

Arquitectura Digital Difusa Embebida en un FPGA para Control de un Péndulo Invertido sobre un Carro

Andrés Flores⁺, Elsa Rubio, Víctor Ponce, Luis Luna^{*}

Centro de Investigación en Computación, Laboratorio de Robótica y Mecatrónica, Instituto Politécnico Nacional, México

⁺ ing.andresfc.2415@gmail.com

Resumen. Se presenta una arquitectura que permite embeber un sistema de control no lineal difuso sobre un FPGA de Xilinx XC3S500E. Este tiene el objetivo de controlar al péndulo invertido sobre un carro. El modelo difuso y los conjuntos de membresía difusos propuestos tienen como objetivo estabilizar al sistema en la posición vertical. Se describe en Verilog y VHDL el modelo. Este modelo es comparado con la respuesta generada por los sistemas difusos simulados en Matlab y la simulación del comportamiento generada en Isim de Xilinx, obteniendo resultados exitosos. La frecuencia máxima del controlador es 24.17 MHz la cual es tomada del resumen de ISE.

Palabras clave: Control difuso embebido, FPGA, lenguajes de descripción de hardware, péndulo invertido sobre carro.

1 Introducción

Un sistema embebido envuelve cómputo, pero este no es su principal objetivo. Por otro lado la lógica difusa tiene un amplio rango de aplicaciones y entonces el hilo común que une todas esas aplicaciones es que un experto (operador humano) puede, al menos en un principio, ejecutar la tarea requerida. Sin embargo, cuando el modelo matemático es demasiado complicado para tener un uso práctico, un modelo lingüístico llega a ser ventajoso [1]. El sistema del péndulo invertido es del tipo lazo abierto inestable pues se necesita un tipo de control para mantener al péndulo en la posición vertical [2]. Existen numerosas aplicaciones embebidas de lógica difusa para control de sistemas mecánicos [3, 4, 5]. La presente investigación se basa en la implementación de un sistema de Control Difuso para resolver el problema conocido del péndulo invertido sobre un carro. Los bloques fundamentales del sistema Difuso embebido se describen en los principales Lenguajes de Descripción de Hardware (HDL): Verilog y VHDL. Se utiliza la técnica de Modulación por Ancho de Pulso descrito dentro del

⁺ Alumno becario CONACyT y PIFI, estudiante de Maestría en Ciencias en Ingeniería de Cómputo del Centro de Investigación en Computación-IPN, México.

^{*} Alumno estudiante de Ingeniería en Control y Automatización de *ESIME-IPN*, México.

sistema embebido en HDL para otorgar la ley de control adecuada a la etapa de potencia electrónica, misma que hace trabajar a un motor de corriente directa que funciona como actuador principal para estabilizar la posición del péndulo invertido. Se utiliza un FPGA de la familia de Xilinx Spartan XC3S500E y la tarjeta de desarrollo Spartan 3E [16].

2 Estructura del sistema

El objetivo principal de este desarrollo tecnológico es embeber en HDL un sistema difuso que sea capaz de estabilizar al péndulo invertido que se encuentra montado sobre un carro, esa es la idea de forma general. En esta sección se explica cada una de las etapas que conforman al sistema, y la Figura 1 muestra el diagrama de bloques general empleado.

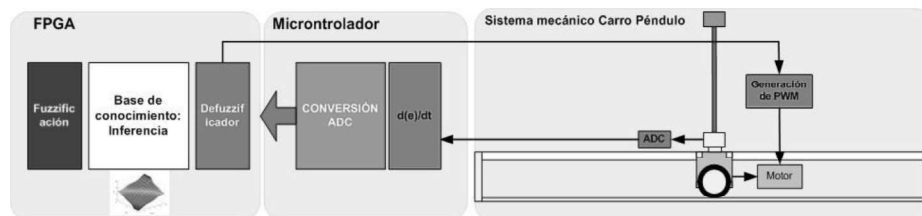


Fig. 1. Diagrama de bloques del sistema diseñado para el control del péndulo invertido.

