

PÉNDULO DE FURUTA

Conchari Cabrera, Christian Ricardo
Quenta Alvarez, Hamed Emmerson
Ticona Coaquira, Franklin Josué

OBJETIVO GENERAL

Aplicar los conocimientos adquiridos durante la materia para investigar, diseñar e implementar el método de control que mejor se ajuste al sistema conocido como Péndulo de Furuta

OBJETIVOS ESPECÍFICOS



Modelar el sistema "Péndulo de Furuta" y definir las ecuaciones que describen su dinámica.



Diseñar e implementar un controlador PID para el sistema modelado



Diseñar e implementar un controlador LQR para el sistema modelado



Diseñar e implementar un controlador LQG para el sistema modelado



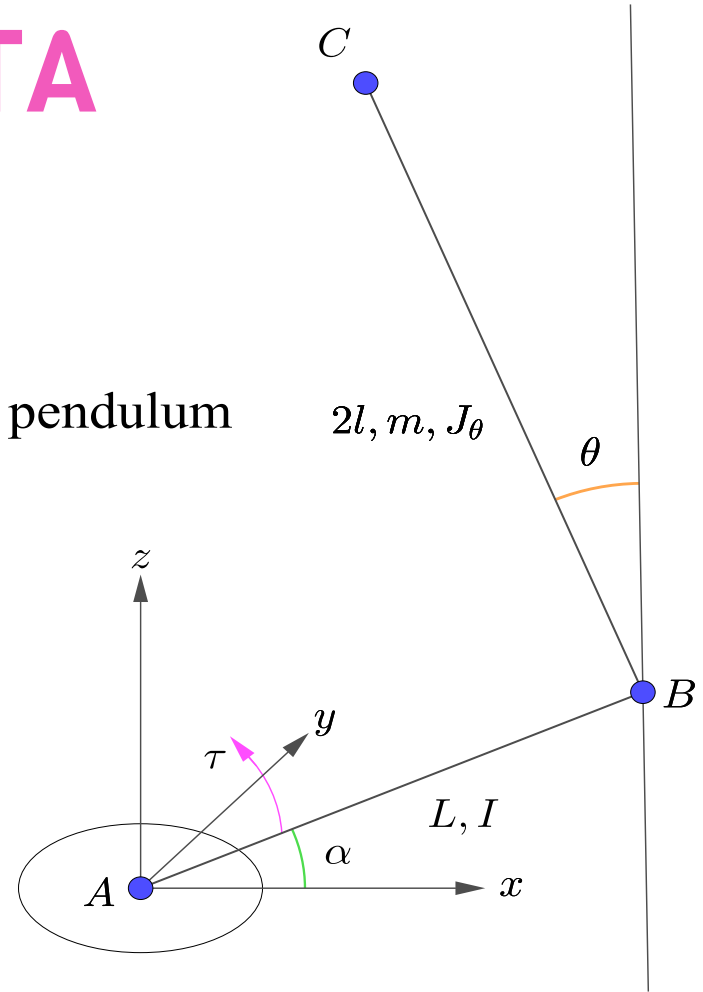
Realizar comparaciones entre los métodos de control PID, LQR y LQG

PÉNDULO DE FURUTA

EL sistema conocido como **PÉNDULO DE FURUTA** se puede clasificar como un péndulo invertido rotatorio. Está conformado por 1 eslabón actuado con movimiento rotatorio sobre un plano horizontal y un eslabón sub-actuado con movimiento rotatorio sobre un plano vertical.

Se trata de un sistema no lineal que en condiciones de lazo abierto es inestable.

Furuta pendulum



DEFINICIÓN DEL PROBLEMA

Siendo que el sistema que estudiaremos se altamente no lineal e inestable, tendremos que trabajar con las siguientes fases.



¿Qué controlador utilizar?

