



## **Desarrollo e Implementación de un Sistema Difuso para un Robot Neumático con software de Monitoreo**

E. Guillén García, O.M. Cora Gallardo, J.M. Ramos Arreguin,  
C.A. Ramos Arreguin, J.E. Vargas Soto, M. Delgado Rosas

**Resumen:** El presente trabajo conjunta diferentes herramientas de software y hardware para controlar el movimiento de un brazo manipulador de un grado de libertad, con actuador neumático en lugar de un actuador eléctrico. La principal aportación del trabajo, es el desarrollo de una arquitectura hardware para la implementación de un algoritmo difuso, programado en un FPGA. El sistema lo complementa una aplicación de software en PC, que nos permite enviar el valor de referencia al sistema, y al mismo tiempo, monitorear el desplazamiento del brazo, a través de la comunicación serial RS232. El software es desarrollado en Visual Basic. Otra herramienta importante en el sistema, es Matlab, utilizado para el desarrollo del algoritmo difuso y la generación de una tabla de acceso a datos para su implementación en hardware. Finalmente, se prueba el sistema de una manera práctica y los resultados son satisfactorios.

**Abstract:** This work presents the application of different software and hardware tools, to achieve the robot manipulator control, considering only one degree of freedom. This robot uses a pneumatic actuator instead of an electrical actuator. The major contribution of this works is the hardware development to implement a fuzzy logic algorithm in hardware, with an FPGA programming device. The system is complemented by Visual Basic software, as PC interface with the hardware. This software let us to send the set point to the control hardware and visual results with graphics behavior are possible, with RS232

communication. Matlab is used to develop the fuzzy logic algorithm. When the fuzzy algorithm is ready, a look-up table is generated using Matlab, which is used as one module for hardware implementation. Finally, the system is tested in a practical way and the results obtained are satisfactory.

### **Introducción**

La lógica difusa está siendo ampliamente utilizada en la solución de problemas que tienen un comportamiento no lineal, donde un control convencional PID no tiene la eficiencia deseada. Sin embargo, el uso de algoritmos de lógica difusa implementados en hardware, ha sido utilizado de muy diversas maneras, especialmente para diversos tipos de control, como lo realizado en años anteriores, cuando se desarrolla la síntesis automática de controladores difusos [1], y posteriormente, su implementación [2]. Un poco después fue reportado un software para el desarrollo de algoritmos difusos, y su implementación en hardware [3], de la misma manera, se han implementado controladores difusos digitales basados en los modelos de una-entrada-una-salida (SISO) y múltiple-entrada-múltiple-salida (MIMO) [4]. En años siguientes, es desarrollada, una implementación en hardware de un sistema difuso tipo 2 [5]. Enseguida, se desarrolla una plataforma basada en FPGA para sistemas difusos con reconfiguración dinámica, basado en la metodología de cooperación co-evolucionaria, para incrementar la velocidad de ejecución del sistema [6]; de igual manera, es implementado un controlador difuso para un vehículo eléctrico [7]. Con el incremento en la popularidad de los controladores lógicos difusos en la industria, especialmente cuando los modelos matemáticos no están disponibles o son inexactos, se llega a la implementación de controladores difusos en hardware, utilizando tecnología FPGA [8]. De igual manera, la lógica difusa también es aplicada en el diseño e implementación de semáforos [9].

El manejo de actuadores neumáticos en el campo de la robótica ha sido aplicado en trabajos recientes [10,13], donde se presentan diversos algoritmos de control, incluyendo el PID tradicional, lógica difusa y redes

---

Emmanuel Guillen García, imanolgg@hotmail.com  
Orlando Marcos Cora Gallardo, orlando\_marcos@hotmail.com  
Juan Manuel Ramos Arreguín, jramos@mecamex.net  
Carlos Alberto Ramos Arreguín, cramos06@alumnos.uaq.mx  
José Emilio Vargas Soto, emilio@mecatronica.net  
Manuel Delgado Rosas, mdelgado@uaq.mx  
Universidad Autónoma de Querétaro, Facultad de Informática,  
Campus Juriquilla, Av. De las Ciencias S/N, Col. Juriquilla,  
Delegación Santa Rosa Jauregui, Querétaro, México.



neuronales, así como resultados de implementación de algoritmos en hardware.

