

Controles Clásico Vs Controles Inteligentes En Una Bola De Levitación Neumática

Classic Controls Vs Smart Controls In A Pneumatic Levitation Ball

Duvan Eugenio Daza Garzon^{1*} 
Universidad de Pamplona

Anderson David Sánchez Rojas 
Universidad de Pamplona

Oscar J. Suarez 
Universidad de Pamplona

Aldo Pardo García 
Universidad de Pamplona

© 2021 Universidad de Córdoba. Este es un artículo de acceso abierto distribuido bajo los términos de la licencia Creative Commons Attribution License, que permite el uso ilimitado, distribución y reproducción en cualquier medio, siempre que el autor original y la fuente se acreditan.

¹ Facultad de Ingenierías y arquitectura Ingeniería Mecatrónica Pamplona Norte de Santander Colombia E-mail: {duvan.daza@unipamplona.edu.co, anderson.sanchez, oscar.suarez, apardo13}@unipamplona.edu.co

RESUMEN

Se realiza una planta no lineal que consiste en mantener una bola a cierta altura dentro de un tubo por medio de un ventilador que servirá para regular esta altura, para esta planta se busca aplicar control inteligente, más específicamente el control difuso y control neuronal, sin embargo para esta primera parte del proyecto se utilizara un controlador PID clásico para su control, en primera medida se tomaran mediciones de la planta en lazo abierto entre la entrada y la salida para realizar una identificación del modelo para posteriormente con la ayuda del software Matlab mediante el comando "Ident" obtener la función de transferencia, luego pasaremos a la interfaz de simulink para realizar un control y adquisición de datos en tiempo real sobre dicha planta.

PALABRAS CLAVE: Control difuso, ultrasónico, neumática, control PID.

ABSTRACT

A non-linear plant is made that consists of keeping a ball at a certain height inside a tube by means of a fan that will serve to regulate this height, for this plant it is sought to apply intelligent control, more specifically diffuse control and neural control, without However, for this first part of the project, a classic PID controller will be used for its control, firstly, measurements of the plant will be taken in open loop between the input and the output to make an identification of the model for later with the help of the MATLAB software. Using the "Ident" command to obtain the transfer function, then we will go to the Simulink interface to perform a control and data acquisition in real time on said plant.

KEYWORDS: Fuzzy control, Intelligent ultrasonic, pneumatic control, PID control.

INTRODUCTION

Actualmente en la industria evoluciona a un camino en el cual se deben realizar procesos más complejos en las plantas industriales, esto causa que muchas veces los procesos sean altamente no lineales de tal forma de que los controladores clásicos o convencionales no sean capaces de hacer el control pertinente del proceso causando un problema mayor, de estos inconvenientes nacieron los llamados métodos de control inteligente que se especializan en controlar precisamente procesos no lineales, algunos de ellos poseen la característica de aprender de los parámetros que se les brinden para luego realizar control con el conocimiento

adquirido.

La planta consiste en una bola que levita por medio de un ventilador dentro de un tubo plástico en el cual el control a realizar es sobre la altura de la pelota dentro del tubo, para obtener la función de transferencia que describe el comportamiento de la planta se tomaran una serie de datos en lazo abierto entre la entrada y la salida de la planta. Usando el comando "Ident" de Matlab y posteriormente en la interfaz de Simulink se realizó una simulación en los diagramas de bloques para certificar que la función de transferencia es correcta y representa la planta real.

Finalmente se realiza un control y una adquisición de datos en tiempo real en la interfaz de Simulink donde tendremos una entrada análoga y una salida PWM para realizar en diagramas de bloques en lazo cerrado con la entrada del sensor que indicara la posición de la pelota y la salida de voltaje para el control de la intensidad del ventilador.

1. OBJETIVOS

Objetivo general:

- Comparar distintos sistemas de control que se van a implementar en la planta Bola de levitación neumática.

Objetivos específicos:

- Generar la función de transferencia que representa el comportamiento de la planta.
- Implementar controladores clásicos e inteligentes P, PI, PD Y PID para controlar el modelo de la planta mediante simulaciones.
- Experimentar y comparar resultados del modelo de la planta en tiempo real.

2. MARCO TEÓRICO

2.1. Controladores clásicos:

Se denominan sistemas de control clásico, cuando la salida no tiene efecto sobre la acción de control, es decir no se compara la salida con la entrada de referencia. Por lo tanto, para cada entrada de referencia corresponde una condición de operación fija. Los tiempos de subida como el tiempo de estabilización el error en estado estacionario son unos puntos muy importantes en la implementación de los controladores.[1]

Control PID:

Un controlador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID. [1]

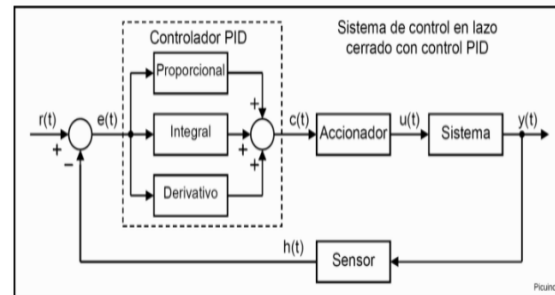


Imagen 1. Esquema de un controlador PID (1).

