

UNIVERSIDAD MILITAR NUEVA GRANADA

LABORATORIO 1 SISTEMA DE CONTROL DIFUSO

AUTORES:

JULIAN CAMILO TORRES , EST.JULIANC.TORRES@UNIMILITAR.EDU.CO

DOCENTE: HECTOR DANIEL BERNAL

FECHA: 30 DE AGOSTO DE 2025



Resumen

El presente laboratorio tiene como propósito principal el diseño e implementación de un sistema de control difuso aplicado a un modelo de péndulo invertido montado sobre un carro. La actividad se desarrolla en el entorno de programación Python y mediante simulaciones en Webots. Se realiza una comparación entre el comportamiento del modelo matemático propio y el modelo del simulador, aplicando tanto controladores clásicos (PID) como controladores basados en lógica difusa. Adicionalmente, se desarrolla una interfaz gráfica que permite evaluar visualmente el rendimiento de los controladores. Finalmente, se optimizan los parámetros de cada controlador con el fin de minimizar el error de control.

Palabras clave: Péndulo invertido, Control difuso, Controlador PID ,Lógica difusa, Simulación, Webots, Python.

This laboratory focuses on the design and implementation of a fuzzy control system applied to an inverted pendulum model mounted on a cart. The activities are carried out using Python programming and Webots simulations. A comparative analysis is performed between the behavior of a self-developed mathematical model and the simulation model, applying both classical (PID) and fuzzy logic controllers. Additionally, a graphical interface is developed to visually evaluate the controllers' performance. Controller parameters are optimized to minimize the control error.

1. Introducción

En el campo de la ingeniería mecatrónica, el desarrollo de sistemas de control inteligentes permite abordar problemas complejos en los que los métodos tradicionales pueden presentar

limitaciones. Uno de estos problemas clásicos es el control de un péndulo invertido, el cual representa un sistema inherentemente inestable.

El laboratorio tiene como objetivo aplicar técnicas de inteligencia artificial, particularmente el control difuso, para estabilizar dicho sistema. A través del uso de Python y simulaciones en Webots, se busca comparar el desempeño de un controlador clásico PID frente a uno difuso. Esta práctica permite reforzar conceptos de modelado dinámico, lógica difusa, optimización de controladores y visualización interactiva de sistemas de control.

2. Métodos e Implementación

2.1. Procedimiento

A) Semana 1 y 2. Implementar Modelo de péndulo invertido en Python y con simulación

y control manual.

Realizar un simulador de péndulo invertido de masa concentrada en el extremo en python con el eje en un carro móvil . Permitir controlar manualmente el carro y tratar de mantener el péndulo arriba.

Para el modelo de solución mediante métodos numéricos investigar e implementar el modelo dinámico del péndulo invertido. Determinar las variables físicas del péndulo invertido y ejecutar el modelo. Encapsular a nivel de función el modelo de tal forma que se le entreguen las variables iniciales, la señal de entrada inicial y luego la señal de entrada del carro. El modelo debe entregar la nueva ubicación del péndulo.

Animación con control manual en python. Realizar una visualización que permita ejercer el control manual, con magnitudes de las variables en valores cercanos a la realidad y variables de entrada configurables como ángulo, velocidad angular, velocidad del vehículo y demás. Implementar el control de fuerza al carro que sostiene el péndulo y poder controlar la dirección del vehículo , por ejemplo con las teclas “A” y “D” para que el vehículo se mueva en la dirección deseada y el usuario intente mantener el péndulo de forma vertical. Usando el modelo generado en el punto anterior como base física.

Animación con control manual en webots usando python. Realizar una visualización que permita ejercer el control manual, con magnitudes de las variables en valores cercanos a la realidad y variables de entrada configurables como ángulo, velocidad angular, velocidad del vehículo y demás. Implementar el control de fuerza al carro que sostiene el péndulo y poder controlar la dirección del vehículo , por ejemplo con las teclas “A” y “D” para que el vehículo se mueva en la dirección deseada y el usuario intente mantener el péndulo de forma vertical. Usando el modelo físico interno de webots como base física.



Fig. 1: Webots

B) Semana 3, Controlador PID sobre simulación en webots implementado en python.