Este artículo se enfoca en el desarrollo de un sistema digital para el posicionamiento de un actuador neumático, el cuál funciona como actuador en un robot manipulador de un grado de libertad. Este sistema incluye el desarrollo de un algoritmo de lógica difusa, su implementación en hardware, y el desarrollo de un software para establecer el valor de referencia, así como el monitoreo de la posición actual del brazo. La principal aportación de este trabajo, es el manejo de un sistema neumático de una manera práctica, utilizando técnicas de lógica difusa, y tecnología FPGA, presentando una técnica para implementar algoritmos difusos en un sistema embebido con FPGA. El sistema digital controla una electro-válvula proporcional 5/3, utilizando la técnica PWM para controlar el flujo y dirección del aire.

### El sistema neumático

La figura 1 muestra el esquema del robot neumático de un grado de libertad, donde el actuador neumático está colocado en posición vertical, y  $X$  representa el desplazamiento del vástago y  $\theta$  el ángulo de inclinación del brazo, respecto a la vertical. En la figura se observa que para colocar el brazo en un ángulo  $\theta$ , es necesario desplazar el vástago una distancia  $X$  respecto al fondo del cilindro, controlando el flujo y dirección del aire.

El movimiento del brazo está limitado dentro del intervalo de  $0^\circ$  a  $164^\circ$ .



Fig. 1 Sistema del robot neumático de un grado de libertad.

En la figura 2 se muestra el sistema neumático utilizado en el esquema de la figura 1, Donde  $EV_1$  y  $EV_2$  son señales digitales que controlan el flujo y dirección del aire a través de la electro-válvula 5/3.

El cilindro utilizado en este trabajo es el mismo que en [10], donde el cilindro neumático cuenta con cámaras de amortiguamiento en ambos lados, como se muestra en la figura 3, en donde  $p_1$  y  $p_2$  son las entradas de presión de aire,  $F_a$  es la fuerza del actuador y  $L$  la longitud de carrera del vástago. La carrera del vástago es 11 mm.

### FPGA (Field Programmable Gate Array)

Un FPGA es un dispositivo programable de propósito general. Integra una gran cantidad de dispositivos lógicos programables en un chip.

El tamaño y velocidad de los FPGAs es equiparable a los Circuitos Integrados para Aplicaciones Específicas (ASICs), pero los FPGAs son más flexibles y su ciclo de diseño es más corto [7].

Un FPGA es un circuito integrado que puede ser programado para realizar cualquier tipo de función digital, este es programable a nivel hardware, por lo que proporciona las ventajas de un procesador de propósito general y un circuito especializado. Su programación se realiza con la ayuda de firmware, y una vez programado el FPGA es capaz de mantener su funcionalidad hasta que se vuelva a reprogramar.

Para implementar un diseño en un FPGA se utilizan lenguajes de descripción de hardware (HDL: Hardware Description Languages), el más conocido es VHDL, sin embargo, no es la única opción, dado que existen otras alternativas como Handel-C.

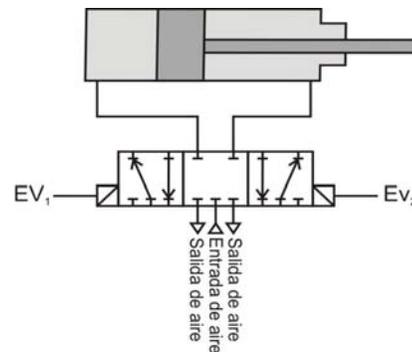


Fig. 2 Sistema neumático utilizado por el robot de un grado de libertad

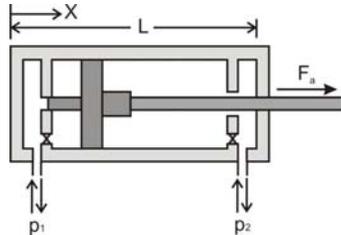


Fig. 3 Diagrama del cilindro neumático utilizado

### Lenguajes de descripción de Hardware (HDL)

HDL describe el comportamiento del Hardware, a diferencia de un lenguaje de programación tradicional donde programa de forma secuencial. En HDL, se ejecutan los procesos en forma paralela y siempre se mantiene en ejecución.

Un diseño digital puede ser creado en cualquier editor que cuente con símbolos gráficos de circuitos o usando un lenguaje de descripción de Hardware como Verilog o VHDL (Very high speed integrated circuit Hardware Description Language).