2.2. Control Inteligente:

Es el resultado de la intersección del Control Automático, la Inteligencia Artificial y los sistemas informáticos distribuidos de tiempo real. En este tipo de sistemas la base de conocimientos de control se obtiene de un experto humano: el operador del proceso a controlar.[2]

2.3. Lógica difusa:

La principal aplicación actual de la lógica difusa son los sistemas de control difuso, que utilizan las expresiones difusas para formular las reglas que controlarán dichos sistemas. Como la lógica difusa sugiere un cierto grado de pertenencia para un dato que se presente dentro de los conjuntos difusos, permite un controlador difuso tomar diferentes grados de acción en un sistema. En los sistemas de control debe tomarse en cuenta el o varias personas para la realización de la base de conocimiento experto de un conocimiento sobre el cual se basará la toma de decisiones. El control difuso puede usar en innumerables

sistemas, tanto sencillos, como brazos. [2]

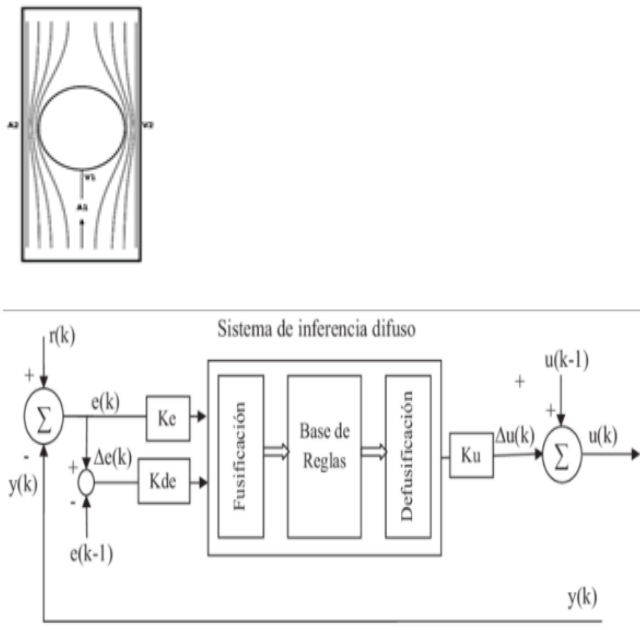


Imagen 2. Esquema de un controlador difuso implementado (2)

2.4. Levitación neumática:

Suspender un cuerpo rígido en el aire dentro de un entorno abierto o cerrado sin que exista de por medio una fuerza física aplicada directamente sobre el cuerpo se denomina levitación, misma que es resultado de la existencia de una fuerza que contrarresta el peso del cuerpo u objeto levitante. [3]

3. METODOLOGÍA

Inicialmente se realiza la construcción de la planta no lineal de levitación de una pelota dentro de un tubo plástico donde se utilizan materiales como un ventilador para la levitación, un tubo plástico, una fuente de alimentación, un sensor Ultrasónico para tomar los datos y para hacer la retroalimentación cuando se realice el controlador y una bola de icopor (Poliestireno expandido). Posteriormente se parametriza el sensor y se empiezan a tomar

las medidas correspondientes para realizar una data que se utilizarán para realizar el modelamiento del sistema que permitirá obtener la función de transferencia que describe la planta.

Finalmente se realiza la implementación en simulink con el fin de realizar una adquisición de datos y control en tiempo real mediante la información recibida por el sensor y el control sobre el ventilador actuador para hacer llegar la pelota al set point especificado para esto se desarrolla distintos controladores clásicos como inteligentes para realizar el control deseado.

4. CONSTRUCCIÓN

4.1. Realización física de la planta:

Los materiales utilizados para la construcción de esta planta fueron los siguientes.

- Sensor Ultrasónico el cual permite medir la distancia a partir de reflejar un rayo de luz infrarroja sobre la superficie.
- Ventilador actuador es el que permite el movimiento dentro del sistema, convierte la energía eléctrica en mecánica y permite realizar el desplazamiento de la pelota dentro de la planta.
- tubo el cual cubre la pelota el sensor y el actuador.
- Arduino es el microcontrolador que nos permite hacer la comunicación entre el proceso y el ordenador y permite también realizar el control del proceso.
- Pelota de icopor es el elemento que se desplaza verticalmente y es aquel se busca controlar.



Imagen 3. Planta de levitación neumática.

El circuito electrónico para representar el proyecto se aprecia en la siguiente imagen en donde se agrupan todos los componentes mencionados para el desarrollo de este proyecto.

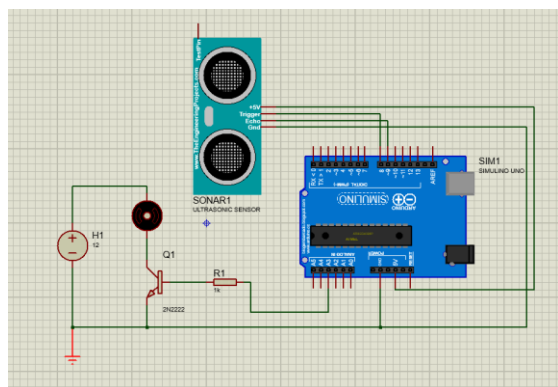


Imagen 4. Circuito electrónico representativo del sensor de proximidad.

