6, 0, -g, 0)^T + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -sin(x_3)/m & 0 \\ cos(x_3)/m & 0 \\ 0 & 2d/J \end{pmatrix} * (u_1, u_2)^T$$

$y = x_2$

3) Calculate all the operating points of the system and explain the obtained result.

Using the state space representation of the system, we calculate the operating points solving for dx = [0 0 0 0 0 0]

Su = [g*m 0]

Sx = [0 0 0 0 0 0]

File answer_3.m

```
syms J m d g x1 x2 x3 x4 x5 x6 u1 u2
A=[x4;x5;x6;0;-g;0]

B=[0 0;
   0 0;
   0 0;
   -sin(x3)/m 0;
   cos(x3)/m 0;
   0 2*d/J]

u=[u1;u2];
dx=A+B*u;

S = solve(dx(1)==0,dx(2)==0,dx(3)==0,dx(4)==0,dx(5)==0,dx(6)==0, ...
    x1,x2,x3,x4,x5,x6,u1,u2,'Real',true);

Su=[S.u1;S.u2]
Sx=[S.x1;S.x2;S.x3;S.x4;S.x5;S.x6]
```

4) Find the operating point that corresponds to $\overline{u}_1 = mg, \overline{u}_2 = 0$. Linearize the system around this operating point.

Using the state space representation of the system, we linearize the system around the operating point [0 0 0 0 0 0] calculated earlier. To do that, we calculate the Jacobian and then substitute the values of the operating point.

A =

[0, 0, 0, 1, 0, 0]

[0, 0, 0, 0, 1, 0]

[0, 0, 0, 0, 0, 1]

[0, 0, -(2*u1)/m, 0, 0, 0]

[0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0]

B =

[0, 0]

[0, 0]

[0, 0]

[0, 0]

[2/m, 0]

[0, (2*d)/J]

As we can see, the matrix A still has u1. Later, we'll have to substitute u1 by the value of u1 around the operating point.

We calculated it earlier, u1 = g*m

File answer_4.m

```
syms J m d g x1 x2 x3 x4 x5 x6 u1 u2

A=[x4;x5;x6;-u1*(sin(x3)/m);-g+u1*(cos(x3))/m;0];

B=[0 0;
   0 0;
   0 0;
   -sin(x3)/m 0;
   cos(x3)/m 0;
   0 2*d/J];

u=[u1;u2];

dx=A+B*u;

A = jacobian(dx,[x1;x2;x3;x4;x5;x6]);
B = jacobian(dx,[u1;u2]);

A=subs(A,[x1 x2 x3 x4 x5 x6], [0 0 0 0 0 0])
B=subs(B,[x1 x2 x3 x4 x5 x6], [0 0 0 0 0 0])
```

5) Is the linearized system controllable using both control inputs u_1 and u_2? Is the linearized system controllable using only the control input u_1?

To figure out if the linearized system is controllable, we have to calculate the controllability matrix, it's obtained by juxtaposing the n matrices B, AB, ... , An-1.

We can use the function ctrb() to calculate the controllability matrix.

The system will be controllable if and only if the rank of the controllability matrix is equal to the dimension of x.

To check if the system is controllable without u2 we can recalculate the B matrix with u2=0.

we get that in the first case the system is controllable, and without u2, the system is uncontrollable, 4 of its states are uncontrollable.

This makes sense, because without u2 (without the forces that allow the plane to turn) we can't control the angle of the plane.

uncontrollable_states = 0

uncontrollable_states = 4

File answer_5.m

```
m=30000;
d=5.5;
g=9.81;
J=10000;
u1_A=m*g;

A=[0 0 0 1 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1;
   0 0 -u1_A/m 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];

B=[0 0;0 0;0 0;0 0;1/m 0; 0 2*d/J];

B_without_u2=[0 0;0 0;0 0;0 0;1/m 0; 0 0];

Co=ctrb(A,B);

uncontrollable_states = length(A) - rank(Co)

Co=ctrb(A,B_without_u2);

uncontrollable_states = length(A) - rank(Co)
```

6) Using the pole placement method, design a state feedback controller to control the landing of the aircraft. We want to steer the aircraft from the state $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (1, 5, 0.0174533, -0.1, -0.2, 0.00174533)^T$ to the state $\mathbf{x} = (0, 2.4, 0, 0, 0, 0)^T$. Give the eigenvalues that have been assigned to the controlled system and illustrate the behaviour of the controller by plotting the relevant state and control variables and by a graphical animation.

