

Controlador difuso parametrizable basado en un núcleo de procesamiento reconfigurable descrito en VHDL

Agenor Polo Zabaleta¹, Mario Esmeral Palacio²

Resumen

En este artículo se describe el diseño de un controlador difuso en hardware. Estos dispositivos normalmente son construidos para una planta en específico, lo cual dificulta su rango de aplicabilidad. El desarrollo presentado se basa en un procesador difuso parametrizable, lo que permite su configuración para ser utilizado en el control de distintas plantas sin variar su hardware. En este caso es posible modificar, utilizando reglas tipo Mamdani, la partición del universo de discurso de cada variable de entrada desde 1 hasta 5 conjuntos difusos, con funciones de membresía triangular o Γ -linear para las variables de entrada y conjuntos tipo singleton para la salida, lo cual reduce la complejidad de cálculo del desfusificador. El desarrollo fue implementado mediante VHDL y sintetizado en un FPGA Cyclone Ep1c20f400c7 de ALTERA. El diseño propuesto tiene una capacidad para dos puertos de entrada y uno de salida, empleando una resolución de 10 bits en cada uno (tanto de entrada como de salida). El artículo presenta los resultados y las conclusiones de la síntesis del diseño, así como una comparación con un controlador PID tradicional.

Palabras claves: Controlador, difuso, hardware, VHDL, parametrizable, FPGA

Abstract

The article describes the design of a hardware fuzzy controller. These devices are usually builded for an specific plant difficulting its range of applicability. The presented development is based on a parametrizable fuzzy processor, it allows being configured for use in the control of vary plants without hardware variation. In this case is possible to modify,

Fecha de recepción: 17 de noviembre de 2007
Fecha de aceptación: 13 de diciembre de 2007

* Grupo de Investigación en Robótica y Sistemas Inteligentes. Programa Jóvenes Investigadores Colciencias 2006. Convenio No. 090-2006. agenorpz@gmail.com

** Profesor Tiempo Completo, Universidad del Norte (Barranquilla, Colombia). Grupo de Investigación en Robótica y Sistemas Inteligentes. mesmeral@uinorte.edu.co

Correspondencia: Universidad del Norte Km 5 vía a Puerto Colombia, Barranquilla (Colombia).
Subvenciones y apoyo: Programa Jóvenes investigadores Colciencias, Convenio N° 090-2006.

using Mamdani type rules, the partition of the discourse's universe for each input variable from 1 to 5 fuzzy sets, with triangular or Γ -linear membership functions for the input variables and singleton type sets for the output, reducing the calculation complexity of the defuzzifier. The development was implemented using VHDL and synthesized inside an ALTERA Cyclone Ep1c20f400c7 FPGA device. The proposed design has a capacity for 2 input and 1 output port, using a 10 bits resolution at each one (input and output). The paper shows the results and conclusions about design's synthesis, as well as a comparison with a traditional PID controller.

Key words: Controller, fuzzy, hardware, VHDL, parameterizable, FPGA.

1. INTRODUCCIÓN

La conveniencia de la utilización de lógica difusa para resolver distintos problemas de ingeniería [1][2] ha motivado el desarrollo de diversos sistemas difusos *hardware* mediante el uso de HDLs [3] y la explotación de las características de flexibilidad y reconfigurabilidad de los dispositivos de lógica programable tipo FPGA, los cuales resultan atractivos como plataforma de diseño de *hardware* digital de bajo costo.

Entre las aplicaciones más importantes de la lógica difusa se encuentra el Control Difuso, y los dispositivos de este tipo muestran su principal fortaleza cuando el proceso es no lineal y/o muy complejo para ser modelado con exactitud [1].

Los sistemas de control difuso están basados en reglas y conjuntos difusos, los cuales son utilizados para emular el razonamiento humano. Debido a lo anterior, el éxito de la estrategia de control dependerá en gran medida del conocimiento que se tenga del proceso, es decir, de que tan bien se conozca el comportamiento de la(s) salida(s) de este último ante las distintas condiciones de entrada [1]. Dicho conocimiento puede provenir ya sea de un modelo dinámico del proceso o incluso, en algunos casos, de la experiencia.

La base de conocimiento de un controlador difuso normalmente está compuesta por una de dos tipos de reglas, a saber: Mamdani o Takagi-Sugeno. Las primeras se caracterizan porque tanto en el antecedente como en el consecuente de éstas existen conjuntos difusos; las del segundo tipo incluyen de igual manera conjuntos difusos en el antecedente pero el

consecuente es una expresión dinámica o algebraica dependiente de las variables del proceso.

El sistema diseñado trabaja con base en reglas tipo Mamdani y es un controlador difuso *hardware* de propósito general, ya que permite que sus parámetros sean configurados de acuerdo con un proceso particular a través de un *software* interfaz de usuario instalado en un PC. Las siguientes secciones del documento muestran la descripción detallada del sistema, pruebas y resultados obtenidos, para finalmente presentar las conclusiones.

