

# Control of the angular position of a direct current motor using PID control, fuzzy control and neural control

José I. Canelón S.<sup>1</sup> and Néstor V. Queipo<sup>2</sup>

<sup>1</sup>Escuela de Ingeniería Eléctrica. <sup>2</sup>Instituto de Cálculo Aplicado  
Facultad de Ingeniería. Universidad del Zulia. Apartado 526. Maracaibo, Venezuela  
e-mail: <sup>1</sup>jcanelon@luz.ve, <sup>2</sup>nqueipo@luz.ve

## Abstract

In this study, the angular position of a direct current motor is controlled using three control approaches: PID Control, Fuzzy Control and two different Neural Control approaches. Then, the performance of these controllers is compared under the following criteria: rise time, settling time, percentage overshoot, integral of the absolute value of the error and easiness of design. Each controller is first designed based on the mathematical model of the control problem of interest and then designed using the corresponding experiment. These different control strategies are evaluated under a variety of operating conditions, using the results obtained from computer simulations and the actual experiment. On the mathematical model, step changes of the set point were used, obtaining a similar performance from PID and Neural controllers. Step and sinusoidal changes were used on the experiment. In the first case the PID controller had a better performance but on the second case the best was the Neural controller.

**Key words:** PID control, fuzzy control, neurales control.

# Control de la posición angular de un motor de corriente directa utilizando control PID, lógica difusa y redes neurales

## Resumen

En este trabajo se realiza el control automático de la posición angular de un motor de corriente directa utilizando tres estrategias de control: PID, basado en Lógica Difusa y dos enfoques de controladores basados en Redes Neuronales. Posteriormente se compara el desempeño de estos controladores según los siguientes criterios: tiempo de subida, tiempo de estabilización, porcentaje de sobrepico, integral del valor absoluto del error y facilidad en el diseño. Inicialmente, cada controlador se diseña basado en un modelo matemático del proceso de interés y posteriormente se diseña basado en el motor. Luego se realizan varias pruebas, tanto en simulación como sobre el experimento, para obtener resultados y establecer comparaciones. En simulación se utilizaron cambios escalón del valor deseado, obteniéndose un desempeño muy similar entre el controlador PID y los controladores basados en Redes Neuronales. Sobre el motor se utilizaron cambios escalón y cambios sinusoidales del valor deseado. En el primer caso se obtuvo un mejor desempeño del controlador PID, mientras que en el segundo el mejor fue el controlador basado en Redes Neuronales.

**Palabras clave:** Control PID, Control difuso, control neural.

## Introducción

Durante los últimos años ha habido un incremento significativo en el número de problemas de control que han sido resueltos utilizando técnicas no clásicas, como por ejemplo, Control basado en Lógica Difusa y Control basado en Redes Neuronales. En el presente trabajo se desea controlar de forma automática la posición de un motor de corriente directa utilizando estas dos técnicas de control.

Existen diversos tipos de motores de corriente directa en uso a nivel industrial. Por ejemplo, los motores con rangos de potencia relativamente pequeños se utilizan en equipos de computación, mientras que los de mayor potencia son de gran utilidad en sistemas robóticos. Por lo tanto, existen muchos problemas en los que es necesario el control de posición de motores de corriente directa de manera precisa.

El problema de control de la posición de un motor ha sido resuelto comúnmente utilizando controladores PID. Puede encontrarse una gran cantidad de trabajos publicados en el área de control de motores de corriente directa utilizando este tipo de controlador [1]. Varios trabajos reportan resultados de control de velocidad, que es un problema más sencillo que el control de posición ya que se trata de una función de transferencia de primer orden [2].

Li y Lau [3] realizan el control de posición de un motor de corriente directa utilizando un controlador basado en Lógica Difusa y comparan su desempeño con el de un controlador PI y el de un controlador adaptativo de modelo de referencia. Los resultados reportados se obtienen de simulaciones computacionales utilizando un modelo matemático. Lee [4] expone los conceptos de la teoría de la Lógica Difusa que son necesarios para las aplicaciones a problemas de control. Incluye una descripción de los componentes de un controlador basado en Lógica Difusa y se especifica un procedimiento para diseñar este tipo de controladores. En este artículo puede encontrarse una lista de referencias en el área.

Existen varios trabajos en los que se reportan resultados obtenidos al aplicar Redes Neuronales para resolver problemas de control. Nguyen y Widrow [5] entrenan una red neural para que estacione en retroceso un camión a la entrada de

un depósito fijo. Los autores reportan resultados exitosos desde varias posiciones iniciales. Las pruebas fueron realizadas mediante simulaciones computacionales.

Khalid y Omatu [6] utilizan una red neural para controlar la temperatura de un baño térmico. La red es entrenada de forma tal que aprenda la dinámica inversa del proceso y los patrones de entrenamiento se generan de acuerdo al modelo asumido para la dinámica inversa de la planta. Los resultados se obtienen de experiencias realizadas sobre un experimento real.

Nahas *et al.* [7] proponen el uso de redes neuronales para obtener modelos tanto directo como inverso de procesos, con el fin de utilizar dichos modelos para la estrategia de control de modelo interno. Se proponen dos posibles enfoques para calcular la inversa del proceso: inversa entrenada e inversa calculada en línea. La estrategia de control propuesta es evaluada para dos problemas altamente no lineales: control de concentración en un tanque y neutralización de pH. En ambos casos se trabajó sobre simulaciones computacionales utilizando modelos matemáticos.

En el grupo de trabajos incluidos dentro de la revisión bibliográfica realizada al inicio de esta investigación, no se encontraron trabajos en los que se hiciera una comparación del desempeño de los tres tipos de controlador: PID, basado en Lógica Difusa y basado en Redes Neuronales sobre el mismo problema de control. Además, la mayoría de los trabajos reportan resultados sobre simulaciones computacionales utilizando modelos matemáticos de los procesos y no sobre experimentos reales. Este trabajo constituye un primer paso para superar estas limitaciones.