We used the linearized system to calculate the K, E, H using the pcontrol vector.

pcontrol are the poles that we'd like the system to have. All of them should be negative. The most important ones are pcontrol(1) and pcontrol(2). At first, we set them to values between -1 and -1.5, but the plane was turning too much, so we set the second one to -0.1, closer to 0. pcontrol=[-1.0 -0.1 -1.2 -1.3 -1.4 -1.5]

With the pcontrol vector, we can use the place function to calculate the K matrix using pole placement. It solves the equation det(sI - A + BK) = Pcon(s)

The E matrix selects the states that we want to control using w, those are x and y, so:

E=

[1 0 0 0 0 0;

0 1 0 0 0 0];

Finally, we can calculate the precompensator matrix H, using the formula

H=-inv(E*inv(A-B*K)*B);

Once we have all the matrices, we can set the goal w, calculate u using the formula:

u=u_bar-K*(x-x_bar)+H*(w-w_bar);

Where x_bar is the state on the operating point

Then we can integrate x numerically using the system equations and plot or draw the plane

To draw the plane we used a dot matrix, a rotation and translation matrix and the plot function, defined in the file answer_6_draw_dibujo.m

File `answer_6_init.m`

```
close all;
clear all;
clc;


%size = get(0,'ScreenSize'); % full screen
figure%('Position',[0 0 size(3)/2 size(4)/2]);
hold
grid
% set(gca,'FontSize',12);
xmin=0;
xmax=50;
ymin=-2;
ymax=6;

axis([xmin xmax ymin ymax]);

axis ('square');
```

File answer_6_f.m

```
function  xdot  = f(x,u)
    d=5.5;
    J=10000;
    m=30000;
    g=9.81;

    B=[0 0;
       0 0;
       0 0;
       -sin(x(3))/m 0;
       cos(x(3))/m 0;
       0 2*d/J;
    ];

    xdot=[x(4);x(5);x(6);0;-g;0]+B*u;
end
```

File answer_6_draw.m

```
function f(t,x,u)
    u(1)/10000
    u(2)
    plot(t,x(1),'r--.')
    plot(t,x(2),'g--.')
    plot(t,x(3),'b--.')
    plot(t,x(4),'k--.')
    plot(t,x(5),'k--.')
    plot(t,x(6),'k--.')
    plot(t,u(1)/100000,'c--.')
    plot(t,u(2)/10,'m--.')
    legend('x','y','theta','dx','dy','dtheta','u1','u2')
end
```

File answer_6_main.m

```
answer_6_init;
x=[1;5;0.0174533;-0.1;-0.2;0.00174533]; % Initial state
goal=[0 2.4]

%define constants
m=30000;
d=5.5;
g=9.81;
J=10000;
u1_A=m*g;

A=[0 0 0 1 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1;
   0 0 -u1_A/m 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];

B=[0 0;0 0;0 0;0 0;1/m 0; 0 2*d/J];

pcontrol=0:size(A,1)-1;
separation=0.1;
offset=1;
pcontrol=(pcontrol*-separation)-offset;
pcontrol(2)=-0.1;
pcontrol
if(max(pcontrol)>0)
    print("warning, unstable")
end
E=[1 0 0 0 0 0;
   0 1 0 0 0 0];

K=place(A,B,pcontrol);
H=-inv(E*inv(A-B*K)*B);

dt=0.05;
frame_counter=0;
for t=0:dt:50
    %x_des=[10;5;0;0;0;0]
    x_bar=[0;0;0;0;0;0];
    w=[goal(1);goal(2)];
    w_bar=[0;0];
    u_bar=[m*g;0];
    %u=u_bar-K*(x-x_des)
    u=u_bar-K*(x-x_bar)+H*(w-w_bar);

    x=x+answer_6_f(x,u)*dt; % Euler x

    % Frame sampling
    if frame_counter == 0
       answer_6_draw(t,x,u);
       %answer_6_draw_dibujo(x);
       pause(0.000000000000000000001);
    end
    t
    frame_counter =frame_counter+1;
    if frame_counter == 10
        frame_counter=0;
    end
end;
```

