
Universitat Politècnica de Catalunya

FACULTAD DE INFORMÁTICA DE BARCELONA

PRÁCTICA DE SISTEMAS DINÁMICOS

Optimización

11 DE JUNIO DE 2024

Aubach Altes, Artur
López García, Daniel

Resumen

Este documento presenta un estudio exhaustivo sobre la aplicación de técnicas de control óptimo en el sistema dinámico del péndulo invertido. El trabajo se ha realizado en el contexto del curso de Sistemas Dinámicos en la Universidad Politécnica de Cataluña (UPC), con el objetivo principal de desarrollar e implementar varios métodos de control para estabilizar un péndulo invertido.

Este proyecto ha explorado tanto enfoques clásicos como el controlador PID, así como métodos más avanzados como el Regulador Cuadrático Lineal (LQR) y el Controlador Lineal Cuadrático Gaussiano (LQG). A través de una simulación detallada utilizando Simulink, se han evaluado y comparado estos métodos para determinar su eficacia en la gestión de la dinámica del sistema.

Los resultados obtenidos demuestran que, aunque los controladores PID y LQR ofrecen una estabilización aceptable, el LQG se presenta como una solución más robusta para la compensación de ruido e incertidumbres en la medición. Finalmente, se discuten las limitaciones y las posibles mejoras para futuros desarrollos en el control óptimo de sistemas dinámicos complejos como el del péndulo invertido.

Palabras clave: Sistemas Dinámicos, Control PID, Regulador Cuadrático Lineal (LQR), Controlador Lineal Cuadrático Gaussiano (LQG), Simulación, Simulink

Índice

Abstract	I
Índice de Figuras	III
Símbolos	IV
1. Introducción	1
2. Objetivos	2
3. Modelo del Péndulo Invertido	3
3.1. Definición del Sistema	3
3.2. Ecuaciones de Movimiento	4
3.3. Linealización	6
3.4. Estabilidad, observabilidad y controlabilidad del Sistema	8
4. PID	10
4.1. Introducción al Control Clásico	10
4.2. Implementación de PID y Ajuste	10
4.3. Resultados del Controlador PID	12
5. Implementación del Controlador LQR	13
5.1. Teoría	13
5.2. Función de Costo	13
5.3. MATLAB y SIMULINK del Controlador LQR	14
5.4. Resultados del Controlador LQR	15
6. Filtro de Kalman	17
6.1. Visión General del Filtro de Kalman	17
6.2. Introducción de Error en Sensor	18
6.3. Aplicación del Filtro de Kalman en el Modelo	19
7. Control LQG	22
7.1. Visión General del LQG	22
7.2. Principio de Separación	22
7.3. Combinación de LQR y Filtro de Kalman	23
7.4. Cálculo de Q y R	23
7.5. Resultados del Controlador LQG	24
8. EXTENSIÓN 1 - Control PID del ángulo y proporcional de la posición	27
8.1. Implementación en simulink y ajuste	27
8.2. Resultados del controlador PID + control proporcional:	27
9. EXTENSIÓN 2 - Control proporcional del péndulo invertido	29
9.1. Implementación en Simulink	29
9.2. Resultados del control proporcional	29
9.3. Conclusiones sobre el control proporcional	30
10. Discusión	31
10.1. Desempeño de los Controladores	31
10.2. Extensiones Propuestas	32
11. Conclusión	33
A. Código MATLAB para la Estabilidad y Observabilidad	35
B. Código MATLAB calcular QR	35

Índice de figuras

1.	Diagrama de Cuerpo Libre del Sistema IP	3
2.	PID controller	10
3.	Planta con PID	11
4.	Resultados PID	12
5.	LQR sitem	13
6.	Planta LQR	15
7.	Resultados LQR	15
8.	Enter Caption	18
9.	Filtro de Kalman	19
10.	Filtro de Kalman	20
11.	Resultados LQG	25
12.	Controlador PID + control porporcional	27
13.	Resultado del controlador PID y proporcional	28
14.	Enter Caption	29
15.	Enter Caption	29

Símbolos

Cuadro 1: Tabla de Simbología

Símbolos	Descripción	Variables de Matlab	Valores con la unidad
M	Masa total del carro	M	0.136 kg
m	Masa péndulo	m	0.1 kg
L	Longitud péndulo	l	0.2 m
g	Gravedad	g	9.8 m/s ²
c	Fricción carro	c	0.63 N/m/s
b	Fricción motor	b	0.00007892 N·m/rad/s
I	Inercia momento barra péndulo	I	0.0007176 kg·m ²
θ	Ángulo del péndulo inicial	alfa	π rad
R	radio del piñón del motor	r	0.0006 m
Rm	resistencia del motor	Rm	12.5 Ω
Kt	par motor constante	kt	0.031 N.m/A
Kb	Constante de la fuerza electromagnética del motor	kb	0.031 V.rad/sec

1. Introducció

En el marc de l'Agenda 2030 per al Desenvolupament Sostenible [2], les empreses industrials es troben davant de desafiaments importants. La gran quantitat de dades recopilades per sensors en les plantes de producció, la manca de coneixement tecnològic adequat per gestionar i analitzar aquestes dades, i la necessitat urgent de reduir l'impacte ambiental són factors que impulsen la recerca i el desenvolupament de noves solucions tecnològiques.

En aquest context, la col·laboració entre *Ercros* [1] i la *Universitat Politècnica de Catalunya* (UPC) ha donat lloc al desenvolupament d'una innovadora solució denominada **OPTIQUIM** (Optimitzador Químic). Aquest projecte exemplifica com un grup d'estudiants, mitjançant l'aplicació de tècniques d'intel·ligència artificial (IA) i la utilització de servidors d'alt rendiment (HPC), pot millorar l'eficiència operativa d'una empresa manufacturera química.

L'objectiu principal d'aquesta iniciativa és millorar la puresa d'una substància química produïda per *Ercros*, mitjançant l'anàlisi i tractament de dades massives obtingudes de sensors en les plantes de producció. A més, es pretén afrontar el repte de la gestió de dades no estructurades emmagatzemades en *data lakes* al núvol, garantint la seva correcta organització i accessibilitat.

El desenvolupament de la solució OPTIQUIM ha estat guiat per tres objectius clau: incrementar la puresa del producte, reduir la contaminació associada al procés de producció i optimitzar la gestió de dades. Aquest projecte no només demostra la capacitat dels estudiants per aplicar coneixements teòrics en un context pràctic, sinó que també presenta un nou enfocament per a l'optimització industrial mitjançant noves tecnologies.

$$\text{OPTIQUIM} = \uparrow \text{puresa} + \uparrow \text{beneficis} + \downarrow \text{contaminació} \quad (1)$$

2. Objetivos

El objetivo principal de este trabajo es explorar y evaluar diferentes métodos de control para estabilizar el sistema dinámico del péndulo invertido. Esta investigación se realiza en el contexto del curso de Sistemas Dinámicos en la Universidad Politécnica de Cataluña (UPC). A continuación, se detallan los objetivos específicos:

- **Desarrollo e Implementación de Controladores:** Diseñar, implementar y sintonizar varios tipos de controladores, incluyendo el controlador PID (Proporcional-Integral-Derivativo), el Regulador Cuadrático Lineal (LQR) y el Controlador Lineal Cuadrático Gaussiano (LQG). Cada controlador se desarrollará y evaluará en términos de su capacidad para estabilizar el péndulo invertido en su posición de equilibrio.
- **Comparación de Desempeño:** Realizar una comparación detallada del desempeño de los controladores implementados utilizando criterios como la estabilidad, la rapidez de respuesta, la robustez frente al ruido y la capacidad de recuperación ante perturbaciones. Este análisis permitirá determinar las fortalezas y debilidades de cada método de control en el contexto del péndulo invertido.
- **Simulación con Simulink:** Utilizar la plataforma de simulación Simulink para modelar el sistema del péndulo invertido y probar los controladores desarrollados. La simulación ayudará a visualizar y analizar el comportamiento dinámico del sistema bajo diferentes condiciones de operación y con distintos controladores.
- **Análisis de Robustez:** Evaluar la robustez de los controladores LQR y LQG frente a incertidumbres en el modelo y a la presencia de ruido en las mediciones. El objetivo es determinar cómo estos controladores pueden manejar las perturbaciones y garantizar un rendimiento estable y confiable en situaciones realistas.

A través de estos objetivos, se espera no solo estabilizar el sistema del péndulo invertido, sino también proporcionar una base sólida para futuros estudios en el control óptimo de sistemas dinámicos complejos.

3. Modelo del Péndulo Invertido

3.1. Definición del Sistema

En el estudio del péndulo invertido, es crucial definir un marco de referencia adecuado para analizar y controlar el sistema. El sistema de péndulo invertido consta de un carro que puede moverse horizontalmente sobre una pista, y un péndulo que pivota libremente desde un punto en el carro. Para la modelación y análisis del sistema, utilizamos el siguiente marco de referencia:

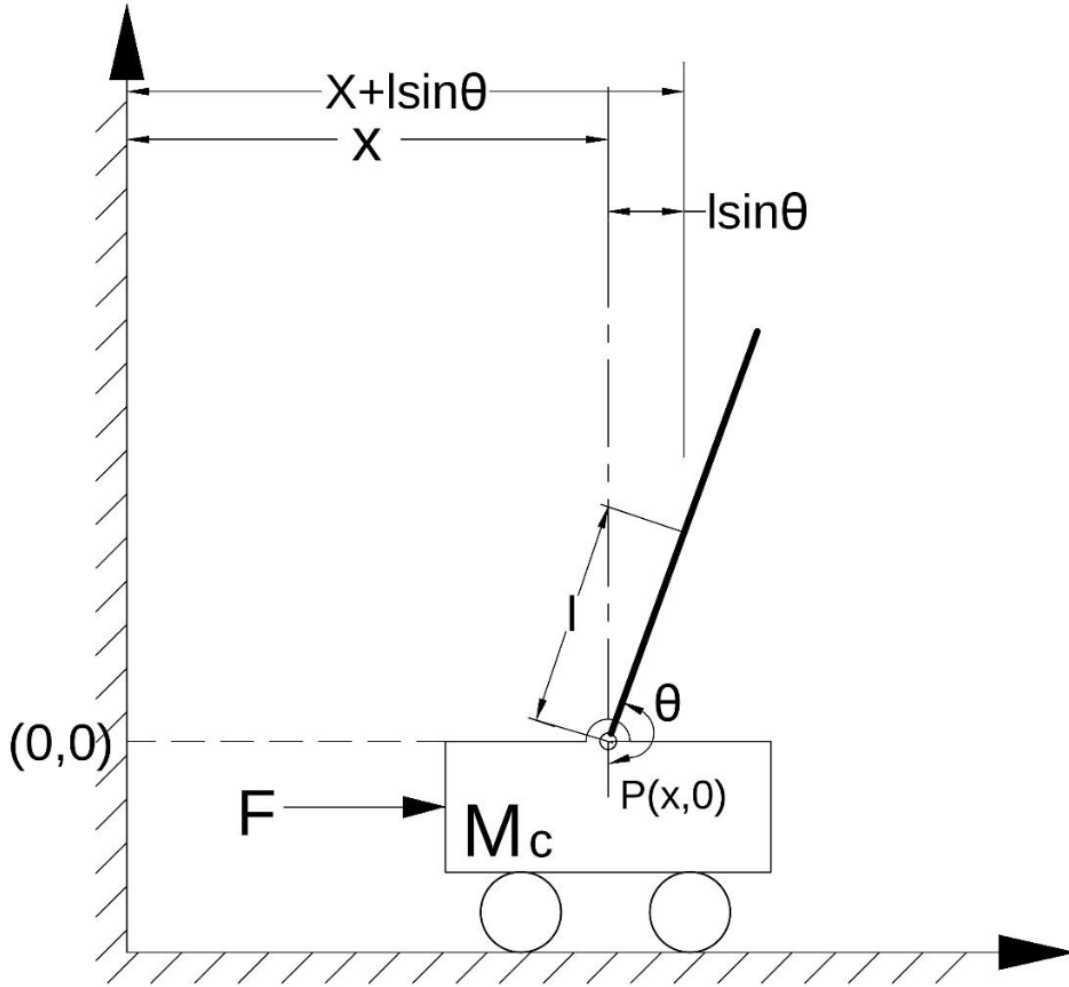


Figura 1: Diagrama de Cuerpo Libre del Sistema IP

El sistema de péndulo invertido se describe mediante las coordenadas del carro x y el ángulo del péndulo θ con respecto a la vertical. En la Figura 1, los parámetros principales son:

- x : Posición del carro
- l : Longitud del péndulo
- θ : Ángulo del péndulo respecto a la vertical
- F : Fuerza aplicada al carro
- M_c : Masa del carro
- M_p : Masa del péndulo
- $P(x,0)$: Posición del pivote del péndulo en el carro

3.2. Ecuaciones de Movimiento

Las ecuaciones de movimiento del sistema de péndulo invertido se derivan utilizando el formalismo de Lagrange, que es un método conveniente para obtener las ecuaciones del movimiento de sistemas complejos. A continuación se presentan las derivaciones detalladas y una explicación de cada paso.

