Universidad Nacional Mayor de San Marcos.
Facultad de Ingeniería de Sistemas e Informática.

PROYECTO FINAL - G8

Automatización y control de Software

Estudiantes:

- Jauregui Diaz, Yajahira Ysabel
- Escudero Principe, Álvaro
- Vasquez Gonzales, Pedro Sebastian
- Sotelo Arce, Jocelyn Estrella
- Collachagua Poma, Airton Wilson
- Victoria Escudero, Raul Antonio

27/11/2024

Semestre 2024-II

Proyecto 1. Control moderno del Péndulo Invertido

1. Introducción

2. Marco teórico

2.1. Modelado del Péndulo Invertido

- **Sistema dinámico no lineal:** El péndulo invertido es un sistema que ejemplifica la no linealidad debido a la presencia de términos como senos y cosenos en sus ecuaciones de movimiento. Esto lo convierte en un desafío para estrategias de control, y se utiliza frecuentemente para probar técnicas modernas debido a su naturaleza inherentemente inestable.
- **Ecuaciones de movimiento:** Se derivan mediante dos enfoques principales las Leyes de Newton y el Método de Lagrange

2.2. Técnicas de Control

- **Control PID:** El controlador Proporcional-Integral-Derivativo (PID) ajusta la respuesta del sistema en función de tres componentes: el error presente (P), acumulado (I), y la tasa de cambio del error (D). Aunque generalmente se aplica a sistemas lineales, puede usarse en el péndulo invertido si este se linealiza alrededor del punto de equilibrio.
- **Control en espacio de estados:** Analizando y diseñando controles mediante matrices que describen el sistema.

2.3. Simulación y Herramientas de Diseño

- **Simulación con MATLAB y Simulink:** Son herramientas estándar para modelar, simular y evaluar el comportamiento de sistemas dinámicos. Permiten verificar estrategias de control antes de su implementación física, ahorrando tiempo y reduciendo errores.
- **Lugar geométrico de las raíces:** Es un método gráfico que muestra cómo las raíces del sistema varían con respecto a los cambios en los parámetros, como la ganancia del controlador. Es útil para analizar la estabilidad y ajustar el diseño del controlador

3. Analisis y explicacion

4. Desarrollo de Actividades

la rayita ¯ es un punto y ^ *es dos puntos, no sé como ponerlo :, v*

4.1. Deriva la función de transferencia del sistema a partir de las ecuaciones (3) y (4). Calcula las raíces del sistema para determinar su estabilidad inicial sin control.

Para derivar la función de transferencia, inicialmente vamos a definir la transformada de Laplace, que es la relación entre la salida y la entrada del sistema, bajo condiciones iniciales nulas. Se expresa de la siguiente forma:

$$G(s) = \frac{Laplace\ (salida)}{Laplace\ (entrada)} = \frac{\theta(s)}{U(s)}$$

Donde:

- θ(s) : Transformada de Laplace de salida $\theta(t)$, que es el ángulo de inclinación del péndulo
- $U(s)$: Transformada de Laplace de la entrada $u(t)$, que es la fuerza aplicada al sistema

Las ecuaciones brindadas que son parte del comportamiento del péndulo son:

(Rotación del Péndulo) (3) $\widehat{Ml\theta} = (M + m)g\theta - u$

(Movimiento lineal del carrito) (4) $\widehat{Mx} = u - mg\theta$

Para derivar la función transferencia, utilizamos la ecuación (3):

$$\widehat{Ml\theta} = (M + m)g\theta - u$$

$$Ml(s^2\theta(s) - s\theta(0) - \overline{\theta}(0)) = (M + m)g\theta(s) - U(s)$$

Se imponen condiciones iniciales nulas, es decir:

$$\theta(0) = 0\,;\ \bar{\theta}(0) = 0$$

donde la ecuación queda de la siguiente manera:

$$Ml(s^2\theta(s)) = (M + m)g\theta(s) - U(s)$$

Despejamos el valor de $U(s)$:

$$Ml(s^2\theta(s)) - (M + m)g\theta(s) = -U(s)$$

Factorizamos $\theta(s)$:

$$\theta(s)(Mls^2 - (M + m)g) = -U(s)$$

Finalmente obtenemos la función de transferencia:

$$\theta(s)(Mls^2 - (M + m)g) = -U(s)$$

$$\frac{\theta(s)}{-U(s} = \frac{1}{(Mls^2 - (M+m)g)}$$

$$\frac{\theta(s)}{-U(s} = \frac{1}{Ml(s^2 - \frac{(M+m)g}{Ml})}$$

Aplicamos diferencia de cuadrados $a^2 - b^2 = (a + b)(a - b)$

$$\frac{\theta(s)}{-U(s} = \frac{1}{Ml(s + \sqrt{\frac{(M+m)g}{Ml}})(s - \sqrt{\frac{(M+m)g}{Ml}})}$$

