

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

El péndulo invertido es uno de los sistemas más fascinantes y desafiantes en el campo de la ingeniería y el control. Este sistema dinámico, que desafía la intuición al mantenerse erguido en un equilibrio inherentemente inestable, ha sido durante décadas un laboratorio perfecto para probar y perfeccionar técnicas de control avanzadas. Desde aplicaciones industriales hasta la robótica moderna, el estudio del péndulo invertido no solo es teóricamente enriquecedor, sino que también sienta las bases para resolver problemas prácticos complejos en sistemas no lineales.

Este proyecto aborda el modelado, simulación y control de un péndulo invertido, guiando al lector desde sus fundamentos matemáticos hasta la implementación de estrategias de control como PID y espacio de estados. A través del uso de herramientas como MATLAB y Python, se analizará el impacto de diversos controladores en la estabilidad del sistema, explorando su comportamiento ante diferentes escenarios.

Además, se busca proporcionar una experiencia integral que combine teoría y práctica, permitiendo que los lectores comprendan tanto los retos técnicos como las soluciones innovadoras necesarias para estabilizar este sistema dinámico. El proyecto no solo promete un aprendizaje técnico profundo, sino que también invita a reflexionar sobre cómo el control y la estabilidad son esenciales en un mundo donde la tecnología avanza a pasos agigantados.

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.

2.4. Implementación Física

Actuadores y sensores:

La implementación física del péndulo invertido requiere componentes como:

- Motores DC: Para generar las fuerzas necesarias.
- Encoders: Para medir posición angular y lineal.

Estos sensores y actuadores permiten capturar datos en tiempo real y aplicar señales de control.

3. Desarrollo de Actividades

- 3.1. *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:

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

Donde:

- $\theta(s)$: 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) $Ml\ddot{\theta} = (M + m)g\theta - u$

(Movimiento lineal del carrito) (4) $M\dot{x} = u - mg\theta$

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

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

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

Se imponen condiciones iniciales nulas, es decir:

$$\theta(0) = 0 ; \dot{\theta}(0) = 0$$

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.

3.2. 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 K_p, K_i y K_d 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

```
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.

```
# 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)

