Práctica 2

Consider the simplified planar model of the control system for self balancing of a motorcycle represented in Figure 1, in which the motorcycle is represented by a bar that rotates around point $\mathbf{b}$. Points $\mathbf{c}_1$ and $\mathbf{c}_2$ are the positions of the centers of mass of the motorcycle and the reaction wheel, respectively. The control system is composed of an actuated reaction wheel, which rotates around point $\mathbf{c}_2$. A motor controls the rotation of the reaction wheel applying a torque τ, which is the only input of the system. Angles q_1, q_2 and their time derivatives are the state variables of the system.

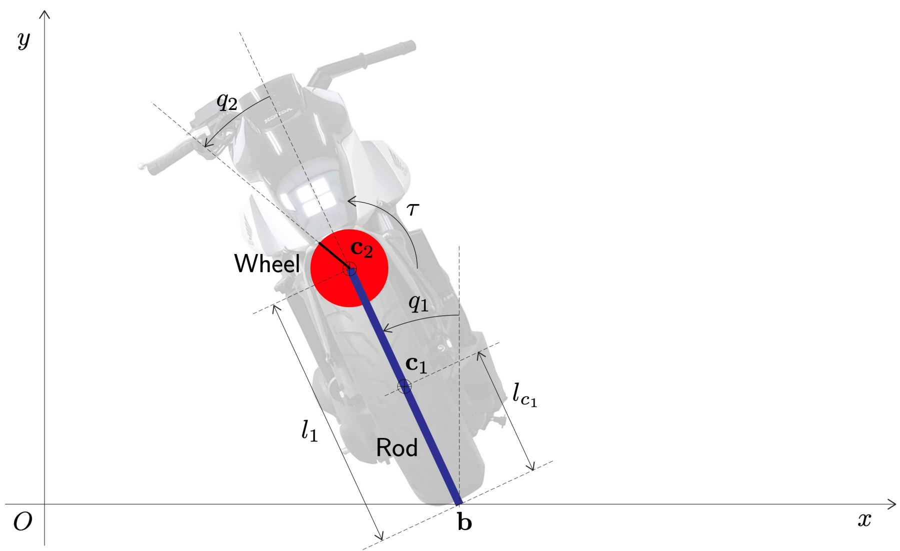

Figure 1: Sketch of the control system for self balancing of a motorcycle

The dynamic model of this system is

$$
\begin{aligned}
(m_1 l_{c_1}^2 + m_2 l_1^2 + I_1 + I_2)\ddot{q}_1 + I_2\ddot{q}_2 - (m_1 l_{c_1} + m_2 l_1) g \sin(q_1) &= 0, \\
I_2\ddot{q}_1 + I_2\ddot{q}_2 &= \tau,
\end{aligned}
$$

with the following parameters

- mass of the motorcycle $m_1 = 200$ [kg],
- mass of the reaction wheel $m_2 = 50$ [kg],
- barycentric moment of inertia of the motorcycle $I_1 = 25$ [kg m^2],
- barycentric moment of inertia of the reaction wheel $I_2 = 5$ [kg m^2],
- length of the rod $l_1 = 1$ [m],

- $l_{c_1} = \frac{l_1}{2} = 0.5$ [m],
- gravity acceleration: $g = 9.81$ [m/s^2].

1) Intentionally omitted.

2) Calculate the state space representation of the system, assuming that $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T$, where angles are expressed in [rad] and angular velocities in [rad/s]. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `statespace.m`)

3) Calculate all the equilibrium points of the system and explain the obtained result. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `equilibrium.m`)

4) Linearize the system around the operating point that corresponds to $\overline{x}_1 = 0, \overline{u} = 0$. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `linearization.m`)

5) Is the linearized system controllable? (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `controllability.m`)

6) Is the linearized system observable by means of the output $y = x_1$? Is the linearized system observable by means of the output $y = x_2$? (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `observability.m`)

7) Using the pole placement method and an observer, design an output feedback controller to control the inclination of the motorcycle. We want to steer the motorcycle from the state $\mathbf{x} = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T = (0.26179, 0, 0, 0)^T$ to the state $\mathbf{x} = (0, 0, 0, 0)^T$. 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. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en la carpeta `controller`)

Originality and completeness of the written answers will be the aspects that will be taken into account in the grading of the exam, and therefore, the `Matlab` code alone will not be considered.

Solution

1) Intentionally omitted.

2) Calculate the state space representation of the system, assuming that $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T$, where angles are expressed in [rad] and angular velocities in [rad/s]. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `statespace.m`)

   First, we need to put $\dot{x} = (\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4)^T = (\dot{q}_1, \dot{q}_2, \ddot{q}_3, \ddot{q}_4)^T$.

   Now, we need to solve for the variables $\ddot{q}_3$ and $\ddot{q}_4$

   For this, we use the dynamic model of this system and the solve function in MATLAB.