### Implementación del sistema digital en FPGA.

El sistema digital implementado se explica, de una manera gráfica, y por medio de bloques descriptivos de hardware.

La entidad principal del hardware llamada "Fuzzy\_Electrovalvula", se encarga del procesamiento de los datos y la comunicación entre el sistema digital y una PC, esto para el correcto manejo del sistema digital y el almacenamiento de la información a través de una comunicación serial RS232.

La figura 4 muestra el diagrama de bloques descriptivo de la parte interna de la entidad Fuzzy\_Electrovalvula. A continuación se describe el funcionamiento de estos módulos.

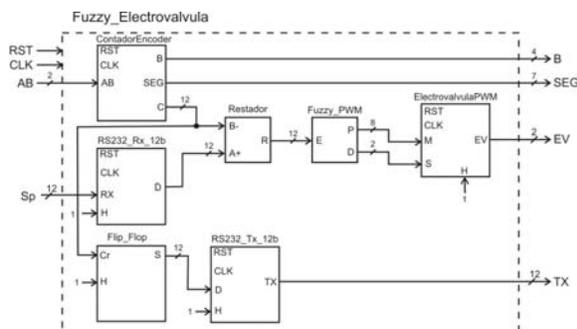


Fig. 4 Diagrama de bloques descriptivo de la entidad Fuzzy\_Electrovalvula

El módulo **CONTADORENCODER** de la figura 5, está formado por cuatro módulos, los cuales son: "DetectorSecuencias", que se utiliza como interfaz con el encoder óptico para determinar si el brazo se encuentra en movimiento y conocer la dirección en que se mueve.

El "ContadorIncDec\_n", utiliza la información del detector de secuencias. Si el brazo se encuentra en movimiento, un contador de 12 bit se incrementa ó decreta dependiendo de la dirección del movimiento, y permite conocer el ángulo de la posición actual del brazo.

El bloque "Bin\_BCD\_12bit" recibe la señal del contador para convertirlo a su equivalente en BCD.

El bloque "Barrido4Exh" ayuda a exhibir el valor de la cuenta en 4 exhibidores de 7 segmentos.

El módulo **RESTADOR** se utiliza para calcular el error de la posición actual respecto a la referencia, donde A representa la señal del valor de referencia, y B el valor de la señal de la posición actual del brazo.

Para el control de la posición del brazo, se utiliza un algoritmo de Lógica Difusa desarrollado en Matlab, el cual es implementado en hardware, y que en la figura 4 se representa por el módulo FUZZY\_PWM.

Este módulo es una tabla de acceso a datos (LUT; Look-Up Table) generada por la descripción de Matlab. Las figuras 6 y 7 muestran las funciones miembro de las señales de entrada y salida.

El programa desarrollado en MatLab está compuesto por una entrada que maneja 3 funciones de mimbrecía de tipo trapezoidal que corresponden a un error positivo, negativo o cero. El error es definido por el modulo restador ya mencionado.

El intervalo activo del bloque FUZZY\_PWM es de 45 a 75 unidades. Mientras mayor es P, mayor velocidad angular del brazo, y a menor valor de P, menor velocidad angular.

