

Control PI difuso de ganancias programables para un sistema mecatrónico de posicionamiento angular-lineal

Luis F. Cerecero-Natale^{1,2}, Julio C. Ramos-Fernández², Marco A. Márquez-Vera²,
Eduardo Campos-Mercado²

¹ Instituto Tecnológico de Cancún, Ingeniería en Mecatrónica,
Cancún, Quintana Roo, México

² Universidad Politécnica de Pachuca, Posgrado en Mecatrónica,
Zempoala, Hgo., México

lcerecero@itcancun.edu.mx, {luisfidelmecca, jramos, marquez, ecampos}@upp.edu.mx

Resumen. En este artículo se muestran los resultados experimentales de la integración sinérgica con: un sistema mecánico, un microcontrolador y una interface con el software EMC2 para aplicaciones de máquinas de control numérico (CNC). La contribución que se muestra en el presente trabajo, es el resultado experimental del diseño y programación de un controlador PI difuso de ganancias programables para el posicionamiento lineal de un sistema mecatrónico. La estrategia de diseño del controlador difuso es de 2-entradas y 2-salidas, el error y la derivada del error y las ganancias proporcional e integral, respectivamente. Se definieron 5 funciones de pertenencia del tipo gaussiana para fusificar el error y su derivada; se utiliza el conectivo AND en la premisa de las reglas difusas, para inferir el grado de disparo de cada regla difusa se codificó el operador producto (T-norm), el método del centroide es el mecanismo de defusificación. La estructura de la base de reglas son del tipo Takagi-Sugeno con los consecuentes de orden cero. Los resultados experimentales del control de posición en lazo cerrado, indican la viabilidad y efectividad de esta variante del controlador PI con ganancias programables para posicionamiento lineal de este tipo de servomecanismos. Este trabajo presenta los resultados experimentales comparativos, usando la regla clásica de sintonización de Ziegler-Nichols y el Controlador PI difuso de ganancias programables, para un tipo de sistema de posicionamiento, que es ampliamente utilizado en aplicaciones industriales.

Palabras clave: Controlador PI difuso, control de posición, máquinas herramientas.

Fuzzy Gain Scheduling PI Controller for Mechatronic Angle-Linear Positioning System

Abstract. In this paper we shows the experimental results using a microcontroller and hardware integration with the EMC2 software, using the Fuzzy Gain Scheduling PI Controller in a mechatronic prototype. The structure of the fuzzy

controller is composed by two-inputs and two-outputs, is a TITO system. The error control feedback and their derivative are the inputs, while the proportional and integral gains are the fuzzy controller outputs. Was defined five Gaussian membership functions for the fuzzy sets by each input, the product fuzzy logic operator (AND connective) and the centroid defuzzifier was used to infer the gains outputs. The structure of fuzzy rule base are type Sugeno, zero-order. The experimental result in closed-loop shows the viability and effectiveness of the position fuzzy controller strategy. To verify the robustness of this controller structure, two different experiments was making: undisturbed and disturbance both in closed-loop. This work presents comparative experimental results, using the Classical tune rule of Ziegler-Nichols and the Fuzzy Gain Scheduling PI Controller, for a mechatronic system widely used in various industries applications.

Keywords: Fuzzy gain scheduling, PI controller, position control, machine tools.

1. Introducción

El controlador PID es el más utilizado para controlar procesos industriales porque este tiene una simple estructura y un desempeño robusto. El diseño de este controlador solo necesita tres parámetros el proporcional, integral y derivativo, el controlador PID puede ser sintonizado usando la técnica bien conocida de Ziegler-Nichols [2]. Existen métodos alternativos para sintonizar las ganancias del PID que tienen estructuras similares a los controladores clásicos PID, donde las ganancias son adaptadas en línea basadas en auto-sintonización de redes artificiales wavenet para la estimación de parámetros [9]. Otra técnica muy utilizada es el control difuso (CD), con descripciones lingüísticas basadas en la experiencia de un experto humano, que se conoce como controlador del tipo Mamdani. Un trabajo de investigación que inspiró la presente propuesta, en donde se ilustra la filosofía y ejemplos para el diseño del controlador difuso PID de ganancias programables, se puede ver en [1].