La primera etapa permite la medición de nuestra variable de posición angular del péndulo. Se acopla mecánicamente al péndulo un potenciómetro y para asegurar que la medición fuese lo más exacta posible se decide utilizar un potenciómetro de precisión industrial de la empresa BI-Technologies. Este sensor tiene una sensibilidad adecuada debido a que es construido de plástico conductivo. La segunda etapa permite la conversión AD del dato proveniente del potenciómetro y es convertido a su resolución de 8 bits. Se utiliza el microcontrolador PIC16F887 [16] para realizar operación de conversión analógica digital. La etapa 3 llamada de Control es donde se encuentra embebido el Sistema Difuso en sus tres etapas: fuzzificación, inferencia y defuzzificación y una etapa extra para realizar la derivada de la señal de error.

Finalmente, la señal defuzzificada es interpretada y convertida a su correspondiente señal PWM con salida de 0 a 5 volts, la cual se direcciona a una frecuencia mayor a 20 KHz hacia una tarjeta de potencia MD03 de la empresa Robot Electronics [15] para controlar al péndulo invertido.

2.1 Definición de la conversión y de la derivada

En la literatura se encuentra la forma de hacer discreto el controlador PID. En la presente investigación se toma en cuenta el enfoque de [14] para describir la función que realiza la derivada del error en forma discreta. Siendo específicos, de la ecuación

(1) correspondiente al controlador PID discreto, se toma la parte derivativa y se describe según la Figura 2, donde k_d es la constante derivativa y $e(n)$ es el error medido.

$$u(n) = k_p e(n) + k_i \sum_{j=0}^n e(j) + k_d (e(n) - e(n-1)) \quad (1)$$

```

always @(clk)          //D=Kd*[ e(n) - e(n-1) ]
begin
  en=entrada;          // e(n)
  en_1=en;             // e(n-1)
  sal=Kd*(en-en_1);
end

```

Fig. 2. Código generado en Verilog que describe la derivada del error.

2.2 Descripción del módulo PWM en VHDL

El módulo PWM se encarga de tomar los 8 bits de salida correspondientes al cálculo del valor de la etapa de defuzzificación, por medio del uso de la técnica del promedio de los centros se genera el pulso PWM a una frecuencia de 20 KHz que es enviado al módulo MD03 que controla el motor de CD, mismo que posiciona al carro para estabilizar al péndulo. Para lograr conseguir el pulso PWM mediante lenguajes de descripción de hardware, se ha tomado la técnica reportada en [6], donde se utiliza un contador y un comparador para generar el pulso con el ancho deseado a la frecuencia deseada. Por lo general, todo sistema que procesa información binaria para controlar un proceso analógico requiere una etapa de entrada analógica digital y una etapa de salida digital analógica (convertidores ADC y DAC). Para reducir costos en los diseños que no requieren alta resolución en la etapa de salida, es posible sustituir el DAC por un algoritmo de Modulación por Ancho de Pulsos (PWM). Una unidad PWM permite asignar cierta duración de tiempo en alto o en bajo a un dato digital de n bits, misma que se considera la salida de la etapa de defuzzificación. Lo anterior se logra conectando un contador y un circuito comparador, tal y como se aprecia en la Figura 3. La Figura 4 muestra un extracto de código en VHDL para conseguir generar el PWM según la descripción dada anteriormente.