Podría ser un controlador PID, un LQR o tal vez un LQG



DESCRIPCIÓN MATEMÁTICA DE LA DINÁMICA DEL SISTEMA



MODELO MATEMÁTICO DEL SISTEMA

Identifiquemos el vector de coordenadas generalizadas del sistema:

$$\dot{q} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Identifiquemos el vector de fuerzas generalizadas:

$$F = \begin{bmatrix} \tau - b_1 \dot{\alpha} \\ -b_2 \dot{\theta} \end{bmatrix}$$

Identifiquemos las energías cinéticas de ambos eslabones

$$T_1 = \frac{1}{2} I \dot{\alpha}^2 \qquad T_2 = \frac{1}{2} J_{\theta} \dot{\theta}^2 + \frac{1}{2} m |\vec{r}|^2$$

Identifiquemos el vector de posición del centro de masa del eslabón subactuado.

$$\vec{r} = \vec{AB} + \frac{1}{2}\vec{BC} = (L \cos \alpha - l \sin \theta \sin \alpha)\hat{i} + (L \sin \alpha + l \sin \theta \cos \alpha)\hat{j} + (l \cos \theta)\hat{k}$$

La energía cinética total del sistema será:

$$T = (\frac{1}{2}I + \frac{1}{2}mL^2)\dot{\alpha}^2 + \frac{1}{2}ml^2 \sin^2 \theta \dot{\alpha}^2 - mLl\dot{\alpha}\dot{\theta} \cos \theta + (\frac{1}{2}ml^2 + 1/2J_{\theta})\dot{\theta}^2 - mgl \cos^2 \theta$$

La energía potencial del sistema:

$$U_1 = 0$$

$$U_2 = mgl \cos \theta$$

Definiremos el Lagrangiano de la siguiente manera:

$$L = (\frac{1}{2}I + \frac{1}{2}mL^2)\dot{\alpha}^2 + \frac{1}{2}ml^2 \sin^2 \theta \dot{\alpha}^2 + mLl\dot{\alpha}\dot{\theta} \cos \theta + (\frac{1}{2}ml^2 + \frac{1}{2}J_{\theta})\dot{\theta}^2 - mgl \cos \theta$$

Las ecuaciones del movimiento de Lagrange para el sistema serán:

$$\begin{cases} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} = \tau - b_1 \dot{\alpha} \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = -b_2 \dot{\theta} \end{cases}$$

Obtendremos el sistema de ecuaciones que definirán la dinámica del sistema.

$$[I + m(L^2 + l \sin^2 \theta)] \ddot{\alpha} + mL \cos \theta \ddot{\theta} + ml^2 \sin(2\theta) \dot{\alpha} \dot{\theta} - mL \sin \theta \dot{\theta}^2 = \tau - b_1 \dot{\alpha}$$

$$[J + ml^2] \ddot{\theta} + mL \cos \theta \ddot{\alpha} - ml^2 \sin \theta \cos \alpha \dot{\alpha}^2 - mgl \sin \theta = -b_1 \dot{\theta}$$

ESPACIO DE ESTADOS LINEALIZADO

Dada la complejidad del modelo obtenido, tendremos que apoyarnos en el software de “Wolfram Mathematica” con las siguientes líneas de código:

Masa Eslabón AB:	0.1 kg
Masa Eslabón AB:	0.1 kg
Longitud Eslabón AB:	0.1 m
Longitud Eslabón CD:	0.2 m
Fricción angular viscosa 1:	0.00001 Nm/s
Fricción angular viscosa 2:	0.00001 Nm/s
Inercia:	0.0013333 kg m ²
Inercia Polar:	0.0003333 kg m ²

```
m = 0.1;
M = 0.1;
l = 0.2 / 2;
L = 0.2;
b1 = 0.00001;
b2 = 0.00001;
i = (1 / 3) * M * L^2;
Jθ = (1 / 12) * m * (2 * l)^2;
g = 9.78;
coordenadas = {α[t], θ[t]};
velocidades = D[coordenadas, t];
fuerzas = {τ[t] - b1 * α'[t], -b2 * θ'[t]};
c1 =  $\frac{1}{2} * i + \frac{1}{2} * m * l^2$ ;
c2 =  $\frac{1}{2} * m * l^2$ ;
c3 = m * L * l;
c4 =  $\frac{1}{2} * m * l^2 + \frac{1}{2} * J_{\theta}$ ;
c5 = m * g * l;
```