En resumen, los objetivos de este trabajo son los siguientes:

- Diseñar e implementar dos tipos de controlador: uno basado en Lógica Difusa y otro basado en Redes Neuronales, para el control de la posición de un motor de corriente directa.
- Comparar el desempeño de los controladores implementados, con el de un controlador PID clásico, bajo ciertos criterios de evaluación previamente establecidos.

### Definición del Problema

#### Control de la posición angular de un motor de corriente directa

Un motor de corriente directa es un dispositivo que rota debido a la interacción que se produce entre los campos magnéticos que generan dos corrientes: la corriente de excitación y la corriente de inducido. En este trabajo se utiliza el control de inducido, en el que se varía la tensión del inducido y se mantiene constante la corriente de excitación. En este caso la variable de salida es la posición angular del motor medida en radianes y la variable de entrada es el voltaje del inducido medido en voltios. La función de transferencia que expresa de manera aproximada el comportamiento del sistema es [8]:

$$\frac{\theta(s)}{E_a(s)} = \frac{K_m}{s \cdot (T_m s + 1)} \tag{1}$$

donde  $K_m$ = constante de ganancia en [°/seg] y  $T_m$ = constante de tiempo en [seg]. El símbolo (°) representa grados.

Si se aplica la Transformada Inversa de Laplace a la función de transferencia del proceso se obtiene la siguiente ecuación diferencial de segundo orden:

$$T_m \frac{d^2\theta(t)}{dt^2} + \frac{d\theta(t)}{dt} = K_m e_a(t) \tag{2}$$

#### Sistema experimental para el control de la posición angular de un motor de corriente directa

En la Fig. 1 se muestra el diagrama de bloques del sistema experimental disponible en el laboratorio.

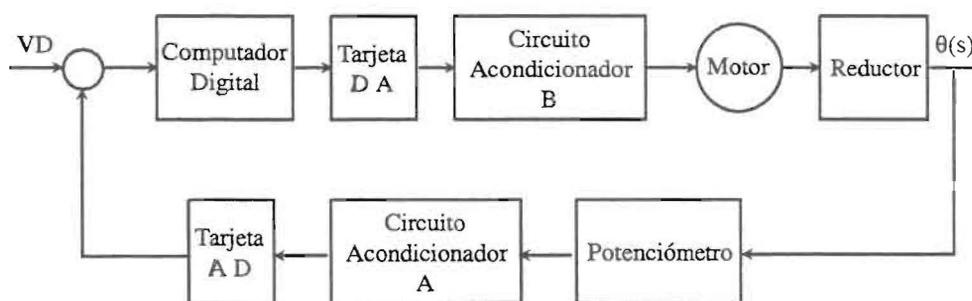


Figura 1. Sistema experimental para el control de la posición angular de un motor de CD.

El motor del laboratorio dispone de un freno magnético móvil que permite aplicar diferentes fuerzas de frenado al motor variando de esta manera los valores de  $K_m$  y  $T_m$ . Para la condición de freno considerada en este trabajo  $K_m = 85.02 \text{ }^\circ/\text{V} \cdot \text{seg}$  y  $T_m = 0.159 \text{ seg}$ . Estos valores fueron tomados del trabajo de Moreno y Montiel [1]. Por condiciones de diseño del motor, su voltaje del inducido debe estar dentro del intervalo [-15, 15]V. El motor tiene una zona muerta entre [-3.5, 3.5]V, que debe ser considerada al diseñar controladores que funcionen sobre el experimento.

Los componentes restantes son típicos de un sistema de control digital: El computador constituye el controlador y puede ser utilizado para implementar diferentes algoritmos de control; el reductor tiene una ganancia de 1/30, es decir, que por cada 30 revoluciones del motor él gira 1 vez; el potenciómetro es el dispositivo que se utiliza para medir la posición y tiene una ganancia de 10.438 °/V, el circuito acondicionador A tiene una ganancia de 1/3; la tarjeta A/D es una OMEGA DAS-8 y tiene como intervalo de lectura [-5, 5]V; la tarjeta D/A es una OMEGA DDA-06 y tiene como intervalo de salida [0, 10]V y el circuito acondicionador B transforma el voltaje de salida al intervalo [-1, 15]V con una ganancia de -1.5.

### Metodología de Solución

El diseño de cada controlador involucra el siguiente procedimiento:

1. Diseño e implementación sobre el modelo matemático:

$$\frac{\theta(s)}{E_a(s)} = \frac{85.02}{s \cdot (0.159 \cdot s + 1)} \tag{3}$$

Tabla 1  
Parámetros para el controlador PID

	K	T <sub>i</sub>	T <sub>d</sub>	T
Modelo Matemático	5	1000	0.1	0.01
Motor	4	100	0.1	1.76 · 10 <sup>-3</sup>

y prueba del controlador implementado en simulaciones computacionales del lazo de control.

La simulación se utiliza para tener una idea del funcionamiento de los lazos de control antes de trabajar con el experimento, con el fin de evitar posibles daños al equipo experimental como consecuencia de una condición de operación inapropiada. Las ecuaciones de diferencias que permiten simular el comportamiento del motor se obtienen a partir de su modelo matemático:

$$x_2(i) = x_2(i-1) + T \frac{85.02 \cdot e(i) - x_2(i-1)}{0.159}$$

$$x_1(i) = x_1(i-1) + T * x_2(i) \quad (4)$$

donde  $x_1$  es la posición angular del motor,  $x_2$  representa la velocidad y T es el tiempo de muestreo que se utiliza en el lazo de control digital en simulación.

2. Diseño e implementación sobre el experimento real.

Una vez implementados, se compara el desempeño de estos controladores con el de un controlador PID, bajo los siguientes criterios:

- (i) Parámetros de Respuesta Transitoria: Tiempo de subida ( $t_r$ ), Tiempo de estabilización ( $t_s$ ), Porcentaje de sobrepico (P.O.).
- (ii) Integral del valor absoluto del error.
- (iii) Facilidad de diseño.