En 1974 Mamdani fue uno de los pioneros [3], usando reglas de composición para inferencia que fueron propuestas por Zadeh [4], para controlar plantas no lineales. Un año después, Mamdani y Assilian desarrollaron el primer CLD, como se muestra en [5], este fue implementado satisfactoriamente para controlar un laboratorio tipo planta motriz de vapor. En un sentido estricto, el primer controlador difuso mostrado en [5] fue equivalente a un sistema difuso de 2 entradas PI, donde el error y la derivada del error, son usadas como entradas para el mecanismo de inferencia [6]. Takagi-Sugeno (TS) proponen una estructura de controladores difusos con los consecuentes en forma de polinomios [7], esta fue una propuesta que tendió un puente para el estudio y desarrollo de la lógica difusa integrando las técnicas de modelado y control clásico, moderno, adaptable y técnicas con descripción diferencial, donde los consecuentes de cada regla difusa están representados con estructuras matemáticas lineales o no lineales. Los controladores PID con diferentes configuraciones tipo TS se muestran con un análisis de estabilidad [8]. En la últimas tres décadas, en el campo de la aplicación de los controladores difusos, obtuvieron prestigio incluyendo aplicaciones: industriales, automotrices, aeroespaciales, médicas, agricultura de precisión, por citar algunas. Trabajos de investigación teórica en este sentido han aportado el soporte científico para

el diseño y desarrollo de controladores difusos, en donde se considera el estudio y análisis de estabilidad, como se ilustra en [15].

En el presente trabajo, la estructura del PI difuso de ganancias programables usa consecuentes lineales de orden cero (Singleton). Así, se muestran los resultados de aplicar el algoritmo de control PI difuso de ganancias programables, embebido en un microcontrolador, para controlar la posición angular-lineal de un sistema mecatrónico. Este artículo está organizado de la siguiente manera: en la Sección 2 se describe la estructura del Controlador PI Difuso de Ganancias Programables, la Sección 3 presenta los componentes de la Plataforma de Experimentación, la Sección 4 describe la Interfaz Hombre-Máquina del Sistema Mecatrónico, la Sección 5 muestra el Diseño del controlador y los Resultados Experimentales. Las Conclusiones están incluidas en la Sección 6.

2. Controlador PI difuso de ganancias programables

En este trabajo se propone un Controlador PI Difuso de Ganancias Programables (CPIDGP), para controlar la posición Angular y en consecuencia el avance Lineal de un sistema mecatrónico, el bloque difuso consta de 2-entradas y 2-salidas, como se ilustra en la Fig. 1, las entradas son el error e y su derivada respecto del tiempo \dot{e} , las 2 salidas son las ganancias proporcional K_{pDif} e integral K_{iDif} , ambas ganancias defusificadas se introducen al controlador PI clásico, de donde se obtiene la variable manipulada u , es decir la salida cambia cada instante dentro del ciclo de control.

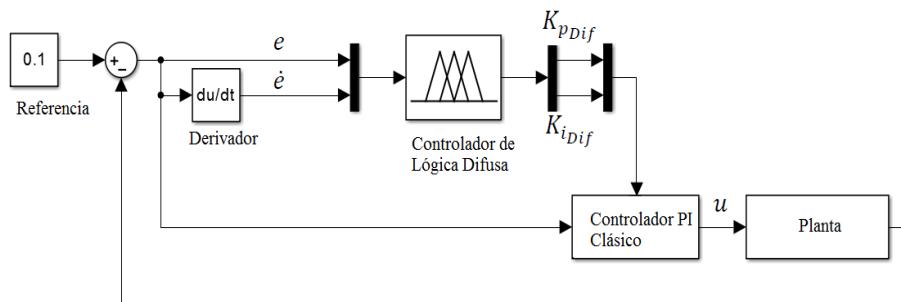


Fig.1. Sistema de Control tipo PI con Difuso de Ganancias Programables

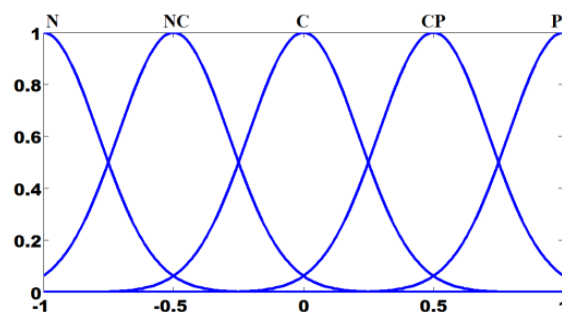


Fig. 2. 5 conjuntos difusos tipo Gaussianas para e y \dot{e} , normalizadas

Las variables lingüísticas e y \dot{e} , son definidas por 5 conjuntos difusos del tipo Gaussianas, mostradas en la Fig. 2, para estas $\sum_{i=1}^5 \mu_i(e) = 1$, $\sum_{i=1}^5 \mu_i(\dot{e}) = 1$, donde $\mu_i(e)$ y $\mu_i(\dot{e})$ son valores de las funciones de pertenencia en el instante dentro del ciclo de control.