Cinemática del Sistema

Las coordenadas del centro de masa del péndulo (x_p, y_p) se pueden expresar en términos de las variables del sistema (x, θ) de la siguiente manera:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x + l \sin \theta \\ -l \cos \theta \end{bmatrix} \quad (2)$$

Esta ecuación describe la posición del péndulo en un marco de referencia fijado al carro. Aquí, x es la posición horizontal del carro, l es la longitud del péndulo, y θ es el ángulo del péndulo con la vertical.

Lagrangiano del Sistema

El Lagrangiano no relativista para un sistema de partículas se puede definir como:

$$\mathcal{L} = T - U \quad (3)$$

donde T es la energía cinética total y U es la energía potencial total del sistema. La energía potencial del sistema es:

$$U = -mgl \cos \theta \quad (4)$$

La energía potencial U se calcula considerando que el péndulo está influenciado por la gravedad, con m siendo la masa del péndulo y g la aceleración debido a la gravedad.

La energía cinética total del sistema es la suma de la energía cinética traslacional (T_t) y la energía cinética rotacional (T_r):

$$T = T_t + T_r \quad (5)$$

Energía Cinética Traslacional

La energía cinética traslacional (T_t) se da por:

$$T_t = T_{tc} + T_{tp} \quad (6)$$

donde T_{tc} es la energía cinética traslacional del carro y T_{tp} es la energía cinética traslacional del péndulo:

$$T_{tc} = \frac{1}{2} M_c \dot{x}^2 \quad (7)$$

$$T_{tp} = \frac{1}{2} m \left[\left(\frac{dx_p}{dt} \right)^2 + \left(\frac{dy_p}{dt} \right)^2 \right] \quad (8)$$

Sustituyendo las expresiones de x_p y y_p y derivando con respecto al tiempo, obtenemos:

$$T_{tp} = \frac{1}{2} m \left[\left(\frac{d}{dt} (x + l \sin \theta) \right)^2 + \left(\frac{d}{dt} (-l \cos \theta) \right)^2 \right] \quad (9)$$

Resolviendo la ecuación anterior, tenemos:

$$T_{tp} = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{x}\dot{\theta}\cos\theta \quad (10)$$

Aquí, la energía cinética del péndulo incluye componentes debido al movimiento lineal del carro y al movimiento angular del péndulo.

Energía Cinética Rotacional

La energía cinética rotacional (T_r) se da por:

$$T_r = \frac{1}{2}I\dot{\theta}^2 \quad (11)$$

donde I es el momento de inercia del péndulo.

Energía Cinética Total

Por lo tanto, la energía cinética total del sistema es:

$$T = \frac{1}{2}M_c\dot{x}^2 + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}I\dot{\theta}^2 \quad (12)$$

A partir de las ecuaciones anteriores, el Lagrangiano se puede escribir como:

$$\mathcal{L} = \frac{1}{2}M_c\dot{x}^2 + \frac{1}{2}m\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{x}\dot{\theta}\cos\theta + \frac{1}{2}I\dot{\theta}^2 + mgl\cos\theta \quad (13)$$

Ecuaciones de Lagrange

Dado que el sistema tiene dos grados de libertad, hay dos ecuaciones de Lagrange del movimiento, dadas por:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right) - \frac{\partial\mathcal{L}}{\partial x} = F - c\dot{x} \quad (14)$$

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\theta}}\right) - \frac{\partial\mathcal{L}}{\partial\theta} = -b\dot{\theta} \quad (15)$$

Estas ecuaciones representan las fuerzas generalizadas aplicadas al sistema. La primera ecuación considera la fuerza F aplicada al carro y la fricción $c\dot{x}$. La segunda ecuación toma en cuenta el torque de fricción $b\dot{\theta}$ aplicado al péndulo.

A partir de las ecuaciones (11), (12) y (13), las ecuaciones de movimiento resultantes son:

$$(M + m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F - c\dot{x} \quad (16)$$

$$(I + ml^2)\ddot{\theta} + ml\ddot{x}\cos\theta + mgl\sin\theta = -b\dot{\theta} \quad (17)$$

Estas ecuaciones describen el movimiento del carro (\ddot{x}) y el movimiento angular del péndulo ($\ddot{\theta}$). La primera ecuación representa el balance de fuerzas horizontales en el carro, mientras que la segunda ecuación representa el balance de torques en el péndulo.

3.3. Linealización

La linealización de un modelo no lineal es un paso esencial para el diseño de controladores lineales, como los controladores PID y LQR. En esta sección, se presenta el proceso de linealización del modelo del péndulo invertido utilizando la expansión en serie de Taylor y la matriz Jacobiana.

Modelo No Lineal

Consideremos un modelo no lineal de la forma:

$$\dot{X} = f(X, U) \quad (18)$$

donde X es el vector de estados y U es el vector de entradas del sistema. Queremos encontrar un modelo lineal local alrededor de un punto de operación (X_0, U_0) .

Expansión en Serie de Taylor y Matriz Jacobiana

Para encontrar el modelo lineal, utilizamos la expansión en serie de Taylor alrededor del punto de operación:

$$(X_0, U_0) \rightarrow (X = X_0 + \delta X, U = U_0 + \delta U) \quad (19)$$

La expansión en serie de Taylor de \dot{X} alrededor de (X_0, U_0) es:

$$\delta \dot{X} = f(X_0 + \delta X, U_0 + \delta U) \quad (20)$$

Usando la expansión en serie de Taylor, obtenemos:

$$\delta \dot{X} = f(X_0, U_0) + \frac{\partial f}{\partial X}(X_0, U_0)\delta X + \frac{\partial f}{\partial U}(X_0, U_0)\delta U + \text{H.O.T.} \quad (21)$$

Puntos de Operación y Negligencia de Términos de Orden Superior

Asumiendo que el sistema está en equilibrio en el punto de operación, tenemos:

$$f(X_0, U_0) = 0 \quad (22)$$

El punto de referencia se elige como:

$$(X_0, U_0) = ([0, \pi, 0, 0], 0) \quad (23)$$

Negligiendo los términos de orden superior, obtenemos:

$$\delta \dot{X} = A\delta X + B\delta U \quad (24)$$

donde las matrices A y B son las matrices Jacobianas de f evaluadas en el punto de operación (X_0, U_0) :

$$A = \left. \frac{\partial f}{\partial X} \right|_{(X_0, U_0)}, \quad B = \left. \frac{\partial f}{\partial U} \right|_{(X_0, U_0)} \quad (25)$$

Redefiniendo $\delta X \equiv X$ y $\delta U \equiv U$, llegamos a la forma estándar del modelo lineal:

$$\dot{X} = AX + BU \quad (26)$$

Matriz Jacobiana

Podemos linealizar nuestro modelo no lineal alrededor del punto de referencia (X_0, U_0) . Las derivadas parciales de f con respecto a X y U nos proporcionan las matrices A y B :

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \\ \frac{\partial f_4}{\partial x_1} & \frac{\partial f_4}{\partial x_2} & \frac{\partial f_4}{\partial x_3} & \frac{\partial f_4}{\partial x_4} \end{bmatrix}_{(X_0, U_0)} \quad (27)$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{ml^2 g}{\alpha} & -\frac{(I+ml^2)c}{\alpha} & -\frac{bml}{\alpha} \\ 0 & \frac{mgl(M+m)}{\alpha} & -\frac{mlc}{\alpha} & -\frac{b(M+m)}{\alpha} \end{bmatrix} \quad (28)$$

donde:

$$\alpha = I(M + m) + mMl^2 \quad (29)$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial U} \\ \frac{\partial f_2}{\partial U} \\ \frac{\partial f_3}{\partial U} \\ \frac{\partial f_4}{\partial U} \end{bmatrix}_{(X_0, U_0)} = \begin{bmatrix} 0 \\ 0 \\ \frac{I+ml^2}{\alpha} \\ \frac{ml}{\alpha} \end{bmatrix} \quad (30)$$

Modelo en el Espacio de Estados

Finalmente, el modelo lineal en el espacio de estados, con F (fuerza) como la entrada U , se expresa como:

$$\dot{X} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{ml^2 g}{\alpha} & -\frac{(I+ml^2)c}{\alpha} & -\frac{bml}{\alpha} \\ 0 & \frac{mgl(M+m)}{\alpha} & -\frac{mlc}{\alpha} & -\frac{b(M+m)}{\alpha} \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ \frac{I+ml^2}{\alpha} \\ \frac{ml}{\alpha} \end{bmatrix} F \quad (31)$$

Pero todavía no hemos acabado ya que nosotros no controlamos la fuerza del motor, sino su voltaje, por tanto todavía debemos introducir un último cambio: hacer que la F sea el voltaje, V_m , y que al multiplicarse por la matriz B (que cambiaremos) se calcule la fuerza. La fuerza aplicada al carro, F , es generada por el motor de corriente continua (PMDC) y la relación entre F y el voltaje aplicado, V_m , se da por la ecuación:

$$F = \frac{k_t V_m r - k_b \dot{x}}{R_m r^2} = \frac{k_t V_m r - k_b \dot{x}_3}{R_m r^2} \quad (32)$$

donde k_t es la constante de par del motor, k_b es la constante de retroalimentación electromotriz, R_m es la resistencia del motor, y r es el radio del tambor del motor. Y esto nos permite ahora reescribir las ecuaciones lineales del modelo, específicamente la parte de B^*F , como:

$$\dot{X} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{ml^2 g}{\alpha} & -\frac{(I+ml^2)}{\alpha} \left(c + \frac{k_t k_b}{R_m r^2} \right) & -\frac{bml}{\alpha} \\ 0 & \frac{mgl(M+m)}{\alpha} & -\frac{ml}{\alpha} \left(c + \frac{k_t k_b}{R_m r^2} \right) & -\frac{b(M+m)}{\alpha} \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ \frac{(I+ml^2)k_t}{\alpha R_m r^2} \\ \frac{ml k_t}{\alpha R_m r^2} \end{bmatrix} V_m \quad (33)$$

Finalmente, tenemos la segunda ecuación de nuestro sistema, $Y = C^*x + D^*u$ donde consideraremos que somos capaces de ver todas las variables de nuestro sistema: $C = Id$ y también consideramos $D = 0$.

3.4. Estabilidad, observabilidad y controlabilidad del Sistema

En esta subsección se presentan los resultados del análisis de estabilidad, controlabilidad y observabilidad del sistema del péndulo invertido, análisis fundamental para el futuro diseño de los controladores y observadores que se implementarán en las siguientes secciones. El código MATLAB utilizado para este análisis se encuentra detallado en el Apéndice. A.