A continuación obtendremos el lagrangiano y el espacio de estados linealizado

```
lagrangiano = c1 * α'[t]2 + c2 * (Sin[θ[t]])2 * α'[t]2 + c3 * α'[t] * θ'[t] * Cos[θ[t]] + c4 * θ'[t]2 - c5 * Cos[θ[t]];
                                     |coseno                                     |coseno
```

```
modelomat = MapThread[{q, v, h} ↦
    |aplica a través
```

```
    D[D[lagrangiano, v], t] - D[lagrangiano, q] == h, {coordenadas, velocidades, fuerzas}];
    |· |deriva |deriva
```

```
eq = modelomat // Simplify;
    |simplifica
```

```
Expand[eq];
|expande factores
```

```
Collect[eq, {α''[t], θ''[t]}];
|agrupa coeficientes
```

```
(*parametros={});*
```

```
modelo = StateSpaceModel[eq, {{α[t], 0}, {α'[t], 0}, {θ[t], 0}, {θ'[t], 0}}, {τ[t]}, {α[t], θ[t]}, t]
    |modelo lineal de espacio de estados
```

El resultado de ejecutar el código anterior será el siguiente:

$$\left(\begin{array}{cccc|c} 0. & 1. & 0. & 0. & 0. \\ 0. & -0.00428571 & -62.8714 & 0.00642857 & 428.571 \\ 0. & 0. & 0. & 1. & 0. \\ 0. & 0.00642857 & 167.657 & -0.0171429 & -642.857 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array} \right) S$$

Podemos escribir el espacio de estados de la siguiente forma:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0043 & -62.8714 & 0.0064 \\ 0 & 0 & 0 & 1 \\ 0 & 0.0064 & 167.6570 & -0.0171 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 428.5710 \\ 0 \\ -642.8270 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Donde:

$$\begin{aligned} x_1 &= \alpha & x_3 &= \theta \\ x_2 &= \dot{\alpha} & x_4 &= \dot{\theta} \end{aligned}$$

Utilizando el software de Matlab y más específicamente la función ss2tf() encontraremos las funciones de transferencia de nuestro sistema.

$$\frac{\theta(s)}{\tau(s)} = \frac{-642.8s^2 - 0.0213s + 1.229e^{-12}}{s^4 + 0.0214s^3 - 167.7s^2 - 0.3185s}$$

$$\frac{\alpha(s)}{\tau(s)} = \frac{428.6s^2 + 3.214s - 3.144e04}{s^4 + 0.0214s^3 - 167.7s^2 - 0.3185s}$$

Es importante notar que se utilizaran las condiciones iniciales de $\theta=10^\circ$ y $\alpha=0^\circ$ para realizar el diseño y las simulaciones correspondientes a los controladores.