- Sobre la planta virtual en webots se debe diseñar, acoplar y simular un controlador clásico de forma que el péndulo sea mantenido en equilibrio en un ángulo deseado por un controlador PID automático, se generan las condiciones iniciales y se procede a dejar que el controlador estabilice el péndulo mediante la fuerza aplicada sobre el carro que sostiene el péndulo.
- Optimizar los parámetros del controlador PID para minimizar el error respecto a unas condiciones retadoras de salida, mediante el ajuste de parámetros y la repetición de la simulación(simulación montecarlo) con variación aleatoria de las entradas.

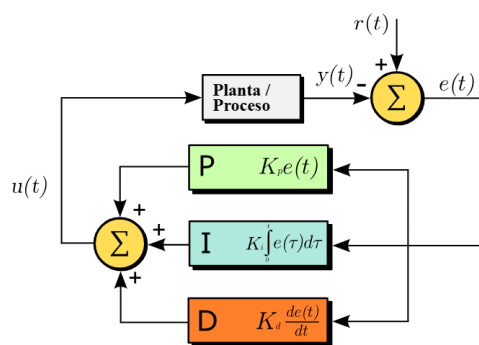


Fig. 2: Control PID

7aC) Semana 5, Controlador Difuso en python

- Implementar conjuntos difusos para el procesamiento de las condiciones del péndulo y tomar acciones sobre el carro de control. Modelar conjuntos difusos de entrada , un motor de inferencia de Mamdani , conjuntos difusos de salida y defusificación , usando python.
- Simular y aplicar el controlador difuso sobre

la planta virtual, de forma que el robot pendulo se mantenga en el angulo deseado a partir del movimiento del vehiculo que lo sostiene, Optimizar los conjuntos difusos para minimizar el error.

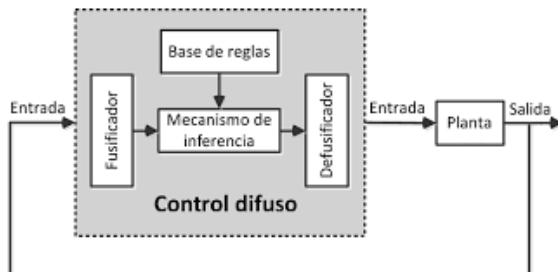
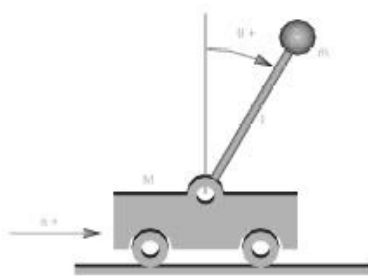


Fig. 3: Control difuso

3. Resultados

Entrega 16 de Agosto:

Para esta practica se inicia con el modelo de un péndulo invertido aplicado a código para poder interpretarlo o simularlo en visual code con el lenguaje de python y en la aplicación de webots usando también el lenguaje de python pero agregándole un nuevo reto de simulación debido a que en esta plataforma se maneja el entorno lo que se llaman ecenas.



Péndulo en un carro.

Fig. 4: Pendulo invertido

Como se puede ver la idea era realizar el modelo de un péndulo invertido matemático que funcionara con los principios del modelo y que claramente se le pudieran aplicar fisicas es por eso que inicialmente se realizo en visual code para facilitar un poco el modelado debido a que era un lenguaje de programación un poco mas conocido y mas fácil de trabajar para poder hacer pruebas en el uso de librerías.

Parámetros del Sistema

Gravedad (g): 9.81 m/s²

Masa péndulo (m): 0.25 kg

Masa carro (M): 0.5 kg

Longitud (L): 0.5 m

Fricción (b): 0.1 N·s/m

Fuerza máxima: 10.0 N

Paso tiempo: 0.02 s

Fig. 5: Valores fisicos

Como se puede ver el modelo matemático inicia teniendo en cuenta factores físicos como lo son la velocidad, gravedad la masa del péndulo masas del carro, longitud de la barra, fricción que tiene la caja con el suelo, fuerza que se aplica cada vez que se mueve y el paso de muestro o tiempo que va ayudar a la tasa de actualización para que podemos animar el comportamiento de este en el paso del tiempo.