Jk

```
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

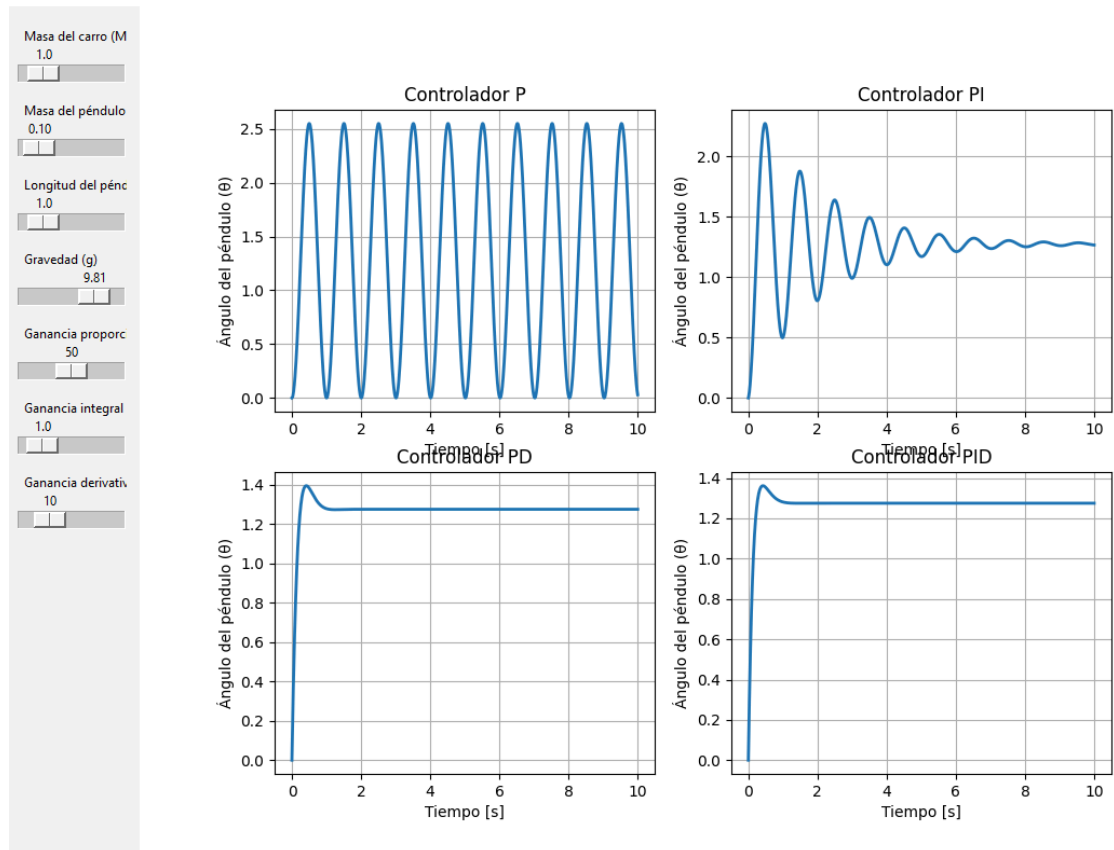
```
sys_p_closed = ctrl.feedback(sys_p * sys_open)
sys_pi_closed = ctrl.feedback(sys_pi * sys_open)
sys_pd_closed = ctrl.feedback(sys_pd * sys_open)
sys_pid_closed = ctrl.feedback(sys_pid * sys_open)
```

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 K_p , K_i y K_d influyen de manera diferente en las características dinámicas del sistema, teniendo un impacto significativo en las respuestas.

K_p :

El K_p 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

K_i :

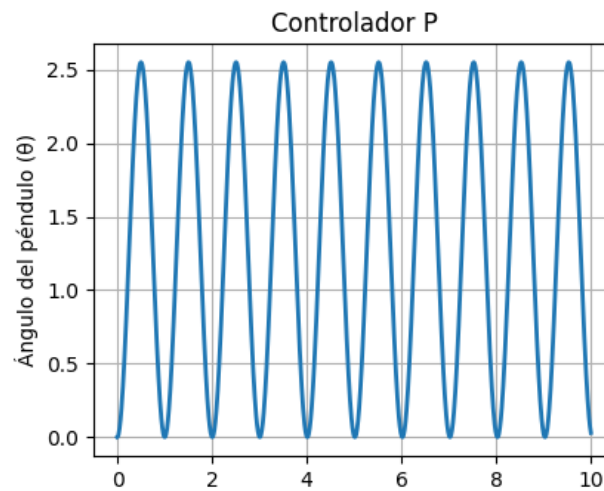
El k_i acumula el error a lo largo del tiempo, si es demasiado alto puede causar oscilaciones persistentes debido a una acumulación excesiva de error.

K_d :

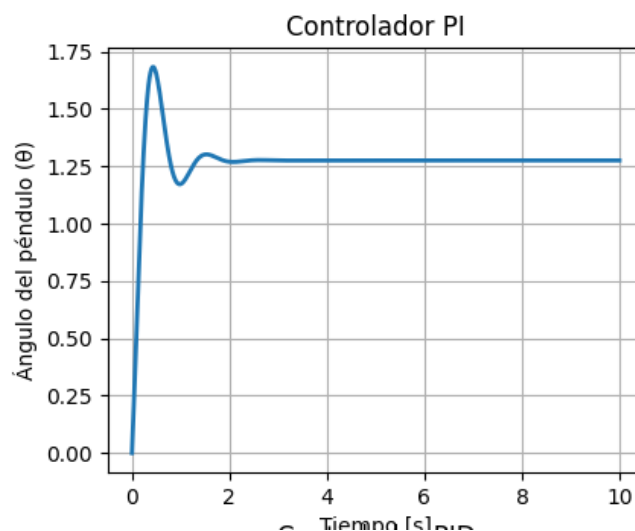