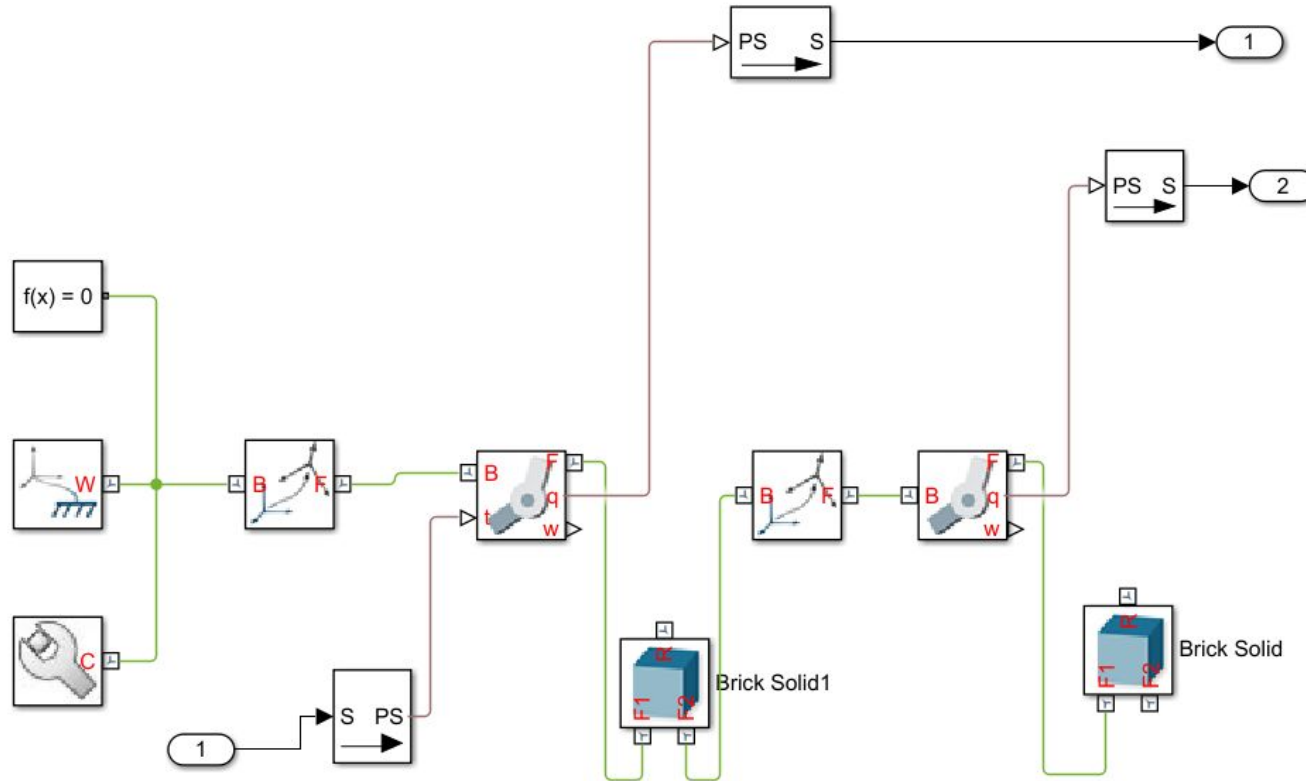
DISEÑO DE CONTROLADORES

SE UTILIZARÁN LOS CONTROLADORES:

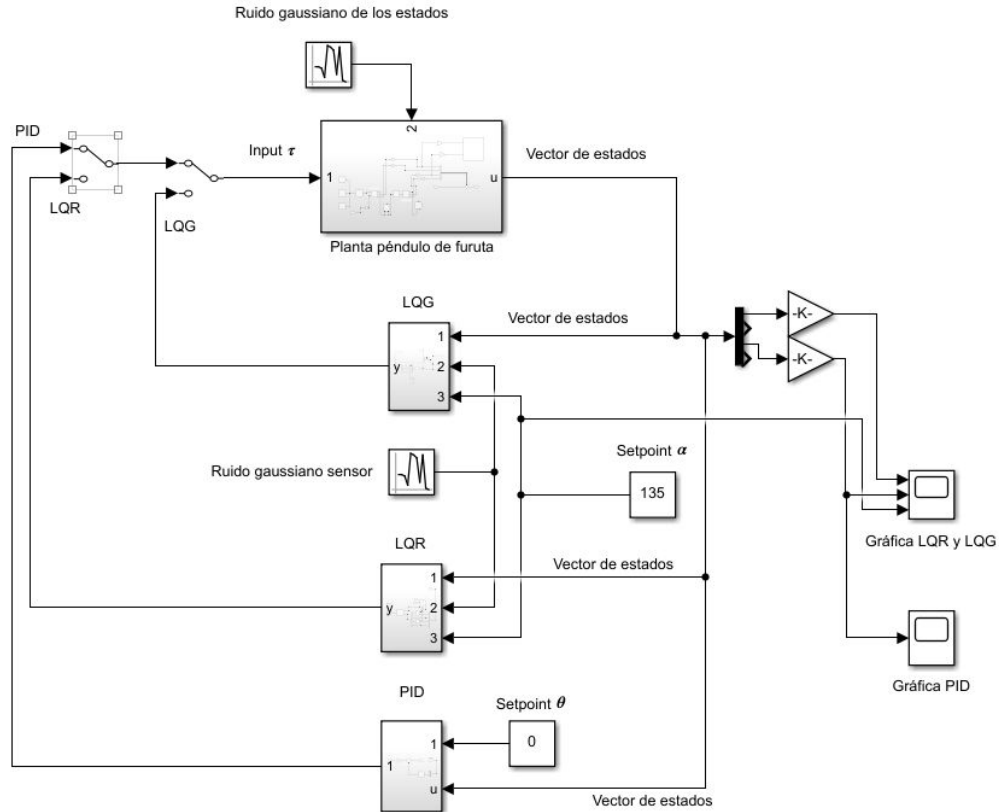
- Proportional Integral Derivative (PID)
- Linear Quadratic Regulator (LQR)
- Linear Quadratic Gaussian (LQG)



IMPLEMENTACIÓN CON SIMULINK Y SIMSCAPE



IMPLEMENTACIÓN CON SIMULINK Y SIMSCAPE



DISEÑO DEL CONTROLADOR PID

Tomando en cuenta que el control PID es aplicable únicamente a sistemas SISO y nuestro sistema cuenta con una entrada y dos salidas, se optó por desestimar la salida correspondiente al ángulo alfa y controlar únicamente la salida del ángulo beta.