### Controlador

#### Proporcional-Integral-Derivativo (PID)

La ecuación que se utilizó para implementar el controlador PID de manera discreta es:

$$u(k) = u(k-1) + K \left[ (e(k) - e(k-1)) + \frac{T}{T_i} e(k) + \frac{T_d}{T} (e(k) - 2e(k-1) + e(k-2)) \right] \quad (5)$$

donde  $u(k)$  y  $e(k)$  son la señal de control y el error respectivamente en el instante  $k$ , K es la ganancia,  $T_i$  es el tiempo integral y  $T_d$  es el tiempo derivativo. El controlador PID requiere del ajuste de esos tres parámetros para lograr un funcionamiento satisfactorio. En la Tabla 1 se muestran los valores de los parámetros para los que se obtuvo el mejor desempeño en cuanto a la integral del error.

### Controlador basado en lógica difusa

Un Controlador Basado en Lógica Difusa intenta proporcionar un algoritmo para transformar la estrategia de control lingüística basada en el conocimiento de un experto, en una estrategia de control automático. En Lee [4] puede encontrarse la descripción del proceso de diseño de un controlador basado en Lógica Difusa.

El procedimiento de diseño del controlador basado en Lógica Difusa es similar para el caso del modelo matemático y del experimento, y se resume a continuación:

- Definición de las Variables de Entrada y Salida al Controlador: la variable de entrada es el error y la variable de salida es el voltaje aplicado.

- Definición del Universo de la Variable de Entrada: se asume que la posición del motor varía entre 70° y -70°, por lo que el intervalo de variación del error es [-13.4, 13.4] considerando la ganancia del potenciómetro.

- Partición del Universo de la Variable de Entrada en Conjuntos Difusos: Se dividió el espacio en siete conjuntos difusos etiquetados: GP, MP, PP, ZE, PN, MN y GN. En la Figura 2 se muestran los conjuntos difusos para la variable error. Las funciones de membresía de los conjuntos difusos son trapezoidales y están definidas de forma funcional a través de cuatro puntos característicos.

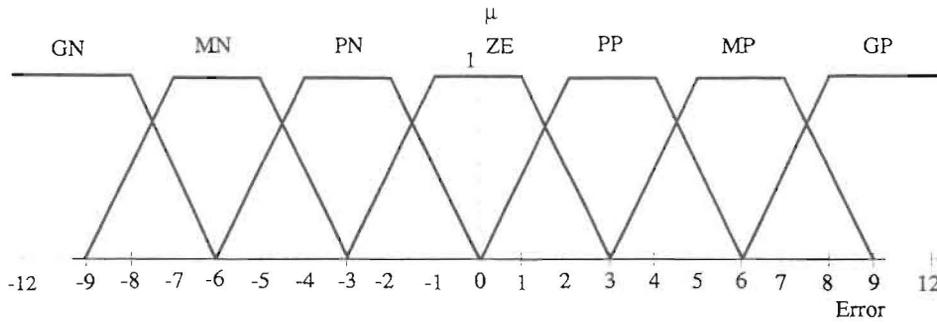


Figura 2. Conjuntos difusos para la variable error.

- Definición del Universo de la Variable de Salida: Aquí hay una diferencia ya que el intervalo de variación de la variable de salida para el modelo matemático es [-15,15]V, mientras que para el motor fue [-11.5, 11.5] debido a la zona muerta.

- Partición del Universo de la Variable de Salida en Conjuntos Difusos: En ambos casos se dividió el universo en siete conjuntos difusos: GP, MP, PP, ZE, PN, MN y GN, pero se utilizaron funciones de membresía distintas. En las Figuras 3 y 4 se muestran los conjuntos difusos del voltaje del inducido para el modelo matemático y para el motor respectivamente.

- Establecimiento de la Base de Reglas: La generación de las reglas se hizo en base al conocimiento que se tiene del comportamiento del modelo matemático y del motor. Las siete reglas son las siguientes:

- IF Error IS GN THEN Voltaje IS GN
- IF Error IS MN THEN Voltaje IS GN
- IF Error IS PN THEN Voltaje IS GN
- IF Error IS ZE THEN Voltaje IS ZE
- IF Error IS PP THEN Voltaje IS PP
- IF Error IS MP THEN Voltaje IS MP
- IF Error IS GP THEN Voltaje IS GP

- Proceso de Inferencia: Se eligió la norma triangular  $x \wedge y = \min\{x,y\}$  para interpretar el operador AND.

- Definición de la Estrategia de Defusificación: En este caso se utiliza el método del centro de gravedad el cual establece que el valor numérico de la señal de control es el centro de gravedad de la distribución de la variable de salida.

Para el modelo matemático se utilizó un tiempo de muestreo de 0.01 seg. En este caso, se

estableció este valor porque fue el tiempo de muestreo que arrojó el menor error al final del proceso de entrenamiento de la red neural. Para el motor se utilizó como tiempo de muestreo  $3.57 \cdot 10^{-3}$  seg., ya que este fue el menor tiempo de muestreo obtenido en el sistema experimental utilizado.

**Controlador de modelo interno**

La configuración general de un controlador de modelo interno se muestra en la Figura 5. En el gráfico, P es el proceso, M es un modelo del proceso, C es el controlador, F es la función de transferencia de un filtro,  $y_{sp}$  es el valor deseado,  $y(k)$  salida del proceso,  $\tilde{y}(k)$  es la salida del modelo,  $d(k)$  es la diferencia entre  $y(k)$  y  $\tilde{y}(k)$ ,  $u(k)$  es la señal de control.