Calcula la tasa de cambio del error, si es demasiado alto puede amplificar el ruido de alta frecuencia presente en el sistema

El comportamiento de los sistemas al usar distintos controladores :

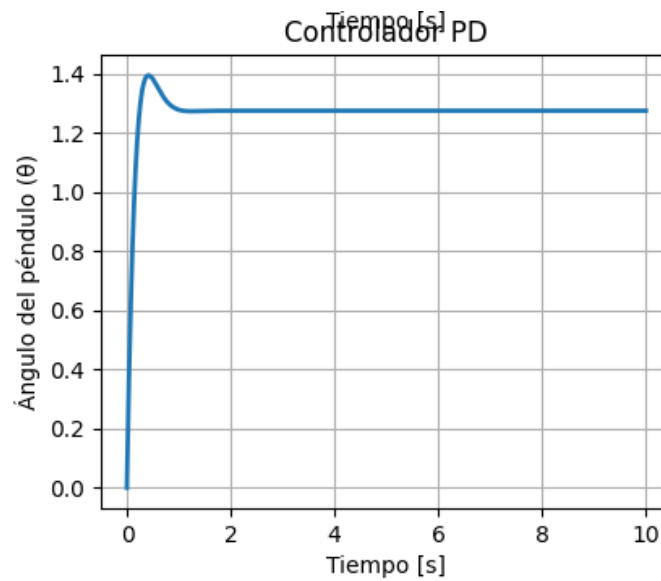
- Controlador P: Genera oscilaciones constantes, por lo que no es adecuado para sistemas que requieren estabilidad.



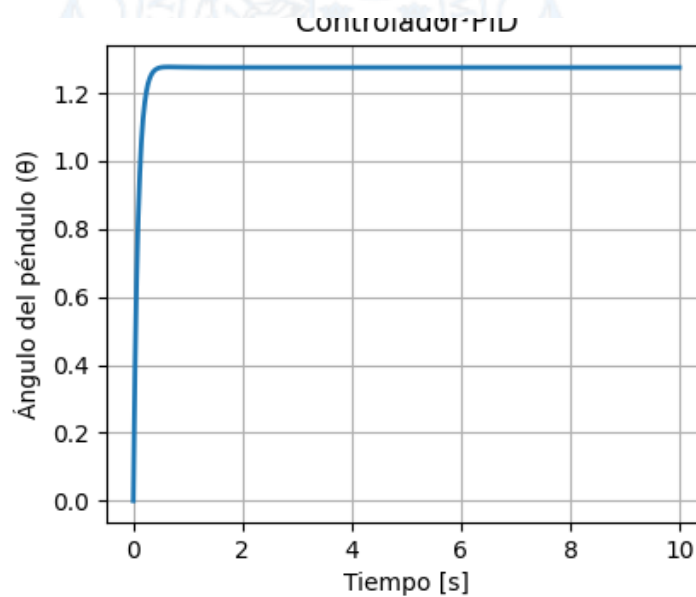
- Controlador PI: Mejora la precisión eliminando el error en estado estacionario, pero introduce oscilaciones temporales.



- 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.



3.3. *Analiza cómo cambia el desempeño del sistema al integrar los algoritmos genéticos para optimizar los parámetros K_p , K_i y K_d del controlador PID. Compara el*

desempeño del PID optimizado frente a uno ajustado manualmente, destacando sus ventajas y limitaciones.

Controlador PID:

Parámetros del manual:

```
modo == "manual":
    Kp, Ki, Kd = self.Kp, self.Ki, self.Kd
```

Resultados:

- Flexibilidad Inicial: El ajuste manual permite cambios rápidos basados en el criterio del ingeniero, siendo adecuado para sistemas simples o bien entendidos.
- Dificultades en Sistemas Complejos: Para sistemas con dinámicas complicadas o múltiples variables interdependientes, el ajuste manual suele ser insuficiente y no logra alcanzar un rendimiento óptimo.
- Dependencia de la Pericia: Este método exige una gran experiencia técnica, ya que errores como inestabilidad o excesivo sobreimpulso son comunes si no se realiza correctamente.

Controlador PID Optimizado:

Parámetros del optimizado:

```
modo == "genetico" and valores_optimos:
    Kp, Ki, Kd = valores_optimos
```

Resultados:

- Precisión Superior: Los algoritmos genéticos realizan una exploración exhaustiva del espacio de parámetros, identificando combinaciones que minimizan métricas específicas de desempeño, como ITAE o ISE. Esto permite optimizar la relación entre rapidez, sobreimpulso y estabilidad del sistema.
- Consistencia Garantizada: Los parámetros óptimos se obtienen siguiendo un enfoque objetivo, eliminando influencias subjetivas y garantizando resultados consistentes y fiables.

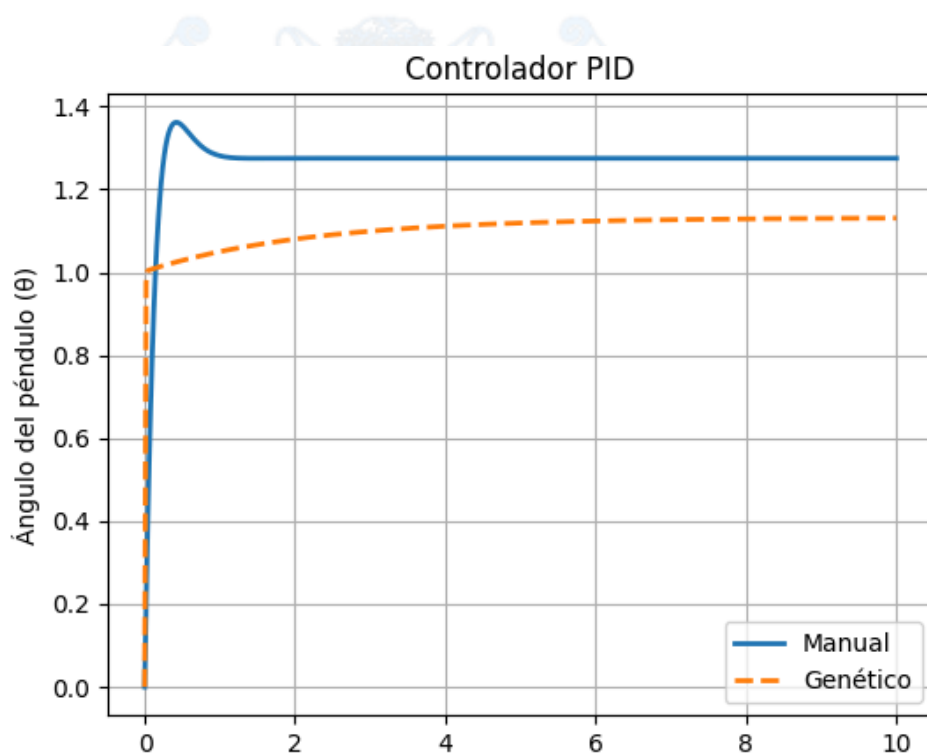
- Adaptación a Sistemas Complejos: En sistemas no lineales o con interacciones complejas, donde el ajuste manual presenta dificultades, el AG encuentra soluciones que no serían evidentes para un ingeniero, maximizando el desempeño del controlador.

Comparativo:

Manual: $K_p:50$, $K_i:1$, $K_d:1$

Optimizado:

- Mejor individuo: [92.30763339601725, 79.89074102995312, 61.538945401990276]
- Mejor fitness: 0.000342



	PID - AJUSTE MANUAL	PID - OPTIMIZADO
Desempeño General	Funcional, pero menos preciso.	Mejor desempeño con respuestas más rápidas y estables.
Estabilidad	Más propenso a oscilaciones y tiempos de estabilización largos.	Estable con menos oscilaciones y un tiempo de estabilización más corto.