2. METODOLOGÍA

Las directrices que fueron definidas como punto de partida para el diseño se resumen en Tabla 1:

Tabla 1
Directrices de diseño

PARÁMETRO DE DISEÑO	VALOR
Capacidad E/S	2 Entradas / 1 Salida
Resolución E/S	10 bits
Máximo número de conjuntos / variable difusa de entrada	5
Máximo número de reglas	25
Formato de regla	Mamdani
Método de inferencia	Operador Mín
Método de desdifusificación	Centroide
Tipos de membresía soportados en la entrada	Triangular, Γ -linear
Tipos de membresía soportados en la salida	Singleton

Visto como un sistema, el controlador difuso puede dividirse en cinco etapas: Difusificador, Base de Conocimiento, Motor de Inferencia, Desdifusificador y una lógica de control principal que regula la interacción entre las anteriores etapas y el usuario (ver Figura 1).

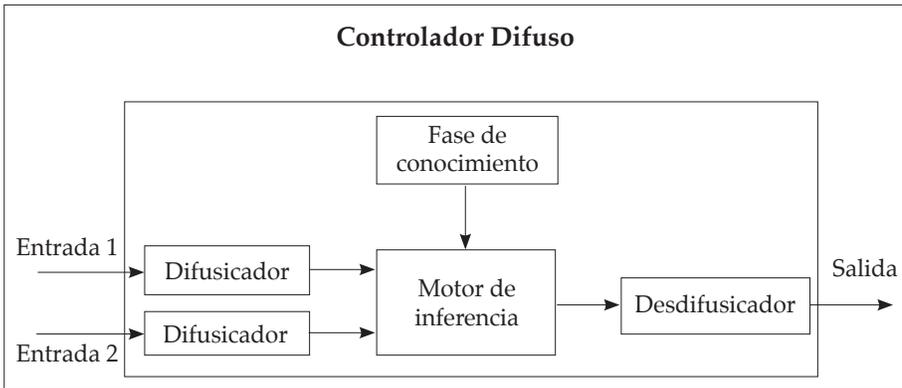


Figura 1. Diagrama del controlador difuso de dos entradas y una salida

2.1 Difusificador

El sistema utiliza un difusificador por cada puerto de entrada, cuya estructura se muestra en la Figura 2.

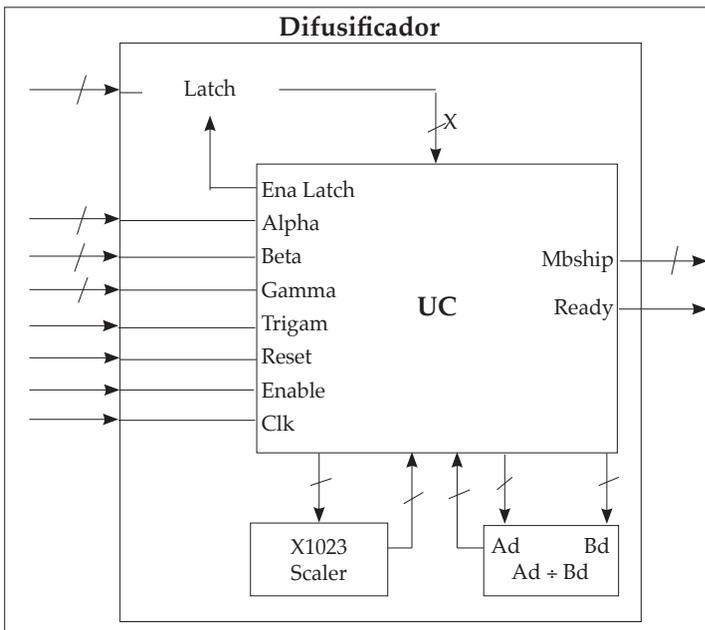


Figura 2. Estructura del difusificador

Cada bloque de este tipo permite seleccionar entre funciones de membresía triangular o Γ -linear mediante una señal denominada Trigam. Las entradas Alpha, Beta y Gamma corresponden a los parámetros de la expresión que se debe evaluar. El *latch* en la entrada mantiene fijo el valor leído del puerto de entrada del controlador mientras se calcula el valor de membresía.

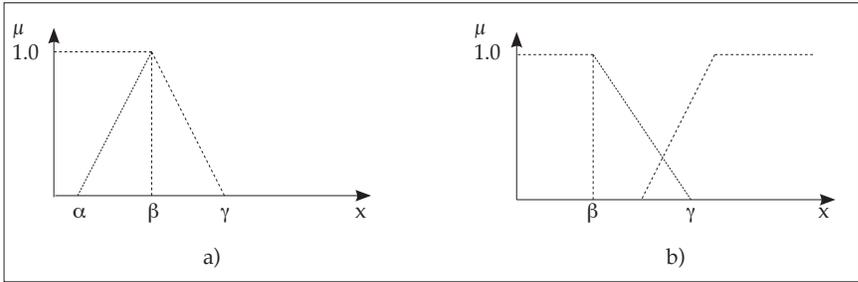


Figura 3. Funciones de membresía para variables difusas de entrada:
 a) Triangular, b) Γ -linear

