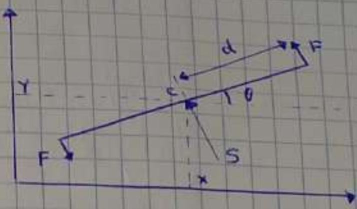
Fórmulas Ingeniería del Control primer parcial:

Examen tipo:

Datos: Centro de masas ($c = (x, y)^T$), ángulo (θ), fuerzas (S y F), entradas (u_1 y u_2).

Modelo dinámico del sistema: $\ddot{x} = -1/m \operatorname{sen}(\theta) S$, $\ddot{y} = -g + 1/m \cos(\theta) S$, $\ddot{\theta} = 2d/J F$

Parámetros: $J = 10000$, $m = 30000$, $d = 5'5$, $g = 9'81$



① Demostrar ecuaciones del modelo dinámico utilizando el método de Lagrange.

$L = T - V = (1/2 m\dot{x}^2 + 1/2 m\dot{y}^2 + 1/2 J\dot{\theta}^2) - (mgy)$

$d/dt(\partial L/\partial \dot{x}) - \partial L/\partial x = f \Rightarrow m\ddot{x} = -\operatorname{sen}(\theta) S \Rightarrow \ddot{x} = -1/m \operatorname{sen}(\theta) S$

$d/dt(\partial L/\partial \dot{y}) - \partial L/\partial y = f \Rightarrow m(\ddot{y} + g) = \cos(\theta) S \Rightarrow \ddot{y} = -g + 1/m \cos(\theta) S$

$d/dt(\partial L/\partial \dot{\theta}) - \partial L/\partial \theta = f \Rightarrow J\ddot{\theta} = 2dF \Rightarrow \ddot{\theta} = 2d/J F$

② Espacio de estados $x = (x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})^T = (x_1, x_2, x_3, x_4, x_5, x_6)^T$

$$x = \begin{pmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ -1/m \operatorname{sen}(x_3) u_1 \\ -g + 1/m \cos(x_3) u_1 \\ 2d/J \cdot u_2 \end{pmatrix}; \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} S \\ F \end{pmatrix}$$

③ Calcular todos los puntos de equilibrio del sistema.

```
close all; clear all; clc;
syms x1 x2 x3 x4 x5 x6 S F m g d J
f1 = x4 == 0; // f2 = x5 == 0; // f3 = x6 == 0; // f4 = -(sen(x3)S)/m == 0; // f5 = -g + 1/m cos(x3)S == 0; // f6 = 2d/J F == 0;
result = solve([f1, f2, f3, f4, f5, f6], [x1, x2, x3, x4, x5, x6, S, F], ReturnConditions = true, IgnoreAnalyticConstraints = true)
```

④ Calcular el punto de equilibrio correspondiente a $u_1 = mg$, $u_2 = 0$. Linealizar el sistema sobre este punto.

```
close all; clear all; clc;
syms x1 x2 x3 x4 x5 x6 S F m g d J
f1 = x4; // f2 = x5; // f3 = x6; // f4 = -(sen(x3)S)/m; // f5 = -g + 1/m cos(x3)S; // f6 = 2d/J F; // y1 = x1; // y2 = x2;
A = jacobian([f1 f2 f3 f4 f5 f6]; [x1 x2 x3 x4 x5 x6]);   | A = subs(A, [x3 S]; [0 g·m])
B = jacobian([f1 f2 f3 f4 f5 f6]; [S F]);                 | B = subs(B, x3, 0)
C = jacobian([y1 y2], [x1 x2 x3 x4 x5 x6]);               | C
```

⑤ Controlabilidad del sistema utilizando ambas entradas (u_1, u_2) o solo una (u_1).

```
close all; clear all; clc;
syms x1 x2 x3 x4 x5 x6 S F m g d J
J = 10000; m = 30000; d = 5'5; g = 9'81;
```

Crear matrices A y B obtenidas en el apartado 4.

Controlabilidad para ambas entradas

```
Co = ctrb(A, B);
unco = length(A) - rank(Co)
```

`Ob = obsv(A, C), unob = obsv(sys)`

Controlabilidad para una entrada

```
B(end, end) = 0;
Co = ctrb(A, B);
unco = length(A) - rank(Co)
```

⑥ Usando el método de desplazamiento de polos o LQR, representar el cambio de estado $(1, 5, 0.0174533, -0'1, -0'2, 0'00174533)^T = (0, 2'4, 0, 0, 0, 0)^T$.

Archivo main

```
init;
m = 30000; g = 9'81; d = 5'5; J = 10000; S = mg;
x = [1; 5; 0.0174533; -0.1; -0.2; 0.00174533]; (Estado inicial)
x_eq = [0; 0; 0; 0; 0; 0]; w_eq = [0; 0]; u_eq = [mg; 0] (P.eq)
t = 0;
```

Crear matrices A y B obtenidas en el apartado 4.

```
poles = [-2 : -0.1 : -2.5];      } Pole placement method
K = place(A, B, poles)
Q = matriz 6x6 (I => 100000, 100, 10000000, 100000)   } LQR
R = matriz 2x2 (I => 0.000000001, 0.6);              } method
K = lqr(A, B, Q, R);
f = matrix 2x6 (I => 1, 1);
H = -inv(f·inv(A-B·K)·B);
frame_counter = 0;
dt = 0.01;
for t = 0 : dt : 10
    w = [0; 2.4] (Estado final)
    u = u_eq - Kx + Hw
    x = x + f(x, u)·dt;
    x = x + dt (0.25 + f(x, u) + 0.75 (f(x + dt·2/3·f(x, u), u)));
    pause(dt);
    frame_counter = frame_counter + 1;
    if frame_counter == 15
        draw(x);
        plot(t, x(1), 'k--.', t, x(2), 'r--.', t, x(3), 'b--.',
             t, u, 'g--.')
        Legend('x(t)', 'y(t)', 'theta(t)', 'u(t)')
        frame_counter = 0;
    end
end
```

Archivo f

```
function xdot = f(x, u)
    J = 10000; m = 30000; d = 5.5; g = 9.81;
    x1 = x(1); x2 = x(2); x3 = x(3); x4 = x(4); x5 = x(5); x6 = x(6);
    S = u(1); F = u(2);
    xp = x4; yp = x5; thetap = x6
    xpp = -1/m sen(x3)S; ypp = -g + 1/m cos(x3)S; thetapp = 2d/J F;
    xdot = [xp; yp; thetap; xpp; ypp; thetapp];
end
```

Archivo draw

```
function draw(x)
    clc(), clf(), hold on, axis square, axis([-10, 17, -10, 17])
    x1 = x(1); y1 = x(2); theta = x(3);
    pendulo = [matrix 3x7 (dibujo)];
    rotacion = [cos(th), -sen(th), 0; sen(th), cos(th), 0; 0, 0, 1];
    traslacion = [1, 0, x1; 0, 1, y1; 0, 0, 1];
    aplica = rotacion · traslacion;
    resultado = aplica · pendulo;
    plot(resultado(1, :), resultado(2, :), 'black', 'Linewidth', 1)
end
```

Archivo init

```
close all; clear all; clc;
figure('Position', [2000 1000 750 750]);
grid
hold
xmin = 0
xmax = ;
ymin = ;
ymax = ;
axis([xmin xmax ymin ymax]);
axis('square');
```