Sobreimpulso	Mayor sobreimpulso, lo que afecta la precisión.	Menos sobreimpulso, mejorando la precisión en el control.

Tiempo de Respuesta	Más lento y menos eficiente en la estabilización.	Respuesta más rápida y eficiente.
Aplicación en Sistemas Complejos	No siempre efectivo en sistemas complejos.	Ideal para sistemas complejos donde se requiere un control preciso.

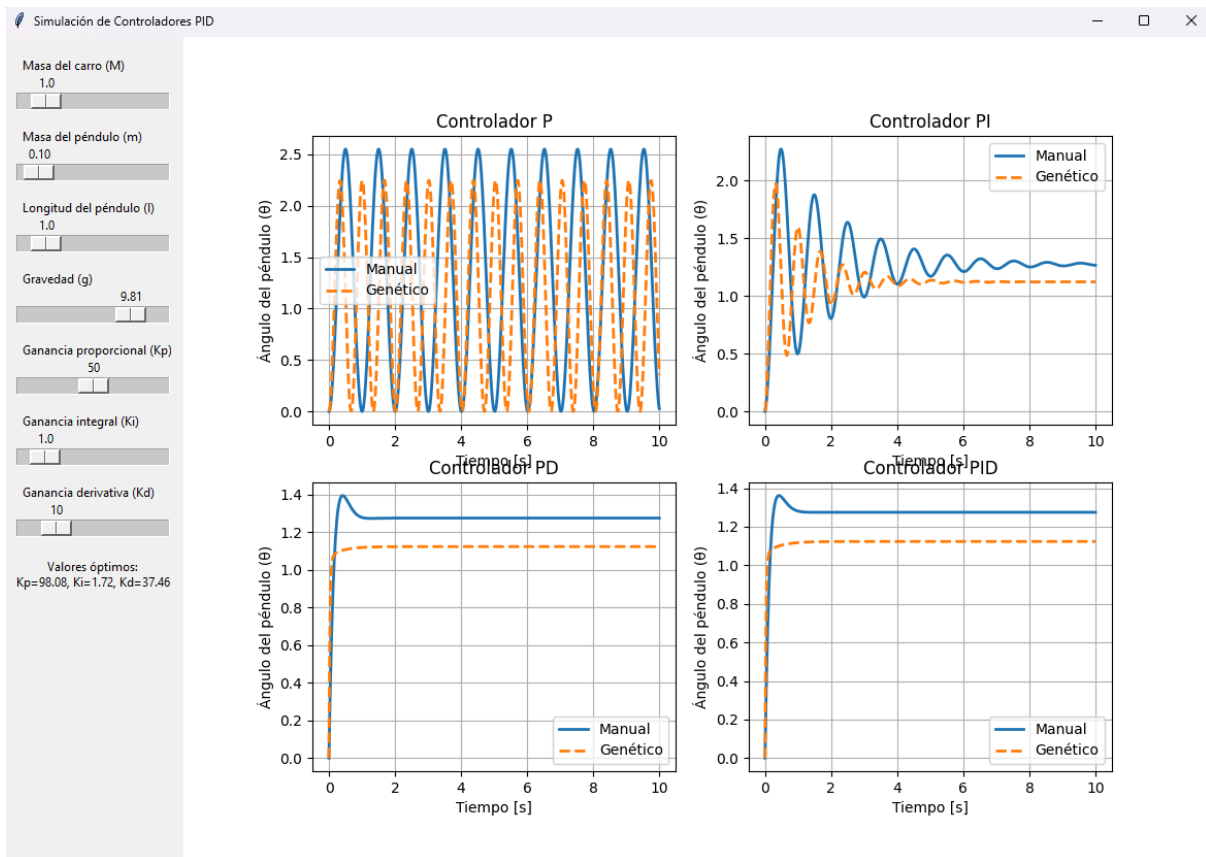
La optimización de los parámetros K_p , K_i y K_d usando algoritmos genéticos demuestra ser una herramienta valiosa y eficaz para mejorar el desempeño de los controladores PID, especialmente en sistemas complejos que demandan un control más preciso y estable.

3.4. *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.*

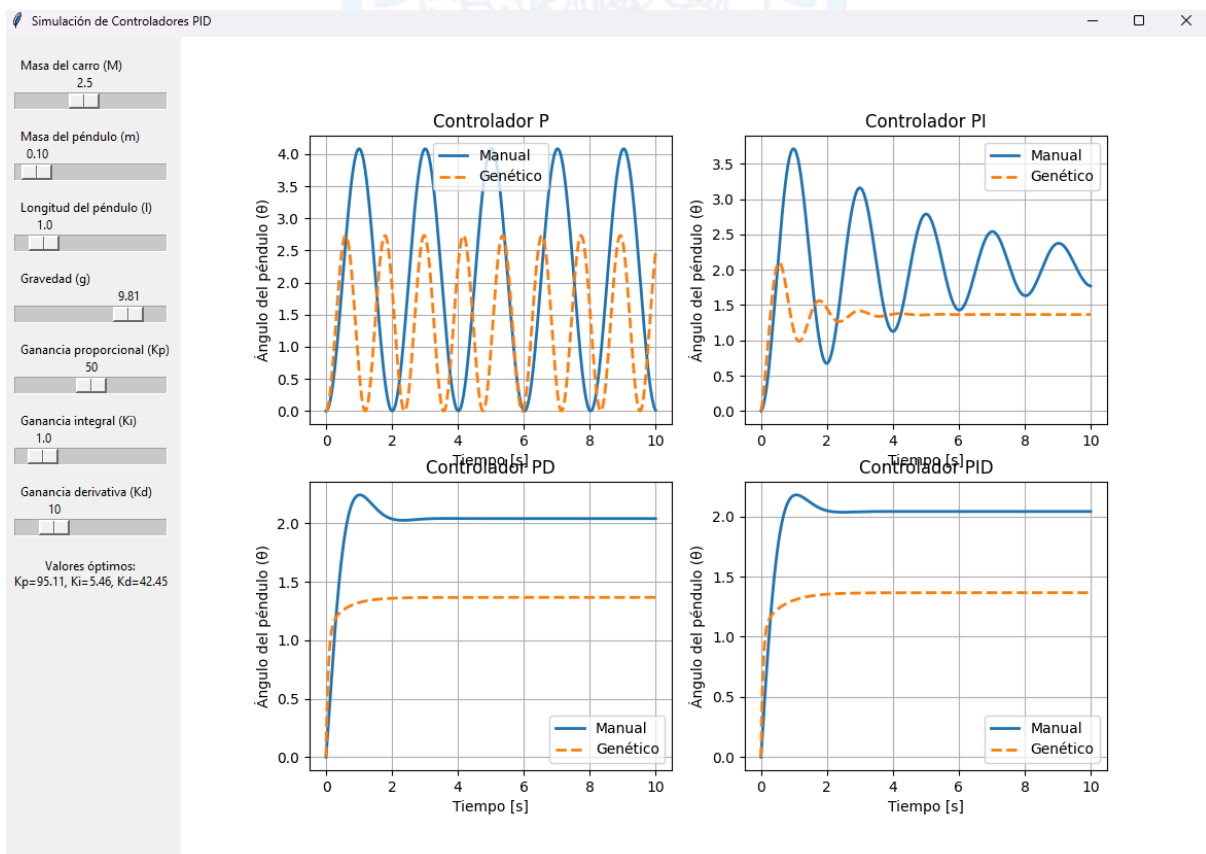
3.4.1. ¿Qué tan sensible es el sistema con el controlador PID (manual u optimizado) ante variaciones en las condiciones iniciales, como un ángulo mayor del péndulo o una posición inicial desplazada del carro?

Para responder a esta pregunta se harán 3 pruebas, la primera con una masa inicial de 1.0, la segunda con 2.3 y la tercera con una masa de 3.5 en el móvil, para finalizar se muestra la gráfica generada del controlador PID tanto manual como optimizado.

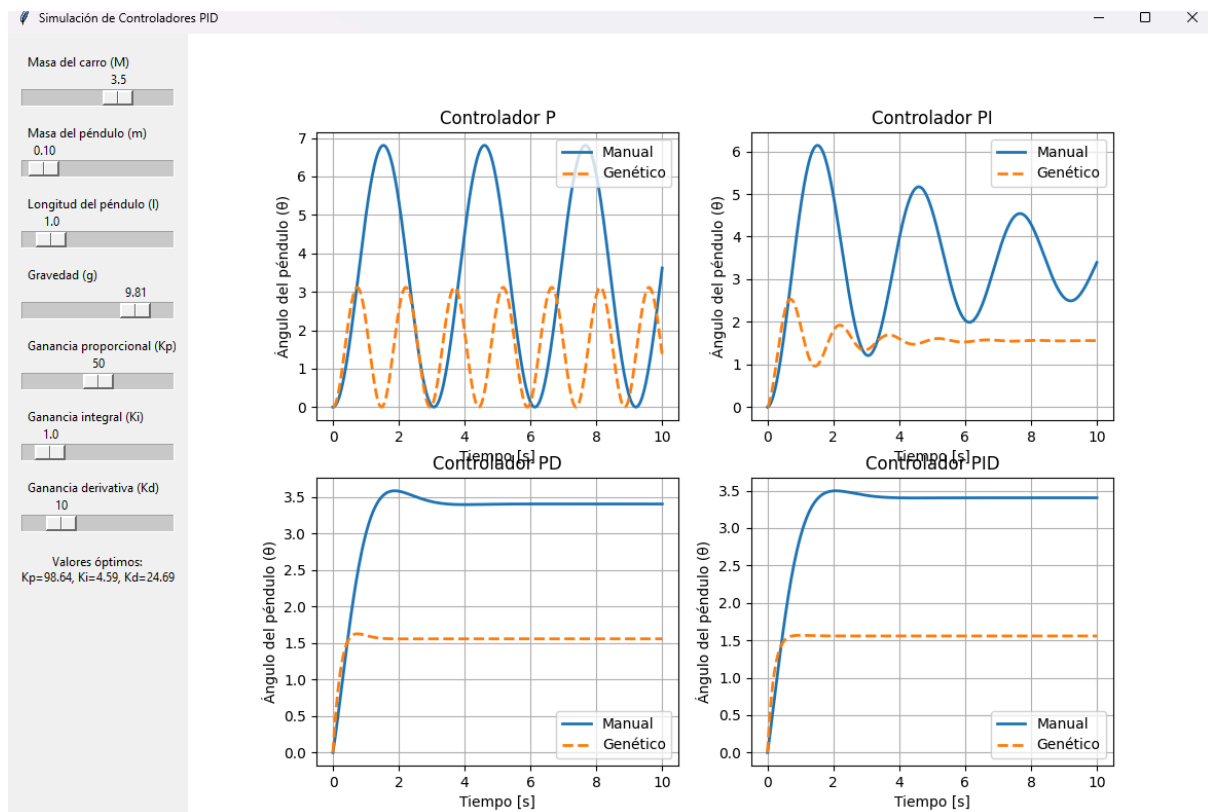
Prueba 1: Masa 1.0



Prueba 2: Masa 2.5



Prueba 3: Masa 3.5



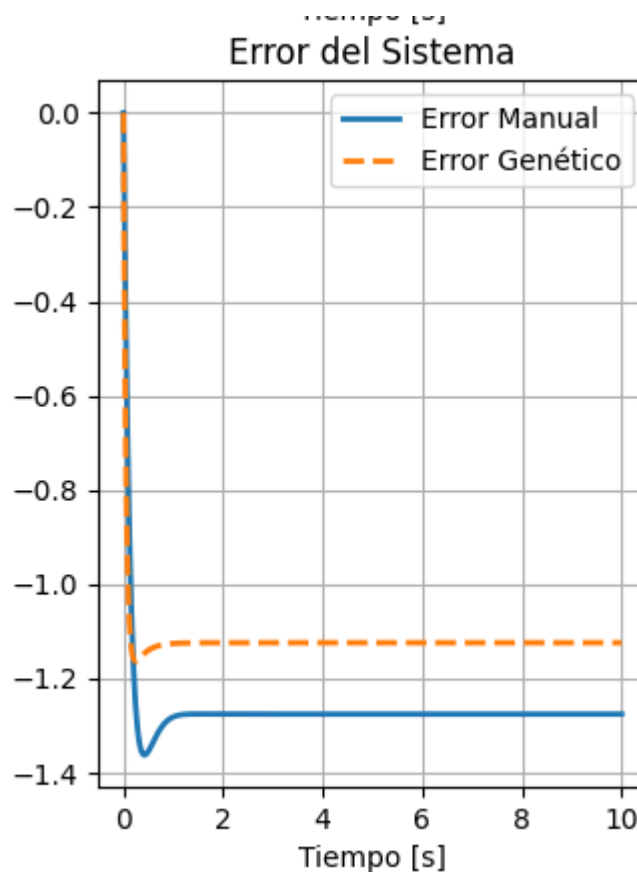
Análisis acerca de las distintas gráficas en relación a la masa del móvil: Se observa que el ángulo de estabilización y el tiempo en el que se estabiliza incrementa conforme la masa del móvil aumenta, esto se explica dado que M afecta directamente a la fórmula sobre la cuál se hacen los cálculos. Uno de los factores en la función de transferencia es M , lo que afecta a la inercia del sistema.

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

El incremento del valor de M , implica que se incremente la fuerza para estabilizar el móvil lo que provoca que el sistema tarde más tiempo en estabilizarse y que las amplitudes iniciales sean más grandes.