Del lazo de control puede obtenerse la siguiente función de transferencia entre  $y(k)$  y  $y_{sp}(k)$ :

$$\frac{y(k)}{y_{sp}(k)} = \frac{F \cdot P \cdot C}{1 + P \cdot C - M \cdot C}$$

Si se asume que el modelo del proceso es perfecto, es decir,  $P = M$  y que la función de transferencia del controlador es igual a la inversa de la función de transferencia del proceso, esto es,  $C = 1/P$  y por lo tanto  $C = 1/M$ , la ecuación (6) se reduce a:

$$\frac{y(k)}{y_{sp}(k)} = F \tag{7}$$

En este caso, el diseñador puede fijar el comportamiento de la salida del proceso  $y(k)$  ante los cambios en el valor deseado  $y_{sp}(k)$ , fijando la función de transferencia del filtro F. En la mayo-

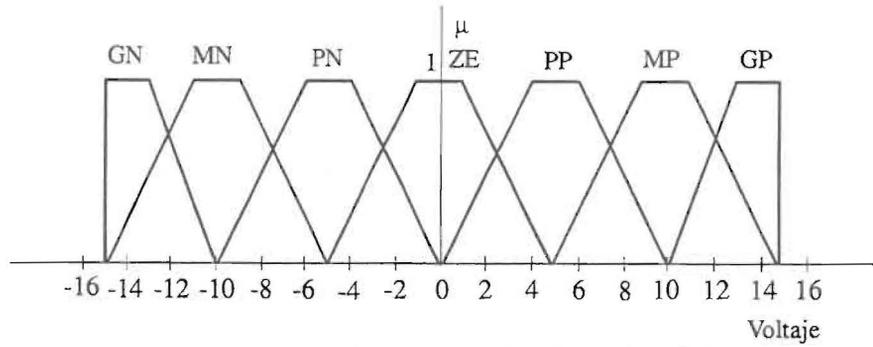


Figura 3. Conjuntos difusos para el voltaje. Modelo Matemático.

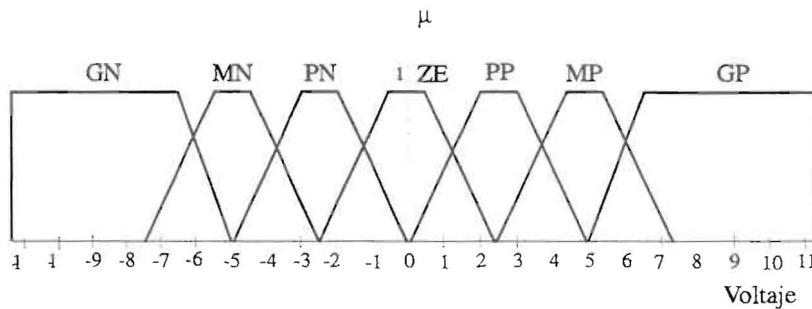


Figura 4. Conjuntos difusos para el voltaje. Motor.

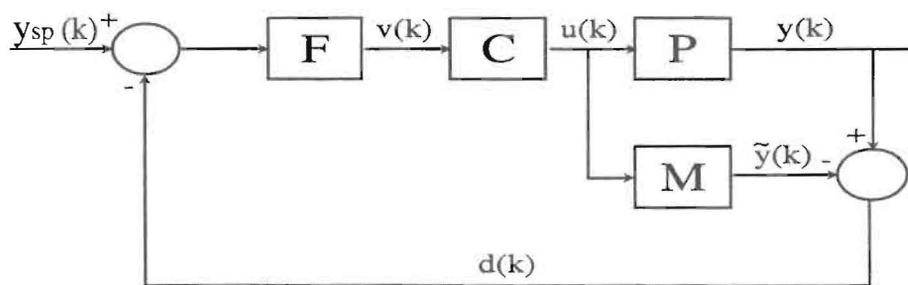


Figura 5. Estructura general del lazo de control de modelo interno.

ría de los casos se elige F igual a la función de transferencia de un filtro de primer orden, es decir:

$$F(z) = \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}} \tag{8}$$

donde  $\alpha$  se denomina constante del filtro. Esta constante varía entre 0 y 1 y es un parámetro que debe determinarse como parte del proceso de diseño del controlador. A menor valor de  $\alpha$ , se le exige al sistema que llegue al valor deseado en menor tiempo y por lo tanto, se requieren mayores valores de la señal de control.

Para implementar el Controlador de Modelo Interno de la Figura 5 se requiere la identificación de un modelo del proceso (M) y de la inversa de dicho proceso. Existe otra configuración de Controlador Basado en Modelo en el que sólo se requiere la identificación de la inversa del proceso que se está controlando. Dicha configuración se denomina Controlador Dead-Beat [9] y se muestra en la Figura 6. En el gráfico G es la función de transferencia de un filtro, cuya forma será determinada posteriormente, C representa la función de transferencia del controlador y P representa el proceso.

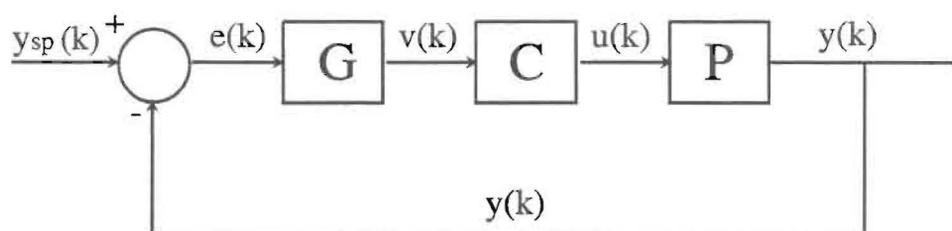


Figura 6. Estructura general del Controlador Dead-Beat.

Si se asume que  $C = P^{-1}$ , puede demostrarse que el lazo de control Dead-Beat presenta un comportamiento equivalente al lazo de Control de Modelo Interno si:

$$G = \frac{F}{1-F} \quad (9)$$

El comportamiento anterior de los controladores basados en modelo es válido porque se asume que el modelo del proceso es perfecto y además que la función de transferencia del controlador es la inversa de la función de transferencia del proceso. En la práctica esto no es posible y entonces se hace necesario utilizar un filtro que tenga una componente integradora para superar las discrepancias entre el modelo y el proceso.

Existen muchas herramientas que pueden utilizarse para realizar la identificación. Una herramienta que ha sido utilizada con frecuencia en los últimos años son las redes neurales. La red neural puede ser entrenada para que aprenda una relación funcional a partir de una data de entrada/salida. Existen muchos algoritmos de entrenamiento como retropropagación, retropropagación con momento y método de Levenberg-Marquardt, que utilizan el método del gradiente para minimizar dicha función del error. En este trabajo se utilizó el método de Levenberg-Marquardt para entrenar las redes neurales. Puede encontrarse mayor información de redes neurales y de estos algoritmos en Hecht-Nielsen [10] y Numerical Recipes [11].