5.2 Datos experimentales del sistema:

La adquisición de datos se llevó a cabo desde el software de Matlab utilizando la herramienta de Simulink y las respectivas librerías para la comunicación del Arduino.



Imagen 5. Distancia de la bola respecto a la variación del PWM.

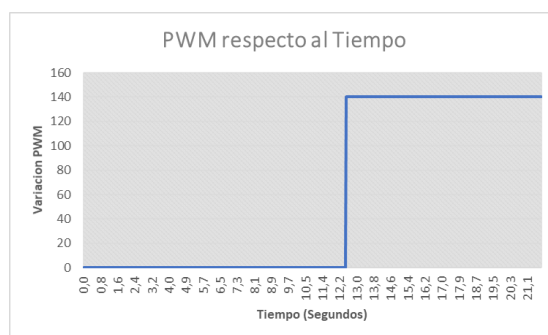


Imagen 6. Variación del PWM respecto al paso del tiempo.

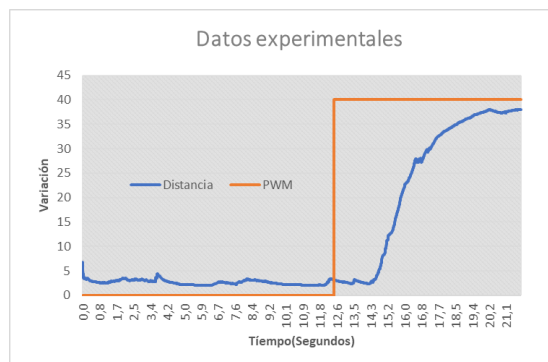


Imagen 7. Distancia vs PWM respecto al tiempo.

5. PROCEDIMIENTO

5.1. Obtención de la función de transferencia

Al obtener los datos adquiridos de distancia y PWM, la herramienta ident de MATLAB permite generar un modelo de función de transferencia a partir de los datos de entrada y salida de una planta.

Este método normalmente se usa para modelar procesos lineales, pero también permite modelar modelos no lineales, gracias a este algoritmo se facilita mucho el proceso de identificación de una planta ahorrando mucho tiempo de trabajo.

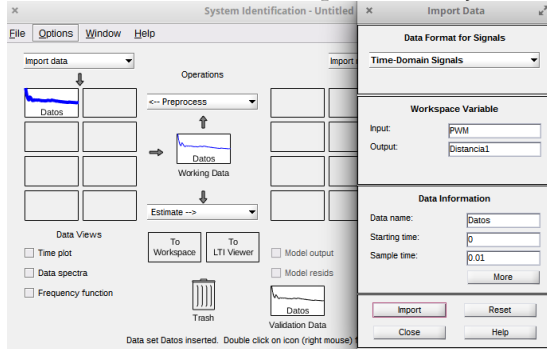


Imagen 8. Herramienta Ident con los datos adquiridos de PWM y distancia.

Posteriormente ya en la herramienta de Ident procedemos a realizar diversos modelos de la planta en búsqueda de encontrar el modelo optimo que representa de mejor manera el comportamiento del sistema.

Parameterization:
Number of poles: 2 Number of zeros: 1
Number of free coefficients: 4
Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.
Status:
Estimated using TFEST on time domain data "mydata".
Fit to estimation data: 86.47% (stability enforced)
FPE: 3.561, MSE: 3.541

Imagen 9. Resultado de parametrización de los avalores para la función de transferencia.

La función de transferencia o modelo generado la cual posee dos polos y un cero y "best fit" de estimación de datos de 86.47% el cual es superior al resto de modelos generados.

```
sys =  
  
7.32e-06  
-----  
s^2 + 0.007929 s + 2.691e-05  
  
Continuous-time transfer function.
```

Imagen 10. función de transferencia obtenida.

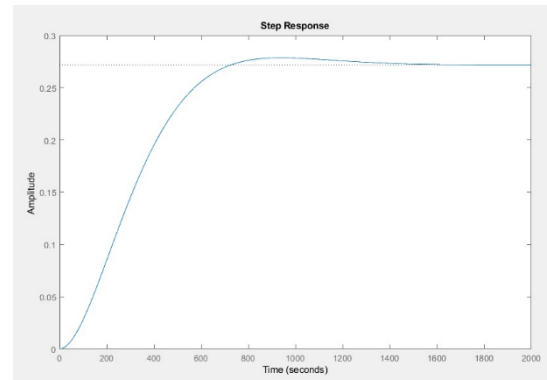


Imagen 11. Grafica del modelo de la función de transferencia generada

5.2. controladores clásicos

5.2.1. método de Ziegler Nichols

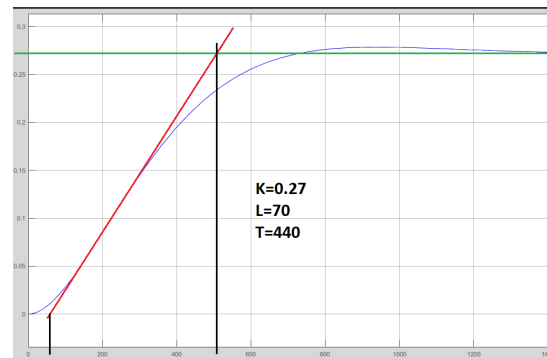


Imagen 12. Parámetros K, L y T de la función.