Valores Propios del Sistema

Los valores propios de la matriz A del sistema se calcularon para evaluar la estabilidad del mismo. Los valores propios obtenidos fueron:

$$\begin{aligned}\lambda_1 &= 0 \\ \lambda_2 &= -19,6327 \\ \lambda_3 &= 6,9178 \\ \lambda_4 &= -5,6006\end{aligned}\tag{34}$$

Estos valores indican la dinámica del sistema de la siguiente manera:

- **Valor propio** $\lambda_1 = 0$: Indica un modo no observable o neutro, lo que significa que el sistema tiene una raíz en el origen del plano complejo. Diremos que el sistema es marginalmente estable en este parámetro el cual se corresponde a la posición, X .
- **Valor propio positivo** $\lambda_2 = -19,6327$: Este valor sugiere que el sistema tiene un modo estable, es decir, las soluciones tenderán a decaer y estabilizarse en un valor. Este se corresponde al parámetro del ángulo del péndulo, θ .
- **Valor propio positivo** $\lambda_3 = 6,9178$: Este valor sugiere que el sistema tiene un modo inestable, ya que las soluciones del sistema crecerán exponencialmente con el tiempo. Este modo se corresponde al parámetro de la velocidad, \dot{X} .
- **Valores propios negativos** $\lambda_4 = -5,6006$: Nuevamente, el valor negativo del eigenvalue nos indica un modo estable, el cual se corresponde a la velocidad angular, $\dot{\theta}$.

En resumen, el sistema presenta un comportamiento mixto con un modo inestable, dos modos estables y uno neutro, lo que implica que el sistema no es completamente estable.

Controlabilidad del Sistema

Para evaluar la controlabilidad del sistema, se calculó el rango de la matriz de controlabilidad, definida como:

$$\text{Cont} = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]\tag{35}$$

El rango de esta matriz se encontró que es 4:

$$\text{rango} = 4\tag{36}$$

Dado que el rango de la matriz de controlabilidad es igual al número de estados del sistema (4), podemos concluir que el sistema es completamente controlable. Esto significa que es posible llevar el sistema desde cualquier estado inicial a cualquier estado final deseado mediante una adecuada selección de las entradas de control.

Observabilidad del sistema

Para evaluar la observabilidad del sistema, se calculó el rango de la matriz de observabilidad, definida como:

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (37)$$

El rango de esta matriz también se encontró que es 4:

$$\text{rango}_{obs} = 4 \quad (38)$$

Dado que el rango de la matriz de observabilidad es igual al número de estados del sistema (4), podemos concluir que el sistema es completamente observable. Esto significa que todos los estados del sistema pueden ser estimados a partir de las salidas medidas.

4. PID

4.1. Introducción al Control Clásico

El controlador Proporcional, Integral y Derivativo (PID) es uno de los métodos más extendidos y efectivos dentro del control clásico de sistemas. Establecido en 1939, el PID ha dominado la ingeniería de control debido a su estructura simple y eficaz, que se adapta bien a una gran variedad de contextos y aplicaciones.

En un sistema de control PID, tres términos interactúan para corregir el error entre un valor deseado y el valor actual medido por el sistema. Estos términos son:

- **Proporcional (P):** Se calcula como el producto del error por una constante proporcional. Este término tiene un efecto directo en la magnitud del error, intentando reducirlo de manera proporcional.
- **Integral (I):** Acumula el error a lo largo del tiempo y lo multiplica por una constante integral. Su principal función es eliminar el error residual en estado estacionario que no puede corregir el componente proporcional.
- **Derivativo (D):** Es el producto de la derivada del error por una constante derivativa. Este término predice la tendencia futura del error, permitiendo una acción correctiva anticipada y mejorando la estabilidad del sistema.

La Figura 2 ilustra el diagrama de bloques básico de un controlador PID, donde cada una de las ganancias ajustables (K_p , K_i , K_d) influye en la respuesta del sistema.

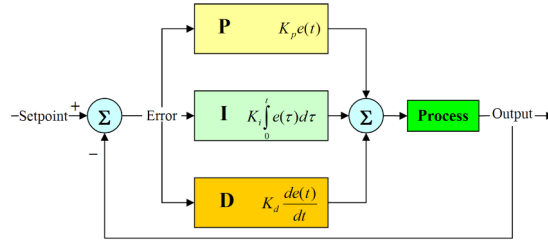


Figura 2: PID controller

La fórmula general del controlador PID se expresa como sigue

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (39)$$

donde $u(t)$ es la señal de control, $e(t)$ es el error de seguimiento, y K_p , K_i , K_d son las ganancias proporcional, integral y derivativa, respectivamente.

Este método ha demostrado ser invaluable en numerosas aplicaciones industriales, desde la regulación de temperatura hasta el control de procesos y sistemas robóticos, ofreciendo una solución robusta y comprensible para la mayoría de los sistemas lineales y algunos sistemas no lineales.

4.2. Implementación de PID y Ajuste

En el diseño del sistema de control para el péndulo invertido, se ha optado por implementar dos controladores PID independientes, uno para controlar la posición del carro y otro para el ángulo del péndulo. Esta elección se basa en la necesidad de manejar dos variables de estado críticas que son esenciales para la estabilidad del sistema. Mientras el controlador de posición se enfoca en mantener el carro en una posición deseada, el controlador de ángulo asegura que el péndulo se mantenga en equilibrio vertical. La combinación de estos dos controladores permite una regulación más precisa y estable del sistema completo.

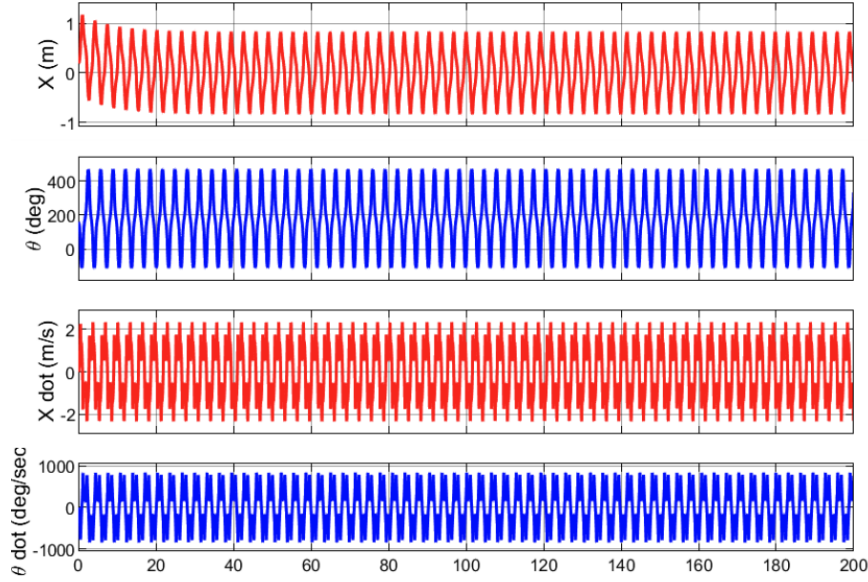


Figura 3: Planta con PID

Configuración en Simulink

Para la implementación en Simulink, se diseñó un esquema donde cada variable controlada (posición y ángulo) es regulada por un controlador PID específico. Además, se utilizó un bloque de ganancia para ajustar y afinar el peso de la salida de cada PID antes de que su acción de control sea enviada a la planta. La Figura 3 muestra el diagrama de bloques de Simulink utilizado para esta configuración.

Ajuste Automático de Parámetros con la Herramienta ‘Tune’ de Simulink

La herramienta ‘Tune’ de Simulink ofrece una metodología robusta para el ajuste automático de los parámetros del controlador PID. Este proceso optimiza las ganancias proporcional (P), integral (I) y derivativa (D) basándose en criterios de diseño especificados, como el tiempo de asentamiento, el sobrepaso y la estabilidad del sistema. A través de simulaciones iterativas y ajustes en el modelo, ‘Tune’ proporciona un conjunto de parámetros que maximizan el rendimiento del controlador bajo diferentes condiciones operativas.

Los valores óptimos obtenidos para los controladores son:

- **PID de posición:** $P = 19,64$, $I = 7,46$, $D = 10,99$
- **PID de ángulo:** $P = 734,85$, $I = 8955,96$, $D = 12,77$

Ajuste de Pesos mediante Prueba y Error

Los pesos para cada controlador PID se establecieron mediante un método de prueba y error, ajustando los valores hasta obtener la mejor respuesta dinámica del sistema. Los pesos finales seleccionados fueron:

- **Peso para el PID de posición:** -1
- **Peso para el PID de ángulo:** 1

Estos pesos ayudan a balancear la influencia de cada controlador en la acción de control final, asegurando que el sistema responda adecuadamente a las perturbaciones y mantenga las condiciones de operación deseadas.

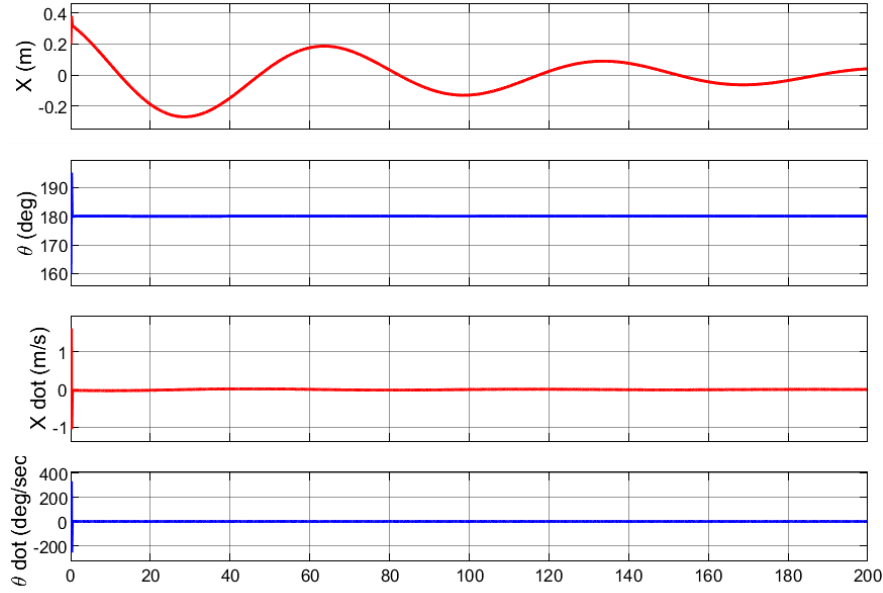


Figura 4: Resultados PID

4.3. Resultados del Controlador PID

Los resultados obtenidos de la implementación del controlador PID para el sistema de péndulo invertido se presentan en la Figura 12. Este conjunto de gráficos ilustra el comportamiento del sistema en términos de desplazamiento del carro (X), ángulo del péndulo (θ), velocidad del carro ($X_{\dot{}}$) y velocidad angular del péndulo ($\theta_{\dot{}}$) a lo largo del tiempo.