Para hacer esto posible la matriz C presentada anteriormente deberá cambiarse por un vector como el siguiente:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad D = 0$$

Llevando este espacio de estados a una función de transferencia obtuvimos lo siguiente:

$$\frac{\theta(s)}{\tau(s)} = \frac{-642.8s^2 - 0.0213s + 1.229e^{-12}}{s^4 + 0.0214s^3 - 167.7s^2 - 0.3185s}$$

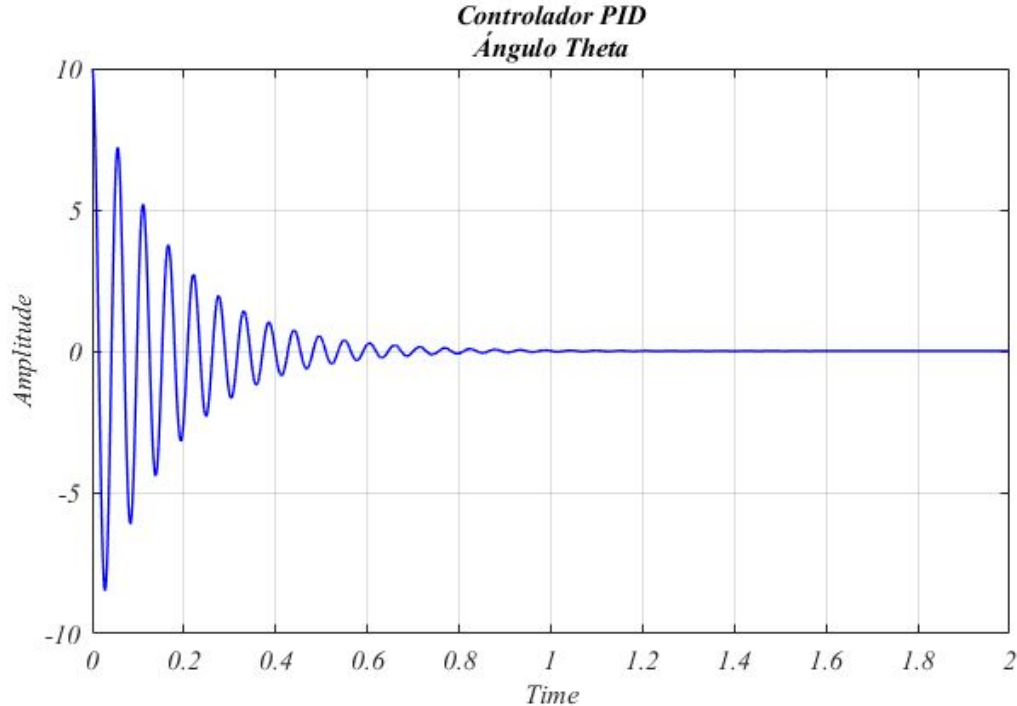
Haciendo uso de la herramienta PID Tuner de Matlab obtendremos las siguientes constantes:

$$\begin{bmatrix} K_P \\ K_I \\ K_D \end{bmatrix} = \begin{bmatrix} -2.2468 \\ -14.6260 \\ -0.0636 \end{bmatrix}$$

Nuestro controlador será el siguiente:

$$G(s) = -2.2468 - 0.0636s - 14.6260\frac{1}{s}$$

Además, se obtuvo el siguiente gráfico al momento de realizar la simulación del sistema en Simulink:



Settling Time = 0.661 s

Peak = 10°

DISEÑO DEL CONTROLADOR LQR

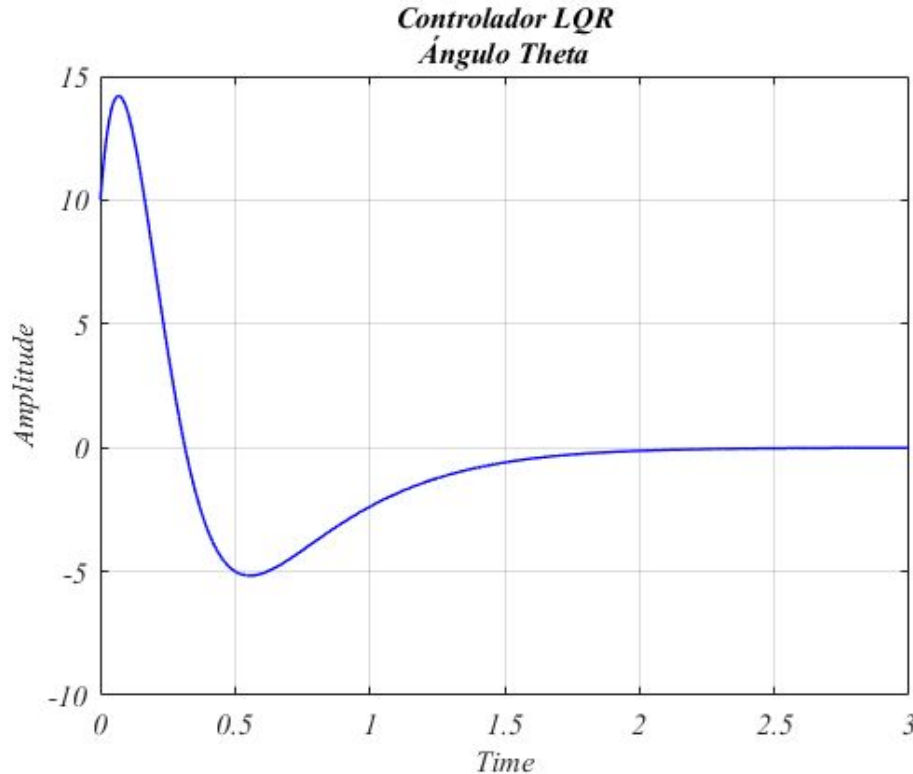
Para el diseñar el controlador LQR se utilizó la siguiente matriz Q y el siguiente valor de R .

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R = 1$$

Con estos valores se encontró la siguiente matriz de retroalimentación K :

$$K = [-3.1623 \quad -2.0889 \quad -26.0826 \quad -2.5986]$$

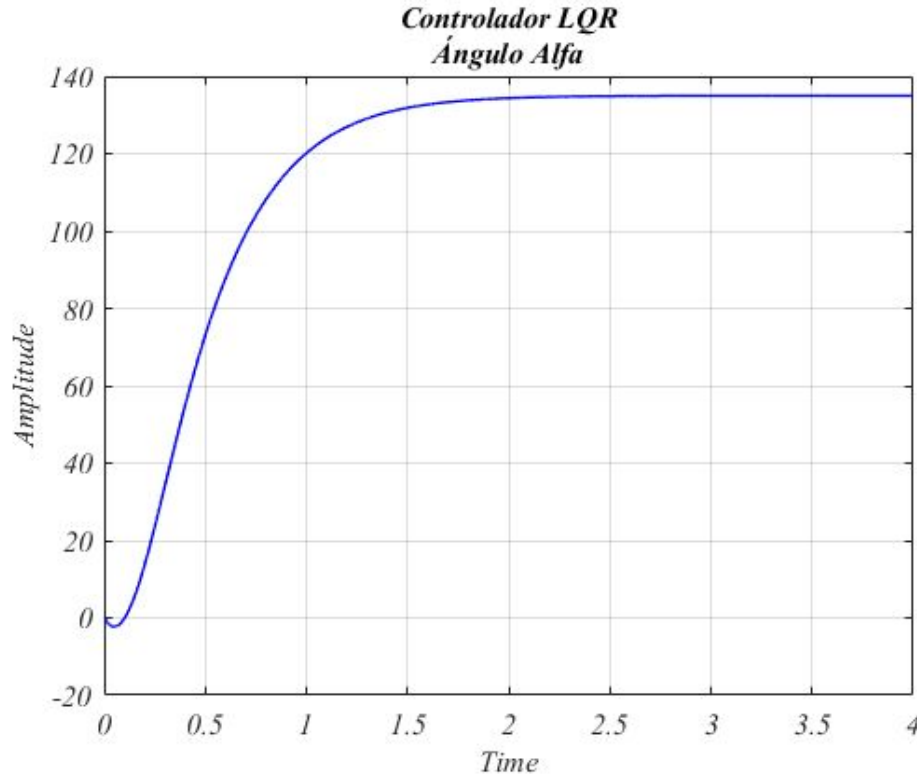
Se llegó a obtener el siguiente gráfico de comportamiento para el ángulo theta al realizar la simulación del controlador LQR diseñado.



