

Diseño e implementación de un sistema de control difuso reconfigurable

Edgar Eliseo franco Ortiz
Departamento de estudios
multidisciplinarios - DICIS
Universidad de Guanajuato
Yuriria, Gto, México
edgar.franco@ugto.mx

Carlos Rodríguez Doñate
Departamento de estudios
multidisciplinarios - DICIS
Universidad de Guanajuato
Yuriria, Gto, México
c.rodriquezdonate@ugto.mx

Ruth Ivonne Mata Chávez
Departamento de estudios
multidisciplinarios - DICIS
Universidad de Guanajuato
Yuriria, Gto, México
ruth@ugto.mx

Abstract—Los controladores difusos han tenido gran uso en distintas disciplinas debido a la facilidad con la que se pueden diseñar sin contar con el modelo matemático del sistema, en este artículo se propone el diseño e implementación de un controlador difuso embebido en un FPGA, reconfigurable mediante una interfaz gráfica de usuario desarrollada en Matlab. Además, esta propuesta puede ser utilizada para cualquier aplicación, ya que solo se requiere que el experto del sistema a controlar proponga los parámetros de diseño en base a la señal de error y derivada del error de la variable a controlar en la interfaz de usuario, que mandará reconfigurar el sistema, lo que implica no será necesario sintetizar el nuevo diseño en el FPGA. De acuerdo a los resultados obtenidos de la síntesis del sistema embebido sintetizado en un FPGA de bajo costo consume el 80% de unidades lógicas, el 1% de los bits de memoria y 52% de multiplicadores.

Palabras clave—control automático, sistema embebido, lógica difusa, Matlab, FPGA.

I. INTRODUCCIÓN

El control automático ha desempeñado un papel vital en el avance de la ingeniería y la ciencia. Se ha convertido en una parte importante e integral para los procesos modernos industriales y de manufactura y, en general, en cualquier operación que requiera el control de alguna variable como presión, temperatura, humedad, flujo, entre otras. Las teorías de control que son utilizadas habitualmente para estas aplicaciones se refieren a las leyes de control clásico, en las cuales se requiere conocer con certeza el modelo matemático que defina el comportamiento dinámico del sistema a controlar. Sin embargo, hoy en día se requieren sistemas de control muy complejos y en algunas ocasiones no es posible o resulta muy complicado obtener su modelo matemático; debido a esto, se han propuesto técnicas de inferencia difusa, ya que permiten general el controlador de un sistema sin la necesidad de obtener su modelo matemático. Los controladores difusos han tenido un gran impacto en el contexto académico e industrial debido a que con el conocimiento experto de una o varias personas, se genera una base de conocimientos que dará al sistema la capacidad de tomar decisiones sobre ciertas acciones que se presenten en su funcionamiento.

La capacidad de la lógica difusa para describir el comportamiento de sistemas complejos mediante reglas

lingüísticas simples ha motivado su aplicación a numerosos sistemas de control. Un sistema de control difuso se basa en reglas de la forma SI – ENTONCES para establecer relaciones entre sus entradas y salidas, debido a esto, no es necesario sintonizar parámetros de un sistema de control no lineal, así como tampoco se requiere del modelo de la planta del sistema y por ello ha tenido gran uso en distintos tipos de aplicación. Uno de los campos crecientes de controladores difusos es el de la robótica, en particular los problemas asociados con la navegación autónoma de un robot móvil constituyen un atractivo espacio de desarrollo de controladores basados en lógica difusa [1], actualmente se han realizado una gran cantidad de investigaciones para el control de motores mediante lógica difusa [2], también se han llevado a cabo distintas investigaciones de control difuso basado en arreglos de compuertas programable en campo (FPGA, *Field Programmable Gate Array*), como la implementación de habilidades de conducción en un robot móvil [3] o el control del desplazamiento de una plataforma omnidireccional [4], los controladores difusos se han aplicado incluso para sistemas de generación eólicos [5] y para analizar el rendimiento de controladores industriales de nivel para tanques con fluidos [6].

