



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas
Ingeniería mecatrónica



Sistemas operativos en tiempo real

- Control Neuro-Difuso de una Plataforma Elevadora con Retroalimentación mediante un sensor de distancia**
- Sistema de levitación mediante aire con PID**

Github:

[S-Chris543/RTOS-pr-cticas-y-examenes: En este repositorio agregaremos las tareas, prácticas y exámenes correspondientes a la materia de RTOS impartida por el maestro Ramón Jaramillo Martínez](#)

Profesor: Ramón Jaramillo Martínez

Grupo: 3MM6

Alumnos:

- Manuel García Torres
- Axel Perea Pinedo
- Christian Emmanuel Castruita Alaniz

Fecha:12/01/2026

Objetivo:

Desarrollar e implementar dos sistemas de control en un entorno RTOS, aplicando un control neuro-difuso con retroalimentación para una plataforma elevadora y un control PID para un levitador neumático mediante un sensor de distancia, con el fin de analizar su desempeño, estabilidad y respuesta temporal bajo condiciones de tiempo real.

Introducción:

El uso de Sistemas Operativos en Tiempo Real (RTOS) se ha vuelto fundamental en el desarrollo de sistemas de control modernos, especialmente en aplicaciones donde la respuesta temporal, la estabilidad y la confiabilidad son críticas. En este tipo de sistemas, la ejecución determinista de tareas, la gestión eficiente de recursos y la correcta sincronización entre procesos permiten implementar algoritmos de control avanzados de manera segura y predecible.

En el presente proyecto se desarrollan e implementan dos sistemas de control en un entorno RTOS. El primer sistema corresponde a un Control Neuro-Difuso con retroalimentación aplicado a una plataforma elevadora, cuyo objetivo es regular la posición o altura de la plataforma ante variaciones de carga y perturbaciones externas. Con lógica difusa se permite obtener un controlador adaptable y robusto frente a no linealidades del sistema.

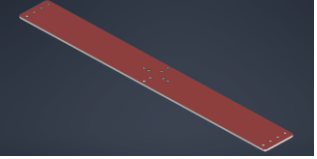


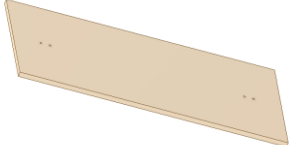


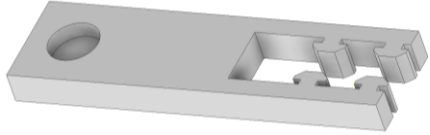
El segundo sistema consiste en un levitador neumático controlado mediante un algoritmo PID, utilizando un sensor de distancia como elemento de retroalimentación. Este sistema permite analizar el desempeño de un controlador clásico en un entorno de tiempo real, evaluando aspectos como la estabilidad, el tiempo de respuesta y el error en estado estacionario bajo la gestión de tareas del RTOS.



Ambos desarrollos integran conceptos de control automático, sistemas embebidos y programación en tiempo real, permitiendo comparar estrategias de control inteligente y clásico dentro de una arquitectura RTOS, así como evaluar su comportamiento frente a restricciones temporales y de procesamiento.

Material y/o equipo:




Plataforma levitadora con lógica difusa.


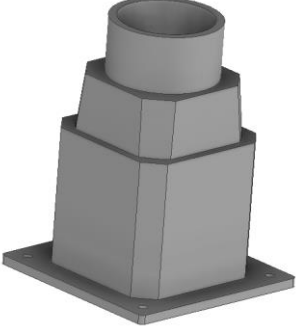
Pieza	Material
-------	----------

		Impresión 3D (PLA)
	Soporte del motor	Alumno
	Perfil de aluminio	Motor
	Motor	Madera
	Base	Impresión 3D (PLA)
	Carro Slider	Impresión 3D (PLA)
	Soporte superior	Impresión 3D (PLA)
	Soporte de resorte	Latón


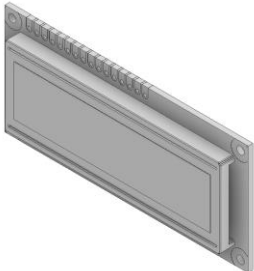
Resorte		
 Tope superior		Impresión 3D (PLA)

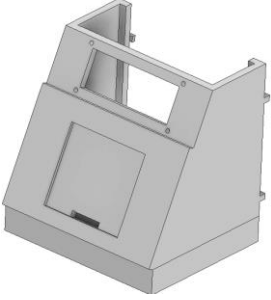
Levitador de aire PID

Pieza	Material	
 Ventilador		Plástico
 Tubo		Plástico
		Unicel.

Pelota	
 <p>Base</p>	Impresión 3D (PLA)
 <p>Soporte de ventilador</p>	Impresión 3D (PLA)

Interfaz gráfica.

Pieza	Material
 <p>Matriz numerica</p>	
	

LCD 16x2	
 <p>Caja</p>	Impresión 3D (PLA)

Desarrollo:

- Procedimiento:

El desarrollo del proyecto inició con la definición de los objetivos generales y del principio de funcionamiento de los sistemas a implementar. En primer lugar, se planteó el diseño de un sistema capaz de controlar la altura de una plataforma elevadora propulsada mediante una hélice, empleando un controlador de lógica difusa con retroalimentación. Para ello, fue necesario establecer diversos criterios de diseño, tales como la selección de los materiales de construcción, la elección de un motor con la potencia suficiente para soportar el peso total de la plataforma y la minimización de fricción y no linealidades mecánicas, con el objetivo de obtener un movimiento lo más lineal posible.