Control P

$$K_p = \frac{T}{LK} = \frac{440}{70 * 0.27} = 23.28$$

Control PI

$$K_p = \frac{0.9T}{LK} = \frac{0.9(440)}{70 * 0.27} = 20.952$$

$$T_i = \frac{L}{0.3} = \frac{70}{0.3} = 233.33$$

$$K_i = \frac{K_p}{T_i} = \frac{20.952}{233.33} = 0.089$$

Control PD

$$K_p = \frac{1.6T}{LK} = \frac{1.6(440)}{70 * 0.27} = 37.24$$

$$T_d = 0.5L = 0.5(70) = 35$$

$$K_d = K_p * T_d = 37.24 * 35 = 1303.4$$

Control PID

$$K_p = \frac{1.2T}{LK} = \frac{1.2(440)}{70 * 0.27} = 27.936$$

$$T_i = 2L = 2(70) = 140$$

$$T_d = 0.5L = 0.5(70) = 35$$

$$K_i = \frac{K_p}{T_i} = \frac{27.936}{140} = 0.199$$

$$K_d = K_p * T_d = 27.936 * 35 = 977.76$$

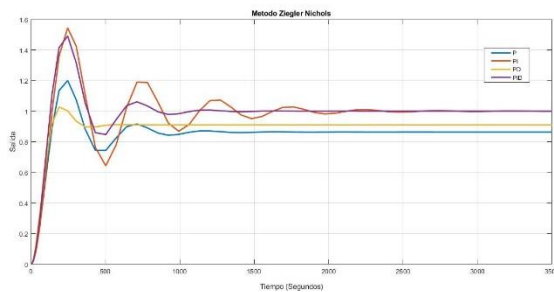


Imagen 13. Respuesta de los controladores P, PI, PD, PID por el método de Ziegler Nichols.

Observamos que los únicos controladores que se estabilizan en la salida requerida es el controlador PI y PID. El controlador PID tiene menos sobresalto y es más rápida su respuesta de estabilización que el PI, su overshoot es muy alto.

5.2.2. Función Pidtune.

```
clear all
clc

num=[7.32e-06];
den=[1 0.007929 2.691e-05];
sys=tf(num,den)
sys1=pidtune(sys,'p')
sys2=pidtune(sys,'pi')
sys3=pidtune(sys,'pd')
sys4=pidtune(sys,'pid')
```

Imagen 14. Código Pidtune en Matlab

```
sys1 =

    Kp = 9.97

P-only controller.
```

Controlador PI

with $K_p = 3.78$, $K_i = 0.0145$

Continuous-time PI controller in parallel form.

Controlador PD

with $K_p = 414$, $K_d = 7.87e+03$

Continuous-time PD controller in parallel form.

Controlador PID

with $K_p = 8.41$, $K_i = 0.0308$, $K_d = 558$

Continuous-time PID controller in parallel form.

Imagen 15. Resultados del Pidtune para los controladores clásicos P, PI, PD, PID.

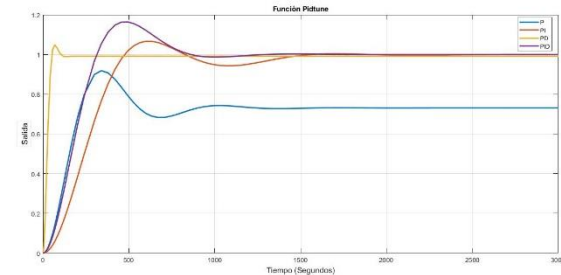


Imagen 16. Respuesta de los controladores, Función Pidtune.

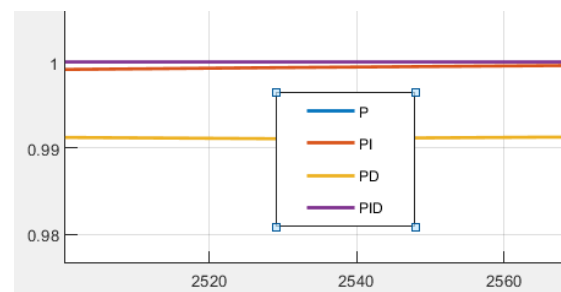
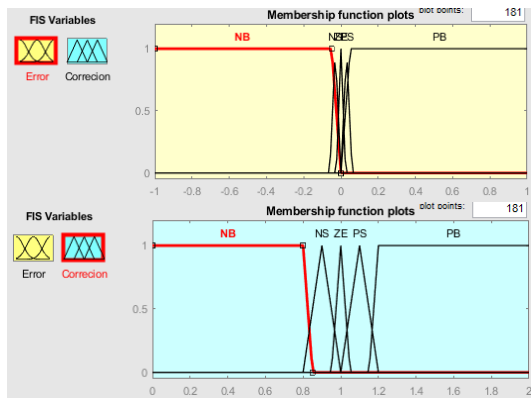


Imagen 17. Zoom grafica de su estabilización en el punto deseado.