Tabla 1. Reglas difusas de sintonización para las ganancias K_{pDif} y K_{iDif}

Conjuntos difusos		Salidas Singleton	
error	derivada del error	ganancias difusas	
e	\dot{e}	K_{pDif}	K_{iDif}
N	N	1.00	1.00
N	NC	1.50	1.50
N	C	2.00	2.00
N	CP	2.50	2.10
N	P	3.00	2.20
NC	N	3.25	2.30
NC	NC	3.50	2.40
NC	C	3.75	2.50
NC	CP	4.00	2.60
NC	C	4.25	2.70
C	N	4.50	2.80
C	NC	4.75	2.90
C	C	5.00	3.00
C	CP	4.75	2.90
C	C	4.50	2.80
CP	N	4.25	2.70
CP	NC	4.00	2.60
CP	C	3.75	2.50
CP	CP	3.50	2.40
CP	C	3.25	2.30
P	N	3.00	2.20
P	NC	2.50	2.10
P	C	2.00	2.00
P	CP	1.50	1.50
P	C	1.00	1.00

El controlador difuso fue diseñado para calcular las salida K_{pDif} y K_{iDif} en cada ciclo de control. La base de conocimiento está compuesta por 25 reglas difusas, para los cinco conjuntos difusos, para las entradas del error y su derivada, normalmente son

las posibles combinaciones. En la Tabla 1 se muestra la base de reglas, que fue implementada en este trabajo.

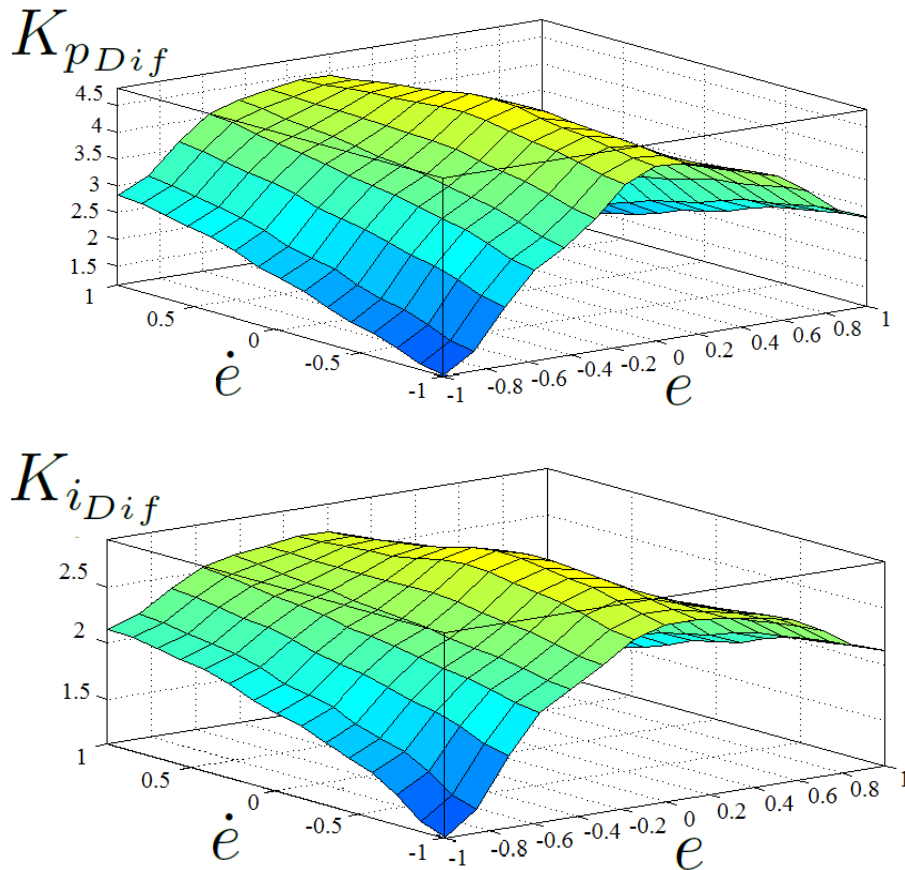


Fig. 3. Superficie de control de las salidas K_{pDif} y K_{iDif} del controlador difuso

La estructura de reglas difusas de CPIDGP se muestra en (1), método fusificación, defusificación y denormalización. La base de reglas difusas es:

$$R_j: \text{Si } e \text{ es } A_j \text{ and } \dot{e} \text{ es } B_j \text{ Entonces } K_{pDif_j} = a_j \text{ y } K_{iDif_j} = b_j, \quad (1)$$

donde A_j y B_j son los conjuntos difusos dados por el diseñador, K_{pDif_j} y K_{iDif_j} son las ganancias proporcional e integral para $j = 1, 2, \dots, n$, donde n son las 25 reglas, a_j y b_j son los consecuentes con los valores de la Tabla I.