El cálculo en los segmentos de recta de las funciones se realiza con expresiones de la forma descrita en (1). El pre-escalamiento por 1023 en el numerador traslada el intervalo normalizado (0, 1) al intervalo (0, 1023) ya que los cálculos se realizarán utilizando aritmética de punto fijo. Esto garantiza que los valores de membresía obtenidos empleen todo el rango de representación de enteros sin signo de 10 bits. Cabe destacar que dicho escalamiento es realizado de manera que no es necesario incluir un multiplicador (2).

$$\mu = \text{Parte Entera} \left(\frac{1023 \times (x - \alpha)}{\gamma - \alpha} \right) \quad (1)$$

$$a \times 1023 = a \times 2^{10} - a \quad (2)$$

El cociente se calcula utilizando lógica combinatoria por medio de un arreglo celular divisor con reestablecimiento [4] (ver Figura 4) de tamaño 20 bits en el dividendo y 10 bits en el divisor, lo cual disminuye significativamente la latencia introducida por la operación si se compara con una implementación secuencial.

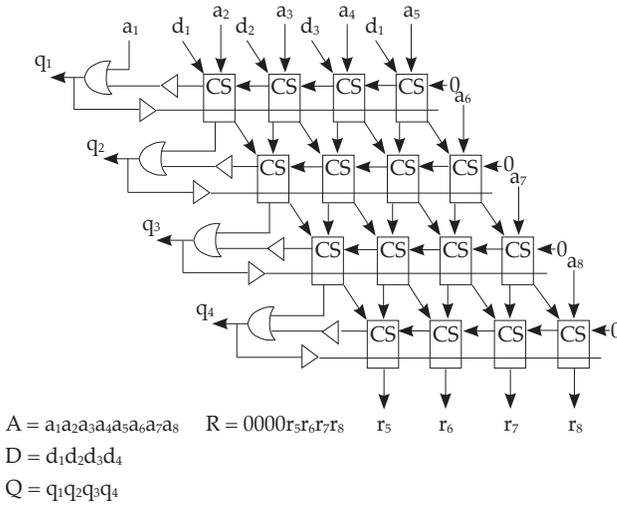


Figura 4. Arreglo celular divisor con reestablecimiento.

Ejemplo para 8 bits en dividendo, 4 bits en el divisor. Figura tomada de [4].

El diagrama de flujo en la Figura 5 resume de manera general la operación de esta etapa del sistema. El proceso completo toma 5 ciclos de reloj.

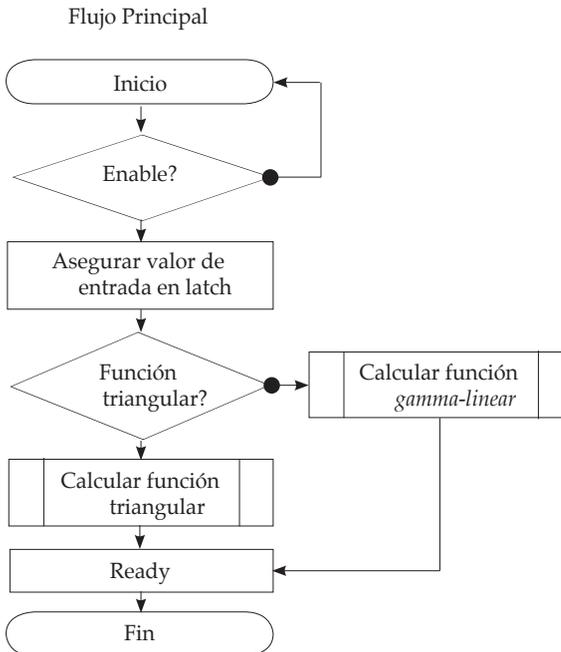


Figura 5. Diagrama de operación del difusificador

2.2 Base de conocimiento y motor de inferencia

La base de conocimiento tiene una capacidad para máximo 25 reglas tipo Mamdani de la forma:

$$\text{If } X_1 \text{ is } S_{1k} \text{ And } X_2 \text{ is } S_{2m} \text{ then } Z \text{ is } S_{oi} \quad (3)$$

Donde X_1 y X_2 son las entradas al controlador, S_{1k} es el k -ésimo conjunto asociado a la entrada X_1 , S_{2m} es el m -ésimo conjunto difuso asociado a la entrada X_2 y S_{oi} es el i -ésimo *Singleton* asociado a la salida Z .

La inferencia en cada regla se efectúa mediante el operador mín (mínimo); por esta razón, el motor de inferencia está constituido por un arreglo de comparadores, que evalúan la operación entre el par de valores de membresía involucrados en el antecedente de cada regla, generando un valor de membresía resultante por la regla evaluada.