Asimismo, se definieron las restricciones físicas y de seguridad del sistema. La plataforma fue diseñada con un solo grado de libertad, permitiendo únicamente el movimiento vertical. Adicionalmente, se estableció un límite máximo de desplazamiento de 30 cm.

Una vez establecidos los objetivos y restricciones del sistema, se procedió al diseño del controlador de lógica difusa. En esta etapa se consideraron parámetros clave, como el voltaje mínimo necesario para vencer la resistencia mecánica del sistema, el voltaje máximo de operación, la selección del número y tipo de funciones de membresía, así como la definición del conjunto de reglas difusas que gobiernan el comportamiento del controlador. Dicho diseño se realizó buscando un compromiso entre estabilidad, rapidez de respuesta y reducción de oscilaciones.

Posteriormente, el controlador difuso fue implementado en el sistema físico, integrándolo dentro de un entorno de Sistema Operativo en Tiempo Real (RTOS). El RTOS permitió la correcta gestión de tareas concurrentes.

De manera paralela, se desarrolló un sistema de levitación neumática controlado mediante un controlador PID, utilizando un sensor de distancia como elemento de retroalimentación. En este caso, el procedimiento inició con la identificación del sistema y la determinación de los parámetros iniciales del controlador PID (ganancias proporcional, integral y derivativa). Posteriormente, se llevó a cabo un proceso de sintonización del controlador, evaluando la respuesta del sistema ante cambios en el punto de referencia, con el objetivo de lograr una levitación estable y precisa.

Ambos sistemas fueron integrados dentro del entorno RTOS, asignando tareas independientes para cada proceso de control, con prioridades definidas y periodos de ejecución adecuados. El RTOS permitió garantizar tiempos de respuesta deterministas, una correcta sincronización entre tareas y una operación segura y confiable de los sistemas. Finalmente, se realizaron pruebas experimentales para analizar el desempeño de cada controlador bajo condiciones de tiempo real, validando la utilidad del RTOS en la implementación de sistemas de control embebido.

Cálculos de la lógica difusa:

Datos:

- Setpoint= 150
- Sensor de distancia= 210
- Error de distancia=-60

Fusificación.

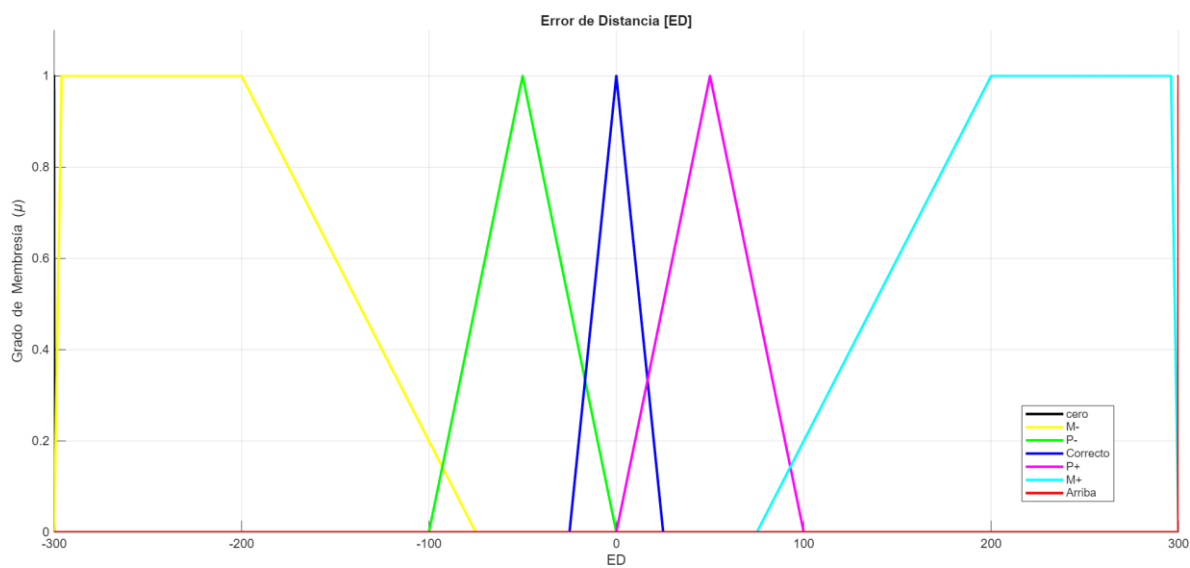
Error de distancia [ED] en mm			[-300,300]	
No. Fp	Etiqueta	Dominio	Tipo de FP	Parámetros
1	cero©	[-300, -300]	Impulso	[-300,-300,-300]
2	Mucho negativo (M-)	[-300,-150]	L	[-299.9,-299.9,-200,-75]
3	Poco negativo (P-)	[-100,0]	Triangular	[-100,-50,0]