Sistema de ecuaciones diferenciales:

$$\ddot{x} = [F - b\dot{x} + mL\ddot{\theta}^2 \sin\theta - mg \sin\theta \cos\theta] / [M + m \sin^2\theta]$$

$$\ddot{\theta} = [g \sin\theta - \cos\theta (F - b\dot{x} + mL\ddot{\theta}^2 \sin\theta) / (M + m \sin^2\theta)] / L$$

Variables de estado:

x_1 = posición del carro (x)
 x_2 = velocidad del carro (\dot{x})
 x_3 = ángulo del péndulo (θ) [rad]
 x_4 = velocidad angular ($\dot{\theta}$) [rad/s]

Representación en espacio de estados:

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = [F - b x_2 + m L x_4^2 \sin x_3 - m g \sin x_3 \cos x_3] / [M + m \sin^2 x_3]$$

$$\frac{dx_3}{dt} = x_4$$

Fig. 6: Modelado

Ya con el modelado se comienza a trabajar para cumplir con el primer reto que es mover el péndulo y hacer un control manual usando las letras del teclado A y D, para poder hacer esto se utilizaron librerías como la de numpy y pygame además de la de matplotlib para poder relacionar el modelo matemático con el péndulo en general.

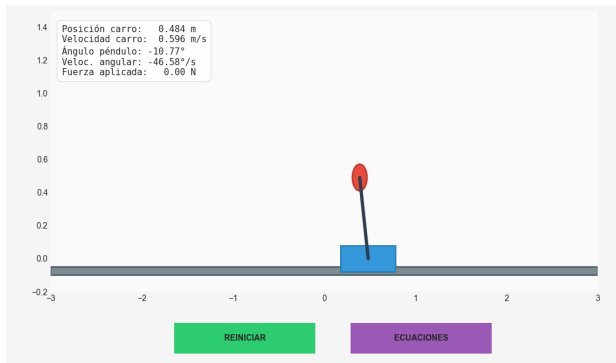


Fig. 7: Resultado final

Ya con esto se puede desarrollar de buena manera esta parte de la practica en resumen se implementa un simulador interactivo de un péndulo invertido usando Python, donde se combinan varias bibliotecas para manejar tanto la física del sistema como la visualización y el control. La lógica física está basada en un modelo matemático de ecuaciones diferenciales que describen el movimiento del carro y del péndulo, integradas numéricamente con de SciPy. La clase principal inicializa parámetros como masas, longitud, gravedad y fricción, define el estado inicial, y establece los elementos gráficos —carril, carro, varilla y masa del péndulo— junto con un panel informativo en tiempo real.

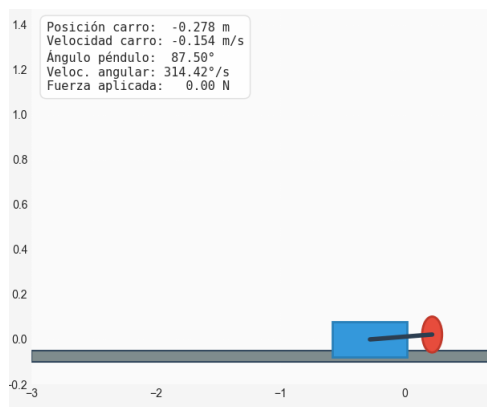


Fig. 8: Resultado final

El funcionamiento es está en la función update, que en cada fotograma resuelve las ecuaciones del sistema y actualiza la posición y el ángulo del péndulo en la animación. La integración numérica calcula cómo cambia el estado en pequeños pasos de tiempo (dt), y esos resultados se traducen en coordenadas gráficas para mover el carro, rotar la varilla y desplazar la masa.

La simulación responde a interacciones del usuario, como presionar las teclas 'a' o 'd' para aplicar fuerzas hacia la izquierda o derecha, logrando que el usuario pueda intentar estabilizar el péndulo. Gracias a matplotlib.animation, se logra un movimiento continuo y realista, mientras que tkinter complementa la experiencia mostrando la teoría y ecuaciones detrás de la simulación.