Un registro de control almacena un código de 4 *bits* asociado a la partición del universo de discurso hecha por el usuario a cada variable de entrada (ver Tabla II); un segundo registro de 10 *bits* contiene la información del tipo de función de membresía que corresponde a cada conjunto, de manera que si un *bit* dado es '0', la correspondiente función de membresía es triangular, de lo contrario, será Γ -linear; esto le permite al sistema conocer cuántas funciones de membresía y de qué tipo debe evaluar por variable de entrada y cuáles membresías resultantes debe tener en cuenta para el cálculo del desfudisificador.

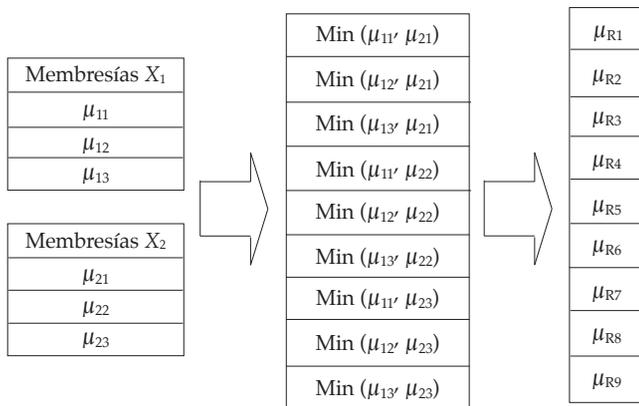


Figura 6. Arreglo de comparadores empleado en el motor de inferencia. En la gráfica de ejemplo existen 3 conjuntos por variable de entrada.

Tabla 2
Posibles asignaciones de Conjuntos para variables de Entrada

CÓDIGO (Hex)	# CONJUNTOS $X_1:X_2$
1	1:1
2	2:1
3	3:1
4	4:1
5	5:1
6	2:2
7	3:2
8	4:2
9	5:2
A	3:3
B	4:3
C	5:3
D	4:4
E	5:4
F	5:5

2.3 Desdifusificador

Luego de obtener las membresías resultantes del proceso de inferencia, los pasos siguientes serían la implicación y la agregación, para luego continuar con la desdifusificación. Sin embargo, llevando a cabo el proceso de implicación en cada regla, aplicando ya sea el operador producto o mínimo entre el valor membresía resultante y el conjunto *singleton* del consecuente de la regla, lo que se obtiene es un *singleton* escalado por el valor de membresía resultante. Siguiendo con el procedimiento, al efectuar la agregación de los conjuntos resultantes de cada regla, mediante una operación OR (máximo) entre los conjuntos obtenidos de la implicación, el conjunto final que se obtiene de todo el proceso es una superposición de los *singletons* involucrados en cada regla, cada uno escalado por el correspondiente valor de membresía inferido. Así, luego de realizar el análisis anterior y al observar la fórmula para calcular el centroide:

$$z * \frac{\sum z * \mu(z)}{\sum \mu(z)} \tag{3}$$

Donde $\mu(z)$ es la función de membresía del conjunto obtenido de la agregación y z asume los valores posibles del universo de discurso de la variable de salida. Puede entonces concluirse que, obtenidos los valores de membresía resultantes de la inferencia, se puede proseguir directamente a la desdifusificación aplicando la siguiente fórmula:

$$z = \frac{\sum_k \mu_k \beta_k}{\sum_k \mu_k} \quad (4)$$

Donde μ_k es la membresía inferida en la k -ésima regla y β_k es la posición (abscisa) del *singleton* del consecuente dentro del universo de discurso de la variable de salida.

Para efectuar este cálculo en *hardware* se recurrió a una lógica digital como la descrita en Figura 7.

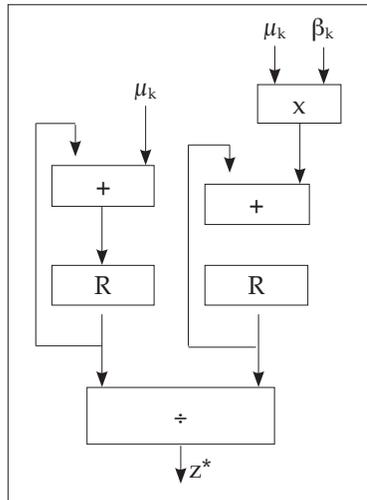


Figura 7. Esquema de lógica digital utilizada para el cálculo del desdifusificador

En esta etapa sí fue necesario implementar un multiplicador de 10 *bits* (con producto de 20 *bits*). Éste fue realizado utilizando lógica combinatoria por medio de un arreglo de 9 sumadores de 10 *bits* en paralelo desplazados 1 *bit* entre sí. Dicho arreglo asemeja a la manera como se realiza la suma en una multiplicación calculada a mano.

La división requerida por la fórmula se implementó mediante un arreglo divisor similar al de la etapa de difusificación, solo que esta vez fue necesaria una capacidad de 24 *bits* en el dividendo y 14 *bits* en el divisor.