Controlador P

5.3. Controlador difuso

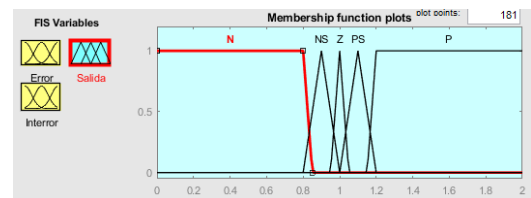
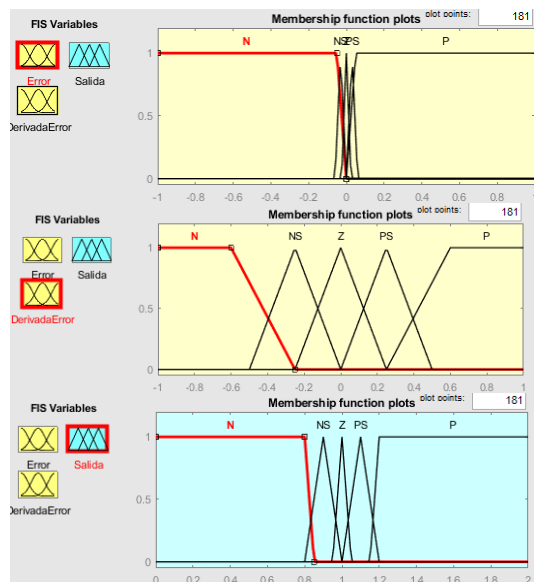
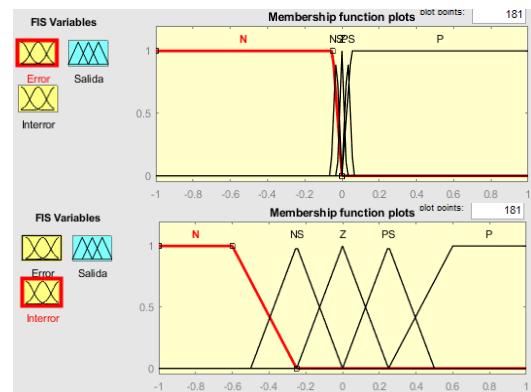


Parámetros Error NB [-2 -1 -0.05 0] NS [-0.06 -0.03 0] ZE [-0.025 0 0.025] PS [0 0.03 0.06] PB [0 0.05 1 2] Error NB [-1 0 0.8 0.85] NS [0.8 0.9 1] ZE [0.95 1 1.05] PS [1 1.1 1.2] PB [1.15 1.2 2 3]	Parámetros Error NB [-2 -1 -0.05 0] NS [-0.06 -0.03 0] ZE [-0.025 0 0.025] PS [0 0.03 0.06] PB [0 0.05 1 2] Error NB [-1 0 0.8 0.85] NS [0.8 0.9 1] ZE [0.95 1 1.05] PS [1 1.1 1.2] PB [1.15 1.2 2 3]
--	--

Imagen 18. Controlador P difuso- 5 funciones de membrecía

Parámetros Error NB [-2 -1 -0.05 0] NS [-0.06 -0.03 0] ZE [-0.025 0 0.025] PS [0 0.03 0.06] PB [0 0.05 1 2]	Parámetros Derivada Error NB [-2 -1 -0.6 -0.25] NS [-0.5 -0.25 0] ZE [-0.25 0 0.25] PS [0 0.25 0.5] PB [0.25 0.6 1 2]
Parámetros Salida NB [-1 0 0.8 0.85] NS [0.8 0.9 1] ZE [0.95 1 1.05] PS [1 1.1 1.2] PB [1.15 1.2 2 3]	

Imagen 19. Controlador PD difuso- 5 funciones de membrecía



Parámetros Error NB [-2 -1 -0.05 0] NS [-0.06 -0.03 0] ZE [-0.025 0 0.025] PS [0 0.03 0.06] PB [0 0.05 1 2]	Parámetros Integral Error NB [-2 -1 -0.6 -0.25] NS [-0.5 -0.25 0] ZE [-0.25 0 0.25] PS [0 0.25 0.5] PB [0.25 0.6 1 2]
---	---

Parámetros	
Salida	
NB	[-1 0 0.8 0.85]
NS	[0.8 0.9 1]
ZE	[0.95 1 1.05]
PS	[1 1.1 1.2]
PB	[1.15 1.2 2 3]

Imagen 20. Controlador PI difuso- 5 funciones de membrecía

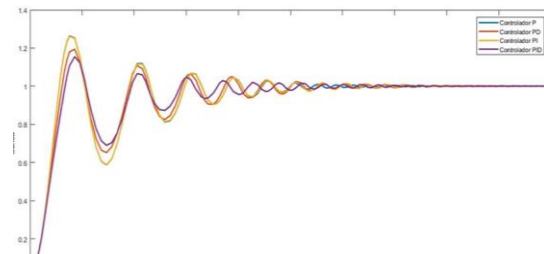


Imagen 22. Gráfico del comportamiento de los cuatro controladores (P, PD, PI, PID).

Observamos un mejor comportamiento en el controlador PID por tener el menor overshoot, todos los controladores tienden a tener el mismo tiempo de estabilización en la salida.