Análisis de Resultados:

- **Desplazamiento (X):** La posición del carro muestra oscilaciones mínimas alrededor de la posición objetivo, estabilizándose rápidamente sin necesidad de reajustes significativos. Esto indica que el controlador PID de posición es altamente efectivo, manteniendo el carro prácticamente inmóvil, lo cual es un indicativo de la precisión y la eficacia del control implementado.
- **Ángulo (θ):** El ángulo del péndulo se mantiene constante en 180 grados, lo que demuestra que el controlador PID para el ángulo es capaz de estabilizar el péndulo en la posición vertical deseada sin desviaciones. Esta estabilidad angular es crucial para el éxito del sistema de péndulo invertido, y el resultado obtenido refleja un excelente desempeño del controlador.
- **Velocidad del Carro ($X_{\dot{}}$):** La velocidad del carro se mantiene cercana a cero durante toda la simulación, indicando que no hay necesidad de correcciones bruscas de velocidad para mantener la posición deseada. Esto sugiere una buena sintonización del controlador, que logra mantener la estabilidad con mínima intervención.
- **Velocidad Angular ($\theta_{\dot{}}$):** Similarmente, la velocidad angular del péndulo es cero, lo que implica que el péndulo no está oscilando y se mantiene en un equilibrio perfecto. Esta estabilidad angular complementa la del ángulo y es un resultado directo de un control PID bien ajustado.

Los resultados presentados demuestran que el diseño y la implementación de los controladores PID para el sistema de péndulo invertido han sido exitosos, logrando estabilidad y control precisos de todas las variables de estado críticas. La capacidad de los controladores para mantener las condiciones deseadas con mínimas correcciones es indicativa de una sintonización óptima y una implementación adecuada.

5. Implementación del Controlador LQR

5.1. Teoría

Tras la implementación del controlador PID, se optó por explorar una solución basada en el control óptimo para mejorar aún más la eficacia del sistema de control del péndulo invertido. El Regulador Cuadrático Lineal (LQR) se presenta como una alternativa robusta y eficiente para sistemas lineales invariantes en el tiempo.

El LQR es un método de control óptimo que utiliza una función de coste cuadrática para determinar los pesos óptimos en una matriz de ganancia, minimizando así un índice de rendimiento cuadrático definido. Este índice de rendimiento, J , se define como:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (40)$$

donde Q y R son matrices de ponderación que se escogen para dar forma al comportamiento deseado en términos de rendimiento y esfuerzo de control, respectivamente.

La ecuación diferencial que describe la dinámica del sistema es:

$$\dot{x} = g(x(t), u(t), t) \quad (41)$$

y el criterio de rendimiento que buscamos minimizar a través de la ley de control óptimo es:

$$J = \int_{t_0}^{t_1} h(x(t), u(t), t) dt \quad (42)$$

Para resolver este problema de control, se utiliza la ecuación de Riccati para hallar la matriz P que permite calcular la ganancia óptima K :

$$PA + A^T P + Q - PBR^{-1}B^T P = 0 \quad (43)$$

Una vez obtenida la matriz P , la ganancia K del LQR se calcula como:

$$K = R^{-1}B^T P \quad (44)$$

La nueva ley de control que minimiza el coste es entonces:

$$u = -Kx \quad (45)$$

Este enfoque proporciona una solución que no solo es óptima en términos de la función de coste sino que también garantiza la estabilidad y la robustez del sistema de control. El diagrama de bloques del LQR se muestra en la Figura 5, ilustrando cómo se integra este controlador en el sistema global.

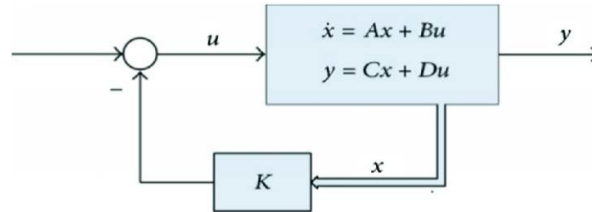


Figura 5: LQR sitem

5.2. Función de Costo

Para el diseño del controlador LQR en nuestro sistema de péndulo invertido, la selección adecuada de las matrices Q y R es crucial, ya que estas determinan el comportamiento y la eficiencia del controlador.

Las matrices Q y R son simétricas y definidas positivas, y su configuración impacta directamente en la penalización de los estados y las acciones de control dentro de la función de costo, que se expresa como:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (46)$$

En nuestro sistema, la matriz Q pondera la importancia relativa de mantener el péndulo en posición vertical y controlar el desplazamiento y la velocidad del carro. Un valor más alto en un elemento particular de Q incrementa la importancia de mantener ese estado cerca del cero, lo cual es crucial para la estabilidad del péndulo. Por otro lado, la matriz R afecta la magnitud del control aplicado, donde un valor menor implica una mayor amplitud de la señal de control permitida.

Para el péndulo invertido, consideramos crítico tanto el ángulo del péndulo respecto a la vertical como la posición del carro. Por lo tanto, hemos configurado Q como:

$$Q = \text{diag}([300, 1500, 0, 0]) \quad (47)$$

Esto significa que se ha dado una alta prioridad a la estabilización del ángulo del péndulo (1500) y una prioridad moderada a la posición del carro (300), mientras que las velocidades (tanto lineal como angular) no se penalizan directamente en esta configuración.

La matriz R , por su parte, se configuró como:

$$R = 0,035 \quad (48)$$

Este valor sugiere que estamos permitiendo una cantidad considerable de actuación del control, equilibrando entre la efectividad del control y el consumo energético o desgaste potencial del actuador.

La elección de estos valores se realizó a través de un proceso de prueba y error, ajustando repetidamente Q y R y observando la respuesta del sistema en simulaciones. Esta metodología nos permitió encontrar una configuración que proporciona un equilibrio óptimo entre la estabilidad rápida del péndulo y el control suave del carro, maximizando la efectividad del sistema sin incurrir en costes excesivos de actuación.

5.3. MATLAB y SIMULINK del Controlador LQR

Para la implementación práctica del controlador LQR en nuestro sistema de péndulo invertido, se utilizó el entorno de Simulink, una herramienta integral de MATLAB que permite la simulación y modelado de sistemas dinámicos de una manera visual e interactiva. La configuración detallada del controlador LQR se llevó a cabo mediante la integración de un bloque específico denominado "LQR_Gain".

5.3.1. Configuración en Simulink

El bloque "LQR_Gain" en Simulink se configuró para implementar la ley de control óptimo calculada en MATLAB. Este bloque utiliza directamente la matriz de ganancias K que se calculó en MATLAB usando las matrices Q y R especificadas anteriormente y la ecuación de Riccati. El flujo de trabajo en Simulink para la implementación del LQR se organiza como sigue:

1. **Cálculo de la Ganancia K :** Se ejecuta un script en MATLAB que resuelve la ecuación de Riccati para obtener la matriz K . Este script utiliza funciones incorporadas como `care` (Control Algebraic Riccati Equation solver) para garantizar una solución precisa y eficiente.
2. **Integración en Simulink:** La matriz K se introduce manualmente en el bloque "LQR_Gain" dentro del modelo de Simulink del sistema de péndulo invertido.
3. **Simulación y Análisis:** Con la matriz K aplicada, se simula el sistema para observar la respuesta del péndulo y del carro bajo la acción del controlador LQR. Se ajustan parámetros y se analizan los resultados para asegurar que el comportamiento del sistema es el esperado y óptimo.

La Figura 6 muestra el diagrama de bloques de Simulink utilizado para la simulación del controlador LQR, destacando la integración del bloque "LQR_Gain" y su conexión con el modelo del sistema.

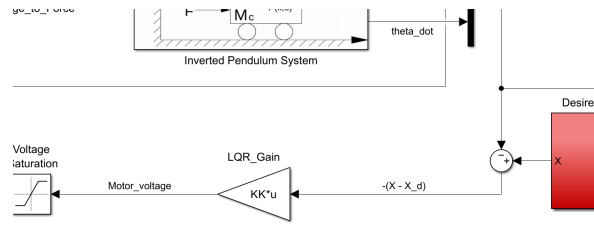


Figura 6: Planta LQR

5.4. Resultados del Controlador LQR

Los resultados de la simulación del controlador LQR para el péndulo invertido muestran una mejora significativa en la estabilidad y la rapidez de respuesta del sistema en comparación con los resultados del controlador PID. Los gráficos siguientes presentan las variables de estado del sistema durante la simulación, Figura 7:

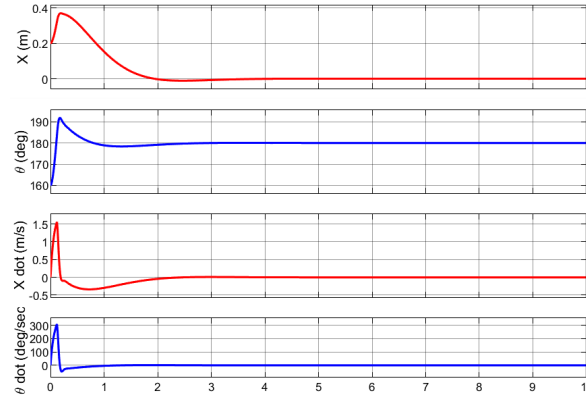


Figura 7: Resultados LQR

Análisis de Resultados:

- **Desplazamiento (X):** El desplazamiento del carro se estabiliza rápidamente a cero dentro de los primeros dos segundos, mostrando una convergencia mucho más rápida en comparación con el PID, donde el carro mostraba movimientos residuales incluso después de 200 segundos.
- **Ángulo (θ):** El ángulo del péndulo alcanza y mantiene la posición vertical deseada (180 grados) sin oscilaciones significativas.
- **Velocidad del Carro (X_{dot}):** La velocidad del carro se reduce a cero casi inmediatamente, indicando un control eficiente y una rápida amortiguación de la velocidad, lo que contribuye a la estabilidad del sistema.
- **Velocidad Angular (θ_{dot}):** La velocidad angular del péndulo se estabiliza a 0 rápidamente, demostrando una excelente capacidad del controlador LQR para manejar la dinámica angular del péndulo.

Comparación con PID:

Los resultados obtenidos con el controlador LQR destacan su superioridad frente al PID en varios aspectos clave:

- **Rapidez de estabilización:** El LQR estabiliza todas las variables de estado dentro de los dos primeros segundos, mientras que el PID requería tiempos considerablemente más largos para alcanzar una estabilidad comparable.
- **Ausencia de oscilaciones:** A diferencia del PID, que resultaba en oscilaciones continuas en algunas variables de estado, el LQR logra una estabilización sin oscilaciones, lo que indica un control más robusto y efectivo.

Los resultados demuestran que el controlador LQR ofrece un rendimiento superior en términos de velocidad de respuesta y estabilidad del sistema. La capacidad del LQR para ajustar rápidamente las variables de estado y mantenerlas estables sin oscilaciones resalta su potencial para aplicaciones en sistemas donde la precisión y la respuesta rápida son críticas. Este análisis subraya la eficacia del LQR en el manejo de las dinámicas complejas asociadas con un sistema de péndulo invertido.

6. Filtro de Kalman

6.1. Visión General del Filtro de Kalman

El Filtro de Kalman es un algoritmo de estimación recursiva que permite inferir el estado de un sistema dinámico a partir de mediciones ruidosas y de un modelo matemático del sistema. Es ampliamente utilizado en control y navegación debido a su capacidad para proporcionar estimaciones óptimas en presencia de ruido y otras incertidumbres.

En aplicaciones de control de sistemas, a menudo existen incertidumbres que deben considerarse durante el proceso de diseño. Es crucial que el ingeniero de control entienda los estados completos del sistema, que son las variables necesarias para describir la dinámica del sistema. Sin embargo, estos estados suelen ser desconocidos, lo que requiere métodos para obtenerlos indirectamente. Uno de estos métodos es el Filtro de Kalman.

El Filtro de Kalman es un estimador de estado óptimo que utiliza los ruidos de proceso y medición, además de las entradas y salidas del sistema, para estimar el estado completo del sistema. Dado que existen numerosas perturbaciones en casos reales, incluyendo el ruido del sensor y del entorno, que pueden afectar el rendimiento de un sistema, el Filtro de Kalman se convierte en una herramienta de análisis poderosa para un ingeniero de control.