El proceso para realizar identificación de modelos utilizando redes neurales es el siguiente: Se asume una forma del modelo, se genera la

data de entrenamiento, se construyen los patrones tanto para el modelo directo como para el modelo inverso de acuerdo a la forma asumida, se entrena la red hasta que el valor de error se estabilice.

Una vez entrenada la red, se implementó el lazo de Control Dead-Beat mostrado en la Fig. 6. Nahas *et al.* [7] plantean dos posibles enfoques para obtener, utilizando una red neural, el modelo de la dinámica inversa del proceso: inversa entrenada e inversa calculada en línea.

- Inversa Entrenada: el modelo de la dinámica inversa de un proceso puede escribirse de la siguiente manera:

$$u(k) = \Gamma[y(k+\theta+1), \dots, y(k+\theta-N_y+1), u(k-1), \dots, u(k-N_u+1)] \quad (10)$$

donde la señal de control  $u$  en el instante  $k$  es una función de  $N_y$  valores anteriores de la salida y  $N_u$  valores anteriores de  $u$ , en un proceso que presenta un tiempo muerto de  $\theta$ .

- Inversa calculada en línea: en este enfoque la inversa se determina en cada intervalo de muestreo resolviendo numéricamente la siguiente ecuación, para el valor  $u(k-\theta)$ :

$$G[u(k-\theta)] = VD - F[y(k), \dots, y(k-N_y+1), u(k-\theta), \dots, u(k-\theta-N_u+1)] = 0 \quad (11)$$

donde  $F$  es la aproximación del proceso utilizando una red neural,  $VD$  es el valor deseado de la salida del proceso en el instante  $k+1$ ,  $y(k+1)$  es la salida real del proceso en el instante  $k+1$ ,  $\theta$  es el tiempo muerto que podría existir y todos los

Tabla 2  
Información del entrenamiento de las redes neurales

	Modelo Matemático	Motor
Modelo directo	$\theta(k+1) = f[\theta(k), \theta(k-1), \theta(k-2), u(k)]$	$\theta(k+1) = f[\theta(k), \theta(k-1), u(k)]$
Error	$9.99 \cdot 10^{-4}$	$2.98 \cdot 10^{-3}$
Error promedio	$8.28 \cdot 10^{-4}$	$1.05 \cdot 10^{-3}$
Nº de iteraciones	200	500
Modelo inverso	$u(k) = f[\theta(k+1), \theta(k), \theta(k-1)]$	$u(k) = f[\theta(k+1), \theta(k), \theta(k-1)]$
Error	4.17	12.32
Error promedio	$5.35 \cdot 10^{-2}$	$6.73 \cdot 10^{-2}$
Nº de iteraciones	10000	20000

demás valores involucrados en la función son conocidos. Existen métodos que permiten determinar soluciones aproximadas de la ecuación (11) como por ejemplo Newton-Raphson y el método de bisección [11].

El método de bisección garantiza convergencia si la raíz existe y si se conoce el intervalo en el que ésta podría estar, que en este caso es  $[-15, 15]$ V. Si en el intervalo no existe ninguna raíz, el método detectará esta condición y en este caso se tomará como señal de control el extremo del intervalo que más acerque al sistema al valor deseado.

Tanto para el Modelo Matemático como para el motor se generó la data de entrenamiento de la siguiente manera: se utilizó un lazo cerrado de control con un controlador proporcional como lo proponen Low y Lee [12], se seleccionó un tiempo de muestreo de 0.01 seg, se generaron aleatoriamente 20 valores deseados en el intervalo  $[-60, 60]^\circ$ , con una duración de tiempo también aleatoria para cada uno de ellos. En algunas ocasiones se permitía que el proceso llegara al valor deseado y así la red podía aprender que debía producir una señal de control cero cuando el valor de posición llegara al valor deseado. La data generada para el modelo matemático tuvo un total de 1462 puntos, mientras que para el motor tuvo 2720 puntos. En la Tabla 2 se presenta información importante del entrenamiento de las redes neurales, tanto para el modelo matemático como para el motor.

- Implementación del Controlador: Para el modelo matemático se implementó el lazo de control dead-beat con la inversa calculada en línea y con la inversa entrenada. En ambos casos se utilizó un tiempo de muestreo de 0.01 seg. Se asumió  $\alpha = 0.98$ . Para el motor se implementó el controlador dead-beat con la inversa entrenada con  $\alpha = 0.985$ .

### Análisis y Discusión de los Resultados

A continuación se presentan y se discuten los resultados obtenidos al realizar algunas pruebas de control tanto en simulación como sobre el motor. En simulación se realizaron los cambios escalón de  $0^\circ$  a  $40^\circ$  y de  $0^\circ$  a  $-40^\circ$ , bajo la influencia de los cuatro controladores: PID, basado en Lógica Difusa (LD), Dead-Beat calculando la inversa en línea (RND) y Dead-Beat con la red neural que aprendió el modelo inverso (RNI).

Sobre el motor se realizaron las pruebas utilizando los controladores PID, basado en Lógica Difusa y Dead-Beat utilizando la red neural que modela la dinámica inversa del proceso. Inicialmente se realizó el siguiente conjunto de pruebas:

- (i) Cambios progresivos de  $0^\circ$  a  $20^\circ$ ,  $20^\circ$  a  $0^\circ$ ,  $0^\circ$  a  $-20^\circ$  y  $-20^\circ$  a  $0^\circ$ ,
- (ii) Cambios progresivos de  $0^\circ$  a  $40^\circ$ ,  $40^\circ$  a  $0^\circ$ ,  $0^\circ$  a  $-40^\circ$  y  $-40^\circ$  a  $0^\circ$  y

Tabla 3  
Valores obtenidos en las pruebas realizadas en simulación

	$t_r$ (seg)	$t_s$ (seg)	P.O.(%)		VF(°)
PID	0.5786	0.5786	0	4.8315	40
	0.5786	0.5786	0	4.8315	-40
LD	0.4471	1.5254	1.25	15.0755	40
	0.4471	1.5254	1.25	15.0755	-40
RND	0.789	0.789	0	5.6107	40
	0.789	0.789	0	5.5966	-40
RNI	0.789	0.789	0	5.5188	40
	0.789	0.789	0	5.5173	-40

(iii) Cambios progresivos de  $0^\circ$  a  $60^\circ$ ,  $60^\circ$  a  $0^\circ$ ,  $0^\circ$  a  $-60^\circ$  y  $-60^\circ$  a  $0^\circ$ . En estos tres casos se compara el desempeño de los tres controladores bajo todos los criterios establecidos.