Debido a la gran variedad de aplicaciones que se pueden llevar a cabo mediante técnicas de lógica difusa, en este trabajo se propone un controlador difuso embebido en un FPGA que puede ser reconfigurado. El sistema propuesto se reconfigura mediante una interfaz gráfica realizada en el software Matlab, sin la necesidad de volver a programar la FPGA, lo cual permite que se generen controladores para distintas aplicaciones, sin tener que realizar cambios en el sistema embebido propuesto. De esta forma, la persona experta del sistema a controlar solo requerirá diseñar en la interfaz de usuario el controlador difuso en base a la señal de error y la derivada del error de la variable a controlar. De acuerdo a los resultados obtenidos de la síntesis del sistema embebido sintetizado en un FPGA de bajo costo consume el 80% de unidades lógicas, el 1% de los bits de memoria y 52% de multiplicadores.

II. MARCO TEÓRICO

A. Lógica difusa

La lógica difusa fue creada para emular la lógica humana y tomar decisiones acertadas a pesar de la información. La lógica difusa es un conjunto de principios matemáticos basados en grados de membresía o pertenencia, cuya función es modelar información. Este modelado se hace con base en reglas lingüísticas que aproximan una función mediante la relación de entradas y salidas del sistema. Esta lógica presenta rangos de membresía dentro de un intervalo entre 0 y 1, a diferencia de la lógica convencional, en la que el rango se limita a dos valores: el cero o el uno.

Una variable lingüística adopta valores con palabras que permiten describir el estado de un objeto o fenómeno; estas palabras se pueden representar mediante conjuntos difusos. Una variable numérica toma valores numéricos, por ejemplo: edad = 65, mientras que una variable lingüística toma valores lingüísticos: edad es “viejo”. Todos los valores lingüísticos forman un conjunto de términos o etiquetas. Para desarrollar un control con estas características, es necesario un experto, en este caso el usuario del sistema a controlar, del cual se tomará un registro de las situaciones que se le presentan, así como de la solución que les da. Esta experiencia se traduce en reglas que usan variables lingüísticas. Para hacer este control es necesario tener las entradas del sistema y éstas se van a mapear a variables lingüísticas. A este mapeo se le llama fusificación. Con estas variables se forman reglas, las cuales serán las que regirán la acción de control que será la salida del sistema.

B. Controlador difuso

Como se muestra en Fig. 1 un controlador difuso está compuesto de cuatro partes principales: fusificación, base de conocimientos, inferencia y defusificación.



Fig. 1. Estructura de un controlador difuso.

Fusificación: Mide los valores de las variables de entrada para realizar un mapeo a escala que transfiere el rango de valores a un universo de discurso difuso. La fusificación convierte los datos de entrada en valores lingüísticos que son las etiquetas de las funciones de pertenencia o conjuntos difusos. Una variable lingüística se asocia a un conjunto de términos, definido en el mismo universo de discurso. Para encontrar cuántos términos son necesarios en un conjunto se emplean particiones difusas. El número de conjuntos difusos determina la complejidad del controlador, y éstos tienen un significado lingüístico como “negativo grande”, “cero”, “positivo pequeño”. La Fig. 2 muestra ejemplos de dos particiones difusas en el mismo universo, normalizado de -1 a +1.

Base de conocimientos: La base de conocimientos contiene toda la información de la aplicación que se va a controlar, así como las metas del controlador. Consta de una base de datos y una base de reglas lingüísticas para controlar la variable. La base de datos proporciona las definiciones para el establecimiento de reglas y la manipulación de datos difusos. La base de reglas caracteriza las metas de control y la política que utilizan los expertos para llevar a cabo el control.

Inferencia: La lógica utilizada para tomar decisiones dentro de un controlador difuso es el núcleo del mismo. A partir de la misma se simula la lógica que utilizan las personas para tomar decisiones, con base en conceptos difusos y en la inferencia de acciones de control, empleando implicaciones y las reglas establecidas según la base de conocimiento.