4	Correcto (c)	[-10,10]	Triangular	[-25,0,25]
5	Poco positivo (P+)	[0,100]	Triangular	[0,50,100]
6	Mucho positivo (M+)	[150,300]	R	[75,200,299.9,299.9]
7	Arriba(A)	[300, 300]	Impulso	[300,300,300]

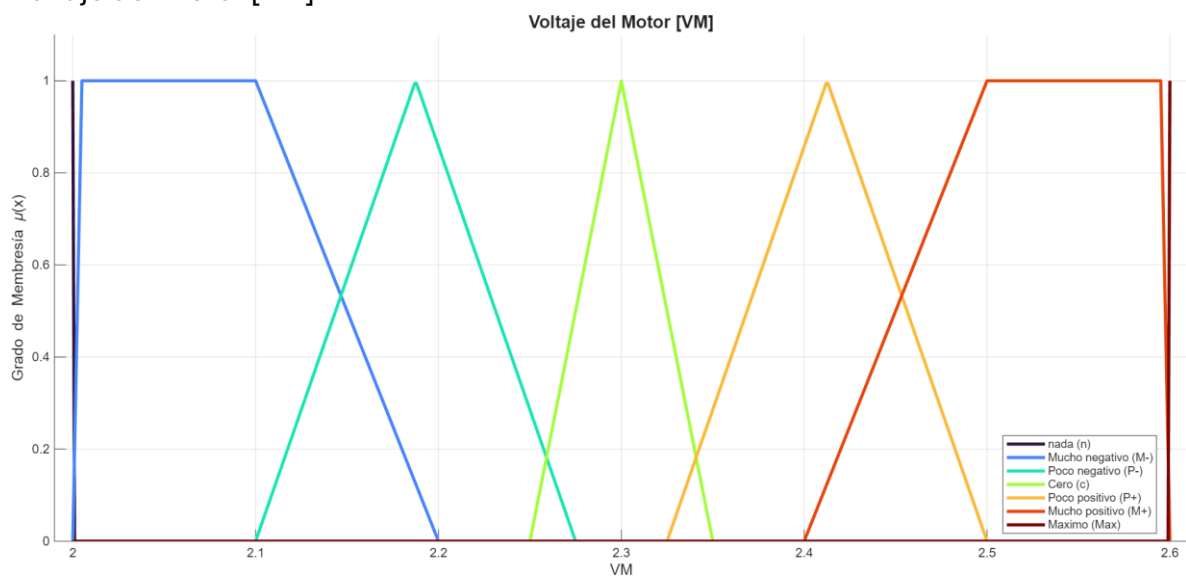
Voltaje del motor [VM]			[2,2.6]	
No. Fp	Etiqueta	Dominio	Tipo de FP	Parámetros
1	nada (n)	[2,2]	Impulso	[2,2,2]
2	Mucho negativo (M-)	[2,2.2]	L	[2.01,2.01,2.1,2.2]
3	Poco negativo (P-)	[2.1,2.275]	Triangular	[2.1 2.1875 2.275]
4	Cero (c)	[2.25,2,35]	Triangular	[2.25,2.3,2.35]
5	Poco positivo (P+)	[2.325,2.5]	Triangular	[2.325,2.4125,2.5]
6	Mucho positivo (M+)	[2.4,2.6]	R	[2.4,2.5,2.59,2.59]
7	Máximo(Max)	[2.6,2.6]	Impulso	[2.6,2.6,2.6]

Graficas:

Error de distancia [ED] gráfica:



Voltaje del motor [VM]:



Cálculos de ED (-60):

$$\mu_{\text{cero}}(-60) = 0$$

$$\mu_{M-}(-60) = 0$$

$$\mu_{P-}(-60) = \frac{x - a}{m - a} = \frac{-60 - (-100)}{-50 - (-100)} = \frac{4}{5}$$

$$\mu_C(-60) = 0$$

$$\mu_{P+}(-60) = 0$$

$$\mu_{M+}(-60) = 0$$

$$\mu_{MAX}(-60) = 0$$

Reglas:

No. De regla		Variable de entrada		Antecedente		Variable de salida		Consecuencia
1	IF	Error de distancia [ED]	ES	cero©	THEN	Voltaje del motor [VM]	ES	nada (n)
2	IF	Error de distancia [ED]	ES	Mucho negativo (M-)	THEN	Voltaje del motor [VM]	ES	Mucho negativo (M-)
3	IF	Error de distancia [ED]	ES	Poco negativo (P-)	THEN	Voltaje del motor [VM]	ES	Poco negativo (P-)
4	IF	Error de distancia [ED]	ES	Correcto (c)	THEN	Voltaje del motor [VM]	ES	Cero (c)
5	IF	Error de distancia [ED]	ES	Poco positivo (P+)	THEN	Voltaje del motor [VM]	ES	Poco positivo (P+)
6	IF	Error de distancia [ED]	ES	Mucho positivo (M+)	THEN	Voltaje del motor [VM]	ES	Mucho positivo (M+)
7	IF	Error de distancia [ED]	ES	Arriba(A)	THEN	Voltaje del motor [VM]	ES	Máximo(Max)