Se puede notar un mayor manejo en el ángulo y tiempo de estabilización cuando se usan algoritmos genéticos, ya que buscan los valores óptimos dependiendo de las variables que se están usando en la fórmula de transferencia.

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



- En la curva de Error Genético se ve que no presenta oscilaciones iniciales a comparación del manual, también muestra más estabilidad y menos propenso a comportamientos erráticos en el sistema, esto quiere decir que el péndulo no se desviará de forma abrupta al intentar mantenerse recto. Otro punto que se ve es que alcanza un estado estable en menos tiempo.
 - En la curva del Error Manual se ve que presenta oscilaciones más altas al inicio lo cual puede generar un comportamiento más inestable haciendo que el péndulo tarde más en alcanzar el equilibrio, Otro punto que se ve es que tarda más en alcanzar un estado estable.
- 3.4.3. ¿Qué diferencias importantes se identifican en el comportamiento del sistema con el controlador optimizado respecto a los ajustes manuales?

Estas diferencias pueden categorizarse en aspectos de desempeño, estabilidad y adaptabilidad:

- Desempeño General

Controlador Ajustado Manualmente: Aunque funcional, este tipo de controlador tiende a ser menos preciso y eficiente. Las respuestas del sistema suelen ser más lentas, con un mayor tiempo de estabilización y un desempeño global menos consistente.

Controlador Optimizado: Utilizando algoritmos genéticos, el PID optimizado realiza una búsqueda exhaustiva en el espacio de parámetros, identificando combinaciones óptimas de K_p , K_i , y K_d . Esto resulta en un desempeño significativamente mejor, con respuestas más rápidas y precisas.

- Estabilidad

Controlador Ajustado Manualmente: Es más propenso a generar oscilaciones, especialmente si los valores de los parámetros no son los ideales. Esto puede llevar a largos tiempos de estabilización o incluso a la inestabilidad del sistema.