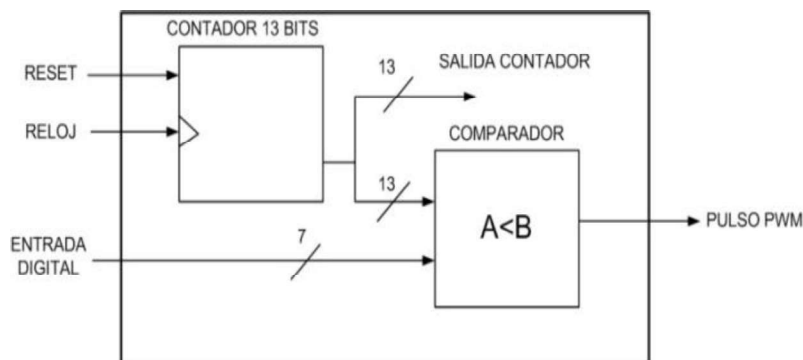


Fig. 3. Diagrama general del módulo que genera el pulso PWM en VHDL.

```

cuenta_inicial: process (reloj, reset, sal,
base, aux)
begin
if reset='0' then
    base <= "00000000000000";
elsif reloj='1' and reloj'event then
    if base /=1 then
        base <= base + 1;
        sal <= '0';
    else
        base <= "00000000000000";
        sal <= '1';
    end if;
end if;
frec_25KHz <= sal;
end process cuenta_inicial;

```

Fig. 4. Extracto de código que describe al módulo PWM.

2.3 Carro péndulo invertido

Recientemente, muchos sistemas robóticos diseñados con péndulo invertido sobre ruedas (WIP²) tales como el Segway PT y el Segway para Robonautas (Robot astronautas de la NASA), han tenido gran popularidad en la comunidad de la disciplina de la Robótica [7].

El péndulo invertido sobre un carro, es un sistema clásico de control bien conocido y estudiado durante años; debido a que sirve como sistema de prueba para el desarrollo de nuevos algoritmos de control, principalmente de control de sistemas no lineales.

El objetivo principal de este sistema es mantener estabilizado el péndulo, el cual se encuentra montado sobre un carro que tiene la capacidad de moverse libremente sobre el eje horizontal (véase Figura 5). Se han reportado varios enfoques de control clásico para el control del carro péndulo invertido, e incluso se han propuesto técnicas de control manejando el enfoque Lagrangiano, con el objetivo de manejar adecuadamente una función de energía, basada en una función de Liapunov para estabilizar dicho sistema no lineal [8]. Se conocen por igual técnicas de Control Neuronal [9] y técnicas de Control Difuso [10, 11, 12].

En la teoría de control se requiere exactitud de los parámetros de la variable controlada. Además, no aplica el modelo lineal cuando el péndulo tiene inclinaciones muy amplias. Por otro lado, el control basado en redes neuronales requiere de sucesivas iteraciones con el objetivo de obtener el conocimiento necesario para su aprendizaje. Sin embargo, en el Control Difuso, podemos usar reglas basadas en la experiencia para aproximar el control de una forma más precisa.

² Por su siglas Wheeled Inverted Pendulum (WIP).

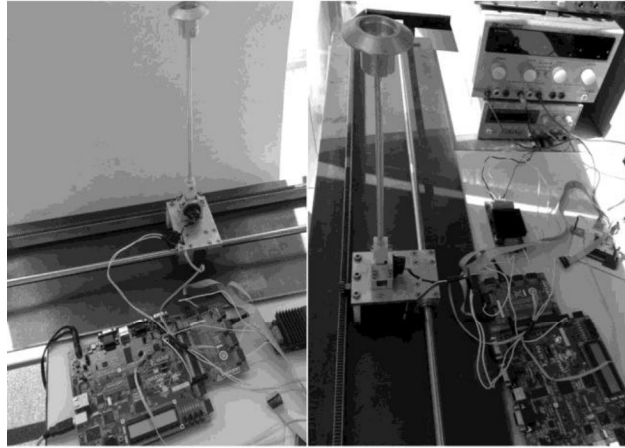


Fig. 5. Péndulo invertido construido para el laboratorio del CIC.

3 Arquitectura del Controlador Difuso en HDL

Las tres principales partes que conforman éste sistema de control descrito en HDL se muestra en la Figura 6.