File answer_6_draw_dibujo.m

```
function f(x)
    clf();
    hold on;
    axis square;
    axis([-10,10,0,20]*1.5)

    x_coor=x(1);
    y_coor=x(2);
    theta=x(3);


    M_plane=[ 5.5 3   2 3   2 1   0 -1 0   1 0   -1 -2  -3 -2   -3 -5.5 -3 -2   -2 -2  0    0  0  2    2  2 3;
                0 0 0.5 3 0.5 1 1.2  1 0.7 1 0.7  1 0.5 3  0.5  0    0  0 -1 -2.4 -1 -1 -2.4 -1 -1 -2.4 -1 0;
            ones(1,28)];

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

    M_translate=[1 0 x_coor;
                0 1 y_coor;
                0 0 1];

    M_plane_transformed=M_translate*M_rotate*M_plane;

    plot(M_plane_transformed(1,:),M_plane_transformed(2,:),'black','LineWidth',1);
end
```

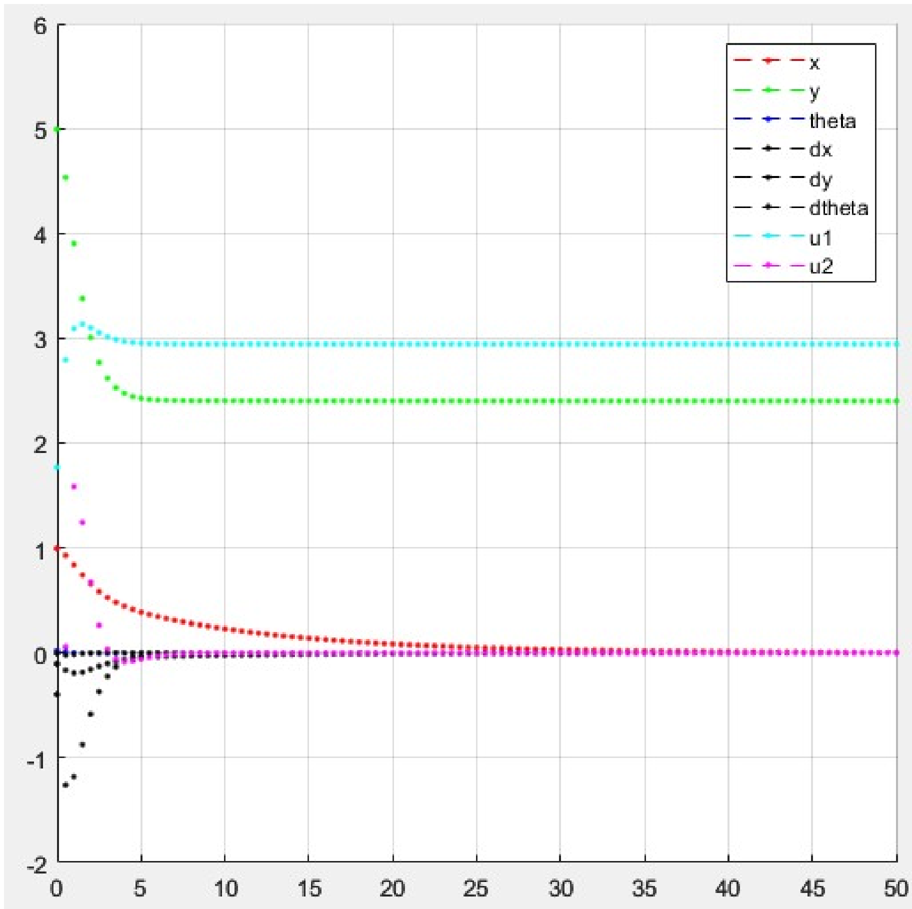

Figure 2: graph representing the state variables of the system

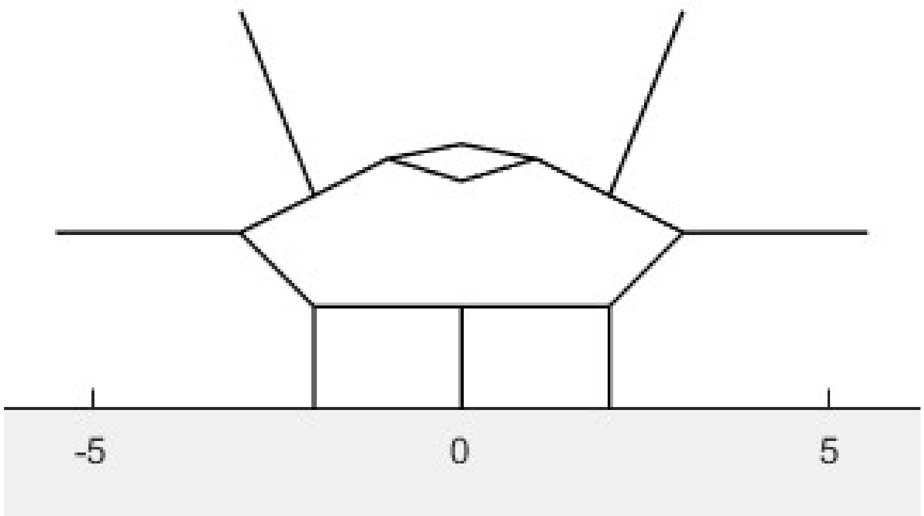

Figure 3: landed aircraft

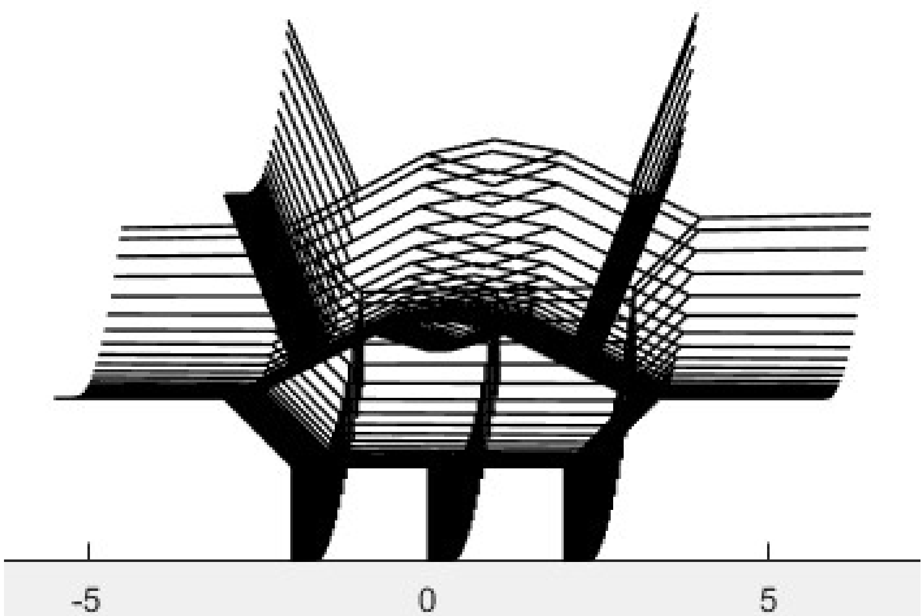

Figure 4: stroboscopic representation of the vtol landing

7) Using the pole placement method, design a state feedback controller to control a lateral dispacement of the aircraft. We want to steer the aircraft from the state $\mathbf{x} = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (0, 5, 0.0174533, -0.1, -0.2, 0.00174533)^T$ to the state $\mathbf{x} = (10, 5, 0, 0, 0, 0)^T$. Give the eigenvalues that have been assigned to the controlled system and illustrate the behaviour of the controller by plotting the relevant state and control variables and by a graphical animation.

Answer 7 is very similar to Answer 6, we only changed the goal, the starting point and adjusted the plot settings.

File answer_7_init.m

```
close all;
clear all;
clc;


%size = get(0,'ScreenSize'); % full screen
figure%('Position',[0 0 size(3)/2 size(4)/2]);
hold
grid
% set(gca,'FontSize',12);
xmin=0;
xmax=50;
ymin=-2;
ymax=10;

axis([xmin xmax ymin ymax]);

axis ('square');
```

File answer_7_f.m

```
function  xdot  = f(x,u)
    xdot=answer_6_f(x,u);
end
```