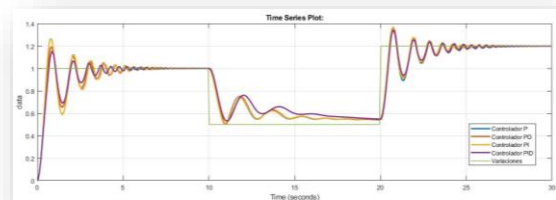
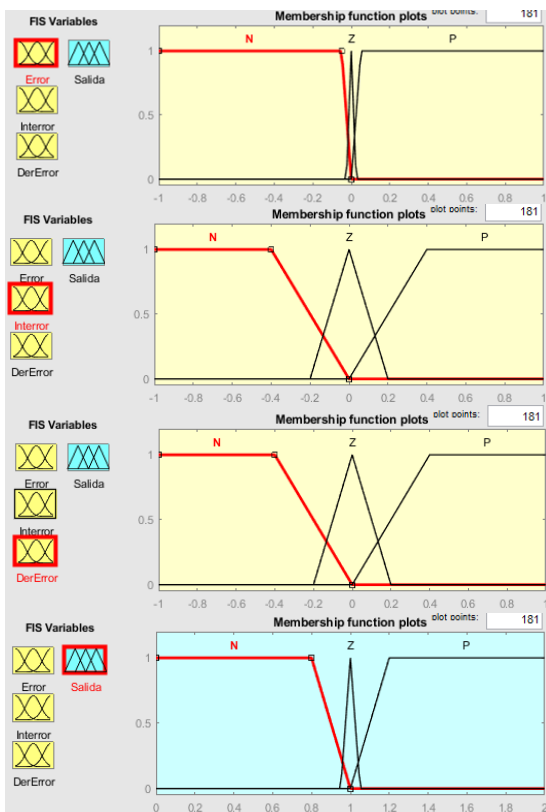


Imagen 23. Variación en la entrada

la variación está en intervalos de 10 segundos lo cual los controladores se establecen antes de cambiar su entrada. No hay estabilización en un rango inferior a 0.6 o mayor a 1.5 debido al diseño del controlador, pero al establecerse en un rango de 1,2 vuelve a estabilizarse en ese punto.



Parámetros Error		Parámetros Integral Error	
N	[-2 -1 -0.05 0]	N	[-2 -1 -0.4 0]
Z	[-0.025 0 0.025]	Z	[-0.2 0 0.2]
P	[0 0.05 1 2]	P	[0 0.4 1 2]
Parámetros Derivada Error		Parámetros Salida	
N	[-2 -1 -0.4 0]	N	[-10 0.8 1]
Z	[-0.2 0 0.2]	Z	[0.95 1 1.05]
P	[0 0.4 1 2]	P	[1 1.2 2 3]

Imagen 21. Controlador PID difuso - 3 funciones de membrecía.

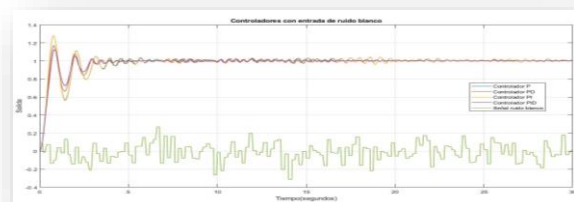


Imagen 24. Gráfico del comportamiento de los controladores con entrada de ruido blanco.

Al tener variaciones muy bruscas, tiende a sacar el controlador del punto de regulación, pero sus variaciones son muy mínimas lo cual el controlador vuelve a responder en su rango de salida.

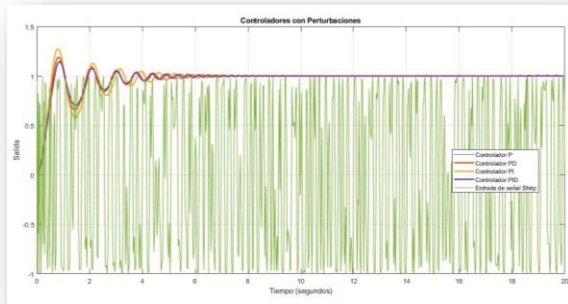


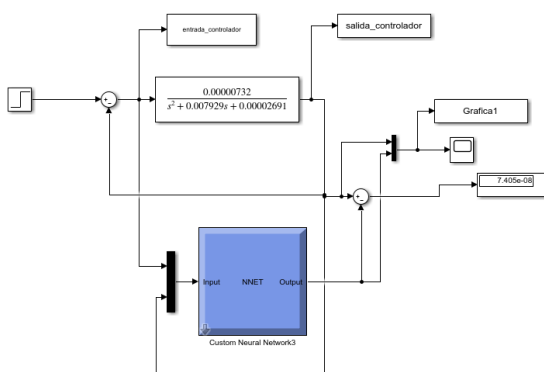
Imagen 25. Gráfico del comportamiento de los controladores con entrada de señal ship.

Los controladores no se ven muy afectados con la perturbación de frecuencia (30Hz), responde bien a ello siendo robusta sus señales en esa frecuencia, pero aun así hay pequeñas variaciones, pero no son muy notables [4].

5.4. Controlador neuronal

Se realiza la clonación de los resultados obtenidos anteriormente con la función del software Matlab Neural Network training, esto para entrenar la red neuronal [5], [6], [7].