Supongamos que tenemos un estado estimado \hat{x} que estima el vector de estado completo x :

$$\dot{\hat{x}} = A\hat{x} + Bu \quad (49)$$

$$\dot{e} = Ax - A\hat{x} \quad (50)$$

El error de estimación del estado se define como:

$$e = x - \hat{x} \quad (51)$$

Combinando las ecuaciones, obtenemos:

$$\dot{e} = Ae \quad (52)$$

Esto significa que si la matriz A es asintóticamente estable, el error convergerá a cero para cualquier entrada, lo que significa que, a medida que el tiempo tiende a infinito, el estado estimado \hat{x} convergerá al estado verdadero x . Alternativamente, si A es inestable, el estado estimado se alejará de los valores reales del estado. Para que el término de error alcance un valor deseado (es decir, cero), se aplica un valor de ganancia de Kalman L :

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (53)$$

$$\hat{y} = C\hat{x} \quad (54)$$

Es importante notar que en la ecuación anterior, a medida que \hat{x} converge a x , \hat{y} también converge a y , negando el valor de la ganancia y llevando el sistema a su verdadero estado. El término de error, por otro lado, se convierte en:

$$\dot{e} = Ax - A\hat{x} - L(Cx - C\hat{x}) = (A - LC)e \quad (55)$$

$$\dot{\hat{x}} = (A - LC)x + Bu + Ly \quad (56)$$

La ecuación anterior es lo que se conoce como el observador de orden completo, que considera tanto la entrada como la salida para su método de estimación.

Para el Filtro de Kalman, un estimador óptimo en comparación con un observador, se introducen dos parámetros adicionales:

$$\dot{\hat{x}} = A\hat{x} + B\hat{u} + Gw(t) \quad (57)$$

$$\hat{y} = C\hat{x} + v(t) \quad (58)$$

donde $w(t)$ y $v(t)$ son ruidos Gaussianos con media cero, siendo Q y R las matrices de covarianza del ruido de proceso y del ruido de medición, respectivamente.

El Filtro de Kalman es, por tanto, una herramienta esencial en el control de sistemas dinámicos, permitiendo una estimación precisa y robusta de los estados del sistema en presencia de ruido y otras incertidumbres.

6.2. Introducción de Error en Sensor

En sistemas de control, la precisión en la medición de las variables de estado es crucial para la implementación efectiva de estrategias de control. En el caso de nuestro sistema de péndulo invertido, una medición precisa del ángulo del péndulo es vital para mantener el equilibrio del sistema. No obstante, en la práctica, las mediciones de los sensores están sujetas a errores y ruidos que pueden degradar el rendimiento del sistema de control.

Para simular el error en la medición del ángulo del péndulo, hemos introducido ruido en la señal del sensor. Esto se ha realizado utilizando el bloque de Simulink *Band-Limited White Noise*, que genera un ruido gaussiano con media cero y una determinada potencia de ruido. Este tipo de ruido es ideal para modelar errores de medición ya que imita la naturaleza aleatoria y no predecible de las perturbaciones en un entorno real.

En el diagrama de Simulink proporcionado en la Figura 8, se observa que el valor del ángulo medido se toma y se combina con una señal de ruido gaussiano antes de ser utilizado en el sistema de control. Específicamente, la señal del ángulo del péndulo pasa por un sumador donde se le añade el ruido generado por el bloque *Band-Limited White Noise*. Esto crea una señal de salida que representa el ángulo medido contaminado con ruido, tal como ocurriría en un sensor real sometido a interferencias o imperfecciones.

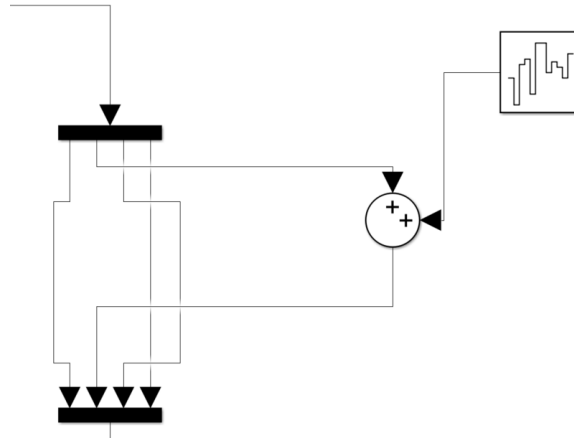


Figura 8: Enter Caption

Justificación del Uso del Filtro de Kalman

El Filtro de Kalman se presenta como una herramienta esencial para corregir el error introducido por el ruido en los sensores. Este filtro permite estimar de manera óptima el estado verdadero del sistema, en este caso, el ángulo real del péndulo, a partir de las mediciones ruidosas.

Las razones por las que el Filtro de Kalman es adecuado para esta tarea son las siguientes:

- **Estimación Óptima en Presencia de Ruido:** El Filtro de Kalman está diseñado específicamente para manejar mediciones ruidosas y proporciona una estimación óptima del estado verdadero del sistema en términos de mínimos cuadrados medios. Esto significa que minimiza la varianza del error de estimación, ofreciendo la mejor aproximación posible al valor real del estado.
- **Recursividad y Eficiencia Computacional:** El Filtro de Kalman actualiza su estimación en cada instante de tiempo de manera recursiva, utilizando tanto el modelo del sistema como las mediciones actuales. Esta característica lo hace computacionalmente eficiente y adecuado para aplicaciones en tiempo real, como el control de un sistema de péndulo invertido.
- **Corrección Dinámica de Desviaciones:** Al ajustar continuamente la estimación basada en las nuevas mediciones, el Filtro de Kalman puede corregir dinámicamente cualquier desviación causada por el ruido o errores de medición. Esto es crucial en sistemas donde los estados cambian rápidamente y se requiere una actualización constante y precisa de las estimaciones.
- **Incorporación de Información del Modelo:** A diferencia de métodos simples de filtrado, el Filtro de Kalman incorpora información del modelo dinámico del sistema, lo que permite una estimación más robusta y precisa incluso cuando el ruido es significativo.

En resumen, la simulación del error de medición del ángulo del péndulo a través del ruido gaussiano modelado en Simulink es una representación realista de los desafíos que enfrentan los sistemas de control en la práctica. El uso del Filtro de Kalman para corregir estos errores de medición es una estrategia efectiva que asegura que el sistema opere con una estimación precisa del ángulo, manteniendo así la estabilidad y el rendimiento deseados.

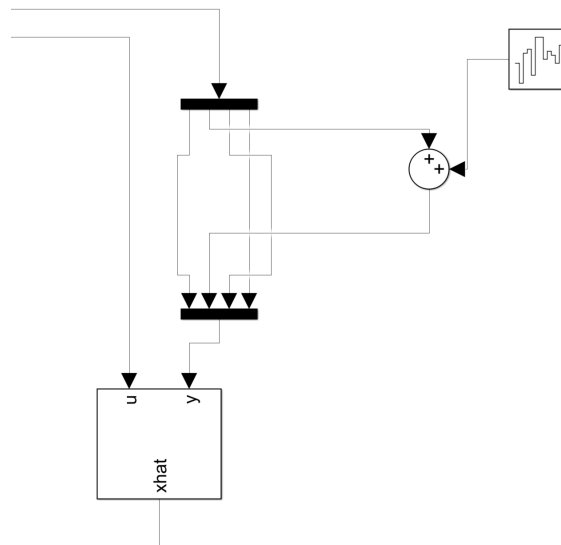


Figura 9: Filtro de Kalman

6.3. Aplicación del Filtro de Kalman en el Modelo

Una vez que hemos introducido el error en la medición del ángulo del péndulo, es esencial implementar un mecanismo que permita estimar de manera precisa el estado verdadero del sistema a pesar de la

presencia de ruido. Para ello, hemos utilizado un Filtro de Kalman, que es un algoritmo óptimo de estimación de estados ampliamente reconocido en el campo del control de sistemas.

En el diagrama de Simulink proporcionado Figura 10, se puede observar cómo se ha implementado el Filtro de Kalman mediante el uso del bloque conocido como *Kalman Filter*. Este bloque es clave para mitigar los efectos del ruido introducido en las mediciones y proporcionar estimaciones más precisas de los estados del sistema.

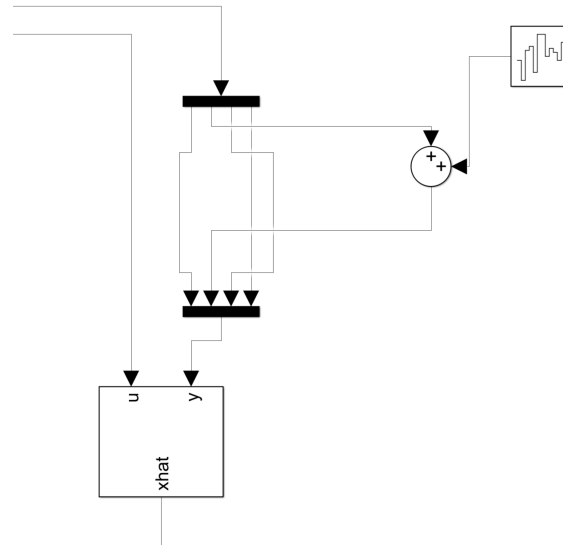


Figura 10: Filtro de Kalman

Descripción de la Implementación

En nuestro modelo, el Filtro de Kalman recibe varias entradas y proporciona estimaciones de los estados del sistema. A continuación, se detallan las entradas y salidas del filtro tal como se han implementado en el diagrama de Simulink:

■ Entradas del Filtro (u y y):

- **u :** Representa la fuerza aplicada al carro. Esta fuerza es un dato de entrada crítico, ya que es una de las variables de control que influye directamente en el comportamiento del sistema.
- **y :** Incluye las mediciones del sistema, que comprenden:
 - La posición del carro.
 - El ángulo del péndulo (con error debido al ruido).
 - La velocidad del carro.
 - La velocidad angular del péndulo.

Estas mediciones son las que se utilizan para evaluar el estado actual del sistema, aunque algunas de ellas, como el ángulo, están contaminadas con ruido.

■ Estimación de Estado (\hat{x}):

- El bloque de Filtro de Kalman procesa estas entradas para proporcionar una estimación optimizada de los estados del sistema. Las salidas estimadas (\hat{x}) incluyen:
 - Posición estimada del carro.
 - Ángulo estimado del péndulo.
 - Velocidad estimada del carro.
 - Velocidad angular estimada del péndulo.

Estas estimaciones son cruciales para la implementación de estrategias de control efectivas, ya que permiten al controlador actuar sobre un estado más preciso y menos afectado por el ruido.

7. Control LQG

7.1. Visión General del LQG

El controlador Lineal Cuadrático Gaussiano (LQG) es una metodología de control óptimo que integra dos técnicas poderosas: el Regulador Cuadrático Lineal (LQR) y el Filtro de Kalman. Esta combinación permite diseñar controladores robustos y eficientes para sistemas dinámicos lineales, incluso en presencia de incertidumbre y ruido en las mediciones.

El objetivo principal del LQG es optimizar el rendimiento del sistema controlado mediante la minimización de un índice de costo cuadrático, mientras se proporciona una estimación precisa de los estados del sistema a partir de mediciones contaminadas con ruido. Esta capacidad lo hace especialmente útil en aplicaciones prácticas donde no todos los estados del sistema son directamente observables y las mediciones disponibles están afectadas por ruidos y perturbaciones.

La estructura de un controlador LQG se basa en dos componentes fundamentales:

1. Controlador LQR (Regulador Cuadrático Lineal):

- Diseñado para minimizar un índice de rendimiento cuadrático que penaliza tanto las desviaciones de los estados del sistema como el esfuerzo de control.
- Calcula la ganancia óptima K que se utiliza para retroalimentar los estados del sistema con el fin de alcanzar el comportamiento deseado.
- La ley de control se define como $u(t) = -Kx(t)$, donde $x(t)$ es el vector de estados del sistema.

$$J = \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt \quad (59)$$

2. Filtro de Kalman:

- Proporciona una estimación óptima de los estados del sistema a partir de mediciones ruidosas.
- Utiliza un modelo del sistema y las mediciones disponibles para corregir y predecir los estados internos del sistema.
- La ecuación de estimación del estado es $\hat{x}(t) = \hat{x}(t-1) + L(y(t) - \hat{y}(t))$, donde L es la ganancia del Filtro de Kalman y $\hat{y}(t)$ es la salida estimada.