Posteriormente se presenta la respuesta del motor a las siguientes señales sinusoidales:

- (i)  $VD(k) = 50 \cdot \text{sen}(2 \cdot \pi \cdot 0.3 \cdot k)$ ,
- (ii)  $VD(k) = 50 \cdot \text{sen}(2 \cdot \pi \cdot 0.5 \cdot k)$  y
- (iii)  $VD(k) = 50 \cdot \text{sen}(2 \cdot \pi \cdot 0.7 \cdot k)$ .

En este grupo de pruebas se realiza la comparación únicamente bajo el criterio de la integral del valor absoluto del error. Por último se establece una comparación entre los diferentes controladores desde el punto de vista de sus respectivos procedimientos de diseño.

#### Pruebas sobre el Modelo Matemático (Simulación)

En la Tabla 3 están los valores de los criterios para el caso de simulación. VF significa el valor final de posición del motor. La primera fila de cada controlador corresponde al cambio de  $0^\circ$  a  $40^\circ$ , mientras que la segunda corresponde al cambio  $0^\circ$  a  $-40^\circ$ . En la Figura 7 se muestra el comportamiento de la posición del modelo matemático para ambos cambios escalón.

Si se observan los valores de la Tabla 3 se puede notar que el controlador PID y los controladores basados en Redes Neuronales exhiben desempeño muy similar en ambos casos, siendo el

PID ligeramente mejor. El controlador basado en Lógica Difusa es el de peor desempeño ya que presenta mayores valores de tiempo de estabilización y de error. Además, es el único controlador que presenta un porcentaje de sobrepico diferente de cero. Al comparar los valores que corresponden a cada controlador para cada cambio escalón, se observa que son iguales lo que demuestra un comportamiento simétrico, característico de un modelo lineal independientemente de la zona de operación. En todos los casos, todos los controladores llevan al modelo matemático hasta el valor deseado y en todos los casos el voltaje final es muy cercano a cero.

#### Pruebas sobre el Motor

A continuación se muestran los resultados obtenidos de las pruebas realizadas sobre el motor utilizando los controladores diseñados: PID, basado en Lógica Difusa (LD) y basado en Redes Neuronales (RN). En la Figura 8 se muestra la respuesta del motor para los cambios escalón  $0^\circ$  a  $60^\circ$ ,  $60^\circ$  a  $0^\circ$ ,  $0^\circ$  a  $-60^\circ$  y  $-60^\circ$  a  $0^\circ$  y para la señal sinusoidal de  $f = 0.7$ . El comportamiento en los casos restantes es similar. Las Tablas 4 y 5 incluyen los valores de error y las medidas de desempeño respectivamente, para los cambios escalón. La Tabla 6 contiene los valores de error para las señales sinusoidales.

En el caso de los cambios escalón, se observa que el controlador PID es el que en general llega y se estabiliza más rápido. El controlador basado en Redes Neuronales es el que generalmente

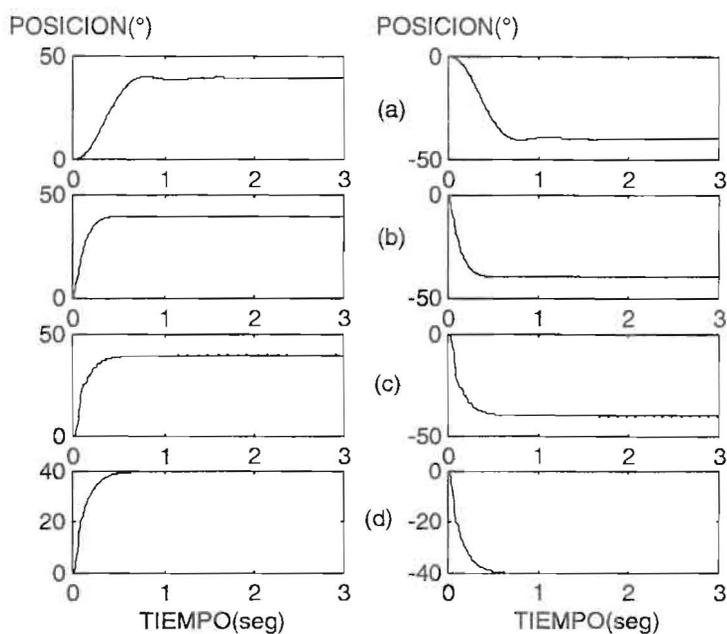


Figura 7. Posición del motor bajo la influencia de los controladores: (a) Lógica Difusa, (b) PID, (c) Dead-beat con inversa calculada y (d) Dead-beat con inversa entrenada.