5.4.1. Clonación función de transferencia en lazo cerrado:



```
in=[entrada_controlador.signals.values salida_controlador.signals.values];
in=in';
Net=newff(minmax(in),[10 10 1],{'tansig' 'tansig' 'purelin'});
Net.trainparam.epochs=600;
Net.trainparam.goal=1e-22;
Net=train(Net,in,salida_controlador.signals.values');
sim(Net,in);
gensim(Net)
```

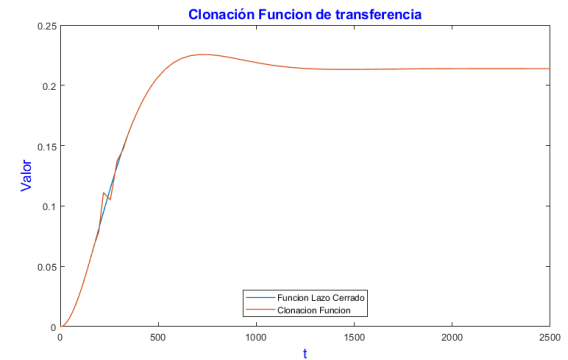
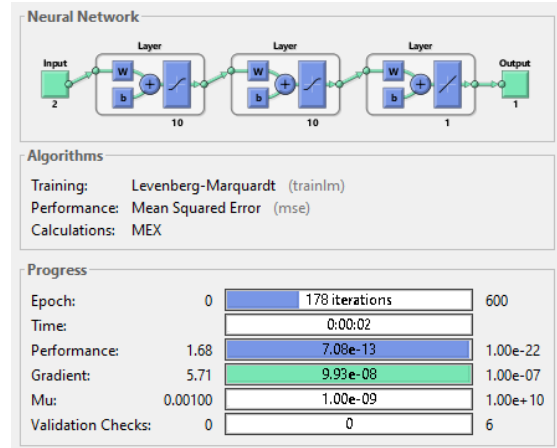
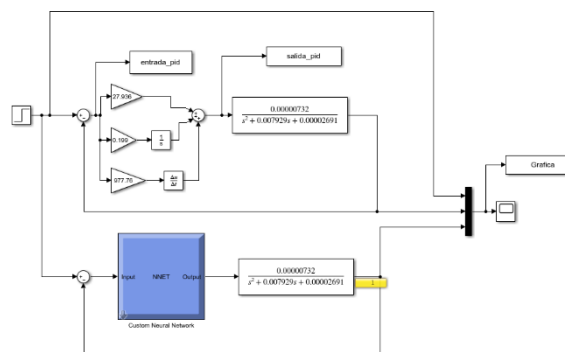
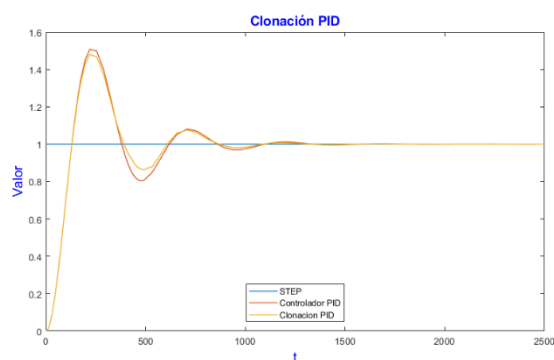


Imagen 26. Gráfico del comportamiento y parámetros de la clonación de la función de transferencia en lazo cerrado.

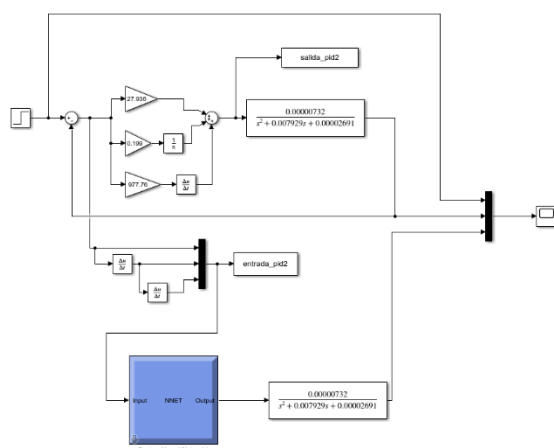
5.4.2. Clonación Controlador PID:



```
in=[entrada_pid.signals.values];
in=in';
Net=newff(minmax(in),[20 20 20 1],{'tansig' 'tansig' 'tansig' 'purelin'});
Net.trainparam.epochs=2500;
Net.trainparam.goal=1e-6;
Net=train(Net,in,salida_pid.signals.values');
sim(Net,in);
gensim(Net)
```



5.4.3. Clonación PID sin constantes solo derivadas:

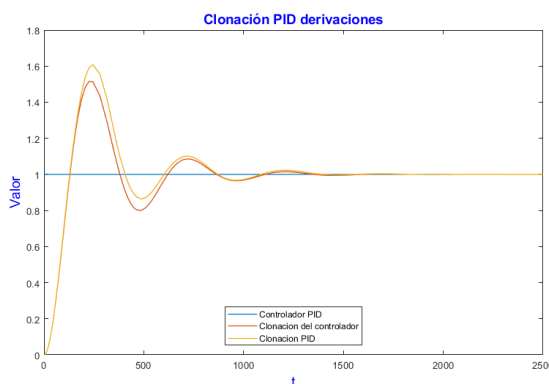


The diagram illustrates a neural network with the following structure:

- Input Layer:** A single input node labeled '3'.
- Hidden Layers:** Two hidden layers, each with 20 nodes. Each layer contains a weight matrix 'W', a bias 'b', an addition node '+', and an activation function (represented by a blue box with a curve).
- Output Layer:** A single output node labeled '1'.