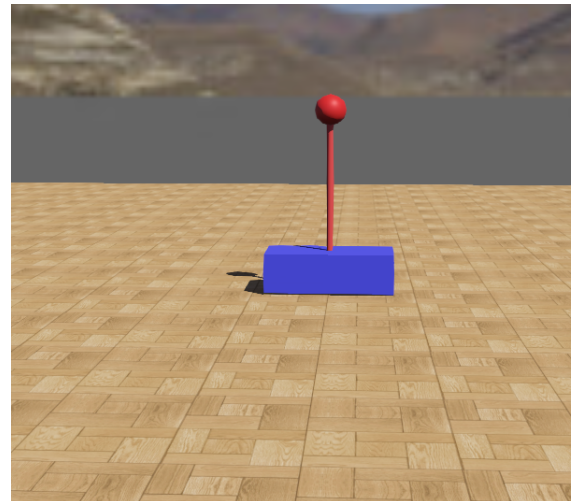


Fig. 9: Resultado webots

Como se ve en la figura anterior, ya se pasó ahora al programa de Webots, donde se encontró con una particularidad que toca usar un archivo wbt que es la extensión de webots pero además de eso toca en la carpeta de controllers colocar un archivo .py que se va a llamar el cual nos va a dar el control del archivo para poder aplicar el mismo modelo que se usó en el trabajo de visual.

```
-----
Tiempo: 24.16s
Posición: -0.537m | Velocidad: -0.045m/s
Ángulo: 107.1° | Vel.Angular: 0.27rad/s
-----
Tiempo: 24.64s
Posición: -0.598m | Velocidad: -0.127m/s
Ángulo: 109.9° | Vel.Angular: 0.10rad/s
-----
Tiempo: 25.12s
Posición: -0.662m | Velocidad: -0.133m/s
Ángulo: 102.2° | Vel.Angular: -0.28rad/s
-----
Tiempo: 25.60s
Posición: -0.717m | Velocidad: -0.114m/s
Ángulo: 113.5° | Vel.Angular: 0.41rad/s
-----
Tiempo: 26.08s
Posición: -0.745m | Velocidad: -0.058m/s
Ángulo: 116.5° | Vel.Angular: 0.11rad/s
-----
Tiempo: 26.56s
Posición: -0.797m | Velocidad: -0.109m/s
Ángulo: 90.9° | Vel.Angular: -0.93rad/s
-----
```

Fig. 10: Impresión de resultados

Ya con esto se puede evidenciar como se imprime el angulo la velocidad pero esta vez en la consola se puede seguir moviendo con las teclas de buena manera se tiene el control de la misma forma con las teclas A y D.

Entrega 23 de Agosto:

Para esta parte ya su usan todos los modelos pero se le agrega un control pid que hace que se pueda mover ya sin usar las teclas y debe mantenerse en el aire el péndulo sin caer y así poder lograr el control la idea de esto es que aguante arto tiempo sin caer.

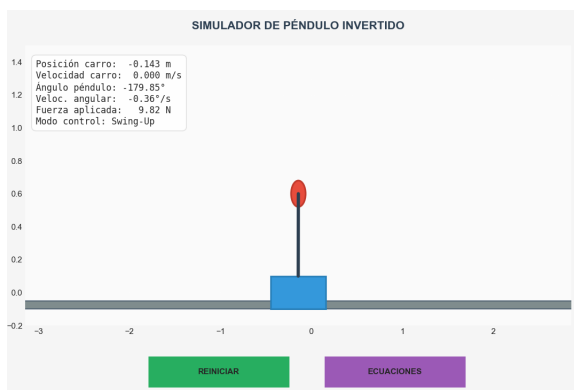


Fig. 11: Controlador en visual

Como se ve se colocan unas constantes con la idea de poder mantener el 0 como regencia que seria noventa grados si se toma el suelo como referencia y así hacer el el proporcional derivativo y integrador mantengan el péndulo sin tener que tocarlo.

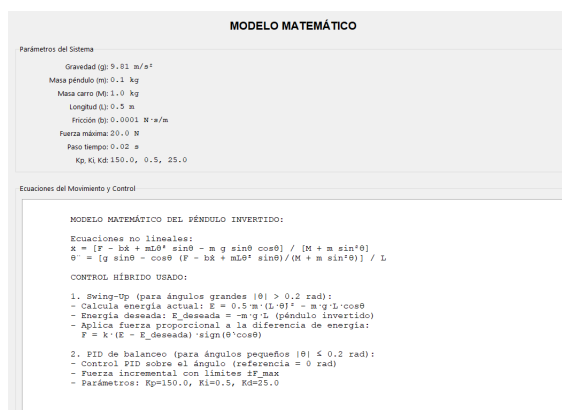


Fig. 12: Evidencia uso del controlador

Este control es un poco mas sencillo que en el de webost debido a que aquí se programaron con

librerías el control ademas de que es un control 3D pero se visualiza en 2D entonces parece que es mejor pero es tema de la visualización.