Obteniendo las raíces del sistema, que son las siguientes:

$$S = -\sqrt{\frac{(M+m)g}{Ml}}$$

$$S = +\sqrt{\frac{(M+m)g}{Ml}}$$

Cuando las raíces del sistema tienen una parte real positiva, esto indica que el sistema es inestable sin control, ya que cualquier perturbación inicial crecerá

con el tiempo. En este caso, encontramos dos raíces: **una positiva y otra negativa**. La raíz positiva confirma que el sistema, en su estado actual sin control, no puede mantenerse estable por sí solo. Por lo tanto, es necesario implementar un controlador para garantizar su estabilidad y funcionamiento adecuado.

4.2. Diseña un controlador PID que permita mantener el péndulo en posición vertical. Simula el comportamiento del sistema con cada tipo de controlador (P, PI, PD y PID). ¿Cómo influyen los parámetros individuales Kp,Ki y Kd en la respuesta del sistema? ¿Qué diferencias observas en el comportamiento del sistema al usar controladores P, PI, PD y PID?

Variables del Sistema :

- M: Masa del carro
- m: Masa del péndulo
- l: Longitud del péndulo
- g: Aceleración de la gravedad

Función de transferencia ya hallada del Sistema:

$$\frac{1}{Ml(s^2 - \frac{(M+m)g}{Ml})}$$

Controlador P:

$$CP(s) = Kp$$

Controlador PI:

$$CPI(s) = Kp + Ki/S$$

Controlador PD:

$$CPD(s) = Kp + Ki * S$$

Controlador PID:

$$CPID(s) = Kp + Ki/S + Kd * S$$

Sistema Cerrado con retroalimentación:

El sistema de control implementado es de retroalimentación negativa, lo que significa que la salida del sistema (el ángulo del péndulo) se retroalimenta al controlador para corregir cualquier error.

Teniendo en cuenta esos detalles se logra a hacer el simulador en python:

1. Primero se definen los parámetros requeridos del sistema

```python
class ControladorPID:
    """Clase que encapsula la lógica de simulación del sistema con diferentes controladores."""

    def __init__(self):
        # Valores por defecto
        self.M = 1.0  # Masa del carro
        self.m = 0.1  # Masa del péndulo
        self.l = 1.0  # Longitud del péndulo
        self.g = 9.81  # Gravedad
        self.Kp = 50  # Ganancia proporcional
        self.Ki = 1   # Ganancia integral
        self.Kd = 10  # Ganancia derivativa

    def actualizar_parametros(self, M, m, l, g, Kp, Ki, Kd):
        """Actualiza los parámetros del sistema y las ganancias del controlador."""
        self.M = M
        self.m = m
        self.l = l
        self.g = g
        self.Kp = Kp
        self.Ki = Ki
        self.Kd = Kd
```

2. Se configura la función de transferencia del péndulo invertido.

```python
# Definición del sistema abierto
num = [1]
den = [self.M * self.l, 0, -(self.M + self.m) * self.g]
sys_open = ctrl.TransferFunction(num, den)
```

3. Se definen los cuatro tipos de controladores (P, PI, PD y PID)

```python
sys_p = self.Kp
sys_pi = self.Kp + self.Ki / ctrl.TransferFunction([1], [1, 0])
sys_pd = self.Kp + self.Kd * ctrl.TransferFunction([1, 0], [1])
sys_pid = self.Kp + self.Ki / ctrl.TransferFunction([1], [1, 0]) + self.Kd * ctrl.TransferFunction([1, 0], [1])
```

4. Se usa un sistema cerrado con retroalimentación para analizar la estabilidad y la respuesta del sistema con el controlador