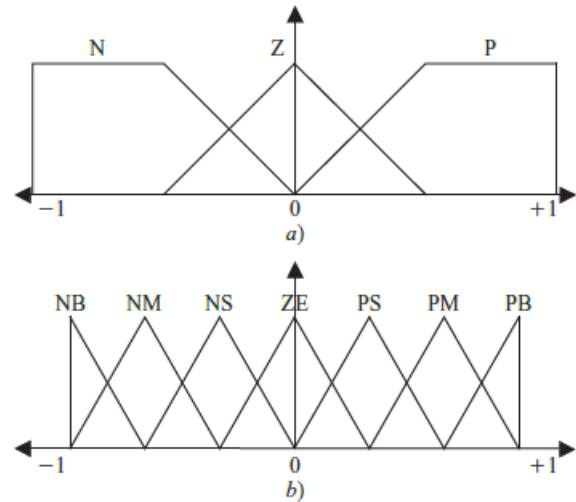


Fig. 2. Particiones difusas con distinto número de términos: a) tres términos; b) siete términos.

Defusificación: La defusificación se encarga del mapeo a escala que convierte el rango de valores de las variables de salida a sus universos de discurso correspondientes. Es la herramienta para obtener la acción de control nítida a partir de una acción de control difusa. Existen diversos métodos para llevar a cabo la defusificación, pero el utilizado en este trabajo fue el método de centro de área. El método de centro de área o centro de gravedad se utiliza para obtener el valor real de la salida. Su metodología es sencilla: corta la función de membresía al grado de la membresía respectiva, es decir, segmenta las funciones de membresía, generando en cada función dos áreas. El área inferior que se forma es la que se toma para hacer el cálculo. Se superponen todas estas áreas y se saca el centroide de la superposición, el cual nos dice la salida real del sistema. El método del centro de área o centro de gravedad se puede representar en forma discreta mediante la ecuación (1).

$$salida = \frac{\sum_{x=a}^b \mu(x) \cdot x}{\sum_{x=a}^b \mu(x)} \quad (1)$$

Donde $\mu(x)$ corresponde a la membresía del conjunto difuso. [7]

III. METODOLOGÍA

La Fig. 3 muestra el diagrama del sistema propuesto, donde la etapa de software realizada en el entorno Matlab será con la cual el usuario interactuará para poder reconfigurar el controlador difuso embebido en el FPGA, ya que en este dispositivo se encuentran embebidos los controladores para comunicarse con la etapa de software, los controladores para el sensor y el actuador, también se encuentra el bloque de reconfiguración del controlador difuso.

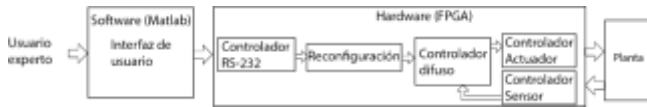


Fig. 3. Esquema del sistema embebido propuesto.

La interfaz de usuario fue diseñada en el entorno Matlab donde el usuario experto del sistema a controlar propondrá todos los parámetros que requiere el controlador difuso que trabaja en base a las señales de error y derivada del error de la variable a controlar; por lo tanto, el usuario propondrá las funciones de membresías de entrada, salida y conjunto de reglas que requiere el controlador difuso. Por otro lado, en el FPGA se encuentra embebido el controlador difuso reconfigurable el cual se diseñó para que el usuario modifique su operación a través de la interfaz de usuario desarrollada en Matlab. La comunicación entre la etapa de software y hardware se realiza mediante el protocolo de comunicación serial RS-232.

A. Matlab

La interfaz gráfica se diseñó en base a los datos que requiere un controlador difuso, la cual se muestra en la Fig. 4. Esta interfaz requiere de los siguientes parámetros: Los conjuntos de entrada correspondientes al error y la derivada del error, los conjuntos de salida, las reglas lingüísticas, las cuales representan el comportamiento que debe tener el controlador, además, se debe ingresar la referencia a la que se desea mantener el sistema y el período de muestreo con el que trabajará el controlador.



Fig. 4. Aspecto de la interfaz gráfica en Matlab.

Para poder ingresar el conjunto deseado se selecciona la entrada a la que pertenece el conjunto y se establece un valor mínimo y máximo, los cuales corresponden al rango que comprenderá el universo difuso. Además, se debe ingresar el número de conjuntos de cada entrada. El sistema embebido

trabaja con dos entradas, las cuales son el error y la derivada del error.