La ecuación (2) mecanismo de inferencia tipo producto:

$$\beta = (\mu_e \cdot \mu_{\dot{e}}). \quad (2)$$

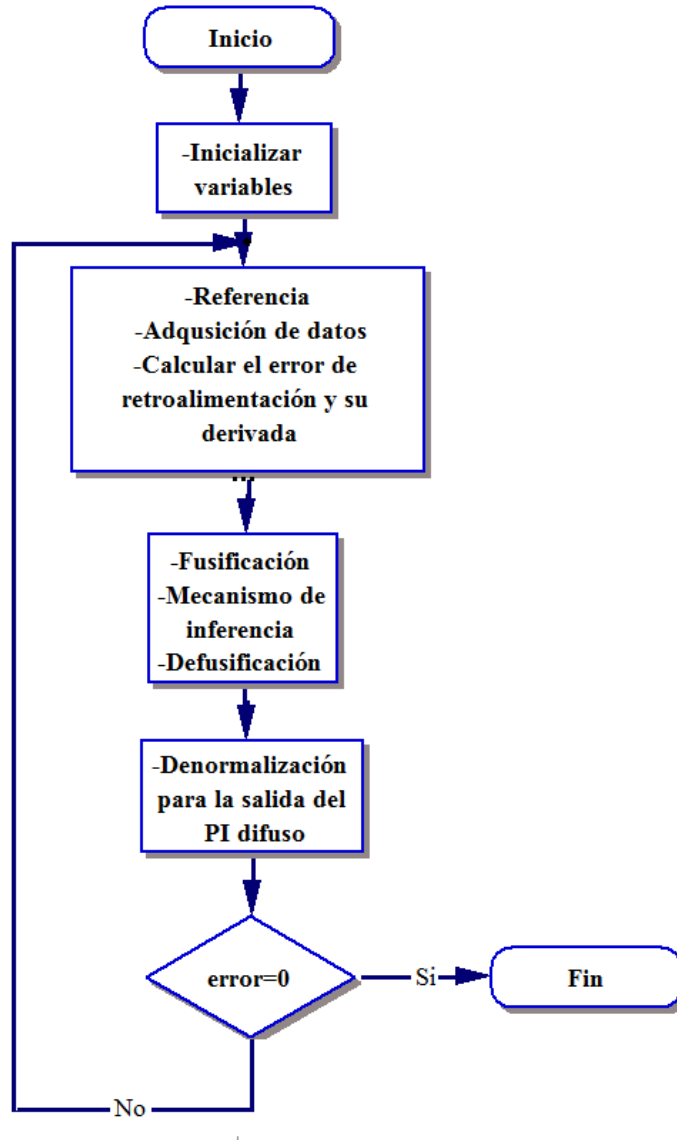


Fig. 4. Diagrama de flujo del controlador PI Difuso de Ganancias Programables

Defusificación de K_{pDif} y K_{iDif} son las ecuaciones (3) y (4):

$$K_{pDif} = \frac{\sum_{j=1}^n \beta_j K_{pDif j}}{\sum_{j=1}^n \beta_j}, \quad (3)$$

$$K_{iDif} = \frac{\sum_{j=1}^n \beta_j K_{iDifj}}{\sum_{j=1}^n \beta_j}, \quad (4)$$

donde $K_{pDif} \in [1, 5]$ y $K_{iDif} \in [1, 3]$.

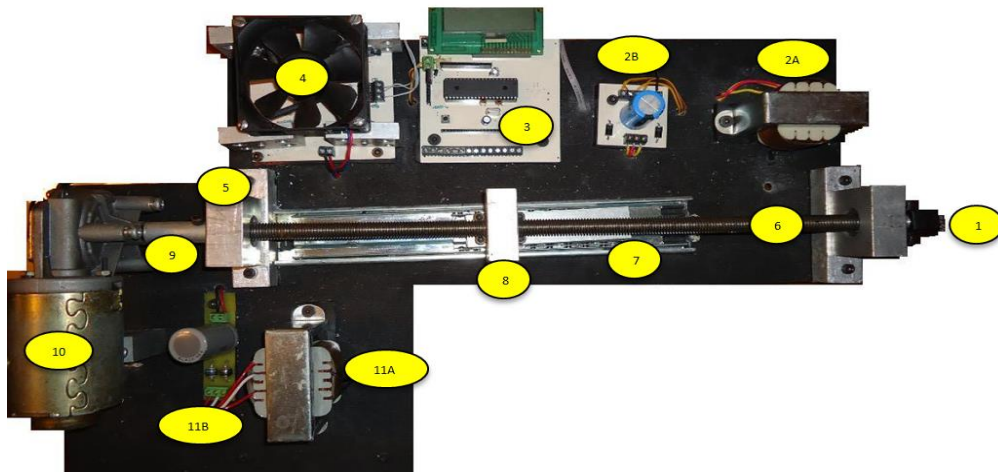


Fig.5. Componentes del sistema mecatrónico

El cálculo de cada ganancia, se realiza por medio de las ecuaciones (5) y (6):

$$K_p = K_{pDif} \cdot K_{pZ-N}, \quad (5)$$

$$K_i = K_{iDif} \cdot K_{iZ-N}, \quad (6)$$

donde K_{pZ-N} y K_{iZ-N} son ganancias calculadas a partir de la respuesta del sistema usando el método de sintonización de Ziegler–Nichols, en este caso, es necesario para obtener la respuesta al escalón del sistema mecatrónico de posicionamiento, la respuesta es de un sistema de primer orden con retardo en el transporte. Esta base de conocimiento es esencial en este artículo para el modelado e identificación del sistema. En este trabajo usamos por conveniencia las ecuaciones de transformación lineal (7) y (8) para normalizar e y \dot{e} entre -1 y 1 :

$$e' = \frac{e - e_{min}}{e_{max} - e_{min}}, \quad (7)$$

$$\dot{e}' = \frac{\dot{e} - \dot{e}_{min}}{\dot{e}_{max} - \dot{e}_{min}}. \quad (8)$$