3. PRUEBAS Y RESULTADOS

Esta sección se divide en dos partes: la primera presenta los resultados de sintetizar el controlador en el dispositivo *Cyclone Ep1c20f400c7* de Altera; la segunda tiene por objetivo ilustrar la capacidad de control del sistema bajo dos procesos de prueba, uno lineal y otro no lineal. En cada caso el rendimiento obtenido fue comparado con el de un control PID tradicional sintonizado por el algoritmo de Dahlin [5].

3.1 Resultados de síntesis

En la Tabla 3 se muestra el resumen de la síntesis del *hardware* en el dispositivo FPGA. La lógica adicional a la que se hace referencia corresponde a la UART encargada de la comunicación bidireccional serial con el PC, circuitería anti-rebote asociada a botones pulsadores necesarios para interactuar con el sistema, entre otros.

Tabla 3
Resumen de Síntesis en FPGA

DESCRIPCIÓN	VALOR
Frecuencia de operación del sistema	25 Mhz
Total LEs consumidos	5.383/20.000
Velocidad Tx/Rx UART	19.200 bps
LEs difusificador	850 (2x425)
LEs motor de inferencia	250
Les desdifusificador	626
LEs lógica adicional y de control del sistema	3657

3.2 Capacidad de control

Se realizaron pruebas en dos escenarios, un proceso lineal y otro no lineal. Las variables de entrada al controlador difuso fueron el error y el cambio del error, y se especificaron 5 conjuntos difusos para cada variable con una base de conocimiento de 25 reglas.

Para llevar a cabo estas pruebas sobre el controlador diseñado, se realizó un montaje como se describe en la Figura 8. La herramienta software LabVIEW v8.0 sirvió para implementar los procesos de prueba y establecer la comunicación con el controlador a través de una interfaz serial RS-232 Full Duplex. El montaje fue realizado de manera que era posible observar el comportamiento en línea del diseño y de un controlador PID sintonizado para cada proceso; esto último con el propósito de realizar comparaciones.

El controlador con el cual se compara el diseño se implementó mediante la librería de control PID disponible en LabVIEW. Dicha comparación fue realizada para múltiples cambios del *Set Point* (SP) (ver Figura 9), teniendo en cuenta los indicadores Error Absoluto Acumulado (IAE) y Variación Absoluta Acumulada en la acción de control (IAdm) (Ver Tabla 4).

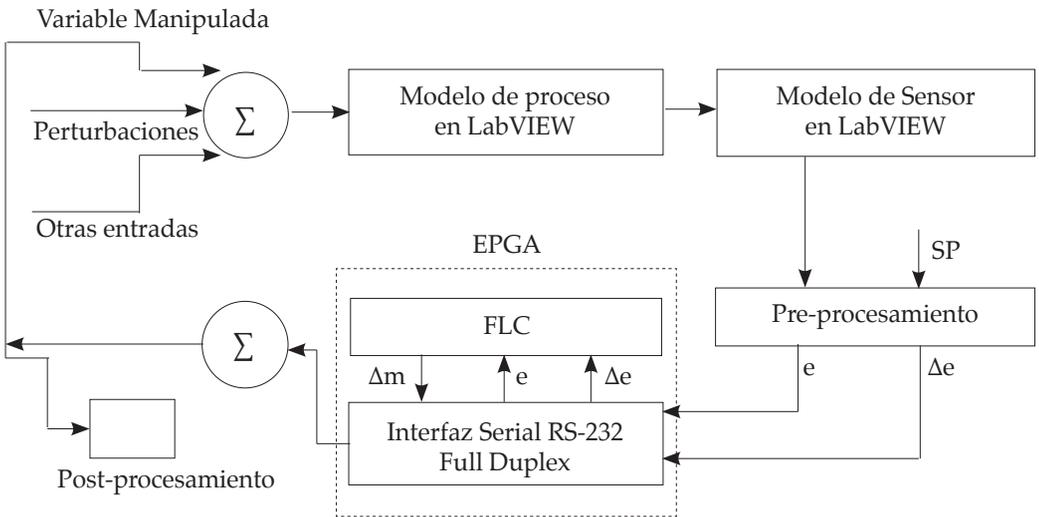
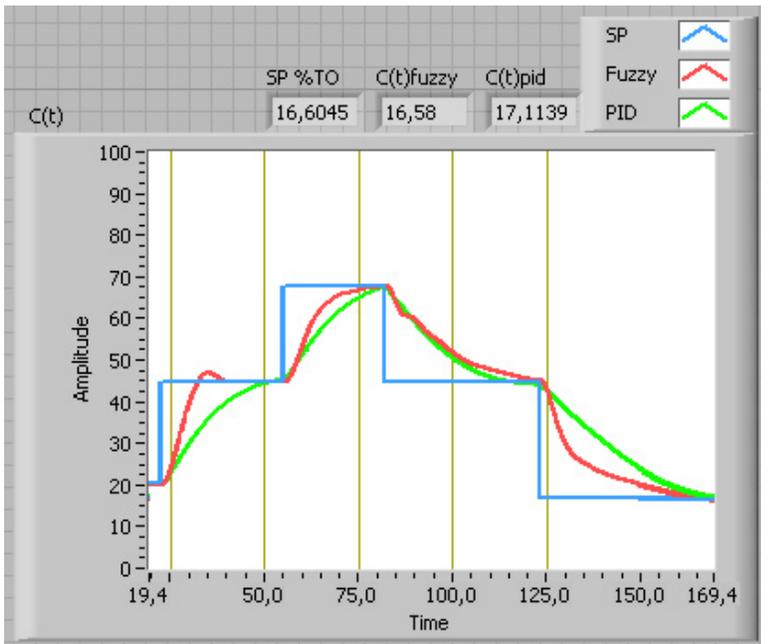