Below the diagram is a table showing the training progress:

Progress			
Epoch:	0	2500 iterations	2500
Time:		0:09:36	
Performance:	236	2.16e-09	1.00e-10
Gradient:	149	2.72e-07	1.00e-07
Mu:	0.00100	1.00e-07	1.00e+10
Validation Checks:	0	0	6



6. CONCLUSIONES

Debido a que el sensor presentaba un gran ruido al momento de hacer las respectivas mediciones para determinar la distancia de elevación de la pelota, se

implementó un filtro de Kalman, el cual permitió obtener datos más precisos pero que a la vez hizo un poco más lento la respuesta del controlador en el sistema.

Mediante la herramienta “ident” de Matlab se pudo establecer la función de transferencia, tomando como valores de entrada el error de posición de dicho controlador y la salida PWM que se ejercía en la planta. Y realizando diferentes pruebas de entrenamiento con distintas configuraciones se logró obtener un modelo de controlador difuso. El cual dio excelentes resultados debido a que su respuesta era más suave y no ocasionaba cambios bruscos al momento de ejercer una acción de control en el sistema.

Se lograron corroborar los diferentes controladores clásicos e inteligentes dentro del trabajo, y se realizó la comparativa de los mismos, aunque si se tuvo que realizar algunas correcciones con respecto al tiempo ya que fue un poco demorado en cuanto a procesamiento computacional en control fuzzy con respecto a la planta.

La clonación de la función de transferencia y los diferentes controladores fue bastante buena como se muestra en las imágenes y de nuevo no se pudo establecer una óptima clonación del controlador inteligente fuzzy ya que el tiempo y consumo computacional se hizo bastante difícil de lograr.

REFERENCIAS

[1]. “Noticias del mercado.” [Online]. Available: <https://franklinlinkmx.wordpress.com/2013/09/05/que-es-el-control-pid/>. [Accessed: 4-Oct-2017].

[2]. “lógica difusa y sistemas de control.” [Online]. Available: http://www.biblioteca.udep.edu.pe/bibvirudep/tesis/pdf/1_185_184_133_1746.pdf/. [Accessed: 4-Oct-2017].

[3]. “Ventilacion industrial.” [Online]. Available: <https://www.airtecnics.com/es/tecnologia/ventiladores-generalidades>. [Accessed: 4-Oct-2017].

[4]. O. Suarez, A. Pardo and E. Sanchez, “Intelligent control for the exciter of a synchronous machine with closed loop voltage,” *World Autom. Congr. Proc.*, vol. 2016-Octob, no. July, pp. 1–6, 2016, doi: 10.1109/WAC.2016.7582983.

[5]. Galarcio, J. D., M. P. Buelvas y otros tres autores, *Una Nueva Metaheurística Aplicada Al Problema De Ruteo De Vehículos Capacitados (Cvrp) Para La Distribución De Productos Perecederos*, *Revista Ingeniería e Innovación*, 5(1), 60-72 (2017)

[6]. Suarez, O. J., Vega, C. J. ., Sanchez, E. N. ., González Santiago, A. E. ., Rodríguez Jorge, O. ., & Pardo Garcia, A. (2020). Abnormal P53 degradation and apoptosis induction in P53-MDM2 network using pinning control strategy. *Revista Colombiana de Tecnologías de Avanzada*, 2(32), 1-7. <https://doi.org/10.24054/16927257.v32.n32.2018.101>

[7]. Rios, Y. Y., García-Rodríguez, J., Sanchez, E. N. ., Alanis, A. Y. ., Ruiz Velázquez, E. ., & Durán Acevedo, C. M. . (2020). Treatment for T1DM patients using neuro-fuzzy inverse optimal control algorithm: a rapid prototyping implementation. *Revista*

- Colombiana de Tecnologías de Avanzada, 2(32), 26-33.
<https://doi.org/10.24054/16927257.v32.n32.2018.104>
- [8]. Gualdrón Guerrero, O. E. ., Reales Osorio, J. J. ., & Acevedo Gata, T. L. . (2016). Implementación de una estrategia de control difuso para aumentar la producción de crudo en pozo petrolero. Revista Colombiana de Tecnologías de Avanzada, 1(27), 98-103.
<https://doi.org/10.24054/16927257.v27.n27.2016.358>
- [9]. W. D. Urango and H. E. Hernández, “Efecto de los decodificadores en la calidad de la solución para un problema de distribución de instalaciones UA-FLP,” Rev. Ing. e Innovación, vol. 5, pp. 73–80, May. 2017. Disponible en <https://revistas.unicordoba.edu.co/index.php/rii/article/view/1256>