La Fig. 3 muestra la superficie de control no lineal para las ganancias proporcional e integral difusas. En la Fig. 4 se muestra un diagrama de flujo del procedimiento que se ejecuta dentro del microcontrolador al compilar y ejecutar el CPIDGP.

3. Plataforma de experimentación

La Fig. 5 ilustra el prototipo donde fueron realizados los experimentos para verificar el desempeño del algoritmo CPIDGP propuesto en este artículo. La Tabla 2 contiene los nombres de los componentes que integran el sistema mecatrónico de posicionamiento Angular-Lineal.

Tabla 2. Componentes del sistema mecatrónico

(1)	Sensor de posición tipo encoder de 256 pulsos por revolución	(7)	Deslizador lineal
(2A, 2B)	Fuente de alimentación para sistema embebido (3)	(8)	Caja de tuerca
(3)	Sistema embebido de control basado en microcontrolador Microchip	(9)	Acoplamiento del Motor(10) con el husillo (6)
(4)	Puente H MOSFET, con sistema de ventilación por aire	(10)	Motor de Corriente Directa de Imán Permanente (PMDCM)
(5)	Sistema suspensión rígida (baleros)	(11A, 11B)	Fuente de alimentación de potencia, para (4) que controla a (10)
(6)	Husillo roscado, $\varphi = 12.7mm$, Resolución lineal $1.958mm = 1$ revolución		

4. Interfaz hombre-máquina del sistema mecatrónico

El software Enhanced Machine Controller (EMC2) es mucho más que un simple programa de fresadora de Control Numérico Computarizado (CNC). Puede controlar máquinas-herramientas, robots u otros dispositivos automatizados. Se puede interactuar para controlar servomotores, motores paso a paso (stepper), relevadores y otros dispositivos relacionados con las máquinas herramientas. Hay cuatro componentes principales en el software EMC2: un controlador de movimiento, un controlador de Entradas/Salidas discretas, un ejecutor de tareas para coordinar las interfaces gráficas de usuario. Además hay una capa llamada HAL (Hardware Abstraction Layer), que permite la configuración de EMC2 sin la necesidad de compilar. La Fig. 6 es un diagrama de bloques que muestra la integración del sistema mecatrónico con la computadora y Linux como sistema operativo, donde se puede programar una trayectoria de posición por medio de Código G, para controlar el motor de corriente continua, la referencia de posición mediante el envío de señales a través del puerto paralelo. Normalmente, los actuadores son motores paso a paso que operan en lazo abierto, la contribución en el presente trabajo es el manejo de las operaciones de avance mecánico con un servomecanismo en lazo cerrado con un controlador del tipo difuso, la EMC2 también puede ejecutar a través de tarjetas de interfaz de servomotores o por medio de un puerto paralelo extendido para conectar con los tableros de control externo [10]. Una de las contribuciones de este desarrollo tecnológico es el control de posición

para máquinas-herramientas y se sometió a prueba el algoritmo embebido en el microcontrolador integrado: el Controlador PI Difuso de Ganancia Programable; el uso de la base de conocimientos mediante el método de sintonización a la respuesta al escalón de Ziegler-Nichols, para identificar los parámetros del Motor de Corriente Directa de Imán Permanente (MCDIP).

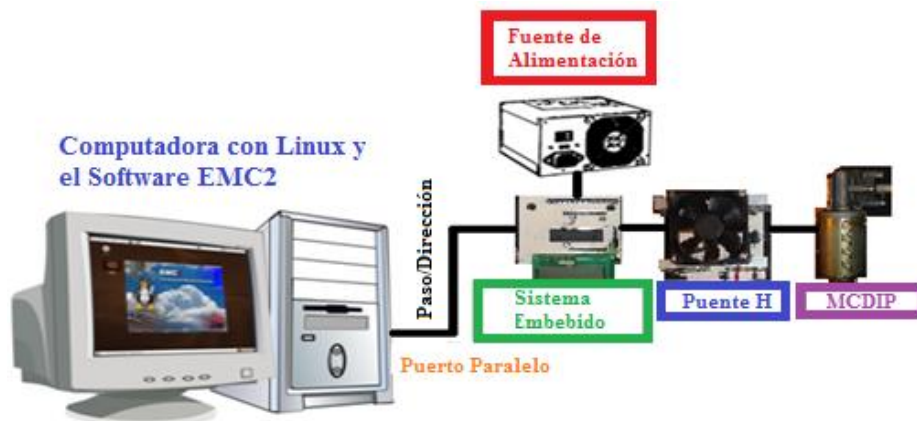


Fig.6. Interfaz Hombre-Máquina con el software EMC2

En este trabajo para la Interfaz Hombre-Máquina (HMI) se utilizó la EMC2, un pin del puerto paralelo con el fin de impulsar el sentido de giro y otro pin se utiliza para asignar la referencia, usando sólo 2 pines del puerto paralelo, por lo tanto puede controlar un sistema embebido, que consiste en el microcontrolador PIC18F4550 de Microchip, en el que se programó el algoritmo CPIDGP.