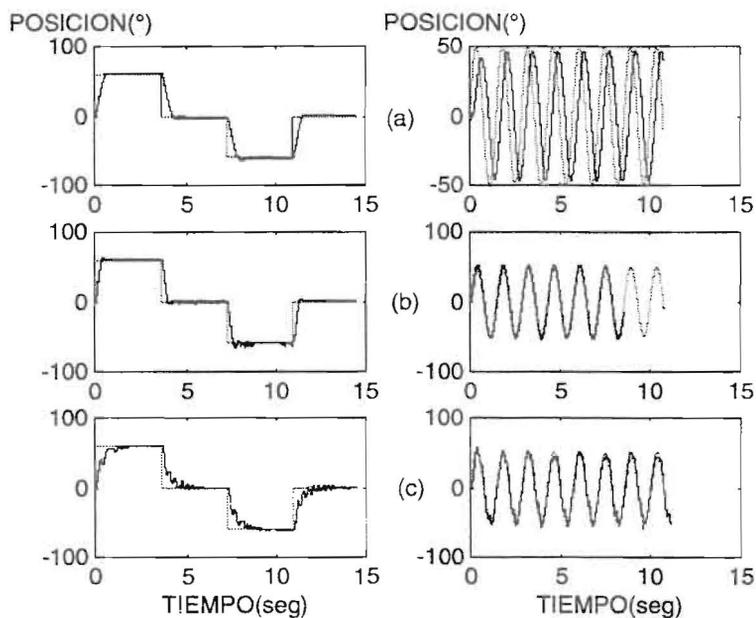


Figura 8. Respuesta del motor para los cambios escalón 0° a 60°, 60° a 0°, 0° a -60° y -60° a 0° y para la señal sinusoidal de  $f = 0.7$ , (a) LD (b) PID y (c) RN.

Tabla 4

Valores del error para los cambios escalón

Cambios Escalón	PID	LD	RN
0° a 20° a 0° a -20° a 0°	14.5019	37.8433	34.0721
0° a 40° a 0° a -40° a 0°	32.1410	56.3624	66.0270
0° a 60° a 0° a -60° a 0°	57.4816	92.6514	98.1658

Tabla 5  
Medidas de desempeño para los cambios escalón

		$t_r(\text{seg})$	$t_s(\text{seg})$	P.O.(%)	VF(°)
PID	0 a 20	0.3948	0.3948	0	20.1148
	20 a 0	0.329	0.658	3.333	0.0386
	0 a -20	0.329	0.658	3.333	-19.9604
	-20 a 0	0.329	0.329	0	0.2703
LD	0 a 20	0.7896	0.7896	0	19.8059
	20 a 0	0.9212	0.9212	0	1.5057
	0 a -20	0.9212	1.2502	1.667	-19.9604
	-20 a 0	0.658	0.658	0	0.6563
RN	0 a 20	0.7896	2.1056	6.667	20.1148
	20 a 0	0.658	0.658	0	-0.1158
	0 a -20	0.7238	2.8952	40	-19.6515
	-20 a 0	0.658	2.3688	11.67	0.0386
PID	0 a 40	0.3948	0.658	5.55	40.1138
	40 a 0	0.3948	0.3948	11.11	0.1930
	0 a -40	0.3948	0.658	5.55	-39.9593
	-40 a 0	0.329	1.316	4.17	0.3475
LD	0 a 40	0.658	0.658	0	40.4226
	40 a 0	0.7896	0.7896	0	-0.1930
	0 a -40	0.658	0.658	0	-39.9593
	-40 a 0	0.5922	0.5922	0	0.3475
RN	0 a 40	2.2372	2.2372	0	39.9593
	40 a 0	1.0528	1.0528	5.55	-0.0386
	0 a -40	0.9212	1.8424	2.78	-39.5733
	-40 a 0	0.7238	1.974	5.55	0.0386
PID	0 a 60	0.3948	0.658	5.8	60.5760
	60 a 0	0.3948	0.5922	8.7	0.3475
	0 a -60	0.3948	0.658	14.5	-60.3444
	-60 a 0	0.3948	0.9212	5.8	0.2703
LD	0 a 60	0.658	1.1844	4.35	61.4254
	60 a 0	0.7238	0.987	4.35	-2.2007
	0 a -60	0.658	1.316	5.8	-60.8849
	-60 a 0	0.658	0.658	0	1.8918
RN	0 a 60	1.974	1.974	0	59.9583
	60 a 0	1.2502	1.2502	2.9	-0.1158
	0 a -60	1.7108	1.7108	0	-60.2672
	-60 a 0	0.7896	1.974	5.8	0.2703

Tabla 6  
Valores del error para los cambios sinusoidales

	PID	LD	RN
f = 0.3	23.3938	156.6498	39.3089
f = 0.5	42.0052	284.8922	37.8031
f = 0.7	70.7589	386.7168	53.0369

presenta mayores tiempos de respuesta. En cuanto al máximo sobrepico, no puede establecerse una conclusión definitiva ya que no hay uniformidad en el comportamiento; en promedio el controlador PID tuvo un sobrepico de 5.7%, el basado en Lógica Difusa 1.35% y el dead-beat 6.5%. El controlador basado en Redes Neuronales lleva al motor al valor deseado a través de movimientos oscilatorios. En todos los casos, la diferencia entre el valor final y el valor deseado se mantiene entre el -1% y el 1% del valor deseado.

En cuanto a los cambios sinusoidales, se nota que a medida que aumenta la frecuencia se degrada la respuesta del motor ya que a éste le es más difícil seguir la entrada. Si se observan los resultados puede notarse que el PID funciona mejor que los otros dos controladores para  $f = 0.3$ , mientras que el controlador basado en Redes Neuronales se comporta mejor para  $f = 0.5$  y  $f = 0.7$ . Las respuestas correspondientes al controlador PID y al controlador basado en redes neuronales están casi superpuestas a la señal sinusoidal deseada. La respuesta del controlador basado en Lógica Difusa es la que mayor desviación presenta, ya que siempre está retrasada con respecto a la señal deseada y además, está generalmente por encima o por debajo de los máximos y de los mínimos de dicha señal.

### Conclusiones y Recomendaciones

Se cubrieron los objetivos propuestos: se logró implementar los controladores tanto sobre el modelo matemático como sobre el motor y comparar su desempeño en ambos casos. En simulación, las pruebas incluyeron un cambio escalón de  $0^\circ$  a  $40^\circ$  y otro de  $0^\circ$  a  $-40^\circ$ . Se demostró el comportamiento simétrico del modelo lineal. El controlador PID es el que logra un

mejor desempeño ya que es más rápido, su tiempo de estabilización es 62% menor que el del controlador basado en Lógica Difusa y 27% menor que el de los controladores basados en redes neuronales, y presenta un valor de la integral del valor absoluto del error 68% y 14% menor que los correspondientes al controlador basado en Lógica Difusa y a los controladores basados en redes neuronales, respectivamente. Los dos controladores basados en redes neuronales presentan un comportamiento casi idéntico. En todas las pruebas realizadas en simulación, todos los controladores llevaron al modelo exactamente hasta el valor deseado.