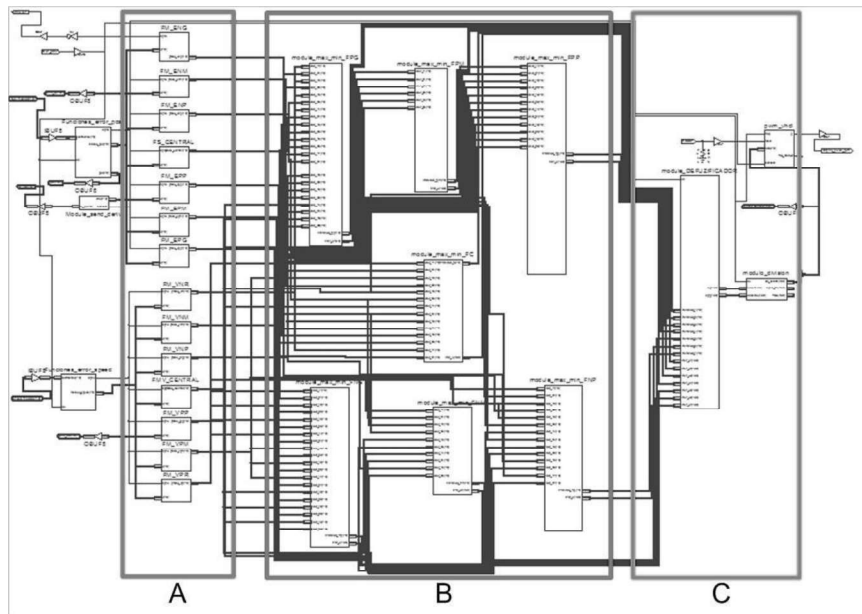


Fig. 6. Arquitectura difusa completa embebida en FPGA. A) Módulos de Fuzzificación, B), Módulos de inferencia difusa ó base de conocimiento, C) Módulo defuzzificador.

La etapa de fuzzificación se realiza a través de cálculos sucesivos para obtener el grado de membresía de la señal de entrada con respecto a las funciones de membresía (FM), que en este caso fueron de la forma triangular y trapezoidal. En la entrada del sistema se tiene almacenada en memoria el punto de referencia para obtener una posición y derivada deseada para mantener al péndulo en la posición vertical. Lo anterior ayuda a obtener un error en la posición y un error en la derivada del error. El FPGA únicamente realiza la operación de diferencia entre la posición deseada y la posición medida, valor que es enviado hacia el microcontrolador que calcula su derivada y envía su valor hacia el FPGA. De esta forma se tienen las dos variables que van a ser fuzzificadas en HDL por el primer módulo. La Figura 7 muestra el diagrama del método descrito para el cálculo del valor de membresía de la señal fuzzificada en su correspondiente universo de discurso.

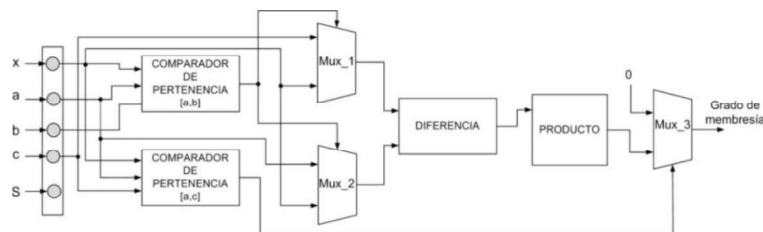


Fig. 7. Diagrama de bloques para permitir calcular el grado de membresía en funciones de membresía triangular y trapezoidal.

El cálculo del grado de membresía se limita a conocer si el dato de entrada se encuentra en alguna de las FM. Si es una función triangular, se detecta en que parte de la función cae el dato para conocer si se encuentra dentro de la pendiente negativa o positiva y realizar la operación necesaria para obtener su correspondiente valor dentro del intervalo difuso [0-255] que es la resolución del controlador de 8 bits. La Figura 8 muestra las funciones de membresía para las variables del fuzzificador del error y de derivada del error, así como el universo de discurso de salida diseñado con funciones de salida Singleton, cuyo valor es enviado al módulo generador PWM.