5. Resultados experimentales

Esta sección presenta la principal contribución de este artículo; la identificación de la planta fue realizada y esta descrita en [11]. Aquí, se desarrollan nuevos experimentos en la misma plataforma, donde la simulación numérica en lazo abierto de la función de transferencia en lazo abierto, se hace por medio del software MATLAB, posteriormente lo aplicamos en el microcontrolador en tiempo real. Presentamos dos clases de experimentos en tiempo real; uno sin perturbación (sin carga) y el segundo con perturbación (aplicando cargas externas).

5.1. Respuesta en lazo abierto

Para encontrar la función de transferencia experimental $\left(\frac{\text{Posición}}{\% \text{ PWM}}\right)$ de la plataforma, se aplica una señal constante del 80% del ciclo útil de trabajo de una señal Modulada en Ancho de Pulso (PWM) con una frecuencia de operación de $13 \frac{\text{rad}}{\text{s}}$. De acuerdo al análisis de la gráfica de Bode como se muestra en [11], La función de transferencia es:

$$\frac{\theta(s)}{U(s)} = \frac{0.1076}{0.318s + 1} \cdot e^{-0.076s}, \quad (9)$$

donde $\theta(s)$ corresponde a la transformación Angular-Linear de la posición que recorre la caja de tuerca (ver Fig. 5 and Tabla II el elemento (8)) y $U(s)$ es el ciclo útil de trabajo de la señal PWM. Al realizar la identificación paramétrica experimental se obtienen las siguientes constantes: ganancia del sistema 0.1076, la constante de tiempo de 0.318 y el retardo en el transporte 0.076, ambos en segundos.

5.2. Diseño del controlador difuso

El principal objetivo en el diseño del controlador difuso descrito en este artículo es regular y controlar la posición del sistema mecatrónico sin sobreimpulso en su posición. Porque en las máquinas herramientas de CNC, una respuesta dinámica con sobreimpulso es crítica y daña las piezas que se maquinan [12]. Usando el método de Ziegler-Nichols [13], se sintoniza el CPICZN, donde las ganancias proporcional e integral son: $K_{pZ-N} = 3.7658$ y $K_{iZ-N} = 14.8650$, respectivamente. La Fig. 7 muestra los resultados de los controladores en una prueba de simulación, para contrastar las salidas del sistema usando el CPICZN contra el CPIDGP, donde la respuesta de tiempo del CPIDGP es más rápida que la del CPICZN, se observa que en la respuesta del control de posición ambas convergen a la referencia.

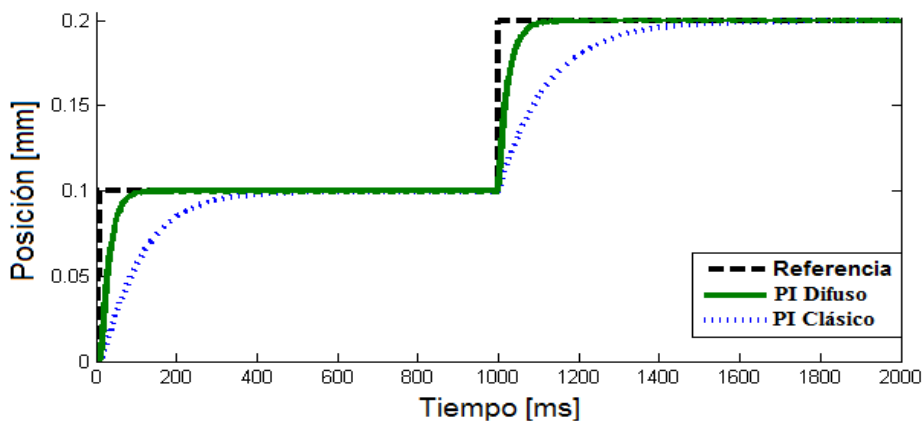


Fig. 7. Simulación del sistema en lazo cerrado bajo un cambio en la señal de referencia

5.3. Resultados experimentales sin perturbación

En esta Subsección se muestran los resultados experimentales de los controladores CPICZN y CPIDGP. Es importante enfatizar que los algoritmos de los controladores están corriendo en un simple microcontrolador PIC18F4550 de Microchip. La señal de salida PWM y de los pulsos del encoder son dados en [11], Donde cada pulso representa

0.007632mm de avance lineal del mecanismo. Con el fin de seguir una trayectoria de referencia de $\frac{1}{10}mm$, es necesario que se acumulen 13 pulsos en el conteo del encoder.