Settling Time = 1.7438 s

Peak = 14.2105°

De igual manera se obtuvo una gráfica de comportamiento para el ángulo alfa al realizar la simulación.



Rise Time = 0.8328 s

Settling Time = 1.5443 s

Peak = 135°

Overshoot = 0°

DISEÑO DEL CONTROLADOR LQG

Gracias al principio de Separación que en resumen trata de que se pueden combinar un LQE y un LQR diseñados por separado y combinarlos para generar un LQG. Podremos utilizar el LQR antes diseñado y obtener los coeficientes del filtro de Kalman que conforman el controlador LQE.

$$K = [-3.1623 \quad -2.0889 \quad -26.0826 \quad -2.5986]$$

Para un controlador LQG de la siguiente forma:

$$\begin{bmatrix} \dot{x} \\ \dot{\epsilon} \end{bmatrix} = \begin{bmatrix} (A - BK) & BK \\ 0 & (A - LC) \end{bmatrix} \begin{bmatrix} x \\ \epsilon \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -L \end{bmatrix} \begin{bmatrix} w_d \\ w_n \end{bmatrix}$$

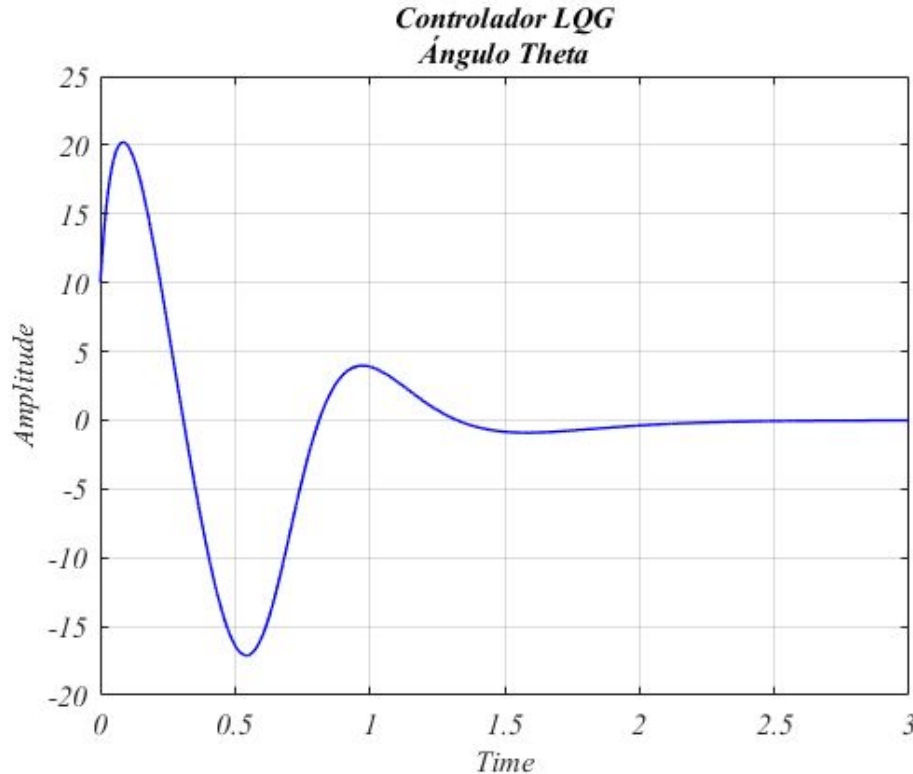
Se utilizaron las siguientes matrices de covarianza:

$$v_n = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix} \quad v_d = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

Es importante resaltar que la covarianza tiene que ser un valor constante y debe tener una media igual a cero para que el ruido sea considerado Gaussiano, y de esa forma podríamos aplicar un controlador LQG.

Finalmente se utilizó la función $[L,P,E]=lqe(A,Vd,C,Vd,Vn)$ de Matlab para obtener las ganancias del estimador. Se utilizó la siguiente función $Kf=(lqr(A',C',Vd,Vn))'$ de Matlab para obtener las ganancias del filtro Kalman.