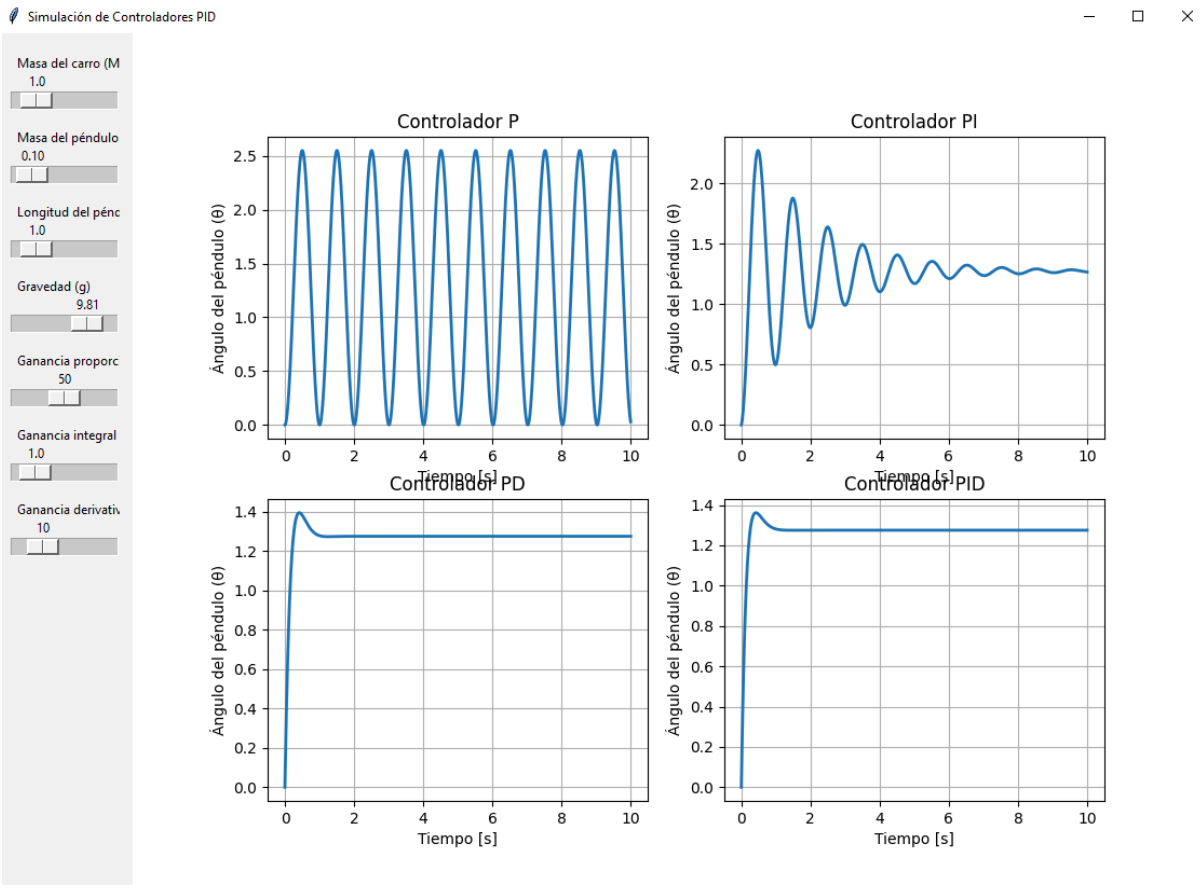
```python
sys_p_closed = ctrl.feedback(sys_p * sys_open)
sys_pi_closed = ctrl.feedback(sys_pi * sys_open)
sys_pd_closed = ctrl.feedback(sys_pd * sys_open)
sys_pid_closed = ctrl.feedback(sys_pid * sys_open)
```

5. Simula la respuesta al escalón del sistema bajo cada controlador.

```
# Tiempo de simulación
t = np.linspace(0, 10, 1000)

# Respuestas al escalón
_, y_p = ctrl.step_response(sys_p_closed, t)
_, y_pi = ctrl.step_response(sys_pi_closed, t)
_, y_pd = ctrl.step_response(sys_pd_closed, t)
_, y_pid = ctrl.step_response(sys_pid_closed, t)
```

6. Por último se grafican las respuestas para poder analizarlas



Los parámetros Kp, Ki y Kd influyen de manera diferente en las características dinámicas del sistema, teniendo un impacto significativo en las respuestas.