   After the calculations, we have the following state space

   $\dot{x}_1 = x_3$

   $\dot{x}_2 = x_4$

   $\dot{x}_3 = (g*l_1*m_2*sin(x_1) - tau + g*lc_1*m_1*sin(x_1))/(m_2*l_1^2 + m_1*lc_1^2 + I_1)$

   $\dot{x}_4 = (m_2 * tau * l_1^2 - I_2 * g * m_2 * sin(x_1) * l_1 + m_1 * tau * lc_1^2 - I_2 * g * m_1 * sin(x_1) * lc_1 + I_1 * tau + I_2 * tau)/(I_2 * (m_2 * l_1^2 + m_1 * lc_1^2 + I_1))$

   Following this, we can conclude that x3 and x4 depend on x1

   File `statespace.m`

```
clear all;
syms m1 m2 I1 I2 l1 lc1 g q1 q2 dotq1 dotq2 ddotq1 ddotq2 tau

eqn1 = (m1*lc1^2 + m2*l1^2 + I1 + I2)*ddotq1 + I2*ddotq2 - (m1*lc1 + m2*l1)*g*sin(q1);
eqn2 = I2*ddotq1 +I2*ddotq2 - tau;
% Resolver el sistema de ecuaciones
sol = solve([eqn1, eqn2], [ddotq1, ddotq2]);

% Mostrar los resultados
x1 = dotq1
x2 = dotq2
x3 = sol.ddotq1
x4 = sol.ddotq2
```

3) Calculate all the equilibrium points of the system and explain the obtained result. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `equilibrium.m`)

   In this part, we need to set all the terms of the state space to 0. After this, we use the solve function with ("'ReturnConditions', true, 'Real', true") to calculate all the equilibrium points of the system. 1) $x_3 == 0$

   2) $x_4 == 0$

3) $(g{*}l_1{*}m_2{*}sin(x_1){-}tau{+}g{*}lc_1{*}m_1{*}sin(x_1))/(m_2{*}l_1^2{+}m_1{*}lc_1^2{+}I_1) == 0$

4) $(m_2 * tau * l_1^2 - I_2 * g * m_2 * sin(x_1) * l_1 ... == 0$

After the calculations, we can conclude that x2 can be in equilibrium in phi*k and x1 can be in all position in x. Is normal that the other variables are 0 since if it is in equilibrium, there is no velocity.

File `equilibrium.m`

```
clear all;
syms u x1 x2 x3 x4 q1 q2 dotq1 dotq2 tau

m1 = 200;
m2 = 50;
I1 = 25;
I2 = 5;
l1 = 1;
lc1 = 0.5;
g = 9.81;

eqn1_q = (g*l1*m2*sin(q1) - tau + g*lc1*m1*sin(q1))/(m2*l1^2 + m1*lc1^2 + I1);
eqn2_q = (m2*tau*l1^2 - I2*g*m2*sin(q1)*l1 + m1*tau*lc1^2 - I2*g*m1*sin(q1)*lc1
    + I1*tau + I2*tau)/(I2*(m2*l1^2 + m1*lc1^2 + I1));

eqn1_q = subs(eqn1_q,[q1 q2 dotq1 dotq2 tau],[x1 x2 x3 x4 u])
eqn2_q = subs(eqn2_q,[q1 q2 dotq1 dotq2 tau],[x1 x2 x3 x4 u])

eqn1 = x3 == 0;
eqn2 = x4 == 0;
eqn3 = eqn1_q == 0;
eqn4 = eqn2_q == 0;
eqns = [eqn1 eqn2 eqn3 eqn4]
s = solve(eqns, [x1 x2 x3 x4 u], 'ReturnConditions', true, 'Real', true);

Su = s.u
Sx1 = s.x1
Sx2 = s.x2
Sx3 = s.x3
Sx4 = s.x4
```

4) Linearize the system around the operating point that corresponds to $\overline{x}_1 = 0, \overline{u} = 0$. (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `linearization.m`)

The only thing you need to do here is change x1 and u to 0.

File `linearization.m`

```
clear all;
syms m1 m2 I1 I2 l1 lc1 g u x1 x2 x3 x4 tau

f1 = x3;
f2 = x4;
f3 = (g*l1*m2*sin(x1) - u + g*lc1*m1*sin(x1))/(m2*l1^2 + m1*lc1^2 + I1);
f4 = (m2*u*l1^2 - I2*g*m2*sin(x1)*l1 + m1*u*lc1^2 - I2*g*m1*sin(x1)*lc1
    + I1*u + I2*u)/(I2*(m2*l1^2 + m1*lc1^2 + I1));

A = jacobian([f1, f2, f3, f4] , [x1, x2, x3, x4])
B = jacobian([f1, f2, f3, f4] , [u])
C = jacobian(x2 , [x1, x2, x3, x4])
D = jacobian(x2 , [u])

%linealizacion x1 = 0, u = 0
A = subs(A,[x1 u],[0 0])
B = subs(B,[x1 u],[0 0])
```

5) Is the linearized system controllable? (Contesta en el informe y sube el código `Matlab` a Aula Virtual en el fichero `controllability.m`)

To check if a linearized system is controllable, one needs to determine matrices A and B using the Lagrange method.