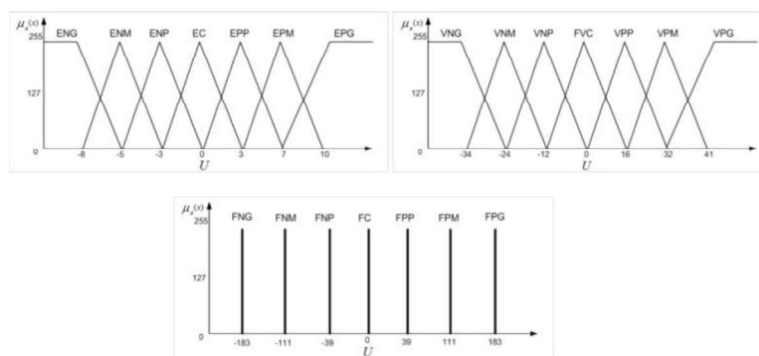


Fig. 8. Funciones de membresía para evaluar a las variables error, derivada y funciones de membresía de Singleton de salida.

El módulo de inferencia toma los grados de membresía calculados, evalúa que reglas difusas se activan y aplica la operación difusa MIN-MAX.

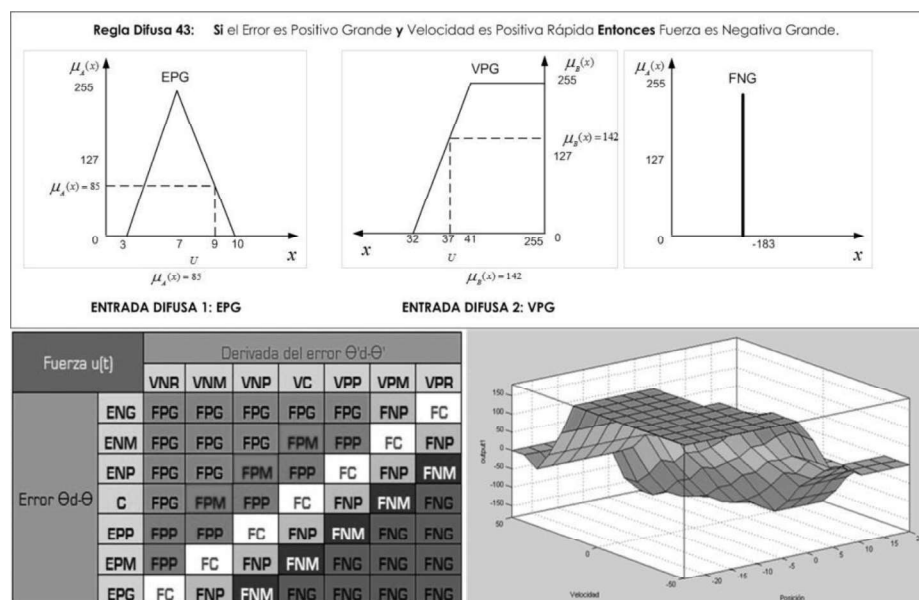


Fig. 9. Ejemplo de cálculo de grado de membresía y tabla de reglas difusas que permiten evaluar el valor del error y su derivada.

El objetivo de la etapa Defuzzificación es obtener la función de membresía final que es la agregación ó la suma de todas las combinaciones de conjuntos difusos enlazados según la base conocimiento y que se activan en un momento dado según la combinación de las dos entradas etiquetadas como error y derivada del error. Se produce una función o una superficie total, ver Figura9, Este función representa la suma de todas las implicaciones y el método de defuzzificación permite obtener el valor adecuado que representa a la función total. La implementación del Método Promedio de los Centros que se utiliza en el presente desarrollo de investigación en tecnología embebida, toma como base la investigación realizada en [13], donde se propone una arquitectura paralela para el cálculo de la salida defuzzificada a partir de la operación correspondiente para encontrar el Centro de Gravedad de la función agregada. La ecuación general para el cálculo del promedio de los centros se muestra en la ecuación (2).

$$Z^* = \frac{\sum_{l=1}^M Z^* w_l}{\sum_{l=1}^M w_l} \quad (2)$$

El valor de la variable de salida Z^* está determinado por el promedio de los centros de los M conjuntos difusos de salida con los pesos w , siendo igual a la altura de los conjuntos difusos correspondientes. En cada uno de los módulos de inferencia difusa correspondiente a cada uno de los consecuentes de salida (FNG, FNM, FNP, FC, FPP,

FPM, FPG), se realizan las dos operaciones difusas MAX-MIN, cuyo resultado es tomado por el módulo defuzzificador.