Kp:

El Kp multiplica el error actual por un factor constante, aumentar este reduce el error en estado estacionario pero si es demasiado alto puede provocar oscilaciones en el sistema o causar inestabilidad

Ki:

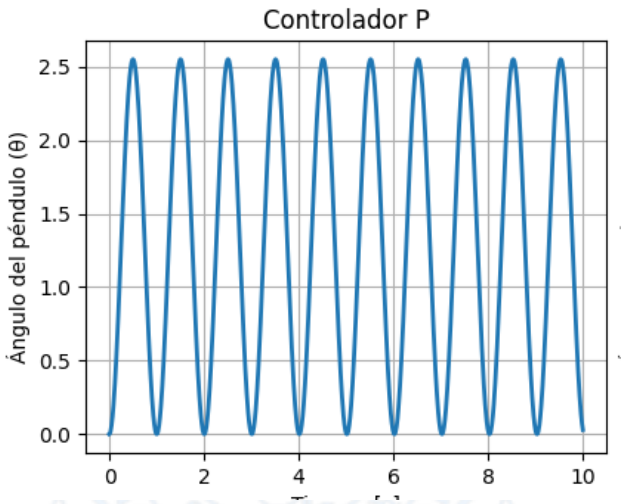
El ki acumula el error a lo largo del tiempo , si es demasiado alto puede causar oscilaciones persistentes debido a una acumulación excesiva de error .

Kd:

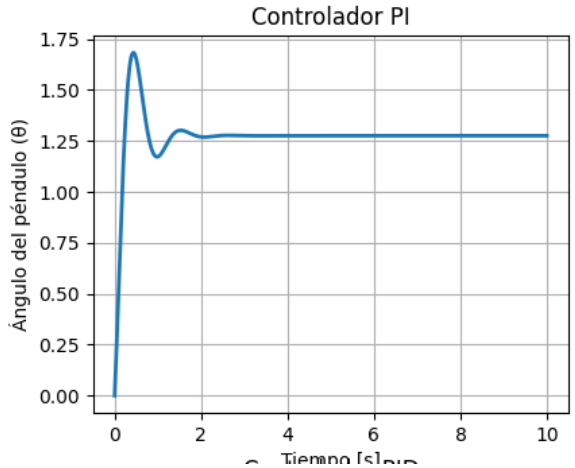
Calcula la tasa de cambio del error, si es demasiado alto puede amplificar el ruido de alta frecuencia presente en el sistema

El comportamiento de los sistemas al usar distintos controladores :

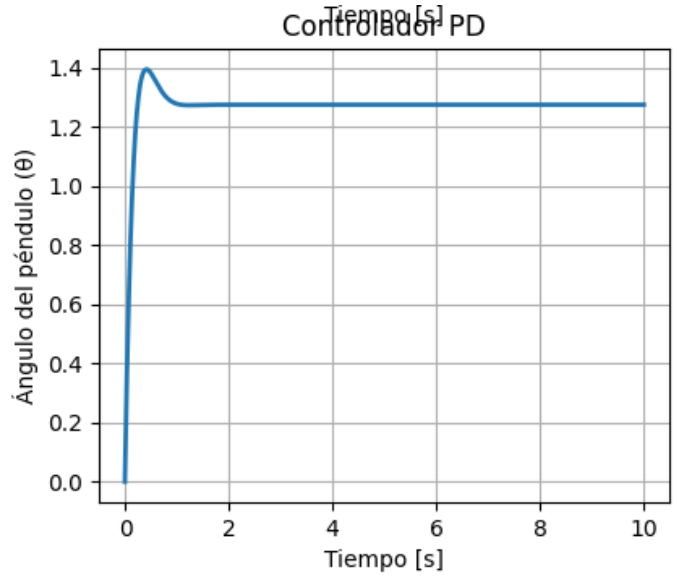
- Controlador P: Genera oscilaciones constantes, por lo que no es adecuado para sistemas que requieren estabilidad.