Alfas cortes:

$$Nada(n) \rightarrow R1: \alpha_{(n)} = \mu_{\text{©}}(-60) = 0$$

$$\text{Mucho negativo (M-)} \rightarrow R2: \alpha_{(M-)} = \mu_{M-}(-60) = 0$$

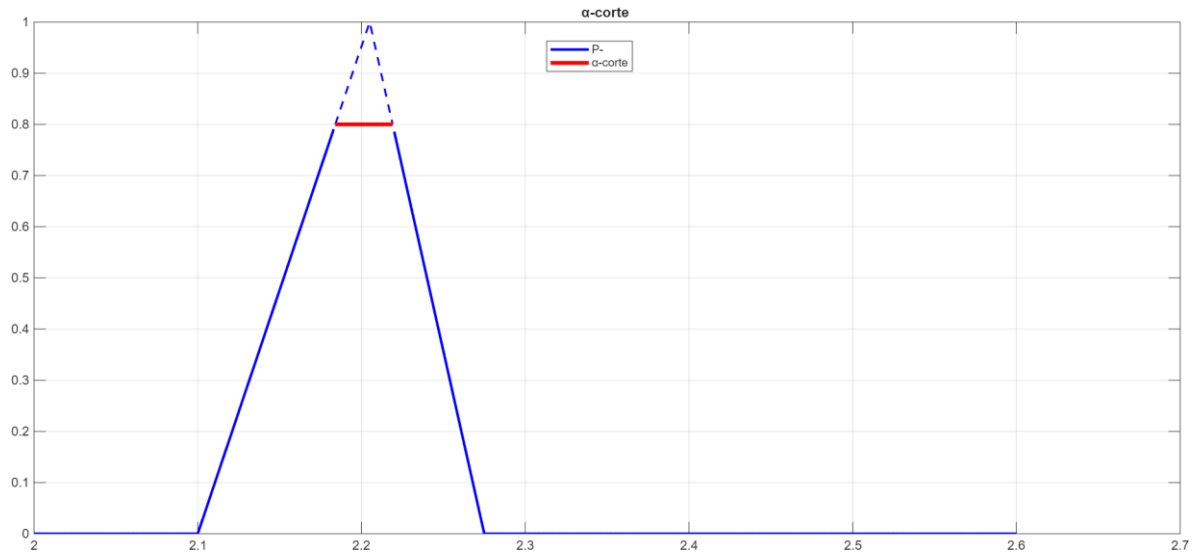
$$\text{Poco negativo (P-)} \rightarrow R3: \alpha_{(P-)} = \mu_{P-}(-60) = \frac{4}{5}$$

$$\text{Cero (c)} \rightarrow R4: \alpha_{(C)} = \mu_C(-60) = 0$$

Poco positivo (P+) $\rightarrow R5: \alpha_{(P+)} = \mu_{P+}(-60) = 0$

Mucho positivo (M+) $\rightarrow R6: \alpha_{(M+)} = \mu_{M+}(-60) = 0$

Máximo (Max) $\rightarrow R7: \alpha_{(Max)} = \mu_a(-60) = 0$



Calculo final:

Muestras: 21

Para saber los valores de cada PM: $S = \frac{2.6-2}{21-1} = 0.03$

La formula

$$\mu_{PM}(2) = \max\{\min[\mu_n(2), \alpha_n], \min[\mu_{M-}(2), \alpha_{M-}], \min[\mu_{P-}(2), \alpha_{P-}], \min[\mu_C(2), \alpha_C], \\ , \min[\mu_{P+}(2), \alpha_{P+}], \min[\mu_{M+}(2), \alpha_{M+}], \min[\mu_{Max}(2), \alpha_{Max}]\}$$

Como solo tenemos un alfa corte en este caso $\alpha_{(P-)}$ la formula queda:

$$\mu_{PM}(2) = \max\{\min[\mu_{P-}(2), \alpha_{P-}]\}$$

Formamos una tabla para cada uno de los PM donde usamos la formula anterior

junto con $\alpha_{(P-)} = \frac{4}{5}$

PM	μ_{p-}	μ_{PM}	PM* μ_{PM}
2	0	0	0
2.03	0	0	0
2.06	0	0	0
2.09	0	0	0
2.12	8/35	8/35	424/875
2.15	4/7	4/7	1 8/35
2.18	32/35	32/35	1 869/875
2.21	26/35	26/35	1 120/187
2.24	2/5	2/5	112/125
2.27	2/35	2/35	86/663
2.3	0	0	0
2.33	0	0	0
2.36	0	0	0
2.39	0	0	0
2.42	0	0	0
2.45	0	0	0
2.48	0	0	0
2.51	0	0	0
2.54	0	0	0
2.57	0	0	0
2.6	0	0	0
		Total Miu (μ)	$\mu_{PM} * Miu$ (μ)
		$2 \frac{32}{35}$	$6 \frac{327}{875}$

Ya con todos estos datos calculamos la potencia del motor con los datos Setpoint= 150 y ED=-60:

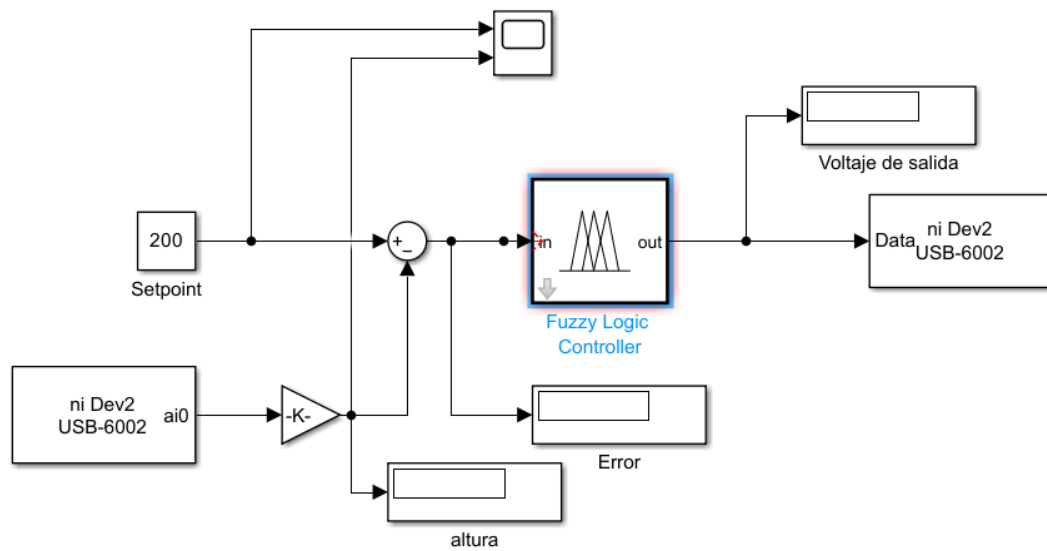
$$PM = \frac{\sum PM \cdot \mu_{PM}}{\sum \mu_{PM}} = \frac{(6 \frac{327}{875})}{(2 \frac{32}{35})} = 2.19$$

- Mediciones

Voltaje al que se rompe la resistencia mecánica	Voltaje de máxima velocidad (Limitado por exceso de velocidad)
2.4V	2.8V
Altura mínima (tomando en cuenta los resortes)	Altura máxima
30mm	300mm

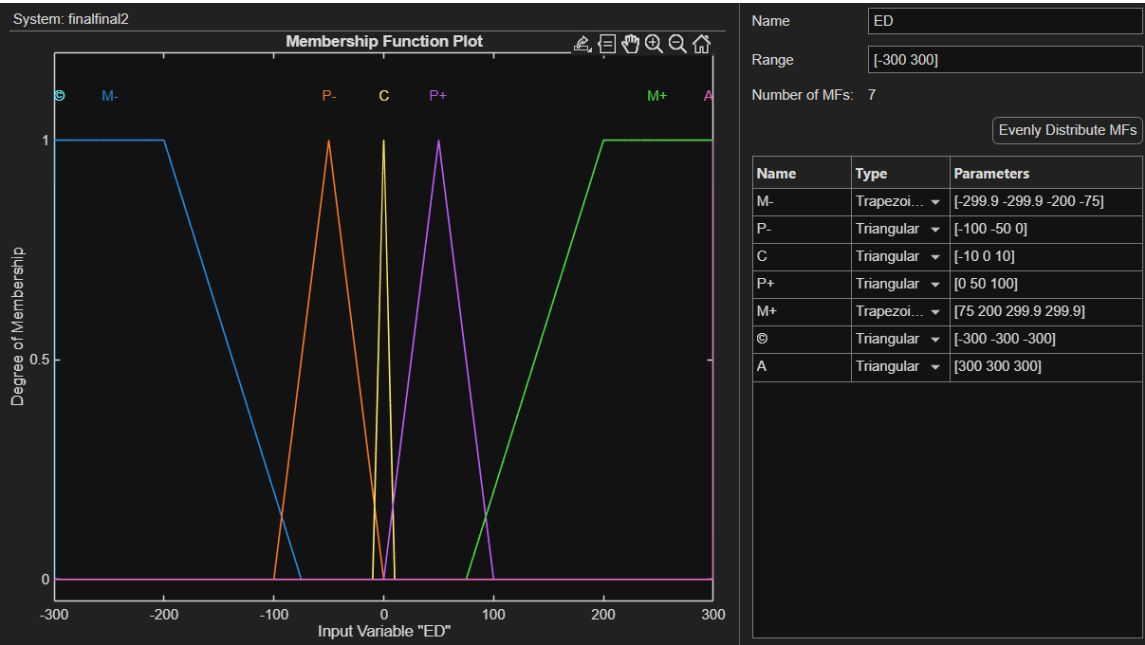
- Diagramas

Diagrama de Simulink:

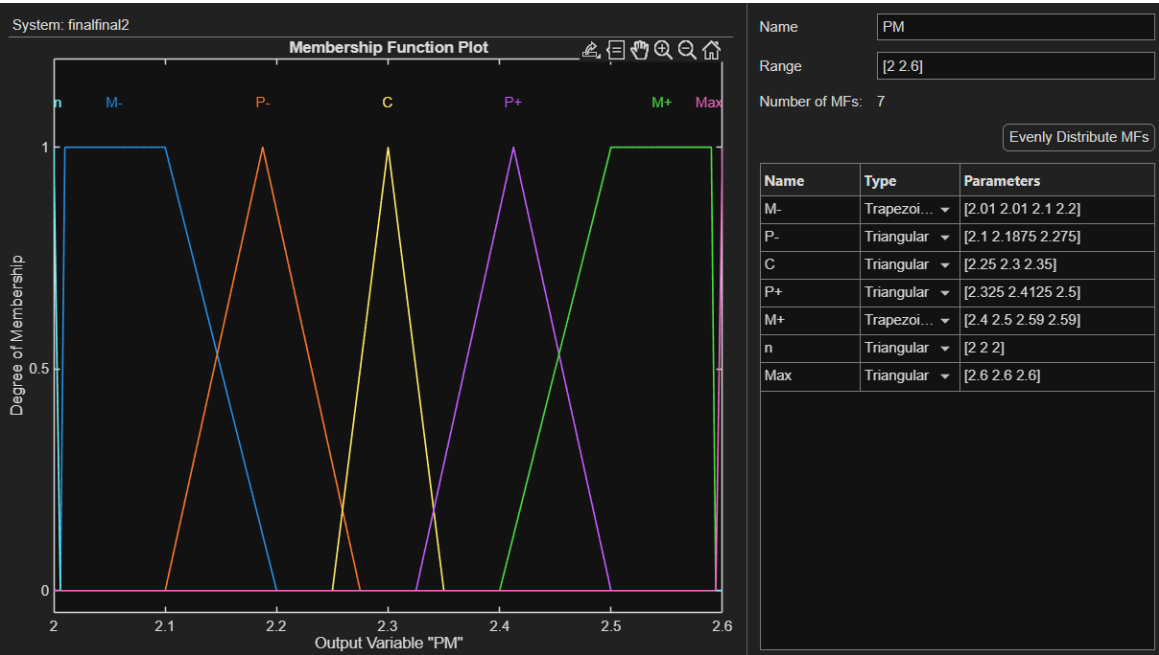


- Simulaciones

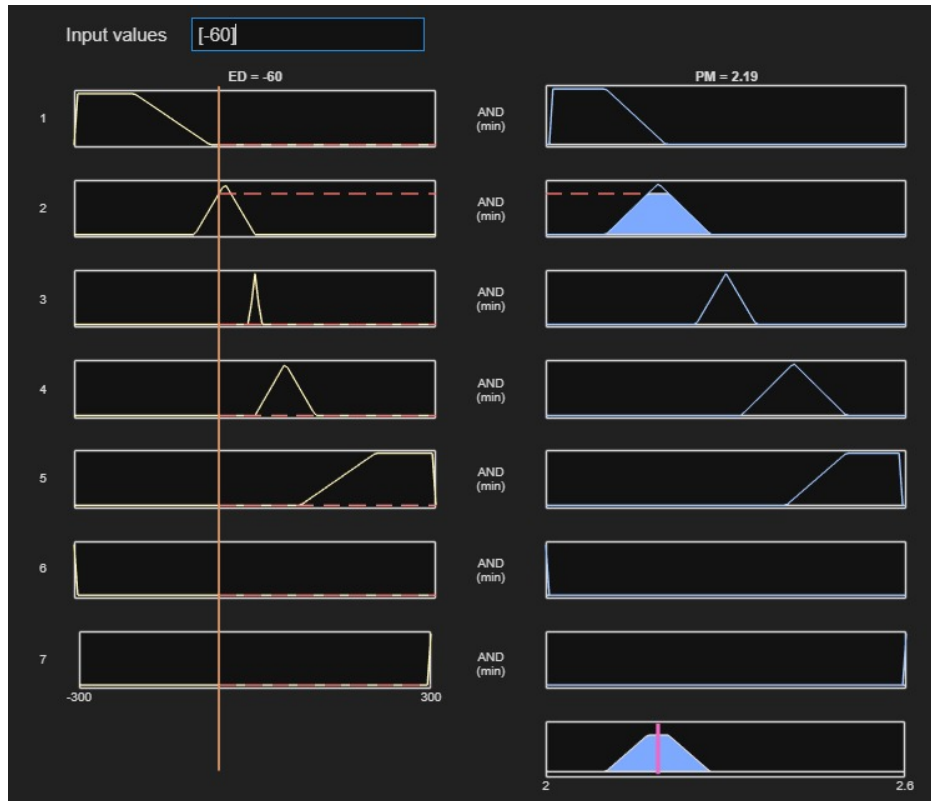
Funciones de membrecía de la entrada



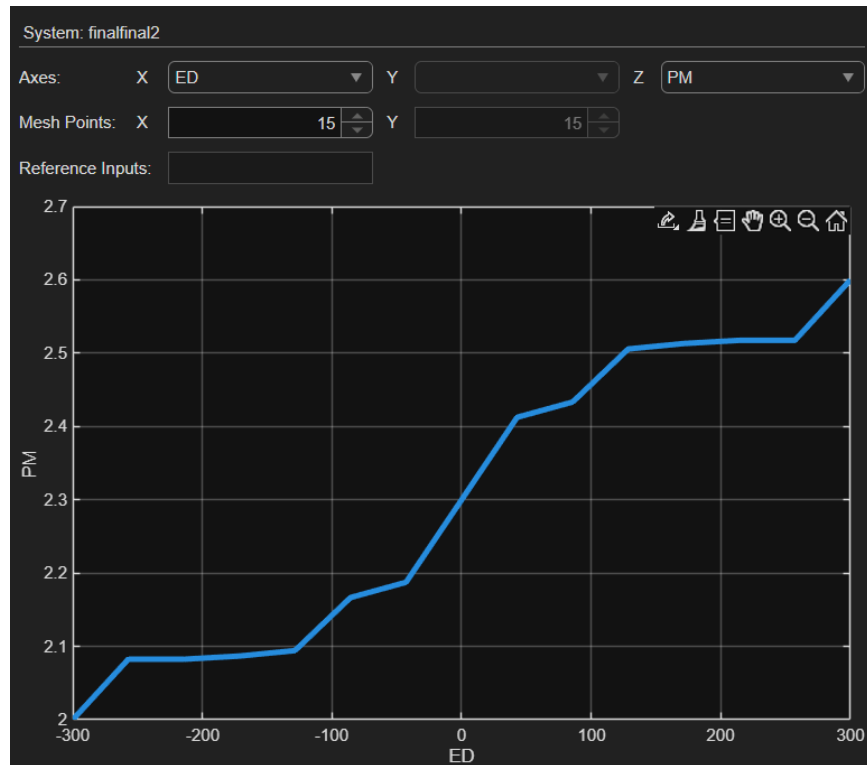
Funciones de membrecía de las salidas.



Rule inference