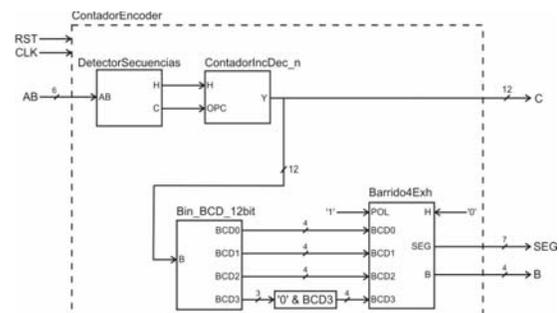


Fig. 5 Diagrama de bloques descriptivo de ContadorEncoder

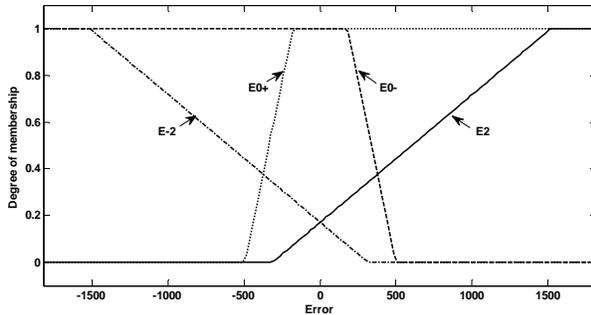


Fig. 6 Funciones miembro de entrada

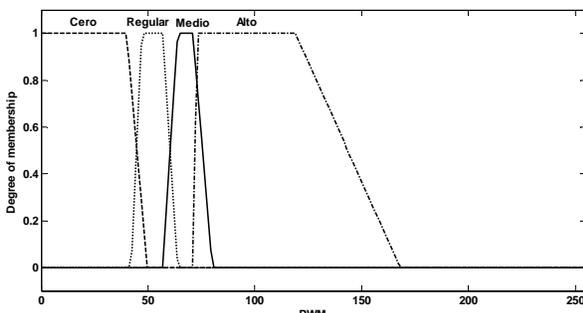


Fig. 7 Funciones miembro de salida

El módulo **ELECTROVALVULAPWM** de la figura 8, genera la señal PWM que se envía a la electroválvula con un ciclo de trabajo definido por el valor en hexadecimal de la entrada D. Este módulo utiliza un divisor de frecuencia que genera un pulso en alto de 20 ns cada 25 ms, que es el tiempo de actualización que se maneja en el sistema; otro módulo maneja el tiempo de duración en bajo y el tiempo de duración en alto; y finalmente, una máquina secuencial se encarga de sincronizar el funcionamiento del generador de la señal PWM.

El módulo **FLIP\_FLOP** retiene el valor del contador de 12 bit de la figura 6, actualizando su valor cada 25 ms para que realicen las correcciones al Generador de PWM.

El módulo **RS232\_TX\_12B** envía el valor del contador de 12 bit, a la PC por medio serial RS232, donde un programa en Visual C# se encarga de manejar la información en bloques de 12 bit, para almacenar y graficar el comportamiento del brazo. Se envía un paquete de 12 bit cada 25 ms.

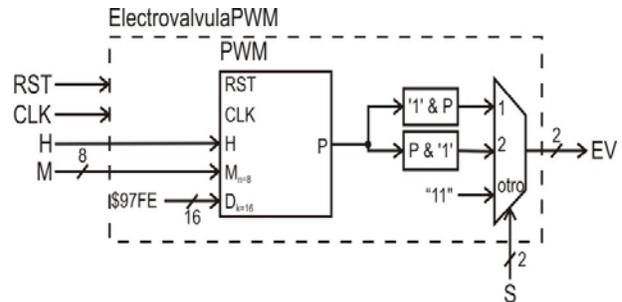


Fig. 8 Diagrama de bloques descriptivo de ElectrovalvulaPWM

El módulo **RS232\_RX\_12B** recibe del programa en Visual C# vía RS232, el valor de referencia deseado por el usuario. Este valor es usado como la entrada B en el módulo restador de la figura 4.

### Proceso de Envío/Recepción en la PC

La interfaz generada por el programa en visual que se encarga del envío del valor de referencia y de la recepción de los datos de la posición actual del brazo por el puerto serial, está hecha en el lenguaje de programación Visual C#.

El medio por el cual se envían los 12 bit del valor de referencia (set point) al sistema digital en el FPGA es serial, con el protocolo RS232. El diagrama de flujo para el proceso de envío de la referencia se muestra en la figura 9, considerando que la transferencia de datos es en paquetes de 8 bit.

Donde la variable “Enviar” almacena el dato a transmitir por el puerto serial, “Pbaja” almacena los 8 bit menos significativos de la referencia a transmitir, y “Palta” almacena los 8 bit más significativos de la referencia. En el proceso de recepción de la posición actual del brazo se deben utilizar solamente 12 bit. El proceso de recepción se muestra en la figura 10.