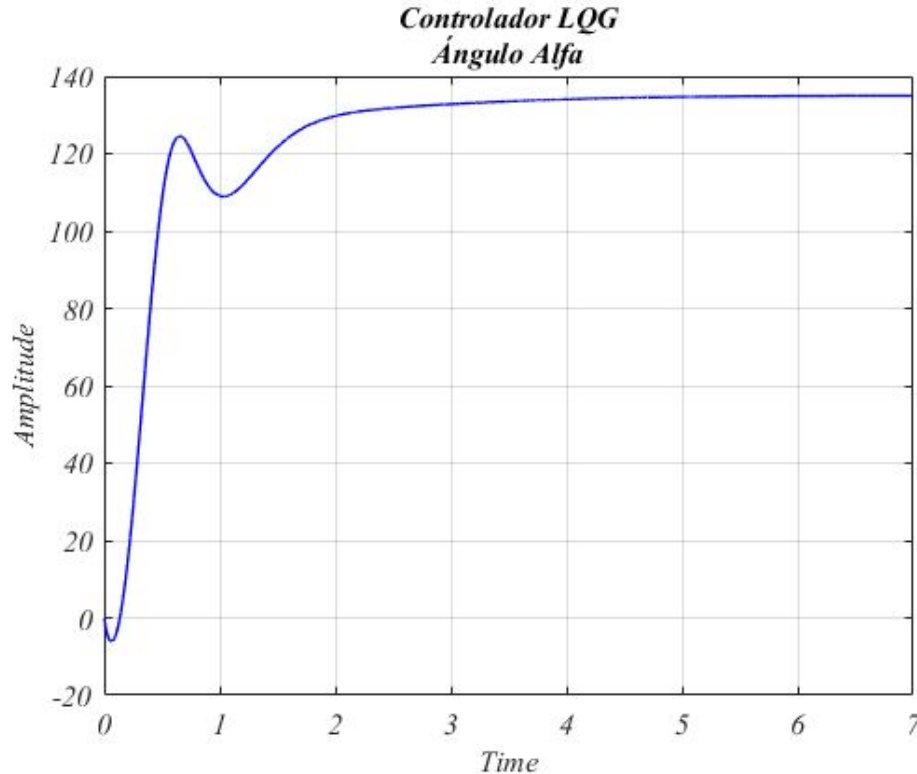
Se llegó a obtener el siguiente gráfico de comportamiento para el ángulo theta al realizar la simulación del controlador LQG diseñado.



Settling Time = 1.9879 s

Peak = 20.1984°

De igual manera se obtuvo una gráfica de comportamiento para el ángulo alfa al realizar la simulación.



Rise Time = 0.3862 s

Settling Time = 2.6555 s

Peak = 135.0008°

Overshoot = 0.0008°

¿Qué controlador
te parece mejor
para nuestro
sistema?



No lo sé,
tendremos que
hacer algunas
comparaciones



COMPARACIONES

TABLA COMPARATIVA

ITEM	PID	LQR	LOG
Setting Time Alfa [s]	-	1.5443	2.6555
Rise Time Alfa [s]	-	0.8328	0.3862
Overshoot Alfa [°]	-	0	0.0008
Setting Time Theta [s]	0.661	1.7438	1.9879
Valor absoluto del pico máximo Theta [°]	10	14.2105	20.1984

CONCLUSIONES

Se logró modelar adecuadamente el sistema conocido como Péndulo de furuta, obteniéndose el modelo matemático que describe la dinámica del sistema, el espacio de estados y las funciones de transferencia correspondientes.

De igual forma, se logró diseñar, implementar y simular los controladores PID, LQR y LQG en la planta obtenida. Se debe mencionar que en el caso del PID solamente se diseñó para controlar el ángulo θ , por las condiciones MIMO de nuestro sistema. Con lo mencionado anteriormente, los controladores LQR y LQG lograron controlar ambas salidas de manera satisfactoria, incluso tomando en cuenta el ruido agregado al sistema.

Finalmente, tomando en cuenta las comparaciones realizadas se determinó que el controlador más adecuado para nuestro sistema es el controlador **LQR**.

[VIDEO EN YOUTUBE](#)





¡GRACIAS POR SU ATENCIÓN!

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

Contactos:

- christian.conchari@ucb.edu.bo
- hamed.quenta@ucb.edu.bo
- franklin.ticona@ucb.edu.bo