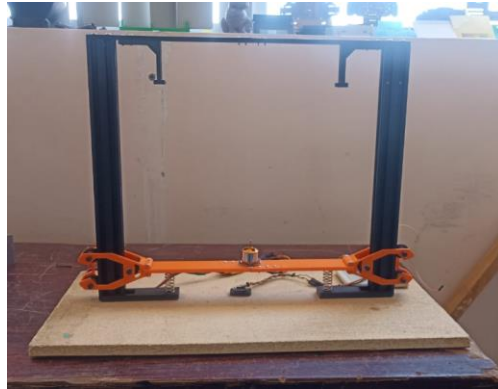


Control Surface

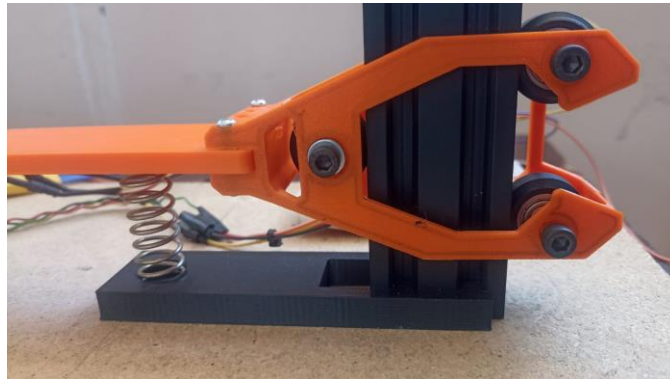


- Imágenes

Plataforma Elevadora:



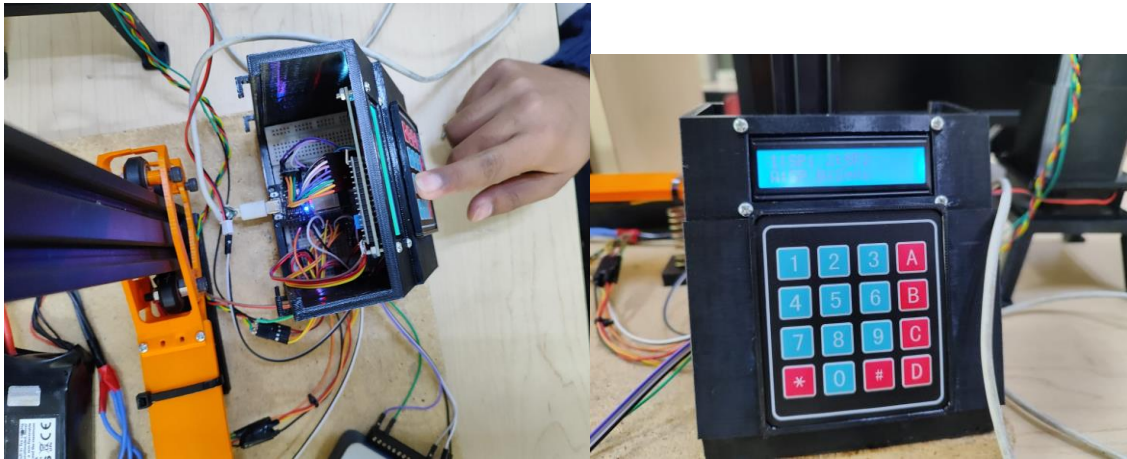
Carro Slider y tope inferior con resorte:



Levitador neumático



Interfaz usuario-máquina



Ambos sistemas en conjunto

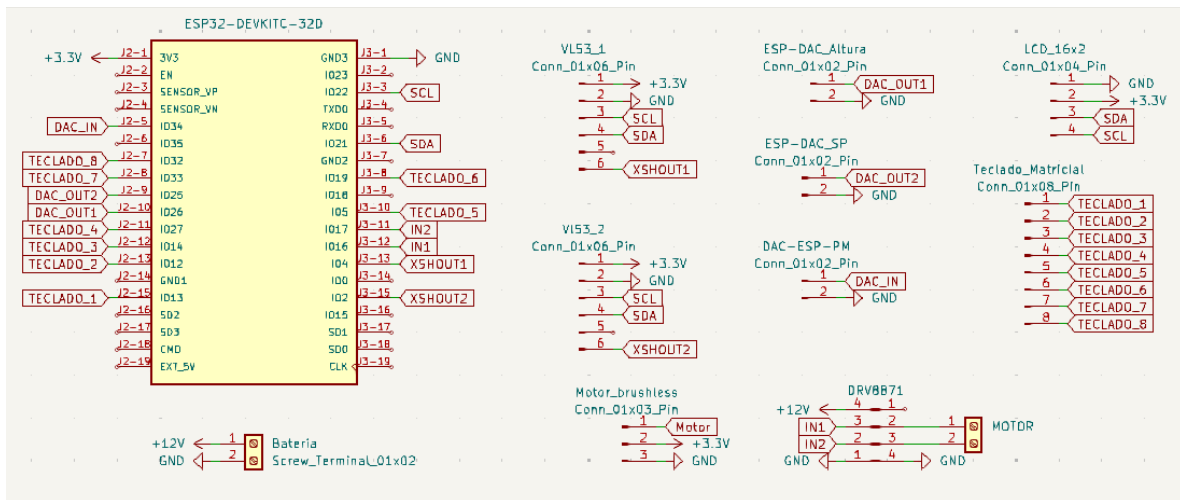


Resultados.

Set-point	Medición obtenida mm	Error mm	Tiempo de oscilación (s)
30	35	5	3
100	110	7.5	4.5
200	215	10	3
300 (maximo)	295	5	2

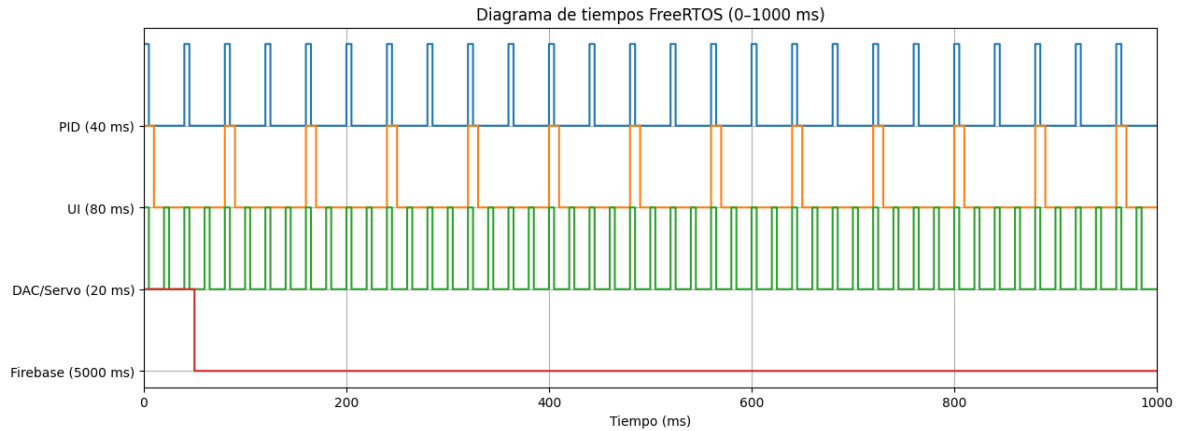
Con esto podemos concluir que tenemos un error promedio de 6.875 mm con un tiempo de oscilación para alcanzar el setpoint promedio de 3.125 s.

Diagrama de electrónico:



Implementación en RTOS:

- DAC / Servo → cada 20 ms (el más rápido)
- PID Sensores → cada 40 ms
- Interfaz (UI) → cada 80 ms
- Firebase → solo aparece al inicio (su período es 5000 ms)



Conclusión:

El uso de una arquitectura multitarea con FreeRTOS en este proyecto permitió gestionar de manera eficiente las tareas críticas de control, como el procesamiento de sensores y los algoritmos PID, al asignarles periodos cortos y alta prioridad para garantizar una respuesta estable en tiempo real. La implementación en un ESP32 facilitó el aislamiento de procesos mediante el uso de mutex y la distribución en diferentes núcleos, lo que evitó interferencias entre las funciones de control, la interfaz de usuario y la comunicación en la nube. Como resultado, se validó la utilidad de los sistemas operativos en tiempo real para desarrollar aplicaciones de control robustas, logrando un desempeño predecible con un error promedio de 6.875 mm y tiempos de respuesta adecuados para las exigencias de los sistemas de levitación y elevación implementados.

Bibliografía

- [1] R. Barry, *Mastering the FreeRTOS™ Real Time Kernel*, Real Time Engineers Ltd., 2021.
- [2] FreeRTOS, “FreeRTOS Documentation,” Amazon Web Services, 2024. [En línea]. Disponible: <https://www.freertos.org>
- [3] K. J. Åström y R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, Princeton University Press, 2010.
- [4] K. Ogata, *Ingeniería de Control Moderna*, 5ª ed., Pearson Educación, 2010.