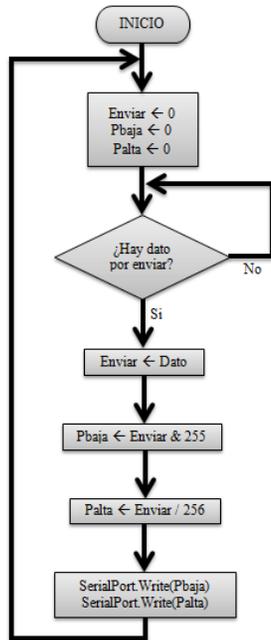


Fig. 9 Diagrama del proceso de envío de datos desde la PC

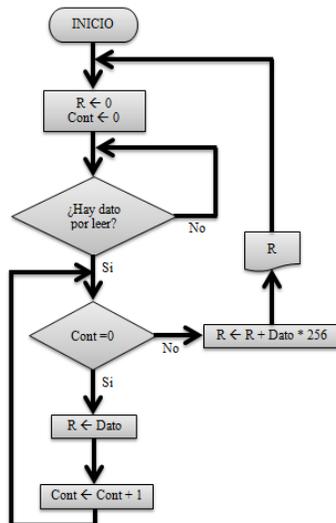


Fig. 10 Diagrama del proceso de recepción de datos desde la PC

En la figura 10, “R” es la variable que almacena los 12 bit de datos de la posición actual del brazo, procesando dos paquetes de 8 bit. La variable “Cont” es un contador que identifica si el paquete recibido es el menos significativo o el más significativo.

### Resultados.

Para probar el funcionamiento del brazo, se realizó la implementación del sistema digital en un FPGA Spartan-3 de Xilinx, con número de dispositivo XC3S200, con capacidad de 200 Kcompuertas lógicas, y con una frecuencia de reloj de 50 MHz. Esta implementación permite un sistema embebido compacto y económico.

La tabla 1 muestra la capacidad consumida del dispositivo en la implementación del sistema completo.

Tabla 1: Recursos utilizados en el FPGA Spartan-3.

Utilización Lógica	Usado	Disponible	% Uso
Registros	122	3840	3
Número de LUTs	559	3840	14
Número de Slices	301	1920	15
Número de Bloques I/O	27	173	15
Número de BUFGMUXs	1	8	12

La figura 11 se muestra el cilindro neumático utilizado, así como el brazo y la compresora que alimenta a la electroválvula.

Las figuras 12, 13 y 14 muestran el resultado de probar el sistema digital, aplicado al brazo manipulador de un grado de libertad, con un actuador neumático. Se probó con diferentes valores de puntos de referencia, los cuales fueron 19°, 80° y 164°.

En los resultados, se puede observar que el posicionamiento del brazo manipulador está dentro del 10% del valor del punto de referencia, lo que es aceptable, tomando en cuenta que es necesario el poder desarrollar las herramientas para retroalimentar la velocidad de desplazamiento del actuador neumático.

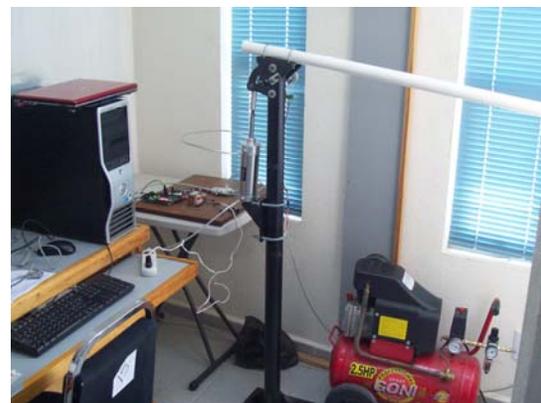


Fig. 11 Brazo con actuador neumático y sistema digital de control.