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \quad (60)$$

7.2. Principio de Separación

Una característica clave del controlador LQG es el principio de separación, que permite diseñar el LQR y el Filtro de Kalman de manera independiente. Esto significa que las ganancias K del LQR y L del Filtro de Kalman se pueden calcular separadamente y luego combinarse para formar el controlador LQG.

Este principio facilita el diseño del controlador porque:

- **Descomposición del problema:** Permite abordar la optimización del control y la estimación de los estados de forma separada. Primero, se diseña el LQR para minimizar el costo de control asumiendo que los estados son conocidos. Luego, se diseña el Filtro de Kalman para estimar los estados del sistema basándose en las mediciones ruidosas.
- **Modularidad:** La modularidad en el diseño simplifica la implementación y el ajuste del controlador, ya que cada componente puede ser optimizado independientemente antes de su integración en el sistema completo.

- **Flexibilidad en la adaptación:** El controlador LQG puede adaptarse fácilmente a cambios en el modelo del sistema o en el entorno de operación simplemente ajustando las matrices Q , R o las covarianzas del ruido del proceso y de la medición.

7.3. Combinación de LQR y Filtro de Kalman

Al combinar las ganancias de control K obtenidas del diseño LQR con las ganancias de estimación L del Filtro de Kalman, se obtiene el controlador LQG. La estructura general del controlador LQG para un sistema de control en espacio de estados se puede expresar de la siguiente manera:

1. **Estimación del estado:** El Filtro de Kalman calcula la estimación de los estados $\hat{x}(t)$ a partir de las mediciones ruidosas $y(t)$.

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + L(y(t) - C\hat{x}(t)) \quad (61)$$

2. **Ley de control:** Utilizando los estados estimados $\hat{x}(t)$, el controlador LQR determina la acción de control óptima $u(t)$.

$$u(t) = -K\hat{x}(t) \quad (62)$$

La combinación de estos dos componentes resulta en una estrategia de control robusta y eficiente que puede gestionar tanto la dinámica del sistema como la incertidumbre en las mediciones. En el contexto del péndulo invertido, el controlador LQG permite mantener el equilibrio del péndulo y controlar la posición del carro de manera óptima, incluso cuando no se dispone de todas las variables de estado de manera precisa.

Ventajas del LQG en el diseño de control del péndulo invertido:

- **Manejo del ruido y las incertidumbres:** El Filtro de Kalman compensa la falta de acceso a todas las variables de estado y proporciona estimaciones precisas en presencia de ruido.
- **Optimización del rendimiento:** El LQR asegura que la acción de control minimiza el costo cuadrático, optimizando la estabilidad y el rendimiento del sistema.
- **Aplicabilidad general:** Aunque el controlador LQG es aplicable a una amplia variedad de sistemas, es particularmente valioso en situaciones donde las mediciones están contaminadas por ruido y no todos los estados del sistema son directamente observables.

En resumen, el controlador LQG ofrece una solución integral para el diseño de sistemas de control óptimos, combinando las fortalezas del LQR y el Filtro de Kalman para manejar tanto la dinámica del sistema como la incertidumbre en las mediciones. Esta combinación lo convierte en una herramienta poderosa y versátil en el campo del control de sistemas dinámicos.

7.4. Cálculo de Q y R

En la implementación del controlador LQG para el sistema de péndulo invertido, es esencial determinar las matrices Q y R adecuadas para el Regulador Cuadrático Lineal (LQR) y el Filtro de Kalman. Estas matrices son fundamentales para definir el rendimiento y la estabilidad del sistema de control.

Inicialmente, los valores de Q y R se seleccionaron para el diseño del controlador LQR, penalizando las desviaciones de los estados y el esfuerzo de control. Sin embargo, al aplicar el Filtro de Kalman, los valores de Q y R deben recalibrarse para optimizar la estimación del estado en presencia de ruido en las mediciones.

Para estimar estos valores, hemos utilizado un proceso de optimización detallado en el apéndice B. Este proceso implica encontrar los valores óptimos de Q y R que minimicen la diferencia entre el estado real y el estado deseado del sistema al final de la simulación.

Proceso de Optimización

El proceso de optimización de Q y R se realiza mediante el siguiente código MATLAB, que está descrito en detalle en el apéndice B:

Configuración Inicial del Sistema: Se carga el modelo de Simulink y se establecen los parámetros del sistema, tales como la masa del péndulo (m), la longitud del péndulo (l), la fricción del carro y del motor (c , b), entre otros. Se definen las matrices A y B específicas para el sistema de péndulo invertido.

Verificación de Controlabilidad: Se verifica la controlabilidad del sistema calculando la matriz de controlabilidad `ctrlMat` y asegurando que su rango sea igual al número de estados del sistema. Esto garantiza que el sistema puede ser controlado completamente mediante las entradas disponibles.

Definición del Estado Deseado: Se define el estado deseado del sistema como mantener el péndulo en la posición vertical ($\theta = \pi$) y el carro en la posición de equilibrio ($x = 0$).

Optimización de Q y R : Utilizando la función ‘`fmincon`’, se optimizan los valores de Q y R . Esta función ajusta iterativamente los parámetros Q y R para minimizar la diferencia entre el estado real y el estado deseado del sistema al final de la simulación.

- **Parámetros Iniciales:** Se establecen los valores iniciales de Q y R para la optimización.
- **Límites de Optimización:** Se definen los límites inferiores y superiores para los parámetros Q y R , asegurando que se mantengan en un rango plausible durante la optimización.
- **Ejecución de la Optimización:** La función ‘`fmincon`’ ajusta Q y R para minimizar el costo, que es la diferencia entre los estados finales real y deseado del sistema.

Evaluación del Desempeño: Al final de la optimización, el código muestra los valores óptimos de Q y R . Estos valores son los que mejoran la capacidad del sistema para mantener el equilibrio y responder adecuadamente a las perturbaciones.

Análisis del Código del Apéndice

El código proporcionado en el apéndice B realiza varias funciones clave:

- **Cálculo de Matrices A y B :** Define las matrices del sistema a partir de los parámetros físicos del péndulo invertido.
- **Verificación de Controlabilidad:** Asegura que el sistema es controlable, lo cual es un requisito para estabilizarlo usando un controlador LQR.
- **Optimización de Q y R :** Utiliza un algoritmo de optimización para ajustar los valores de Q y R de manera que minimicen la diferencia entre el estado real y el estado deseado al final de la simulación.
- **Asignación de Ganancia K :** Calcula la ganancia K del LQR y la asigna al espacio de trabajo de MATLAB para su uso en Simulink.
- **Simulación y Evaluación:** Ejecuta la simulación del modelo de Simulink y evalúa el rendimiento del sistema basado en la diferencia entre los estados reales y deseados.

Este enfoque nos permite obtener una configuración óptima para las matrices Q y R , asegurando que el sistema de control LQG puede manejar de manera efectiva la dinámica del péndulo invertido en presencia de ruido y otras incertidumbres. Los valores óptimos de Q y R obtenidos son fundamentales para garantizar un desempeño robusto y preciso del sistema.

7.5. Resultados del Controlador LQG

En esta sección, analizamos los resultados obtenidos con la implementación del controlador LQG en el sistema de péndulo invertido. A continuación, se presenta la gráfica de las respuestas del sistema para las variables de estado: posición del carro (X), ángulo del péndulo (θ), velocidad del carro (\dot{X}) y velocidad angular del péndulo ($\dot{\theta}$).

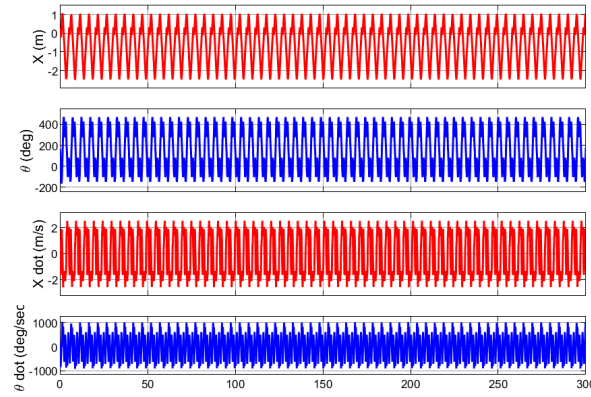


Figura 11: Resultados LQG

Como se puede observar en la Figura 11, el sistema con el controlador LQG no logra estabilizarse. Las respuestas muestran oscilaciones continuas en todas las variables de estado a lo largo del tiempo, indicando una inestabilidad persistente.

Análisis de los Resultados

■ Posición del Carro (X):

- La posición del carro presenta oscilaciones significativas entre -2 y 1 metros. Estas oscilaciones son grandes y persistentes, sugiriendo que el controlador LQG no está logrando mantener el carro en una posición fija o cercana al equilibrio.

■ Ángulo del Péndulo (θ):

- El ángulo del péndulo muestra oscilaciones amplias, variando entre aproximadamente -200 y 400 grados. Esto indica que el péndulo está rotando más allá de su punto de equilibrio vertical y no se mantiene estable, girando excesivamente hacia adelante y hacia atrás.

■ Velocidad del Carro (\dot{X}):

- La velocidad del carro oscila de manera similar a la posición, con variaciones significativas. Esto refuerza la idea de que el carro no se está estabilizando y está en constante movimiento oscilatorio.

■ Velocidad Angular del Péndulo ($\dot{\theta}$):

- La velocidad angular del péndulo también muestra grandes oscilaciones. Estas fluctuaciones indican que el péndulo está experimentando rápidas aceleraciones y desaceleraciones, contribuyendo a la inestabilidad del sistema.

7.5.1. Comparación con los Resultados del PID y LQR

En comparación con los resultados obtenidos con los controladores PID y LQR, el desempeño del controlador LQG es notablemente inferior en términos de estabilización del sistema:

■ PID:

- Los resultados con el controlador PID mostraron una estabilización parcial con algunas oscilaciones menores. El PID logró mantener el carro y el péndulo más cerca de su posición de equilibrio aunque con cierta inestabilidad residual.

■ LQR:

- El controlador LQR demostró ser mucho más efectivo, logrando una estabilización más rápida y precisa de las variables de estado, con menores oscilaciones y un comportamiento mucho más estable en comparación con el LQG y PID.

La inestabilidad observada en el sistema controlado por LQG puede deberse a varios factores:

■ **Sintonización de Q y R :**

- Los valores de Q y R utilizados en el LQR y el Filtro de Kalman podrían no estar optimizados adecuadamente para el sistema específico del péndulo invertido. Esto puede provocar una mala estimación de los estados y, por lo tanto, un control deficiente.

■ **Ruido y Perturbaciones:**

- El Filtro de Kalman, aunque diseñado para manejar ruido en las mediciones, podría no estar compensando eficazmente las perturbaciones presentes en el sistema. Si el ruido es mayor o más complejo de lo que se anticipó, la estimación de estado puede ser inexacta, llevando a una respuesta incorrecta del controlador.

■ **Dinámica No Lineal:**

- El péndulo invertido es un sistema altamente no lineal. Si la linealización utilizada para el diseño del LQG no captura adecuadamente la dinámica del sistema, el controlador puede no ser capaz de estabilizar el péndulo en condiciones reales.

■ **Modelado Inadecuado:**

- Cualquier discrepancia entre el modelo matemático del sistema y la realidad física puede resultar en un mal desempeño del controlador. La precisión del modelo es crítica para la efectividad del LQG.