Figura 8. Diagrama de montaje realizado para llevar a cabo las pruebas sobre el controlador difuso

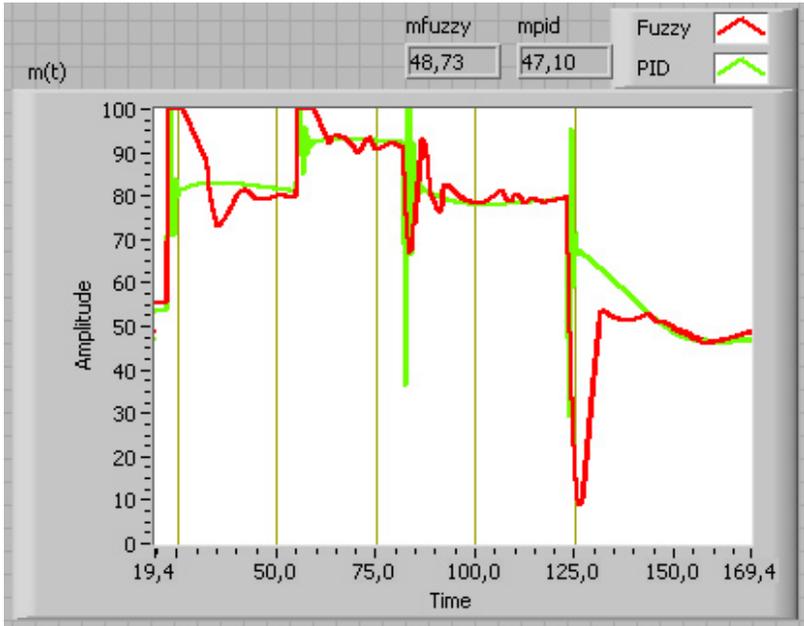
Tabla 4
Resultado Pruebas de Capacidad de Control

Proceso	ΔSP (%TO)	IAE Fuzzy (%TO)	IAE PID (%TO)	IAdm Fuzzy (%CO)	IAdm PID (%CO)
Lineal	+31,19	58,42	64,99	78,50	63,31
Lineal	+37,35	124,73	144,91	28,22	24,50
Lineal	-18,67	20,85	21,86	96,44	84,65
Lineal	-65,36	130,34	177,87	118,67	114,13
No lineal	+24,35	143,73	276,5	85	99
No lineal	+23,32	181,62	249,30	44	65
No lineal	-23,32	331,53	324,60	100	210
No lineal	-27,99	331,48	533,80	125	202

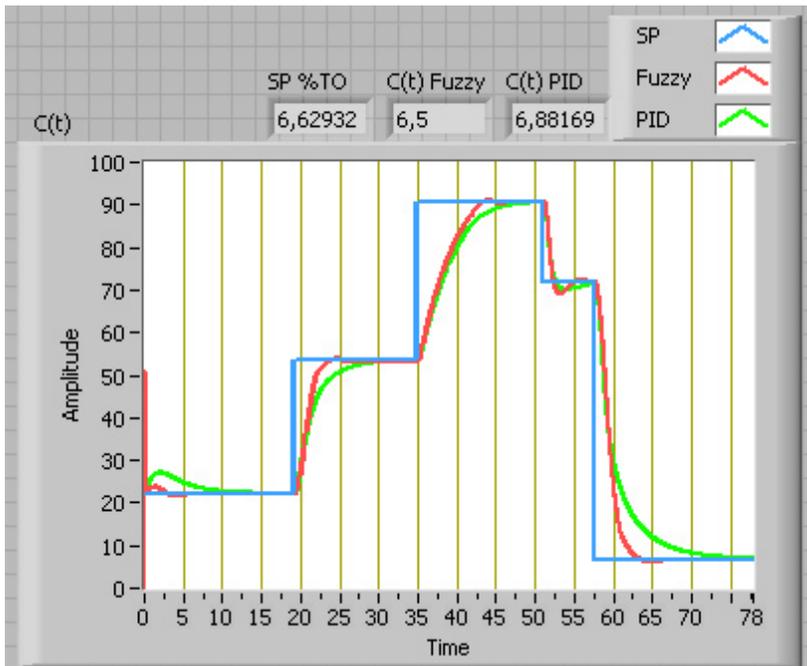
Los procesos de prueba son ampliamente conocidos en la literatura; estos son: el tanque de mezclado uniforme fluido frío/fluido caliente (lineal) y el tanque reactor no isotérmico continuamente agitado (CSTR) (no lineal). Para mayor detalle de estos procesos puede referirse entre otros a [5].