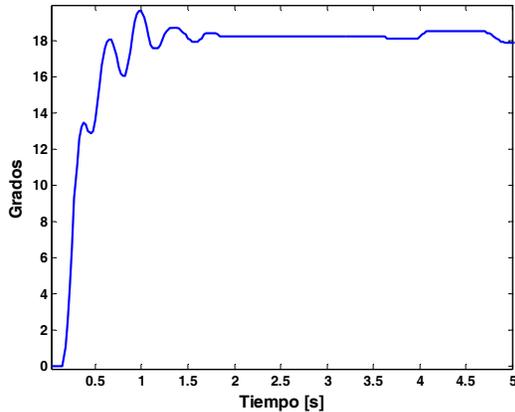


Fig. 12 Prueba realizada con un punto de referencia de 19°

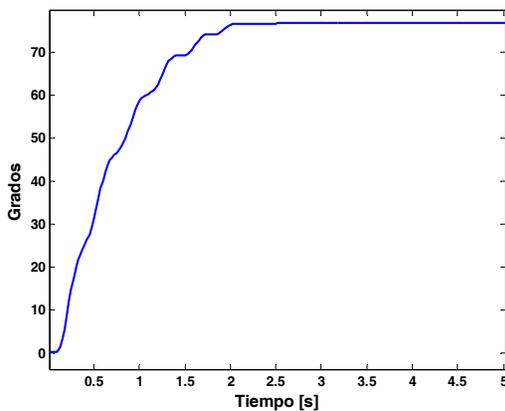


Fig. 13 Prueba realizada con un punto de referencia de 80°

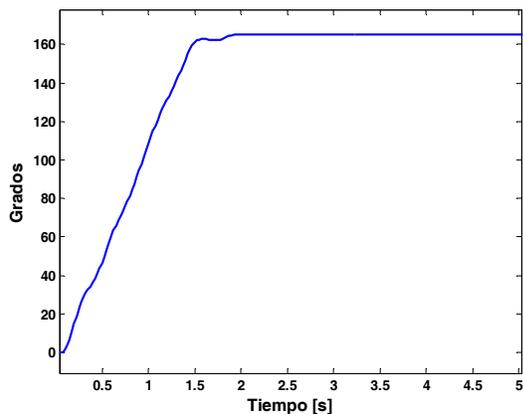
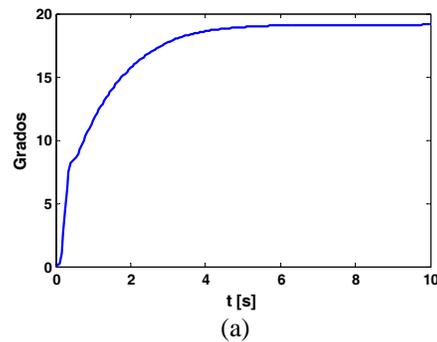


Fig. 14 Prueba realizada con un punto de referencia de 164°

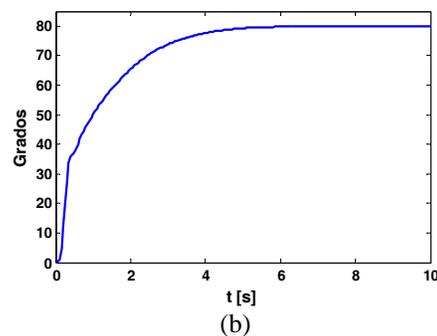
El sistema digital toma una muestra de la posición del brazo cada 25 ms, debido a que el tiempo de respuesta de la electro-válvula 5/3 es de 19 ms. En pruebas realizadas, no se puede tener periodos de muestreo menores a 25 ms, debido a que el comportamiento del sistema no tiene un buen desempeño, pues en ocasiones no se tenía flujo de aire, y el valor DC debe ser demasiado alto. Es importante destacar que en todo momento, el sistema muestra una estabilidad, lo cual lo hace confiable.

En trabajos previos de simulaciones de control difuso, donde el algoritmo difuso reajusta los valores de las constantes del control PID [13], se obtienen resultados como los mostrados en la figura 15.

Las diferencias importantes, son que el sistema práctico llega más rápido al punto de referencia, que en el caso de las simulaciones, sin embargo, se presentan oscilaciones en los resultados prácticos, que en la simulación no se tienen, por lo que es evidente se debe incluir en el sistema difuso la velocidad de desplazamiento del brazo.



(a)



(b)

Fig. 15 Resultados de la simulación del uso de lógica difusa en el control del actuador neumático. (a) Referencia de 19°. (b) Referencia de 80°.



En pruebas prácticas realizadas con un controlador PID [12], las cuales se muestran en la figura 16, donde se puede apreciar una serie de respuestas escalón, presentándose una oscilación cuando la referencia es mayor de 90°, debido al efecto de gravedad en el sistema, y al actuador le es más difícil mantener la posición. En este aspecto, la implementación con lógica difusa ha mejorado la respuesta, permitiendo llegar a los 160° de rotación.

### Conclusiones.

Este trabajo presenta la implementación en hardware de un algoritmo difuso para controlar la posición de un actuador neumático. Se utiliza un FPGA Spartan-3 para realizar las pruebas prácticas, aprovechando las ventajas de los dispositivos lógicos reprogramables.