En la interfaz se pueden seleccionar funciones triangulares y trapezoidales, señalando los puntos que deben ingresarse para generar cada función como se muestra en Fig. 5. A cada conjunto ingresado se le puede asignar el nombre deseado para poder identificarlo al momento de ingresar la matriz con las reglas de inferencia.

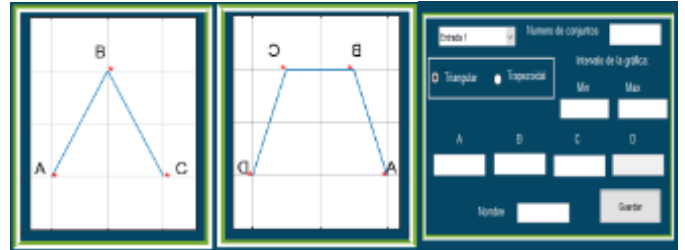


Fig. 5. Puntos para los valores de las funciones triangular y trapezoidal.

Una vez que se hayan ingresado los datos para un conjunto se selecciona el botón "Guardar" y a continuación aparecerá la gráfica de dicho conjunto, como se muestra en la Fig. 6.

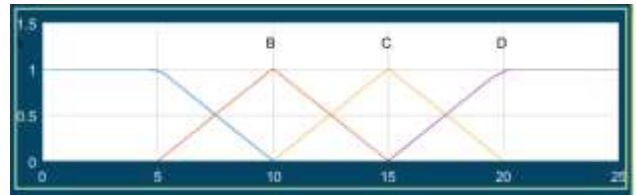


Fig. 6. Gráfica de los conjuntos para una entrada.

Al terminar de ingresar los conjuntos para ambas entradas y los conjuntos de salida se podrán ingresar las reglas llenando la matriz con los nombres de los conjuntos de salida. En Fig. 7 se muestra la matriz de las reglas, en la cual aparecerán los nombres de los conjuntos de ambas entradas, las filas corresponden a los conjuntos de la primera entrada y las columnas a los conjuntos de la segunda entrada. El tamaño de la matriz se modificará automáticamente de acuerdo con el número de conjuntos de las entradas.

	D	E	F
A			
B			
C			

Fig. 7. Matriz para las reglas de inferencia.

Una vez llenada la matriz de las reglas se puede presionar el botón "Enviar" y se establecerá la comunicación con la tarjeta FPGA para enviar todos los datos mediante el protocolo RS-232.

La interfaz permite actualizar el valor de referencia del controlador y las reglas de inferencia sin la necesidad de volver a ingresar todos los conjuntos de entradas y salida, por lo cual se pueden realizar distintas pruebas cambiando solo la matriz de las reglas de inferencia.

B. FPGA

En Fig. 8 se muestra el diagrama general de la arquitectura propuesta para el FPGA. En ella se muestran las etapas de la lógica difusa con las señales principales de cada etapa. El bloque de "Reconfiguración" es el que contiene toda la información ingresada a Matlab, esta información es almacenada en el FPGA utilizando memorias RAM. En el bloque "Er/Der" es donde se calculan los valores del error y la derivada del error, para calcular el error es simplemente la diferencia de la referencia y el valor medido por el sensor. La derivada del error se calcula con la diferencia del error actual y el error anterior.

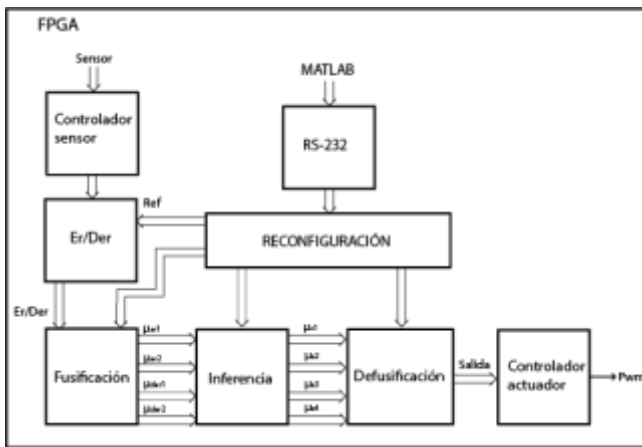


Fig. 8. Diagrama general del diseño.