La primera prueba experimental fue realizada sin carga (sin perturbación), el comportamiento de la plataforma con ambos controladores es mostrada en la Fig. 8, como se puede ver existe un retardo en ambos controladores, en el CPICZN son de 0.18s y 0.028s, para la primera y segunda referencia respectivamente, mientras que para el CPIDGP son de 0.116s y 0.104s. Sin embargo el tiempo de convergencia del CPIDGP es más rápido que el CPICZN y sin sobreimpulso.

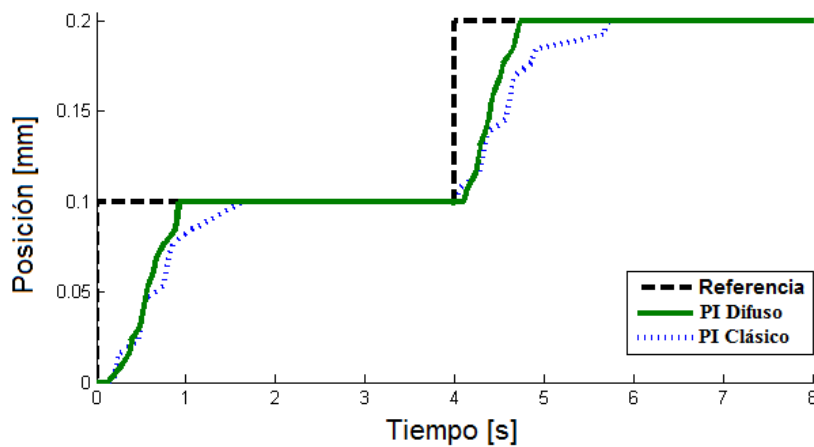


Fig.8. Desempeño de los controladores sin carga

5.4. Resultados experimentales con perturbación

La segunda prueba experimental se realiza con una carga de 2.788kg, esta masa es colocada sobre la caja de tuerca. La respuesta dinámica con ambos controladores es mostrada en la Fig. 9 como se puede ver en esta figura, existe retardo en el transporte para ambos controladores. En el CPICZN son 0.212s y 0.06s para la primera y segunda referencia respectivamente. Mientras que para el CPIDGP son de 0.088s y 0.128s. Sin embargo en el controlador CPICZN hay error en el estado estacionario, porque los parámetros de sintonización no son robustos.

La tercera prueba experimental se realiza con una carga de 4.934kg sobre la caja de tuerca, el comportamiento del CPICZN y CPIDGP son comparados en la Fig. 10, donde el retardo en el transporte para el CPICZN es: 0.212s y 0.264s para la primera y segunda referencia respectivamente, mientras que para el CPIDGP los retardos en la primera y segunda referencia son: 0.104s y 0.132s. Al igual que el experimento anterior, el comportamiento del controlador CPIDGP en lazo cerrado, converge más rápido que el CPICZN, no reside error de retroalimentación en el estado estacionario y siempre llega a la señal de referencia deseada.