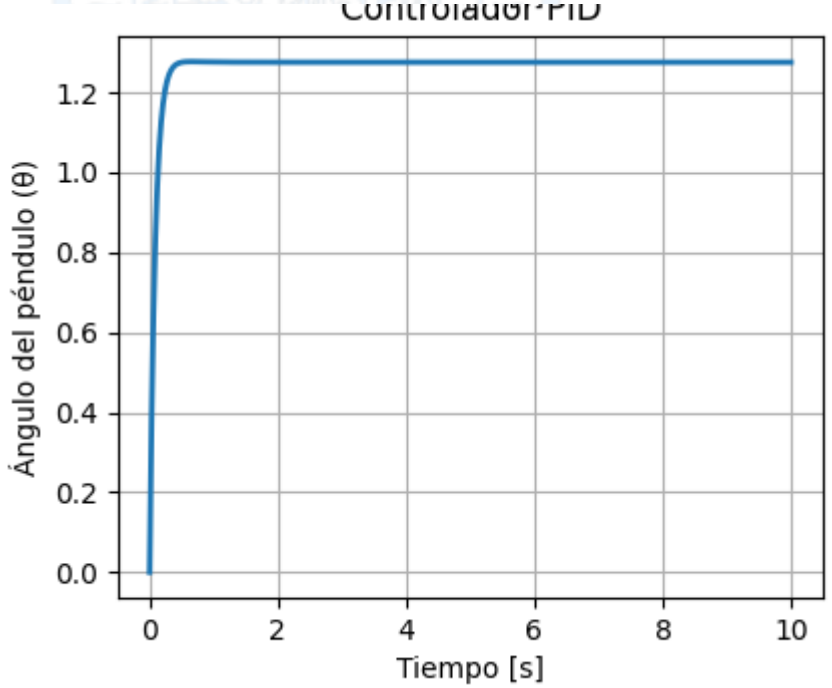
- Controlador PI: Mejora la precisión eliminando el error en estado estacionario, pero introduce oscilaciones temporales.



- Controlador PD: Controla las oscilaciones rápidamente, pero no corrige el error en estado estacionario.



- Controlador PID: Combina las ventajas de PI y PD, logrando una respuesta equilibrada con estabilidad, rapidez y precisión. Es la mejor opción para un sistema que busca minimizar el sobre impulso y alcanzar la referencia de forma rápida.



4.3. Analiza cómo cambia el desempeño del sistema al integrar los algoritmos genéticos para optimizar los parámetros Kp,Ki y Kd del controlador PID. Compara el desempeño del PID optimizado frente a uno ajustado manualmente, destacando sus ventajas y limitaciones.

4.4. Genera gráficos que permitan visualizar las dos leyes de control diseñadas (PID ajustado manualmente y PID optimizado), evaluando el comportamiento del sistema en términos de tiempo de estabilización y sobre impulso.

4.4.1. ¿Qué tan sensible es el sistema con el controlador PID (manual u optimizado) ante variaciones en las condiciones iniciales, como un ángulo mayor del péndulo o una posición inicial desplazada del carro?

4.4.2. Después de la optimización, ¿qué cambios observas en la curva de error del sistema?

4.4.3. ¿Qué diferencias importantes se identifican en el comportamiento del sistema con el controlador optimizado respecto a los ajustes manuales?

5. Bibliografía

1. Cañas, D.; Henao, P y Vidales, F. (2016) *Diseño, Elaboración e Implementación de un Control De Pendulo Invertido para el Laboratorio de P.L.C*. Informe Final de Trabajo de Grado - Institución Universitaria ITM - Colombia https://repositorio.itm.edu.co/bitstream/handle/20.500.12622/6047/Control_Pendulo_Invertido_Ca%C3%B1as_Henao_Vidales_2016.pdf?sequence=1
2. Saqib, Ii; Liangyu, Z; Safeer, U. Adee, M. y Muhammad, F. *Control strategies for inverted pendulum: A comparative analysis of linear, nonlinear, and artificial intelligence approaches*. (2024) PLOS ONE. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0298093
3. Slotine J, y Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, New Jersey. https://lewisgroup.uta.edu/ee5323/notes/Slotine%20and%20Li%20applied%20nonlinear%20control-%20bad%20quality.pdf
4. Ogata, K. (2010). Modern Control Engineering (3ra edición). Upper Saddle River, New Jersey: Prentice Hall. https://plcsitemiz.wordpress.com/wp-content/uploads/2009/03/modern-control-engineering-kogata-3rd-edition.pdf
5. Tinoco Romero, R. y Orces, E. (2004) *Modelado, Simulación y Control de un Péndulo Invertido usando componentes análogos simples* - Tesis de Grado - California Institute of Technology - USA https://www.dspace.espol.edu.ec/bitstream/123456789/6118/36/CICYT.pdf
6. Triviño Macias, L. (2020) *Modelado, simulación y control de un péndulo invertido* - Trabajo de Fin de Grado en Ingeniería Electrónica de Telecomunicaciones - Universidad Autonoma de Barcelona - España https://ddd.uab.cat/pub/tfg/2020/234238/TFG_LuisGeovannyTrivinoMacias.pdf
7.