File answer_7_draw.m

```
function f(t,x,u)
    u(1)/10000
    u(2)
    plot(t,x(1),'r--.')
    plot(t,x(2),'g--.')
    plot(t,x(3),'b--.')
    plot(t,x(4),'k--.')
    plot(t,x(5),'k--.')
    plot(t,x(6),'k--.')
    plot(t,u(1)/100000,'c--.')
    plot(t,u(2)/10,'m--.')
    legend('x','y','theta','dx','dy','dtheta','u1','u2','Location','southeast')
end
```

File answer_7_main.m

```
%answer_7_init;
x=[0; 5; 0.0174533; -0.1; -0.2; 0.00174533]; % Initial state
goal=[10 5]

%define constants
m=30000;
d=5.5;
g=9.81;
J=10000;
u1_A=m*g;

A=[0 0 0 1 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1;
   0 0 -u1_A/m 0 0 0;
   0 0 0 0 0 0;
   0 0 0 0 0 0];

B=[0 0;0 0;0 0;0 0;1/m 0; 0 2*d/J];

pcontrol=0:size(A,1)-1;
separation=0.1;
offset=1;
pcontrol=(pcontrol*-separation)-offset;
pcontrol(2)=-0.1;
pcontrol
if(max(pcontrol)>0)
    print("warning, unstable")
end
E=[1 0 0 0 0 0;
   0 1 0 0 0 0];

K=place(A,B,pcontrol);

H=-inv(E*inv(A-B*K)*B);

dt=0.05;
frame_counter=0;
for t=0:dt:50
    %x_des=[10;5;0;0;0;0]
    x_bar=[0;0;0;0;0;0];
    w=[goal(1);goal(2)];
    w_bar=[0;0];
    u_bar=[m*g;0];
    %u=u_bar-K*(x-x_des)
    u=u_bar-K*(x-x_bar)+H*(w-w_bar);

    x=x+answer_6_f(x,u)*dt; % Euler x

    % Frame sampling
    if frame_counter == 0
       %answer_7_draw(t,x,u);
       answer_7_draw_dibujo(x);
       pause(0.000000000000000000001);
    end
    t
    frame_counter =frame_counter+1;
    if frame_counter == 1
        frame_counter=0;
    end
end;
```

File answer_7_draw_dibujo.m

```
function f(x)
    answer_6_draw_dibujo(x)
end
```