Como las funciones que se utilizan en la interfaz gráfica son solo triangulares y trapezoidales, en la fusificación se puede utilizar la ecuación de la recta (2) para obtener los valores de pertenencia de los conjuntos difusos.

$$y = mx + b \quad (2)$$

En donde:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (3)$$

Despejando y_2 de (3):

$$y_2 = m(x_2 - x_1) + y_1 \quad (4)$$

Sustituyendo los datos de (4) por los mostrados en Fig. 9 se obtiene el valor correspondiente al grado de membresía del conjunto (5). De esta forma, mediante la ecuación de la recta, es posible obtener todos los valores de membresía correspondientes a los conjuntos del error y la derivada del error.

$$\mu_2 = m(Er - C_2) \quad (5)$$

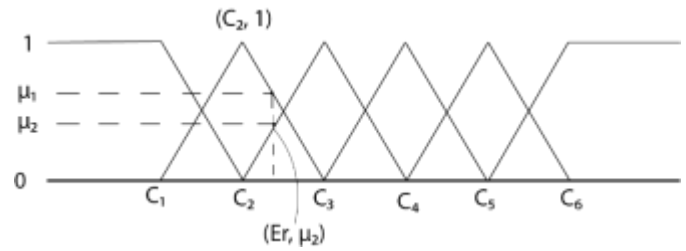


Fig. 9. Cálculo de los valores de membresía mediante la ecuación de la recta.

Una vez realizada la fusificación, se utilizan los valores obtenidos para llevar a cabo la inferencia. En la inferencia se accede a los datos recibidos de Matlab correspondientes a la matriz de las reglas lingüísticas para determinar cuáles son los conjuntos de salida que corresponden a los valores del error y la derivada del error. Una vez encontrados los conjuntos de salida, se les asigna un valor de membresía usando los valores obtenidos en la fusificación para poder calcularlos. En la defusificación se usó el método del centroide y el valor de salida se calculó con la ecuación (1), para ello se utilizaron los conjuntos de la salida y los valores de membresía obtenidos en la inferencia. Finalmente, la salida del controlador difuso se envía al actuador del sistema a controlar, para este caso esta señal es enviada a un modulador de ancho de pulso (PWM) el cual es la señal que requieren diversos actuadores como relevadores y servomotores.

IV. RESULTADOS

A continuación, se muestra la interfaz gráfica en Fig. 10 al haber ingresado todos los datos de un sistema a controlar.

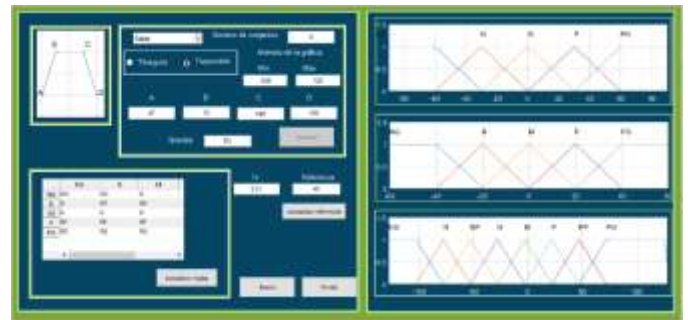


Fig. 10. Datos ingresados en la interfaz.

La interfaz permite observar las gráficas de los conjuntos ingresados, lo cual facilita el llenado de las reglas lingüísticas.

	NG	N	M
NG	NG	NG	N
N	N	NP	NP
M	N	N	M
p	PP	PP	PP
PG	PG	PG	PG

Fig. 11. Matriz de reglas lingüísticas.

Como se observa en Fig. 11, la matriz de las reglas de inferencia se llena con los nombres asignados a los conjuntos, de esta forma para modificar alguna regla y cambiar el comportamiento del controlador solamente hay que cambiar la relación de los conjuntos de salida respecto a los de entrada.