El controlador LQG, en su configuración actual, no ha logrado estabilizar el sistema de péndulo invertido. Para mejorar el desempeño, se recomienda una revisión detallada de la sintonización de Q y R , y una evaluación más profunda del modelo y las suposiciones subyacentes. Adicionalmente, podría ser beneficioso explorar métodos de control más robustos o ajustar los parámetros del Filtro de Kalman para una mejor estimación en presencia de ruido.

8. EXTENSIÓN 1 - Control PID del ángulo y proporcional de la posición

Decidimos como una de las extensiones intentar controlar el sistema mediante un control PID del ángulo del péndulo y un control proporcional de la posición.

8.1. Implementación en simulink y ajuste

Implementamos este mecanismo de control mediante un PID independiente para controlar específicamente el ángulo y un 'gain'(multiplicador) para multiplicar el error de la posición por un escalar que después añadiremos al resultado del PID. La combinación del PID con un proporcional permite un cierto control que, sin embargo, se ve incapaz de estabilizar por completo la posición del carro.

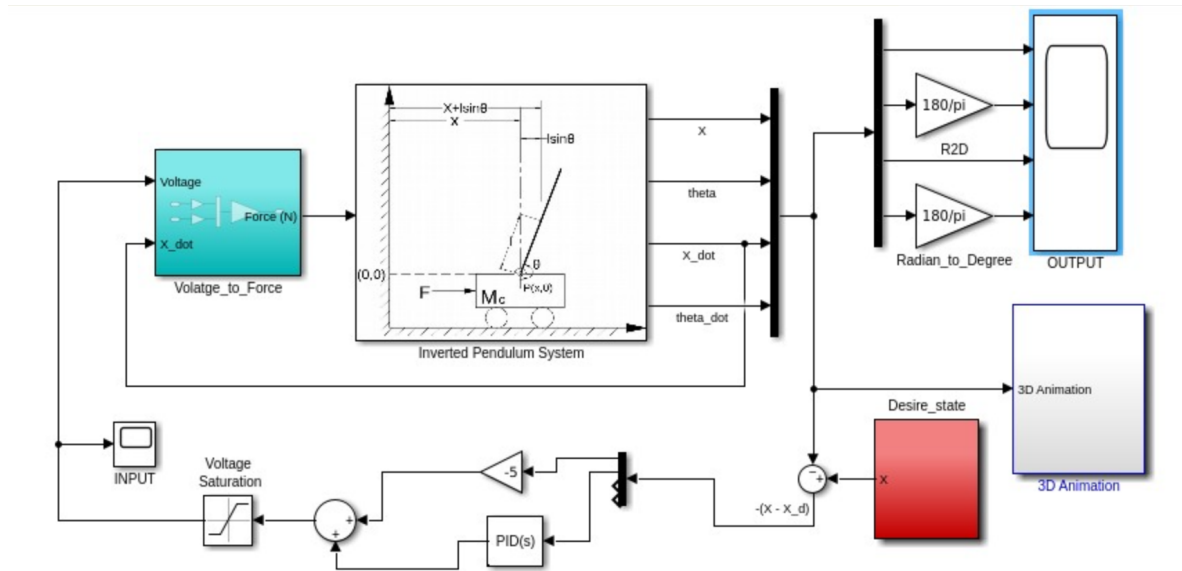


Figura 12: Controlador PID + control proporcional

Ajuste de los valores:

Debido a que ya se ajustó el controlador PID anteriormente para el ángulo, simplemente usamos el mismo controlador PID. Mientras tanto, para el valor escalar del controlador proporcional probamos a base de prueba y error hasta haber un valor que nos satisface, en nuestro caso, el -5.

8.2. Resultados del controlador PID + control proporcional:

Los resultados obtenidos de la implementación de este controlador para el sistema de péndulo invertido se presentan en la Figura 13. Este conjunto de gráficos ilustra el comportamiento del sistema en términos de desplazamiento del carro (X), ángulo del péndulo (θ), velocidad del carro (\dot{X}) y velocidad angular del péndulo ($\dot{\theta}$) a lo largo del tiempo.

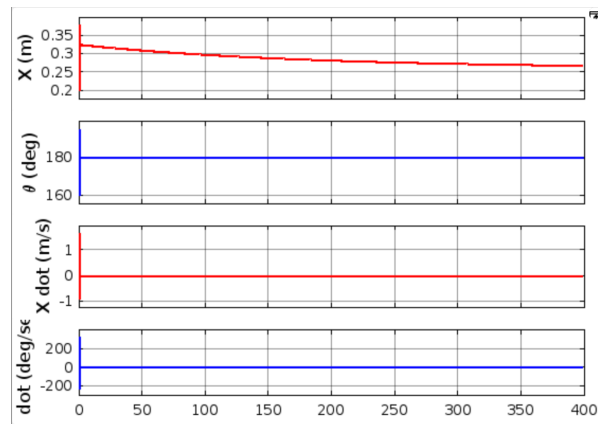


Figura 13: Resultado del controlador PID y proporcional

- **Desplazamiento (X):** La posición del carro muestra una variación mínima pero constante hacia el 0 sin llegar a ver si se estabiliza o no pese a una t de 400 lo que nos muestra que el control proporcional puede no ser suficiente para un buen control.
- **Ángulo (θ):** El ángulo del péndulo se estabiliza rápidamente y se mantiene constante en 180 grados, lo que demuestra que el controlador PID para el ángulo es capaz de estabilizar el péndulo en la posición vertical deseada sin desviaciones muy rápidamente. Esta estabilidad angular es crucial para el éxito del sistema de péndulo invertido, y el resultado obtenido refleja un excelente desempeño del controlador para el control del ángulo.
- **Velocidad del Carro (\dot{X}):** La velocidad del carro se mantiene cercana a cero durante toda la simulación tras un pico inicial (primera acción de control), indicando que no hay necesidad de correcciones bruscas de velocidad para mantener la posición deseada tras esta primera fase. Esto sugiere una buena sintonización del controlador, que logra mantener la estabilidad con mínima intervención.
- **Velocidad Angular ($\dot{\theta}$):** Similarmente, la velocidad angular del péndulo es cero, lo que implica que el péndulo no está oscilando y se mantiene en un equilibrio perfecto. Esta estabilidad angular complementa la del ángulo y es un resultado directo de un control PID bien ajustado. Podemos además añadir que aunque no podamos ver que la posición finalmente se acabe estabilizando podemos suponer que finalmente lo hará ya que el desplazamiento es muy pequeño, la velocidad es 0 y el ángulo es estable por lo que no hace falta ninguna intervención y por lo tanto acabará deteniéndose.

9. EXTENSIÓN 2 - Control proporcional del péndulo invertido

En esta sección, exploramos la implementación de un controlador proporcional para gestionar la dinámica de un péndulo invertido. El objetivo es evaluar la efectividad de un controlador proporcional (K) en la estabilización del sistema.

9.1. Implementación en Simulink

La implementación consiste en un sencillo controlador proporcional que multiplica la matriz de errores por un escalar (el error de cada uno de los parámetros).

9.1.1. Ajuste del valor proporcional

El ajuste de la ganancia proporcional K_p se realizó mediante un enfoque de prueba y error. Empezamos con valores bajos de K_p y los incrementamos gradualmente hasta observar una respuesta razonable del sistema. En nuestro caso, un valor de $K_p = 2,35$ proporcionó una respuesta inicial que parecía estabilizar el ángulo del péndulo, aunque con limitaciones evidentes.

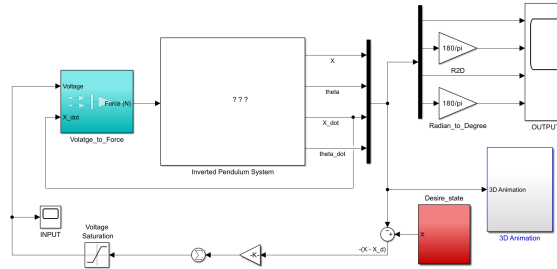


Figura 14: Enter Caption

9.2. Resultados del control proporcional

Los resultados obtenidos con el controlador proporcional se muestran en la Figura 15.

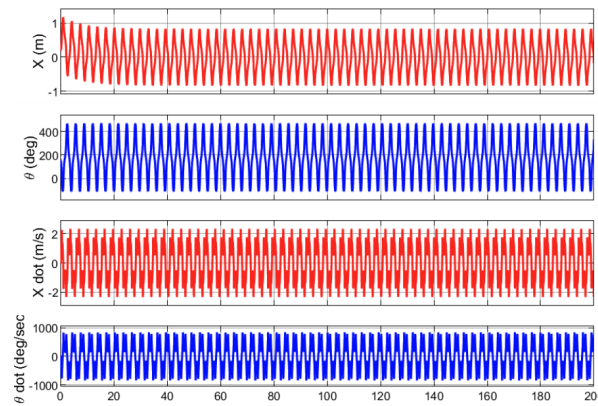


Figura 15: Enter Caption

- **Desplazamiento (X):** La posición del carro exhibe oscilaciones significativas alrededor de la posición de equilibrio, sin alcanzar una estabilización clara. Esto indica que la simple aplicación de un controlador proporcional no es suficiente para llevar el sistema a un estado estacionario.

- **Ángulo (θ):** El ángulo del péndulo se aproxima a la posición vertical deseada, pero con oscilaciones persistentes. Aunque el controlador logra acercar el ángulo a 180 grados, no consigue eliminar las oscilaciones ni mantener una estabilidad continua.
- **Velocidad del Carro (\dot{X}):** La velocidad del carro también presenta variaciones continuas, reflejando la ineficacia del controlador proporcional para estabilizar la posición del carro de manera sostenible.
- **Velocidad Angular ($\dot{\theta}$):** La velocidad angular del péndulo sugiere oscilaciones que no se amortiguan adecuadamente, lo que es consistente con el comportamiento observado en el ángulo del péndulo. Esto confirma que el control proporcional no es suficiente para lograr la estabilidad deseada en el sistema de péndulo invertido.

9.3. Conclusiones sobre el control proporcional

Los resultados demuestran que un controlador proporcional simple no es adecuado para estabilizar completamente el sistema de péndulo invertido. Las oscilaciones persistentes en la posición del carro y el ángulo del péndulo indican que la multiplicación del error por una constante no proporciona el control necesario para mantener el sistema en equilibrio.

10. Discusión

En esta sección se analiza en profundidad el desempeño de los diferentes métodos de control implementados para el sistema del péndulo invertido. El análisis se centra en la efectividad de los controladores PID, LQR y LQG, así como en las extensiones propuestas, evaluando sus capacidades para estabilizar el péndulo invertido y manejar las incertidumbres y perturbaciones.

10.1. Desempeño de los Controladores

10.1.1. Controlador PID

El controlador PID, que combina acciones proporcionales, integrales y derivativas, ha demostrado ser una herramienta eficaz para estabilizar sistemas dinámicos simples. En el contexto del péndulo invertido, el controlador PID fue capaz de mantener el ángulo del péndulo cerca de la posición vertical deseada con una respuesta razonablemente rápida. Sin embargo, el PID mostró ciertas limitaciones en términos de:

- **Oscilaciones Residuales:** Aunque el PID logró estabilizar el sistema, se observaron oscilaciones residuales en la posición del carro y el ángulo del péndulo. Estas oscilaciones indican una falta de amortiguación suficiente, lo que podría afectar la estabilidad a largo plazo.
- **Sensibilidad al Ruido:** El controlador PID es relativamente sensible al ruido en las mediciones, lo que puede conducir a ajustes erráticos y una mayor dificultad para mantener el equilibrio en presencia de perturbaciones.
- **Ajuste de Parámetros:** La necesidad de sintonizar manualmente los parámetros K_p , K_i y K_d para obtener una respuesta óptima puede ser un proceso tedioso y no siempre garantiza el mejor desempeño en todas las condiciones operativas.