Sobre el motor se utilizaron variaciones escalón y variaciones sinusoidales del valor deseado. Ante los cambios escalón progresivos del valor deseado, se obtuvo un mejor desempeño del controlador PID. Los valores de tiempo de estabilización de este controlador son, en promedio, el 50% de los valores de tiempo de estabilización para el controlador basado en Lógica Difusa y el 33% de los valores correspondientes al controlador dead-beat con redes neuronales. Los valores de error para el controlador PID son el 50% de los correspondientes a los otros dos controladores. Los valores promedio de porcentaje de sobrepico fueron los siguientes: para el PID 5.7%, para el basado en Lógica Difusa 1.35% y para el dead-beat 6.5%. En todos los casos, los valores finales de la posición angular del motor presentaron un error entre -1% y 1% del valor deseado.

Ante los cambios sinusoidales del valor deseado se obtuvo lo siguiente: para  $f = 0.3$  el valor del error para el controlador PID es el 50% del valor correspondiente al controlador dead-beat y el 14.3% del valor del controlador basado en Lógica Difusa. Para  $f = 0.5$  el dead-beat presenta un valor de error 10% menor que el del PID y 87% menor que el del basado en Lógica Difusa.

Para el caso  $f = 0.7$  se observa un comportamiento similar al caso anterior, ya que el valor de la integral del absoluto del error para el dead-beat es 25% menor que el valor para el PID y 87% que el valor para el basado en Lógica Difusa. Se concluyen entonces que el controlador dead-beat tuvo el mejor desempeño para este conjunto de pruebas.

En cuanto al diseño, el controlador PID requiere la determinación de los valores de tres parámetros. Puede obtenerse un comportamiento aceptable para un intervalo de valores de estos parámetros, sin embargo, en algunas ocasiones se hace difícil lograr el desempeño óptimo. El controlador basado en Lógica Difusa, al igual que el controlador PID, presenta un comportamiento aceptable para varios valores de sus parámetros pero es complicado lograr el conjunto que permite obtener el óptimo.

El controlador dead-beat permite decidir qué forma tiene la respuesta si se halla una aproximación aceptable de la inversa, lo cual se hizo utilizando redes neurales. La implementación de la función inversa calculada en línea tiene la siguiente ventaja sobre la inversa entrenada: en el aprendizaje del modelo directo, la red realiza una mejor aproximación en un menor número de iteraciones. El controlador dead-beat con redes neurales demostró ser robusto, ya que se realizaron pruebas con diferentes arquitecturas de redes y en general se logró un control satisfactorio.

Luego de realizar este trabajo, han surgido posibles temas de investigación entre los cuales pueden citarse: considerar problemas de control no lineales, analizar cuáles son las condiciones de entrenamiento que se requieren para asegurar que un controlador basado en Redes Neuronales funcione y optimizar las funciones de membresía en el caso del controlador basado en Lógica Difusa.

### Agradecimiento

Los autores queremos expresar nuestro agradecimiento al Laboratorio de Controles e Instrumentación de la Escuela de Ingeniería Mecánica de la Universidad del Zulia, por facilitarnos el equipo experimental utilizado, en especial, al Profesor Javier Bastidas, por su colaboración

en la realización de este trabajo. Asimismo queremos agradecer al Profesor Carlos Vinante por su invaluable contribución técnica.

### Referencias Bibliográficas

1. Moreno, R., Montiel, E. "Estudio Experimental del Método del Relay para la Identificación del Modelo de un Servomotor". Trabajo Especial de Grado. Escuela de Ingeniería Mecánica. Universidad del Zulia. Abril 1992. 153 pp.
2. Anonymous, 1977. "DC Motor Speed Controls and Servo Systems". An Engineering Handbook, by Electro-Craft Corporation.
3. Li, Y. F. and Lau, C. C. "Development of Fuzzy Algorithms for Servo Systems". IEEE Control Systems Magazine. April 1989. pp 65-72.
4. Lee, C. C. "Fuzzy Logic in Control Systems". Part I and Part II. IEEE Transactions on Systems, Man and Cybernetics. Vol. 20, No. 2. March/April 1990. pp 404-435.
5. Nguyen, D., and Widrow, B. "Neural Network for Self-Learning Control Systems". IEEE Control Systems Magazine. April 1990. pp 18-23.
6. Khalid, M. y Omatu, S. "A Neural Network Controller for a Temperature Control System". IEEE Control Systems Magazine. June 1992. pp 58-64.
7. Nahas, E. P., Henson, M. A., Seborg, D. E.. "Nonlinear Internal Model Control Strategy for Neural Network Models". Computer and Chemical Engineering Magazine. Vol. 16, No. 12, 1992. pp 1039-1057.
8. Ogata, Katsuhiko. "Ingeniería de Control Moderna". Editorial Prentice Hall. Segunda Edición. 1993. pp 1020.
9. Vinante, C., Bastidas, J., Febres, G., Abud, L. "Controladores Predictivos: Análisis y Resultados Experimentales". Editado por la Oficina de Publicaciones de la Facultad de Ingeniería de la Universidad del Zulia. 1992. pp. 69.
10. Hecht-Nielsen, R. "Neurocomputing". Addison-Wesley Publishing Company. 1990. 433 pp.
11. Press, W., Flannery, B., Teukolsky, S. Vetterling, W. "Numerical Recipes. The Art of Scien-

- tific Computing". Cambridge University Press. First Edition. 1986. 1528 pp.
12. Low, T. and Lee, T. "A Methodology for Neural Network Training for Control of Drives with Nonlinearities". IEEE Transactions on Industrial Electronics. Vol. 39, No. 2, April 1990. pp 243-249.

Recibido el 18 de Febrero de 1997  
En forma revisada el 28 de Julio de 1997