La implementación realizada en este trabajo, a diferencia de otros presentados previamente [13] [12], aplicado a un manipulador neumático, muestra un sistema más económico y compacto, sobre todo en comparación con [12], donde se utilizaron dos válvulas de flujo proporcionales, las cuales consumen 1A de corriente y cada válvula requiere una interfaz de potencia. Estos elementos son costosos y no se consiguen fácilmente en el país, por lo que es necesario utilizar elementos más comunes. Este trabajo ha reducido la complejidad de la electrónica al utilizar únicamente una electro-válvula controlada utilizando la técnica de PWM, reduciendo el consumo de energía y costos del sistema. La implementación en hardware permite que la evaluación del algoritmo difuso sea en 20 ns, independientemente de la complejidad de las reglas y las funciones miembro, teniendo solamente como una limitante, la capacidad del dispositivo difuso. Se logra combinar de una manera práctica las técnicas de control difuso a un sistema neumático, para controlar la posición, teniendo como principal dificultad la alta no linealidad del aire, así como el efecto de la gravedad en el sistema neumático.

Como parte de los trabajos futuros, está el mejorar el proceso difuso y comparar resultados prácticos con los resultados de utilizar un algoritmo PID tradicional, para poder realizar planeación de trayectorias en actuadores neumáticos.

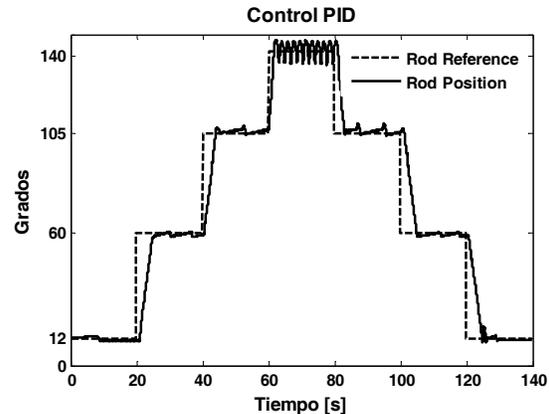


Fig. 16 Resultados de la implementación de un control PID para el actuador neumático.

### Referencias

- [1] A. Barriga, S. Sánchez-Solano, C. J. Jiménez, D. Gálan and D. R. López, "Automatic Synthesis of Fuzzy Logic Controllers", *Mathware & Soft Computing*, vol. 3, no. 3, pp. 425-434, 1996.
- [2] E. Lago, M. A. Hinojosa, C. J. Jiménez, A. Barriga, S. Sánchez-Solano, "FPGA Implementation of Fuzzy Controllers", in XII Conference on Design of Circuits and Integrated Systems (DCIS'97), pp. 715-720, 1997.
- [3] D. R. López, C. J. Jiménez, I. Baturone, A. Barriga and S. Sánchez-Solano, "Xfuzzy: A Design Environment for Fuzzy Systems", in *Seventh IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'98)*, pp. 1060-1065, Anchorage - Alaska, 1998.
- [4] M. J. Patyra and J. L. Grantner, "Hardware implementations of digital fuzzy logic controllers", *Information Sciences: an International Journal*, volume 113, Issue 1-2, pp. 19-54, January, 1999, ISSN:0020-0255.
- [5] M. A. Melgarejo and C. A. Peña-Reyes, "Hardware Architecture and FPGA Implementation of a Type-2 Fuzzy System", *Proceedings of the 14th ACM Great Lakes symposium on VLSI*, pp. 458 - 461, 2004, ISBN 1-58113-853-9.
- [6] G. Mermoud, A. Upegui, C. A. Peña and E. Sanchez, "A Dynamically-Reconfigurable FPGA Platform for Evolving Fuzzy Systems", in IWANN 2005, LNCS 3512, pp. 572-581, 2005.
- [7] S. Poorani, T. V. S. Urmila-Priya, K. U. Kumar and S. Renganarayanan, "Fpga Based Fuzzy Logic Controller for Electric Vehicle", *Journal of The Institution of Engineers*, Singapore, vol. 45 Issue 5, pp. 1-14, 2005.
- [8] K. M. Deliparaschos, F. I. Nenedakis and S. G. Tzafestas, "Design and Implementation of a Fast Digital Fuzzy Logic Controller Using FPGA Technology", *Journal of Intelligent and Robotic Systems*, vol. 45, pp. 77-96, 2006.