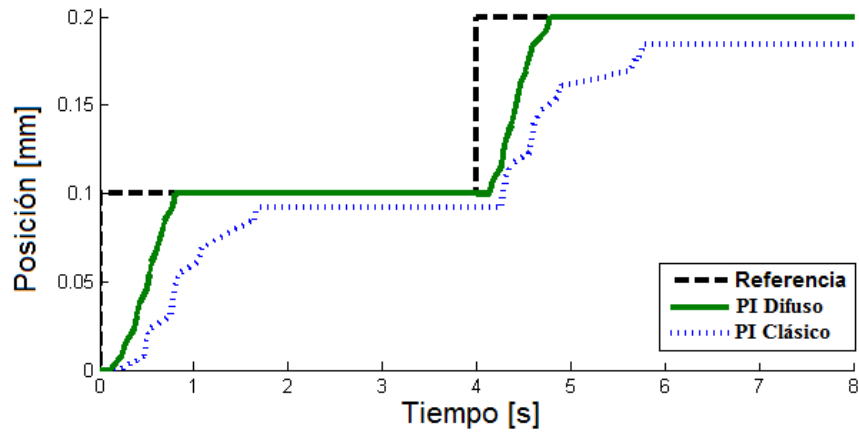


Fig. 9. Desempeño de los controladores con una carga de 2.788 kg

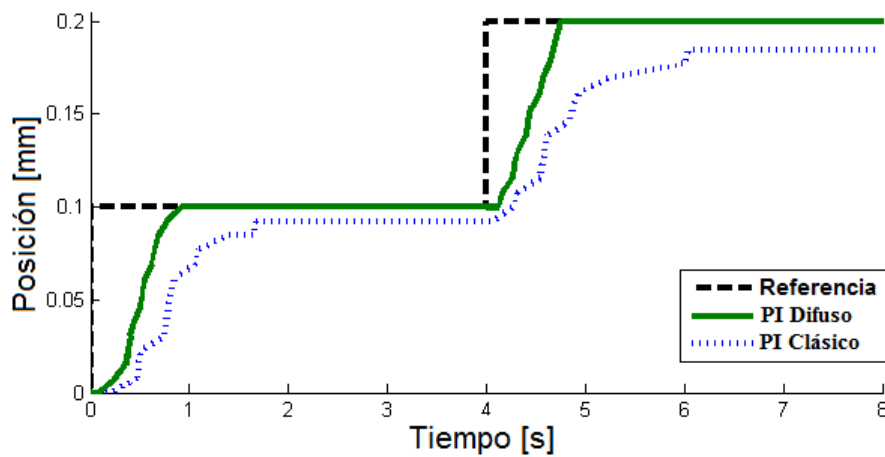


Fig. 10. Desempeño de los controladores con una carga de 4.934 kg

6. Conclusiones

La principal contribución en este artículo son los resultados experimentales usando el algoritmo embebido en un microcontrolador: CPIDGP con nulas respuestas de sobreimpulso y de error estacionario, estos efectos no se permiten en máquinas de desbaste o maquinados de control numérico. Estas dos características son importantes en los sistemas mecatrónicos, ya que puede ser utilizado en máquinas herramientas de alta precisión.

El algoritmo CPIDGP fue implementado en un microcontrolador de bajo costo 18F4550 de Microchip. Fueron utilizadas Funciones de membresía Gaussianas como entradas del CPIDGP, porque presentan una superficie más suave para inferir las

ganancias proporcional e integral difusas, sin sobreimpulso y con mínima oscilación en relación a la funciones de pertenencia triangular y trapezoidal. La principal contribución en este artículo es el uso de la interfaz EMC2 en lazo cerrado con el MCDIP, esto es una nota importante ya que los motores paso a paso son usados en la mayoría de las máquinas herramientas; sin embargo en este desarrollo tecnológico fueron reemplazados por MCDIP.

Referencias

1. Zhao, Z.-Y., Tomizuka, M., Isaka, S.: Fuzzy gain scheduling of PID controllers. *IEEE Transactions on systems, Man and Cybernetics*, Vol. 23, No. 5, pp.1392–1398 (1993)
2. Hang, C.C., Aström, K.J., Ho, W.K.: Refinements of the Ziegler-Nichols tuning formula. *IEE Proc. D, Control Theory and Applications*, Vol. 138, pp. 111–118 (1991)
3. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Elect. Eng. Contr. Sci.*, Vol. 121, pp. 1585–1588 (1974)
4. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst., Man, Cybernetic.*, Vol. SMC-3, No. 1, pp. 28–44 (1973)
5. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man-Mach. Stud.*, Vol. 7, No.1, pp. 1–13 (1975)
6. Mann, G.K.I., Hu, B.-G., Gosine, R.G.: Analysis of direct action fuzzy PID controller structures. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 29, No. 3, pp. 371–388 (1999)
7. Takagi, T., Sugeno, M.: Fuzzy identification of system and its applications to modeling and control. *IEEE Trans. Syst., Man Cybernetics*, Vol. 15, No.1, pp. 116–132 (1985)
8. Ding, Y., Ying, H., Shao, S.: Typical Takagi-Sugeno PI and PD fuzzy controllers: analytical structures and stability analysis. *Information Sciences*, Vol. 151, pp. 245–262, (2003)
9. Gómez, O.I., Ramos-Velasco, L.E., Ramos-Fernández, J.C., García-Lamont, J., Espejel, M.A.: Implementation of different wavelets in an auto-tuning wavenet PID controller and its application to a DC motor. *IEEE Electronics, Robotics and Automotive Mechanics Conference*, pp. 301–306 (2011)
10. Organization Linux: Human-Machine Interface EMC2, <http://www.linuxcnc.org/> [Accessed: 26 Oct. 2015]
11. Cerecero-Natale, L.F., Ramos Fernandez, J.C., Ramos-Velasco, L.E., Pedraza Vera, V.E.: Fuzzy gain scheduling PI controller for a mechatronic system. *World Automation Congress (WAC)*, pp.1–5, Puerto Vallarta México (2012)
12. Overby, A.: *CNC Machining Handbook, Building, Programming, and Implementation*. McGraw-Hill (2011)
13. Aström, K.J., Wittenmark, B.: *Adaptive Control*. Addison-Wesley Series in Electrical and Computer Engineering: Control Engineering, (1989)
14. Aracil, J., Gordillo, F.: *Stability Issues in Fuzzy Control*. Physica-Verlag Heidelberg, Series Studies in Fuzziness and Soft Computing (2000)