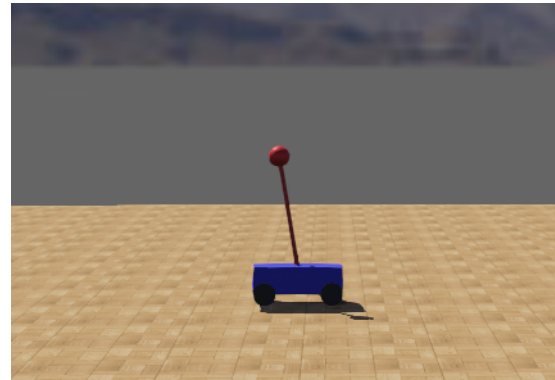


Fig. 13: Controlador en webots

Como para el caso de visual se maneja el mismo modelo solo se ajusta el controlador al diseño de webots con sus características para poder hacer que funcione de igual manera que en de visual code, como se evidencio se logra controlar y dura un poco menos tiempo que en visual debido a que es un poco mas realista es decir las fisicas no solo estan hechas con el modelado si no que se integran con un paack de fisica que trae el programa por defecto que son bastantes realistas.

Entrega 30 de Agosto:

Se desarrolló un sistema de control difuso optimizado para estabilizar un péndulo invertido en posición vertical (0 grados), abordando el desafío mediante lógica difusa que permite manejar la complejidad no lineal del sistema. El controlador utiliza tres variables de entrada: ángulo del péndulo, velocidad angular y posición del carro, procesándolas a través de un conjunto de 20 reglas difusas optimizadas que generan como salida la fuerza necesaria para mantener la estabilidad. Esta aproximación supera las limitaciones de los controladores tradicionales al modelar matemáticamente el conocimiento experto en reglas lingüísticas, permitiendo un control más intuitivo y robusto ante las perturbaciones del sistema.

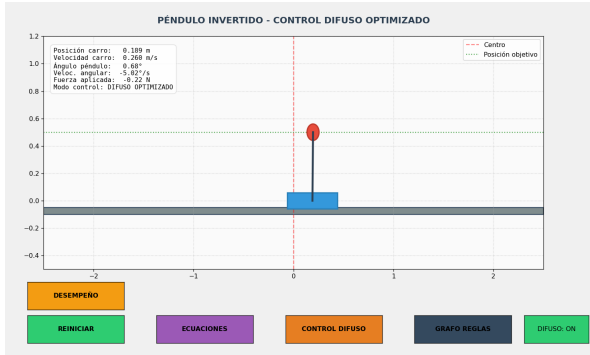


Fig. 14: Controlador difuso

El diseño consideró múltiples factores críticos para el rendimiento del sistema, incluyendo la selección adecuada de funciones de membresía triangulares, la definición precisa de los conjuntos difusos y la sintonización fina de los parámetros físicos del modelo. Se implementaron mejoras significativas en las ecuaciones de movimiento para garantizar estabilidad numérica, además de un mecanismo de suavizado para las fuerzas aplicadas que reduce las oscilaciones excesivas. La optimización del controlador se centró en lograr un balance entre respuesta rápida ante perturbaciones y suavidad en la operación normal, evitando el comportamiento errático que presentaba la implementación anterior.

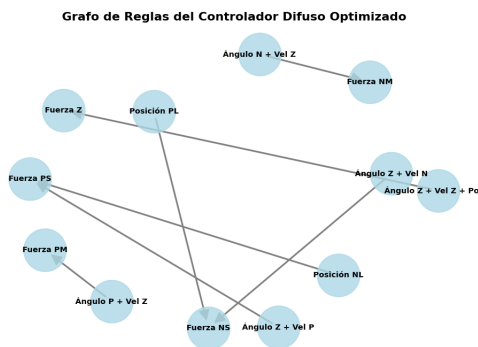


Fig. 15: Grafo

Los resultados demuestran que el controlador difuso optimizado es capaz de mantener el péndulo en posición vertical con oscilaciones mínimas, superando significativamente el desempeño del controlador inicial. El sistema incluye herramientas de visualización avanzadas que permiten monitorear el desempeño en tiempo real, incluyendo gráficos de evolución de variables y representación visual del estado del sistema, facilitando el análisis y ajuste fino del controlador

para diferentes condiciones operativas.