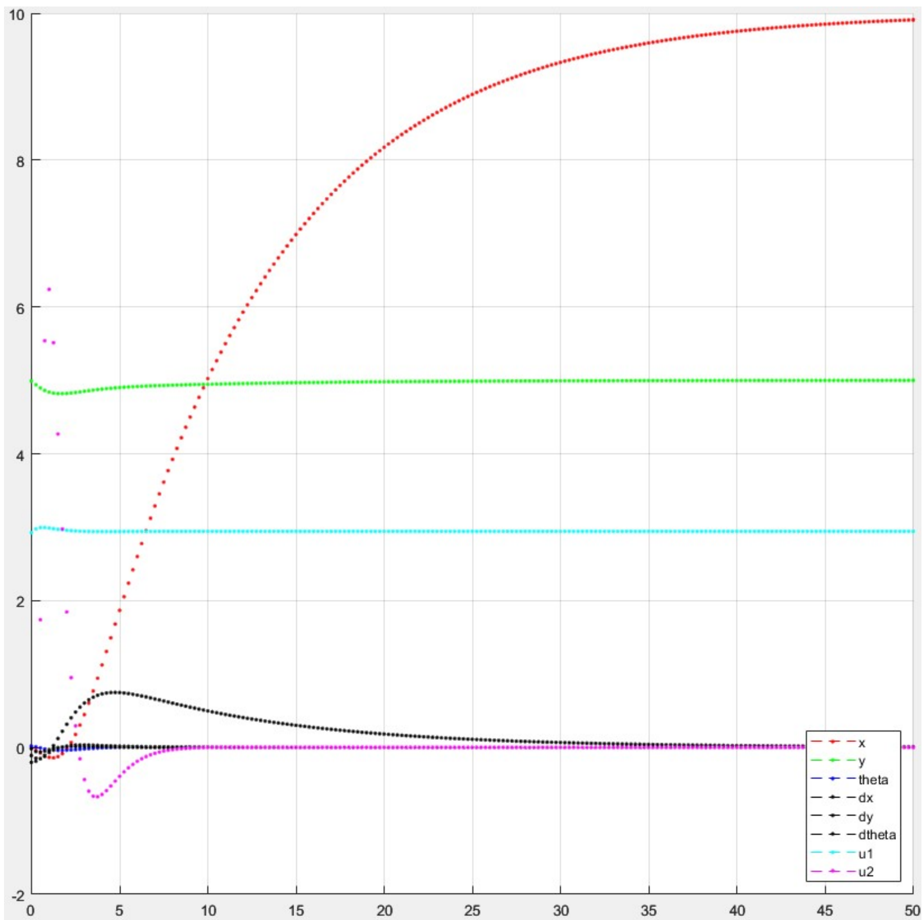

Figure 5: graph representing the state variables of the system

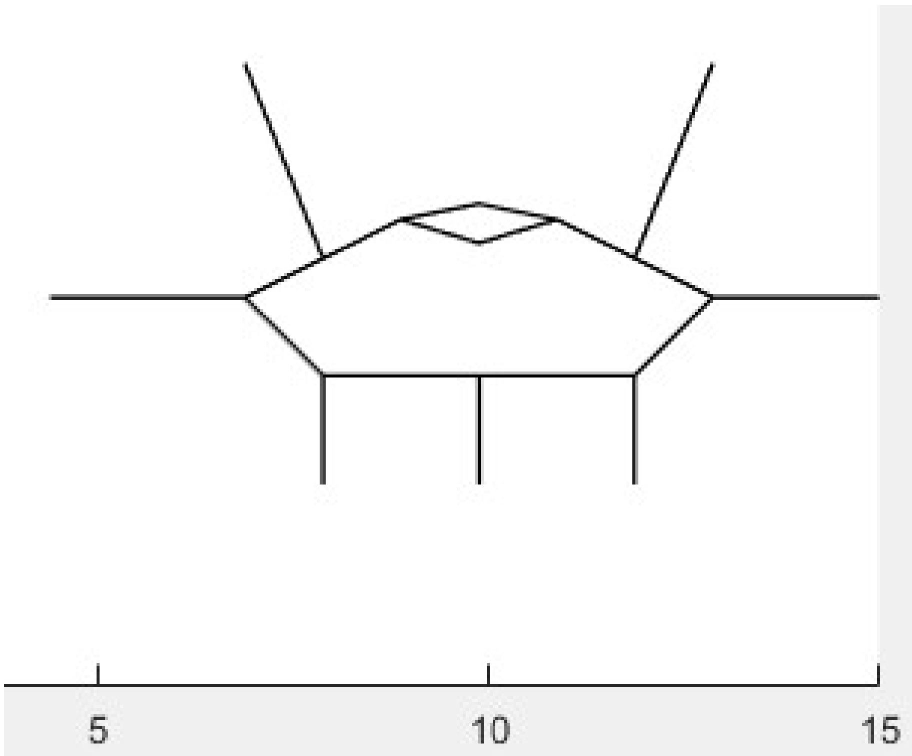

Figure 6: hovering aircraft

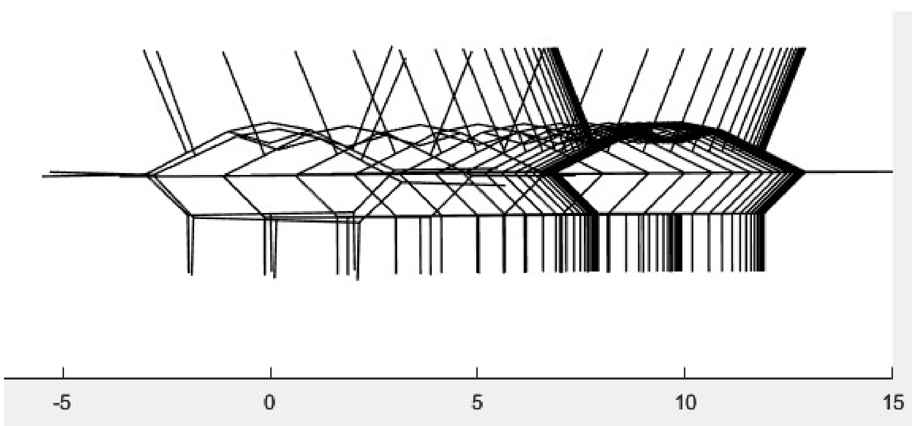

Figure 7: stroboscopic representation of the horizontal movement