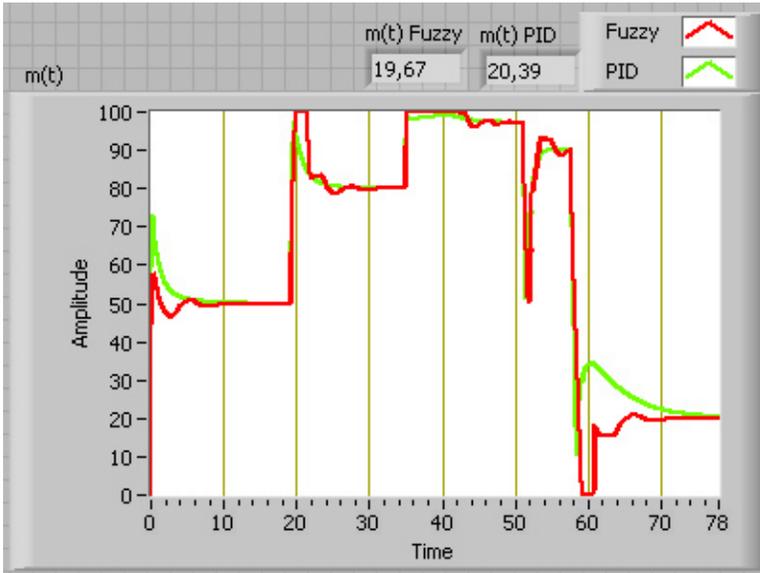
a)



b)



c)



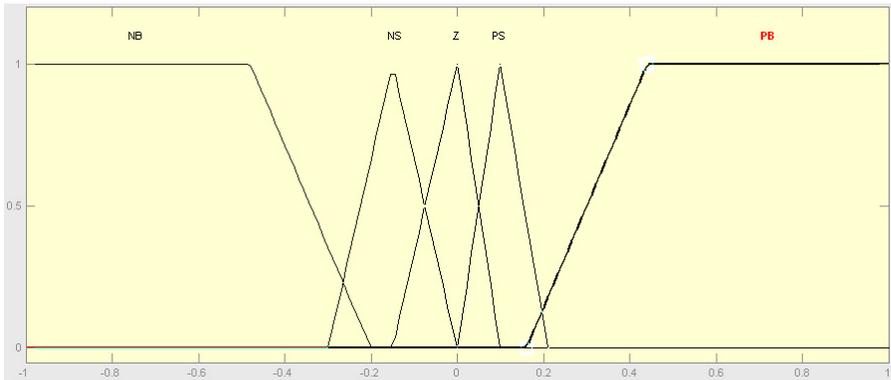
d)

Figura 9. Gráficas de variable controlada y acción de control para los procesos no lineal (a y b) y lineal (c y d).

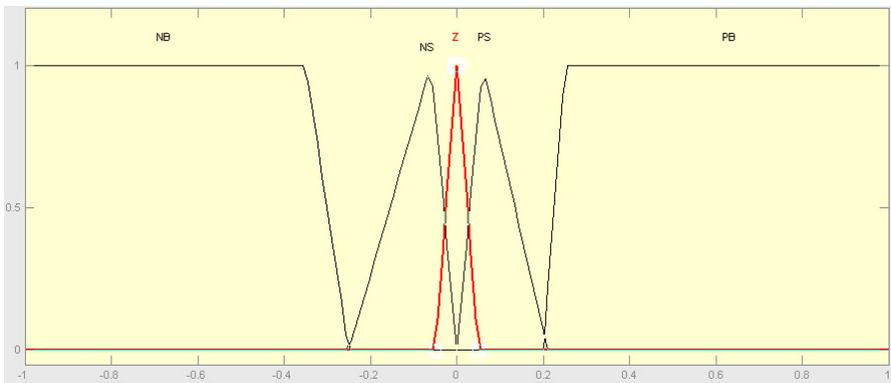
Una interfaz de usuario ejecutada desde un PC permite configurar a través del puerto serie RS-232 la base de conocimiento (Tabla 5) y las funciones de membresía (ver Figuras 10 y 11) deseadas para cada proceso.