Controlador Optimizado: Garantiza una mayor estabilidad, reduciendo las oscilaciones iniciales y permitiendo una convergencia más suave hacia el equilibrio. Esto se debe a que los algoritmos genéticos ajustan los parámetros de manera que se minimicen métricas como el sobreimpulso y el error.

- Sobreimpulso

Controlador Ajustado Manualmente: Generalmente presenta un sobreimpulso elevado, lo que afecta negativamente la precisión del sistema y puede generar esfuerzos adicionales en los actuadores.

Controlador Optimizado: Minimiza el sobreimpulso al buscar valores de K_d y K_p que controlen eficazmente la oscilación inicial, mejorando la precisión en el seguimiento de la referencia.

- Adaptación a Sistemas Complejos

Controlador Ajustado Manualmente: En sistemas con múltiples interacciones o dinámicas no lineales, los ajustes manuales suelen ser insuficientes para alcanzar un rendimiento óptimo. Además, este método depende en gran medida de la experiencia del operador.

Controlador Optimizado: Los algoritmos genéticos permiten encontrar soluciones adaptativas para sistemas complejos, donde el ajuste manual no identifica fácilmente las combinaciones ideales de

parámetros. Esto asegura un control más robusto frente a variaciones en las condiciones iniciales o perturbaciones externas.

- Tiempo de Respuesta

Controlador Ajustado Manualmente: El sistema responde más lentamente y presenta ineficiencias en la estabilización.

Controlador Optimizado: Reduce significativamente el tiempo de respuesta, estabilizando el sistema más rápidamente incluso ante condiciones iniciales desfavorables.

4. Conclusiones

1. El estudio y control del péndulo invertido permitió profundizar en conceptos fundamentales de sistemas no lineales y técnicas de control modernas. Su comportamiento dinámico inestable lo convierte en un modelo ideal para probar y validar estrategias de control en la ingeniería.