4 Resultados

Para poder validar que la salida del control difuso es la adecuada para estabilizar al péndulo es necesario corroborar los resultados generados en las tres situaciones descritas a continuación:

1. Los resultados de las simulaciones del control difuso generado en el Toolbox de Matlab.
2. Los resultados de las simulaciones del control difuso embebido en el software ISIM del fabricante del FPGA XILINX.
3. Los resultados del control real en el motor de CD por el pulso PWM generado por el FPGA.

Los dos primeros resultados permiten comparar que tan cercanos están los sistemas difusos descritos; dicha comparación permite entrever que tan preciso es el método de inferencia y de defuzzificación entre ambas simulaciones. En ésta simulación se toma en cuenta que se hace cero la variable derivada y se mantiene variante la posición en su intervalo de movimiento. La Figura 10 muestra la comparación entre las situaciones en simulación y el aporte real de la salida del sistema difuso para control del péndulo, según se observa en la gráfica, el comportamiento del sistema será devolver la posición del péndulo al cero digital.

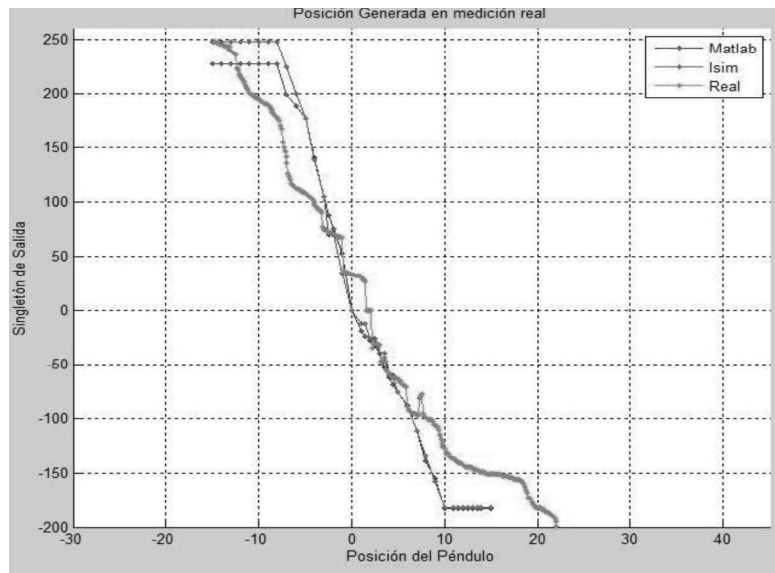


Fig. 10. Comparación del comportamiento del sistema difuso embebido en FPGA, el generado en Matlab y la señal PWM generada en el sistema real.

Finalmente se presentan los recursos utilizados en el FPGA de Xilinx XC3S500E para lograr embeber el sistema difuso de control que estabiliza el péndulo invertido. Como se observa, únicamente se utilizó el cien por ciento de los elementos lógicos con que cuenta el FPGA, usados como módulos de multiplicación y de los demás módulos se tienen suficientes recursos para hacer mejoras e incluso hacer más robusto al sistema difuso, por medio de otros operadores de inferencia o compartir el sistema de control con un algoritmo PID.

Table 1. Recursos lógicos utilizados en la descripción del sistema de control difuso en el FPGA XC3S500E de Xilinx.