- [9] C. Karakuzu and O. Demirci, "Fuzzy Logic Based Smart Traffic Light Simulator Design and Hardware Implementation", *Applied Soft Computing*, volume 10, Issue 1, pp. 66-73, January, 2010, ISSN:1568-4946.
- [10] J. M. Ramos, E. Gorrostieta, J. C. Pedraza, R. J. Romero and B. Ramírez, "Pneumatic Cylinder Control for a Flexible Manipulator Robot", in *12<sup>th</sup> IEEE International Conference on Methods and Models in Automation and Robotics*, pp. 637-641, 2006.
- [11] E. Gorrostieta, J. M. Ramos and J. C. Pedraza, "Fuzzy and Neuronal Control to Flexible Manipulator", in *International Journal of Factory Automation, Robotics and Soft Computing*, Issue 2, pp. 155-160, ISSN 1828-6984, April 2007.
- [12] J. M. Ramos-Arreguin, J. C. Pedraza-Ortega, E. Gorrostieta-Hurtado, E. J. Romero-Troncoso, J. E. Vargas-Soto and F. Hernandez-Hernandez, "Pneumatic Fuzzy Controller Simulation vs Practical Results for Flexible Manipulator", in *Automation and Robotics*, I-Tech Education and Publishing, pp. 191-200, ISBN 978-3-902613-41-7, 2008.
- [13] J. M. Ramos-Arreguin, J. C. Pedraza-Ortega, E. Gorrostieta-Hurtado, R. J. Romero-Troncoso, "Artificial Intelligence Applied into Flexible Manipulator", *Seventh Mexican International Conference on Artificial Intelligence*, pp. 339-345, 2008.

## Currículo corto de los autores

### **Emmanuel Guillén García**

Alumno de 6° semestre de la carrera de Ingeniería en computación INC07 de la Facultad de Informática, Universidad Autónoma de Querétaro.

### **Orlando Marcos Cora Gallardo**

Alumno de 8° semestre de la carrera de Ingeniería en computación INC07 de la Facultad de Informática, Universidad Autónoma de Querétaro.

### **Juan Manuel Ramos Arreguín**

Ingeniero en Electrónica egresado de la Facultad de Ingeniería Mecánica, Eléctrica y Electrónica (FIMEE) en 1994, de la Universidad de Guanajuato. Termina la Maestría en Ingeniería Eléctrica en la FIMEE, en 1997. Sus estudios de Doctorado en Ciencias con opción Terminal en Mecatrónica, fueron en el Centro de Ingeniería y Desarrollo Industrial (CIDESI), en el 2008. Actualmente es Profesor en la Universidad Autónoma de Querétaro y pertenece al Sistema Nacional de Investigadores en el nivel de Candidato.

### **Carlos Alberto Ramos Arreguín**

Alumno de 9° semestre de la carrera Ingeniería en Computación INC99 de la Facultad de Informática,

Universidad Autónoma de Querétaro, realiza su estancia profesional en el Centro de Investigación y Desarrollo en Informática (CIDIT).

### **José Emilio Vargas Soto**

Ingeniero Mecánico Electricista egresado de la Universidad Nacional Autónoma de México en 1988. Obtuvo el grado de Maestría en Tecnologías de Fabricación por la Universidad Politécnica de Madrid en 1992, el Doctorado en Ciencias Físicas por la Universidad Complutense de Madrid en 1994, y el Postdoctorado en Sistemas de Telepresencia por la Universidad de Telecomunicaciones de Tokio en 1997. Actualmente se desempeña como profesor-investigador en la Facultad de Informática de la Universidad Autónoma de Querétaro.

### **Manuel Delgado Rosas**

Egresado del Instituto Tecnológico de Querétaro como Ingeniero Industrial Mecánico en 1986 y del Instituto Tecnológico de Celaya, Gto., como Maestro en Ciencias en Ingeniería Mecánica en 1999. Realizó estudios de Doctorado en Ingeniería en Procesos Industriales en la Universidad de La Rochelle (Francia), de 1999-2002. Del 2002 al 2007 trabajó como profesor-investigador en el Centro de Ingeniería y Desarrollo Industrial (CIDESI), adscrito al Posgrado Interinstitucional en Ciencia y Tecnología. Actualmente es profesor-investigador de la Facultad de Informática de la Universidad Autónoma de Querétaro (UAQ). Pertenece al SNI como Investigador Nivel 1 y es miembro del Registro CONACYT de Evaluadores Acreditados (RCEA).