2. El controlador PID, en su configuración básica, ofreció una solución funcional para estabilizar el péndulo invertido, pero con limitaciones evidentes como el sobreimpulso y oscilaciones iniciales. Al optimizar los parámetros K_p , K_i y K_d mediante algoritmos genéticos, se alcanzó un nivel de desempeño notablemente superior. El controlador optimizado logró reducir el tiempo de respuesta, menor sobreimpulso y mejor robustez ante perturbaciones. De esta manera demostrando que la combinación de estrategias clásicas como el PID con técnicas modernas de optimización puede resolver limitaciones prácticas y garantizar un control más eficiente y confiable.
3. Los algoritmos genéticos destacaron como una herramienta poderosa para sistemas complejos. Proporcionaron soluciones óptimas y consistentes para el ajuste de parámetros, superando las limitaciones del ajuste manual, especialmente en escenarios con condiciones iniciales desafiantes o interacciones no lineales.
4. El análisis de sensibilidad mostró cómo los cambios en las condiciones iniciales (como la masa del móvil) afectan la respuesta del sistema. A mayor masa, se observaron tiempos de estabilización más largos y mayores amplitudes iniciales, evidenciando la necesidad de ajustar los parámetros del controlador para mantener el equilibrio.

5. Bibliografía

- 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%Blas_Henao_Vidales_2016.pdf?sequence=1
- Saqib, Ii; Liangyu, Z; Safeer, U. Adeer, 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>
- 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>
- 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>
- 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>
- Triviño Macías, L. (2020) *Modelado, simulación y control de un péndulo invertido* - Trabajo de Fin de Grado en Ingeniería Electrónica de Telecomunicaciones - Universidad Autónoma de Barcelona - España
https://ddd.uab.cat/pub/tfg/2020/234238/TFG_LuisGeovannyTrivinoMacias.pdf