Dispositivo seleccionado: 3s500efg320-5			
Slices:	2045 de 4656	43%	
Slice Flip Flops:	1002 de 9312	10%	
4 input LUTs:	3428 de 9312	36%	
Logic:	3394		
Shift registers:	34		
IOs:	60		
Bonded IOBs:	60 de 232	25%	
MULT18X18SIOs:	20 de 20	100%	
GCLKs:	1 de 24	4%	

5 Conclusiones

El uso de FPGA como sistema de procesamiento es viable para el uso de sistemas de control que requieren tiempo real. El sistema difuso embebido fue del tipo Mamdani y el uso de HDL para su descripción facilitó su implementación al usar sólo una parte de todos los elementos lógicos del FPGA de Xilinx. La estabilización del péndulo invertido en una vecindad, se consiguió utilizando el sistema anterior, como es posible ver en la Figura 10, se obtienen resultados adecuados en el control. Como trabajos futuros se piensa embeber el sistemas difuso en un Soft-Processor, para su comparación con un modelo Takagi-Sugeno-Kang y un modelo neurodifuso implementados en la misma plataforma.

Agradecimientos

Se agradece el apoyo brindado al IPN, a través del programa PIFI y COFAA; a la SIP-IPN a través de los proyectos: SIP 20131505, SIP20131182, SIP20131514 y al CONACyT 155014.

Referencias

1. M. Togai and H. Watanabe, A VLSI implementation of fuzzy inference engine: Toward an expert system on a chip. *International Journal of Information Sciences*, vol. 38, pp. 147-163, (1986).
2. John H. Lilly. *Fuzzy control and identification*. Wiley, pp. 231, (2010).
3. Chen Yong, Huang Sheng-hua, Wan Shan-ming and WU Fang, A novel fuzzy logic direct torque controller for a permanent magnet synchronous motor with a field programmable gate array. *Journal of Chongqing University*, vol. 7, no.3, pp. 228-233, (2008).
4. Oscar Montiel, Yazmin Maldonado, Roberto Sepúlveda, and Oscar Castillo, Simple Tuned Fuzzy Controller Embedded into an FPGA, *Conference of the North American Fuzzy Information Processing Society - NAFIPS*, pp.1-6, (2008).
5. Jacobo Alvarez, Alfonso Lago and Andres Nogueiras, FPGA Implementation of a Fuzzy Controller for Automobile DC-DC Converters, *IEEE International Conference on Field Programmable Technology*, pp. 237-240, (2006).
6. Lozada, H. Juan Carlos, Apuntes del curso impartido sobre Diseño de Procesadores Dedicados. Centro de Innovación y Desarrollo Tecnológico en Cómputo, Instituto Politécnico Nacional, México, (2013)
7. Zhijun Li, Chenguang Yang, Liping Fan. *Advanced Control of Wheeled Inverted Pendulum Systems*. Springer-Verlag London 2013, pp. 218.
8. C. Aguilar-Ibáñez, O. Octavio Gutiérrez F., and H. Sossa A. Controlled Lagrangian approach to the stabilization of the inverted pendulum system. *Revista Mexicana de Física*. Vol.54, No. 4, pp. 329-335, (2008).
9. Tomasz Praczyk. Using assembler encoding to solve inverted pendulum problem. *Computing and Informatics*, vol. 28, pp. 895–912, (2009).
10. S. Kizir, Z. Bingul and C. Oysu. Fuzzy control of a real time inverted pendulum system. *Journal of Intelligent & Fuzzy System*. Vol. 5177, pp. 674–681, (2008).
11. Seiji Yasunobu and Munechito Mori. Swing up Fuzzy Controller for Inverted Pendulum Based on a Human Control Strategy. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, pp. 1621-1625, (1997).
12. Tzue-Hseng S Li, Ming-Yuan Shieh. Switching-type fuzzy sliding mode control of a cart-pole system. *Mechatronics* Vol. 10, pp. 91-109, (2000).
13. Bit Technologies. hoja de especificaciones. Internet: <http://www.bitechnologies.com/pdfs/6180.pdf>. (2006).
14. R. Isermann, *Digital Control Systems*, Springer-Verlag, (1989). Maurice Gaymer Road, Devantech Ltd (Robot-Electronics). Hoja de especificaciones. Internet: <http://www.robot-electronics.co.uk/htm/md03tech.htm>. [Noviembre 2013].
15. MPLAB, hoja de especificaciones. Internet: <http://ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf>. [Noviembre 2013].