In MATLAB we use the jacobian function.

After this, the function 'ctrb' gives us a matrix, and we check if its rank is equal to the dimensions of x.

If this numbers are equals, the linearized system is controllable.

File controllability.m

```
clear all;

syms u x1 x2 x3 x4

m1 = 200;
m2 = 50;
I1 = 25;
I2 = 5;
l1 = 1;
lc1 = 0.5;
g = 9.81;


f1 = x3;
f2 = x4;
f3 = (g*l1*m2*sin(x1) - u + g*lc1*m1*sin(x1))/(m2*l1^2 + m1*lc1^2 + I1);
f4 = (m2*u*l1^2 - I2*g*m2*sin(x1)*l1 + m1*u*lc1^2 - I2*g*m1*sin(x1)*lc1
    + I1*u + I2*u)/(I2*(m2*l1^2 + m1*lc1^2 + I1));

A = jacobian([f1, f2, f3, f4] , [x1, x2, x3, x4]);
B = jacobian([f1, f2, f3, f4] , [u]);

A = subs(A,[x1 x2 x3 x4 u],[0 0 0 0 0]);
B = subs(B,[x1 x2 x3 x4 u],[0 0 0 0 0]);

Co = ctrb(A,B); %calculo matriz controlabilidad

if (rank(Co) == 4)  %si es 4 es controbable ya que x tiene 4 dimensiones
    disp("Es controlable")
else
    disp("No es controlable")
end

%Es controlable
```

6) Is the linearized system observable by means of the output $y = x_1$? Is the linearized system observable by means of the output $y = x_2$? (Contesta en el informe y sube el código Matlab a Aula Virtual en el fichero observability.m)

To check if a linearized system is controllable, one needs to determine matrices A and C using the Lagrange method.

In C, we use x1 and x2 to check which of the two is observable.

In MATLAB we use the jacobian function.

After this, the function 'obsv' gives us a matrix, and we check if its rank is equal to the dimensions of x.

If this numbers are equals, y is equal x1 or x2 and the system is observable.

In this case, the system is observable and y = x2.

File observability.m

```
clear all;

syms u x1 x2 x3 x4

m1 = 200;
m2 = 50;
I1 = 25;
I2 = 5;
l1 = 1;
lc1 = 0.5;
g = 9.81;


f1 = x3;
f2 = x4;
f3 = (g*l1*m2*sin(x1) - u + g*lc1*m1*sin(x1))/(m2*l1^2 + m1*lc1^2 + I1);
f4 = (m2*u*l1^2 - I2*g*m2*sin(x1)*l1 + m1*u*lc1^2 - I2*g*m1*sin(x1)*lc1
    + I1*u + I2*u)/(I2*(m2*l1^2 + m1*lc1^2 + I1));


% y = x1???
A = jacobian([f1, f2, f3, f4] , [x1, x2, x3, x4]);
C = jacobian(x1 , [x1, x2, x3, x4]);

A = subs(A,[x1 x2 x3 x4 u],[0 0 0 0 0]);
C = subs(C,[x1 x2 x3 x4 u],[0 0 0 0 0]);

Ob = obsv(A,C); %calculo matriz observador

if (rank(Ob) == 4)  %si es 4 y = x1
    disp("y = x1")
end

% y = x2???
C = jacobian(x2 , [x1, x2, x3, x4]);
C = subs(C,[x1 x2 x3 x4 u],[0 0 0 0 0]);

Ob = obsv(A,C); %calculo matriz observador

if (rank(Ob) == 4)  %si es 4 y = x2
    disp("y = x2")
end
```

7) Using the pole placement method and an observer, design an output feedback controller to control the inclination of the motorcycle. We want to steer the motorcycle from the state $\mathbf{x} = (q_1, q_2, \dot{q}_1, \dot{q}_2)^T = (0.26179, 0, 0, 0)^T$ to the state $\mathbf{x} = (0, 0, 0, 0)^T$. 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. (Contesta en el informe y sube el código Matlab a Aula Virtual en la carpeta controller)

First of all, we place 4 poles at -2.

After, we put K, L and H for the observer.

$K = place(A, B, pcon)$

$L = (place(A^T, C^T, pobs))^T$

$H = -(E * (A - B * K)^{-1} * B)^{-1}$

Now we initialize x and $\hat{x}$ to the requested values. We also set w = 0 since we aim for all values to be 0. Within the for loop, we use the following functions.

$\dot{\hat{x}} = (A - BK - LC) * \hat{x} + BH(w - \bar{w}) + L(y - \bar{y})$

$u = \bar{u} - K\hat{x} + H(w - \bar{w})$