10.1.2. Regulador Cuadrático Lineal (LQR)

El LQR se basa en la optimización de un índice de costo cuadrático y ha mostrado un rendimiento superior en la estabilización del péndulo invertido en comparación con el PID. Los principales puntos a destacar sobre el LQR son:

- **Estabilidad y Rapidez de Respuesta:** El LQR proporcionó una estabilización más rápida y precisa del sistema, logrando reducir significativamente las oscilaciones y manteniendo las variables de estado cerca de sus valores deseados.
- **Robustez:** Este controlador mostró una mayor robustez frente a perturbaciones externas y ruido en las mediciones, gracias a su diseño basado en la minimización de un criterio de rendimiento que penaliza tanto el error de estado como el esfuerzo de control.
- **Simplicidad en el Ajuste:** A diferencia del PID, el LQR no requiere un ajuste manual exhaustivo de los parámetros. La elección adecuada de las matrices Q y R permite obtener un desempeño óptimo de forma más sistemática.

10.1.3. Controlador Lineal Cuadrático Gaussiano (LQG)

El LQG, que combina el LQR con el Filtro de Kalman para estimar los estados del sistema a partir de mediciones ruidosas, tenía el potencial de ofrecer el control más robusto. Sin embargo, en nuestra implementación, presentó desafíos significativos:

- **Inestabilidad:** A pesar de la teoría que respalda al LQG, nuestra implementación no logró estabilizar el sistema del péndulo invertido. Se observaron oscilaciones significativas en todas las variables de estado, lo que indica una inestabilidad persistente.
- **Problemas de Sintonización:** La dificultad para sintonizar adecuadamente las matrices Q y R del Filtro de Kalman, así como la integración de la estimación de estados con el control, puede haber contribuido a la falta de desempeño esperado.

- **Complejidad del Modelo:** La complejidad inherente del sistema del péndulo invertido y la aproximación lineal utilizada para el diseño del LQG pueden no haber capturado adecuadamente las dinámicas no lineales del sistema, afectando negativamente el rendimiento del controlador.

10.2. Extensiones Propuestas

10.2.1. Control PID del Ángulo y Control Proporcional de la Posición

Esta extensión combinó un controlador PID para el ángulo del péndulo con un control proporcional para la posición del carro. Los resultados mostraron que, aunque el controlador PID fue efectivo para mantener el ángulo del péndulo estable, el control proporcional no logró estabilizar completamente la posición del carro. Esto sugiere que un control más sofisticado es necesario para manejar de manera efectiva ambas variables.

10.2.2. Control Proporcional del Sistema

La implementación de un control proporcional simple para todo el sistema demostró ser insuficiente para la estabilización del péndulo invertido. Las oscilaciones continuas y la falta de amortiguación adecuada indicaron que este enfoque no es viable para sistemas dinámicos complejos como el del péndulo invertido.

11. Conclusión

En este trabajo, hemos investigado y evaluado diversos enfoques de control para estabilizar el sistema dinámico del péndulo invertido, un problema clásico en la teoría de control debido a su naturaleza inherentemente inestable y no lineal. A través de la implementación y análisis de los controladores PID, LQR y LQG, así como de las extensiones propuestas, hemos logrado obtener una visión completa de las fortalezas y limitaciones de cada método en el contexto del péndulo invertido.

Logros Clave

Uno de los logros más significativos de este estudio ha sido la implementación exitosa de los controladores PID y LQR, que demostraron ser capaces de estabilizar el péndulo invertido de manera efectiva bajo ciertas condiciones. El controlador PID, a pesar de sus oscilaciones residuales, fue útil para manejar la estabilización básica del sistema, mientras que el LQR proporcionó una respuesta más rápida y precisa, destacándose por su robustez y eficiencia en la optimización del costo de control.

El LQG, aunque presentó desafíos en su implementación, mostró el potencial de manejar tanto el control como la estimación de estados en presencia de ruido. La capacidad del Filtro de Kalman para proporcionar estimaciones precisas en entornos ruidosos es particularmente valiosa, aunque requiere una sintonización más fina para alcanzar el rendimiento esperado en sistemas dinámicos complejos.

Contribuciones y Valor del Estudio

Este trabajo ha contribuido significativamente al entendimiento de cómo se pueden aplicar y evaluar diferentes técnicas de control óptimo en sistemas dinámicos inestables. La comparación directa entre los métodos clásicos y los avanzados ha proporcionado una base sólida para comprender mejor las condiciones bajo las cuales cada controlador es más efectivo.

Reflexiones sobre la Robustez y la Adaptabilidad

La robustez de los controladores ante el ruido y las perturbaciones es un aspecto crítico que ha sido abordado de manera diversa por los métodos evaluados. Mientras que el LQR ha demostrado una notable resistencia a las perturbaciones externas, el LQG ofrece un camino prometedor para integrar la estimación de estados con el control en sistemas con incertidumbres significativas. La capacidad del LQG para adaptarse a condiciones cambiantes del entorno y manejar el ruido en las mediciones es una característica que merece una exploración más profunda y refinamiento en futuras investigaciones.

Impacto y Aplicabilidad

Los resultados de este estudio tienen implicaciones importantes para el diseño y la implementación de sistemas de control en una amplia gama de aplicaciones industriales y tecnológicas. La capacidad de estabilizar sistemas inestables como el péndulo invertido es crucial en la robótica, la aviación y muchas otras áreas donde la precisión y la estabilidad son esenciales. Las metodologías y los conocimientos obtenidos aquí proporcionan un marco valioso para abordar desafíos similares en sistemas dinámicos complejos en el futuro.

Referencias

- [1] Ercros: Ercros s.a. (2024), <https://www.ercros.es/ca>
- [2] Nations, U.: La asamblea general adopta la agenda 2030 para el desarrollo sostenible (2015), <https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible/>, consulted on 27 May 2024

A. Código MATLAB para la Estabilidad y Observabilidad

```
1
2 % Definicion de los parametros del sistema
3 m = 1;           % Masa del pendulo
4 M = 1;           % Masa del carro
5 l = 0.5;         % Longitud del pendulo
6 g = 10;          % Aceleracion debido a la gravedad
7 c = 0.5;         % Coeficiente de friccion del carro
8 b = 0.0001;      % Coeficiente de friccion del pendulo
9 I = 0.0007;      % Momento de inercia del pendulo
10 alfa = pi / 2;   % Angulo del pendulo
11
12 % Definicion de la matriz A del sistema
13 A = [0 0 1 0;
14       0 0 0 1;
15       0 (m^2*l^2*g)/(alfa) -(I+m*l^2)*c)/(alfa) -(b*m*l)/(alfa);
16       0 (m*g*l*(M+m))/(alfa) -(m*l*c)/(alfa) -b*(M+m)/(alfa)];
17
18 % Definicion de la matriz B del sistema
19 B = [0;
20       0;
21       (I+m*l^2)/(alfa);
22       (m*l)/(alfa)];
23
24 % Definicion de la matriz C del sistema
25 C = [1 0 0 0;
26       0 1 0 0;
27       0 0 1 0;
28       0 0 0 1];
29
30 % Mostrar los valores propios de la matriz A
31 disp(eig(A));
32
33 % Calcular la matriz de controlabilidad
34 cont = ctrb(A, B);
35
36 % Calcular el rango de la matriz de controlabilidad
37 rango = rank(cont);
38 disp(rango);
39
40 % Calcular la matriz de observabilidad
41 O = obsv(A, C);
42
43 % Calcular el rango de la matriz de observabilidad
44 rango_obs = rank(O);
45 disp(rango_obs);
```

Listing 1: Código MATLAB para la Estabilidad y Observabilidad

B. Código MATLAB calcular QR

```
1
2 % Estado deseado
3 desiredState = [0, pi, 0, 0]; % Modifica seg n tus requerimientos
4
5 % Valores iniciales de Q y R para la optimizaci n
6 initialParams = [300, 500, 0, 0, 0.0035]; % Inicial Q y R
7 lb = [0, 0, 0, 0, 0.00035]; % L mites inferiores
8 ub = [3000, 3000, 0, 0, 5]; % L mites superiores
9
10 % Configuraci n de la optimizaci n
11 options = optimoptions('fmincon', ...
12     'Algorithm', 'sqp', ...           % Algoritmo de optimizaci n
13     'Display', 'iter', ...           % Mostrar progreso iterativo
14     'MaxFunctionEvaluations', 1000, ... % M ximas evaluaciones de funci n
15     'FiniteDifferenceStepSize', 100, ... % Tama o de paso de diferencia finita
16     'StepTolerance', 0.01, ...       % Tolerancia de paso reducida
17     'OptimalityTolerance', 0.01);    % Tolerancia de optimalidad reducida
```



```

18
19 % Llamada a fmincon para optimizar Q y R
20 optimalParams = fmincon(@(params) lqrCostFunction(params, A, B, modelName,
    desiredState), initialParams, [], [], [], [], lb, ub, [], options);
21
22 % Valores ptimos de Q y R
23 Q_optimal = diag(optimalParams(1:4));
24 R_optimal = optimalParams(5);
25
26 % Mostrar los resultados ptimos
27 disp('Valores ptimos de Q:');
28 disp(Q_optimal);
29 disp('Valor ptimeo de R:');
30 disp(R_optimal);
31
32 % Cerrar el modelo de Simulink al finalizar
33 close_system(modelName, 0);
34
35 % Definir la funci n de costo
36 function cost = lqrCostFunction(params, A, B, modelName, desiredState)
37     try
38         % Extraer Q y R de los par metros
39         Q = diag(params(1:4));
40         R = params(5);
41
42         % Verificar que Q y R sean adecuadas
43         if any(diag(Q) < 0)
44             error('Todos los elementos de Q deben ser no negativos.');
```

```

45         end
46         if R <= 0
47             error('R debe ser positiva.');
```

```

48         end
49
50         % Calcular la ganancia LQR K
51         K = lqr(A, B, Q, R);
52
53         % Asignar K al espacio de trabajo de MATLAB para que Simulink lo use
54         assignin('base', 'KK', K);
55
56         % Configurar el tiempo de simulaci n
57         simulationTime = 30; % Duraci n de la simulaci n en segundos
58
59         % Ejecutar la simulaci n
60         simOut = sim(modelName, 'SimulationMode', 'normal', 'SaveOutput', 'on', '
    SaveTime', 'on', 'StopTime', num2str(simulationTime));
61
62         % Verificar las se ales disponibles en la salida de simulaci n
63         signalNames = simOut.who;
64         disp('Se ales disponibles en la salida de simulaci n:');
65         disp(signalNames);
66
67         % Extraer los datos de las se ales correctas
68         % Actualiza estos nombres seg n las se ales correctas en tu modelo
69         realState = simOut.get('realState'); % Ajusta seg n tu modelo
70         desiredStateSim = simOut.get('desiredState'); % Ajusta seg n tu modelo
71
72         % Asegurarse de que los datos sean matrices
73         realState = realState.Data;
74         desiredStateSim = desiredStateSim.Data;
75
76         % Extraer los valores al final de la simulaci n
77         finalRealState = realState(end, :); % ltimo estado real
78         finalDesiredState = desiredStateSim(end, :); % ltimo estado deseado
79
80         % Calcular el costo como la diferencia entre el estado real y el deseado
81         cost = norm(finalRealState - finalDesiredState); % Usa la norma para evaluar
    la diferencia
82
83     catch ME
84         % En caso de error, mostrar el mensaje y devolver un costo alto
85         disp(['Error en lqrCostFunction: ', ME.message]);

```

```
86     cost = 1e10; % Valor alto para indicar fallo en la evaluaci n
87     end
88 end
```

Listing 2: Código MATLAB calcular QR