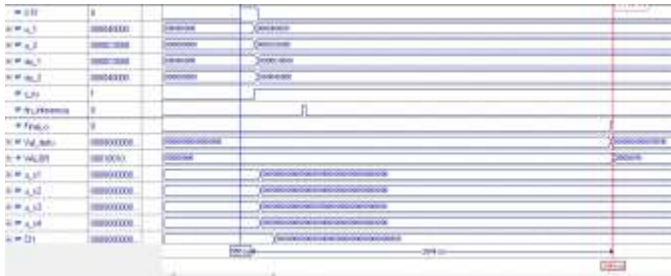


Fig. 12. Simulación de las señales de la arquitectura.

En Fig. 12 se muestra el comportamiento de algunas de las señales principales del diseño para la FPGA. La simulación fue realizada con una señal de reloj de 50 MHz y como se puede observar, el tiempo que se tarda el diseño en obtener el valor final una vez que ha comenzado la etapa de fusificación es de 2644 ns.

El sistema embebido se sintetizó en el quid de desarrollo DE0-Nano, el cual cuenta con un FPGA de bajocosto Cyclone IV EP4CE22F17C6. El consumo de recursos del dispositivo se muestran en la tabla I, donde se puede observar que requiere el 80 % de las unidades lógicas disponibles.

TABLA I. Consumo de hardware del sistema embebido en el FPGA.

Cyclone IV EP4CE22F17C6		
	Total	Usado
Elementos lógicos	22,320	80%
Bits de memoria	608,256	1%
Multiplicadores	132	52%

V. CONCLUSIONES

Como se ha mencionado, en la actualidad los controladores difusos se han utilizado en gran variedad de aplicaciones por lo cual este trabajo propone el diseño e implementación de un controlador difuso reconfigurable y embebido en un FPGA. Por lo tanto, la propuesta permite diseñar el controlador difuso en la interfaz gráfica para cualquier aplicación, de esta forma el usuario que conoce el sistema a controlar tendrá que proponer los conjuntos de entrada, conjuntos de salida y las reglas de la máquina de inferencia, ya que el controlador difuso embebido en el FPGA se reconfigurará en base al diseño propuesto por el usuario. Adicionalmente, con esta propuesta el usuario podrá estar manipulando todos los parámetros hasta lograr obtener la sintonización deseada del sistema a controlar. El consumo de las unidades lógicas en el FPGA en el cual fue implementado el sistema fue del 80% de las disponibles en este FPGA de bajo costo.

REFERENCIAS

- [1] A. Cabrera, S. Sánchez-Solano, I. Baturone, A. Barriga, F. J. Moreno-Velo, p. Brox, "Control difuso de navegación de un robot móvil mediante dispositivos programables." XI congreso latinoamericano de control automático, La Habana, 2004.
- [2] J. J. Galvis López y J. I. Madrid V., "Sistema de control difuso para motor de corriente continua sin escobillas (BLDC) sobre hardware embebido", Tekhnê, vol. 13, n.º 2, pp. 43–48, dic. 2016.
- [3] Tzue-Hseng S. Li, Shih-Jie Chang, Yi-Xiang Chen, "Implementation of Human-Like Driving Skills by Autonomous Fuzzy Behavior Control on an FPGA-Based Car-Like Mobile Robot" IEEE transactions on power electronics, Vol. 50, No. 5, pp. 867-880, 2003.
- [4] M. Peña, J. A. Gómez, R. Osorio, I. López, V. Lomas, H. Gómez, G. Lefranc, "Fuzzy Logic for Omnidirectional Mobile Platform Control Displacement using FPGA and Bluetooth Communication Devices" IEEE. Latin America Transactions, vol. 13, No. 6, pp. 1907-1914, 2015.
- [5] D. Hoyos, R. Farfan, F. Tilca, C. Cadena, "Propuesta de controlador con lógica difusa para sistema eólico" Avances en Energías Renovables y Medio Ambiente, Vol. 10, pp. 56-61, 2006.
- [6] G. Ampuño, W. Agila, H. Cevallos, "Implementación y análisis de rendimiento de un control industrial de nivel para tanques con fluidos, basado en lógica difusa" Maskana, Vol. 5, pp. 27-36, 2016.
- [7] P. Ponce Cruz, Inteligencia artificial con aplicaciones a la ingeniería, 1ra ed. México: Alfaomega, 376 pp.