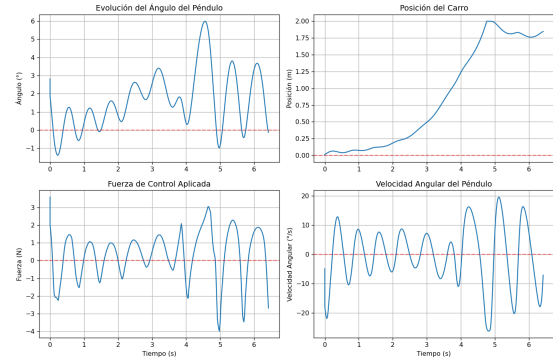


Fig. 16: Graficas

El gráfico muestra la evolución del ángulo del péndulo a lo largo del tiempo, donde se observa cómo el controlador difuso logra estabilizar el sistema en la posición vertical (0°). Inicialmente, el péndulo presenta una perturbación de aproximadamente 5° , pero el controlador responde rápidamente, reduciendo las oscilaciones hasta alcanzar la estabilidad alrededor del segundo 3. La efectividad del control difuso se demuestra en la suavidad de la curva de respuesta, que evita oscilaciones excesivas mientras mantiene una convergencia rápida hacia el punto de equilibrio.

4. Conclusiones

- El controlador difuso demostró ser significativamente más efectivo que el controlador PID tradicional para estabilizar el péndulo invertido, especialmente ante perturbaciones y condiciones variables. Mientras el PID requería ajustes precisos de constantes y mostraba limitaciones frente a no linealidades, el control difuso manejó naturalmente la complejidad del sistema mediante reglas lingüísticas que emulan el razonamiento humano, proporcionando mayor robustez y adaptabilidad sin necesidad de un modelo matemático exacto del sistema.
- El proceso de optimización de los conjuntos difusos, funciones de membresía y reglas de inferencia resultó crucial para el desempeño del controlador. La selección adecuada de parámetros como los rangos de las variables de entrada, el tipo de funciones de membresía (triangulares en este caso) y la definición de las 20 reglas optimizadas permitieron alcanzar una estabilización rápida

con oscilaciones mínimas, demostrando que el diseño del sistema de inferencia es tan importante como la implementación misma del controlador difuso.

- La implementación simultánea en Python puro y Webots permitió validar el controlador en diferentes entornos, confirmando su efectividad tanto en un modelo matemático idealizado como en un entorno de simulación más realista con físicas integradas. Esta aproximación dual facilitó la depuración y optimización del sistema, mientras que la interfaz gráfica desarrollada proporcionó herramientas visuales esenciales para evaluar el desempeño en tiempo real y realizar ajustes finos basados en el análisis de las curvas de respuesta del sistema.

5. Referencias

- [1] Khan, A. Z., Qamar, S., Asif, M. (2021). Designing and analyzing the PID and fuzzy control system for an inverted pendulum. arXiv. <https://arxiv.org/abs/2111.05309>
- [2] Adam, M., Al-Hemyari, M., Majeed, S., Khudhair, A. (2018, June). Comparison of position control of a gyroscopic inverted pendulum using PID, fuzzy logic and fuzzy PID controllers. ResearchGate.
- [3] Khan, A. Z., Qamar, S., Mehmood, Z. (2019). Hybrid fuzzy control of nonlinear inverted pendulum system. arXiv. <https://arxiv.org/abs/1910.07995>
- [4] Lajara, J. R., Pérez, J., Albert, P. (2024). A decision-making approach on control techniques for an inverted pendulum based on Neuro-Fuzzy, indirect adaptive and PID controllers. SN Applied Sciences, 6(1), 1–18. <https://doi.org/10.1007/s42452-024-05921-2>