Tabla 5
Resultado pruebas de capacidad de control

REGLAS PROCESO NO LINEAL					
de/e	nb	Ns	z	ps	pb
nb	-0,67	-0,39	-0,02	-0,02	0,67
ns	-0,61	-0,14	-0,02	0,02	0,75
z	-0,56	-0,12	0,00	0,14	0,83
ps	-0,50	-0,02	0,02	0,17	0,92
pb	-0,44	0,10	0,02	0,58	1,00
REGLAS PROCESO LINEAL					
de/e	nb	Ns	z	ps	pb
nb	-1,00	-0,58	-0,04	-0,35	0,50
ns	-0,92	-0,25	-0,04	0,04	0,75
z	-0,83	-0,21	0,00	0,21	0,83
ps	-0,75	-0,04	0,04	0,25	0,92
pb	-0,50	1,00	0,04	0,58	1,00

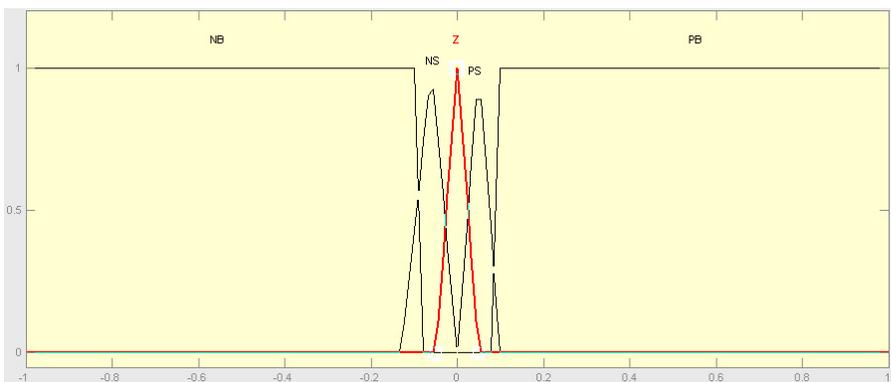


a)



b)

Figura 10. Funciones de membresía definidas para las variables error (a) y cambio del error (b); caso proceso no lineal.



a)

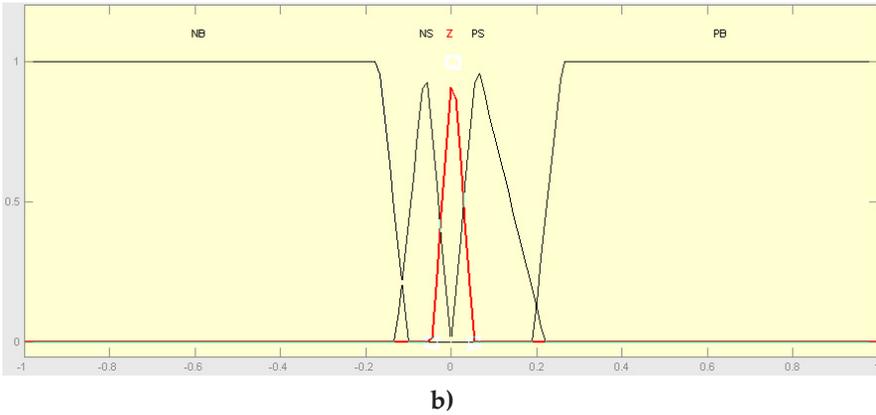


Figura 11. Funciones de membresía definidas para las variables error (a) y cambio del error (b); caso proceso lineal.

4. CONCLUSIONES

El presente trabajo muestra que el controlador diseñado puede configurarse para aplicarse exitosamente en más de un proceso, sin que esto implique modificar su *hardware*, lo cual es consecuencia de la parametrizabilidad de la base de conocimiento y de las funciones de membresía.

El uso de *singletons* en el consecuente de las reglas simplifica el proceso de desdifusificación, ya que permite calcular la salida directamente como un promedio ponderado de éstos, con lo cual se obvian las operaciones de implicación y agregación.

En cuanto a la capacidad de control, sobre escenario no lineal, el controlador difuso implementado produjo en la mayoría de los casos menor cantidad de error acumulado y una variación mucho menos brusca en la acción de control (menor IAdm). Por otro lado, en el proceso lineal, los resultados mejoraron significativamente para el PID por tratarse de su escenario natural; sin embargo, el controlador difuso mantuvo en la mayoría de los caso una ligera superioridad en el IAE al costo de un mayor IAdm.

La cantidad de recursos consumidos por el sistema en el dispositivo FPGA particular sugiere que puede sintetizarse en otro de mucha menor densidad.

REFERENCIAS

- [1] MEZA, Araceli, Observadores difusos y control adaptable difuso basado en observadores. Tesis de maestría, Instituto Politecnico Nacional, México D.F. Octubre de 2003.
- [2] Fuzzy Control and Modeling Analytical Foundation and Applications, IEEE Press Series on Biomedical Engineering, John Wiley and Sons, 2000
- [3] SAMEEP SINGH and KULDIP S. Rattan, Implementation of a Fuzzy Logic Controller Using VHDL, Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American
- [4] MI LU, "Arithmetic and Logic in Computer Systems", Ed. Hoboken, New Jersey: John Wiley & Sons, 2004.
- [5] SMITH, C. y CORRIPIO, A., Principles and practice of automatic control, New York: John Wiley and Sons, 1997.

