

**DESIGN OF A PI EMBEDDED CONTROLLER APPLIED TO LEVEL
CONTROL OF COUPLED TANKS****DISEÑO DE UN CONTROLADOR PI EMBEBIDO APLICADO A CONTROL DE
NIVEL DE TANQUES INTERACTIVOS**

Ing. Jairo Contreras Blanco*, **Ing. Oscar López G.***, **Ing. John Ramírez Mateus***
Esp. Sergio Quintero Ayala*, **MSc. José Bermúdez Santaella****

*** UFPS - Grupo de Investigación y Desarrollo en Microelectrónica Aplicada.**

Universidad Francisco de Paula Santander, Ave. Gran Colombia No. 12E-
96B Colsag, San José de Cúcuta, Norte de Santander, Colombia.

Tel. (+577) 5776655-5751359 Ext. 202

Email: jairo.contreras.blanco@hotmail.com, osmalogue@hotmail.com,
jhonjairorm@ufps.edu.co, sergioivanqa@ufps.edu.co

****UFPS - Grupo de Investigación y Desarrollo en Procesos Industriales- GIDPI.**

Email: joserocardobs@ufps.edu.co

Abstract: In the development of this project is carried out design, simulation and implementation of digital control of a traditional rate on a gifted plant level controller ultrasonic DL14 sensors which consists of two interactive tanks coupled by solenoid Danfoss EV260B all this by programming the softcore Microblaze microprocessor card Nexys 2 Spartan 3E using programming languages C / C ++ and VHDL on Xilinx EDK platform for inclusion, creation and implementation of IP's hardware as well as software file in XPS.

Keywords: MicroBlaze; Nexys; level; controller; IP CORE's.

Resumen: En el desarrollo de este proyecto se lleva a cabo el diseño, simulación e implementación de un control digital de tipo tradicional aplicado a una planta controladora de nivel dotada de sensores ultrasónicos DL14 la cual está conformada por dos tanques interactivos acoplados mediante electroválvulas Danfoss EV260B, todo esto mediante la programación del microprocesador softcore Microblaze de la tarjeta Nexys 2 Spartan 3E utilizando los lenguajes de programación C/C++ y VHDL sobre la plataforma EDK de Xilinx para la inclusión, creación e implementación de hardware IP's así como también el archivo de software en XPS.

Palabras clave: MicroBlaze; Nexys; Nivel; Controlador; IP CORE's.

1. INTRODUCCIÓN

Hasta principios de los años 80, los esquemas de control de las plantas industriales eran centralizados, debido al gran costo y tamaño que podía tener en la época un único computador central encargado de realizar todas las tareas. Con el paso de los años la reducción de tamaño y costo de los dispositivos

semiconductores trajo consigo un cambio en la orientación de las estrategias de control. En la actualidad el control distribuido es de mayor difusión por ofrecer ventajas tales como mejor desempeño que un esquema centralizado, mayor confiabilidad ante fallos, escalabilidad y flexibilidad (Ibrahim, 2006) además de la reducción del factor de forma y consumo de potencia, altas velocidades de operación

y posibilidad de procesamiento paralelo (MSc. Eduardo Avendaño Fernández, 2011).

En esta filosofía de simplificación y descentralización encajan los sistemas embebidos con sus ventajas, los cuales vemos a diario, ubicuos, presentes en todas partes (cámaras, celulares, herramientas, electrodomésticos) para facilitarle al ser humano sus actividades y realzar su mundo.

Un sistema embebido es un computador diseñado a medida para realizar una tarea específica, con limitadas características y periféricos necesarios para comunicarse con el mundo real y que por su simplicidad, bajo costo y eficiencia es atractivo en la aplicación de esquemas de control distribuido.

Dichos sistemas embebidos además de realizar tareas normales de control digital directo (DDC por sus siglas en inglés) permiten llevar a cabo funciones de supervisión, tales como leer datos de un teclado, mostrar información de sensores en un display, activar alarmas y además comunicarse con otros computadores existentes dentro del complejo industrial por medio de redes de área local.

En el transcurso del proyecto se tratará el modelamiento matemático del sistema, hecho en Simulink de MATLAB donde se incluyen las ecuaciones que modelan cada componente del sistema y se apreciará la curva de respuesta del mismo, la creación del controlador PI discreto y el proceso que se llevó a cabo para la creación del sistema embebido en EDK 10.1 de XILINX y la implantación de la lógica de control en el microprocesador MicroBlaze.

2. MODELO DEL SISTEMA DE TANQUES

En la figura 1 se puede apreciar el sistema de tanques interactivos, el cual consta de un primer tanque que recibe suministro de líquido constante; éste se comunica con un segundo tanque por medio de una electroválvula proporcional Danfoss EV260B, sobre la cual se aplicará la acción de control para regular el nivel del tanque superior.

El tanque inferior recibe el líquido proveniente del tanque superior y desaloja líquido hacia un tanque de depósito por medio de otra electroválvula del mismo tipo, sobre la cual se aplica la acción de control independiente para regular el nivel del tanque inferior.



Fig. 1. Sistema de tanques interactivos

Las ecuaciones mostradas a continuación modelan el sistema de tanques interactivos, estas se deducen a partir de la disposición física de los tanques, sus conexiones y el modelo matemático del comportamiento de las válvulas. (Fig. 2).

$$Q_i - C_d \cdot a \cdot \sqrt{2g(H_1 - H_2)} = A \frac{dH_1}{dt} \quad (1)$$

$$C_d \cdot a \cdot \sqrt{2g(H_1 - H_2)} - C_d \cdot a \cdot \sqrt{2gH_2} = A \frac{dH_2}{dt} \quad (2)$$

La respuesta de dichas ecuaciones ante un escalón unitario, puede ser visualizada en la figura 3, la cual fue obtenida usando el software de Simulink, en cuya plataforma se obtienen las curvas de reacción del sistema acoplado en el dominio del tiempo y que además permiten caracterizar el sistema con el fin de realizar la sintonización de los controladores.

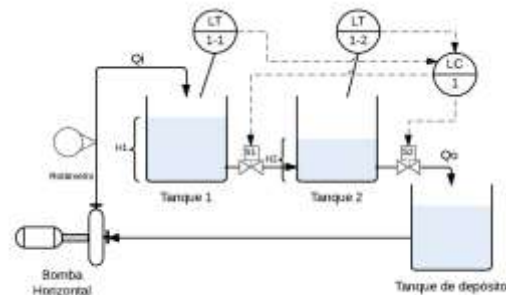


Fig. 2. Diagrama de control del sistema de tanques.

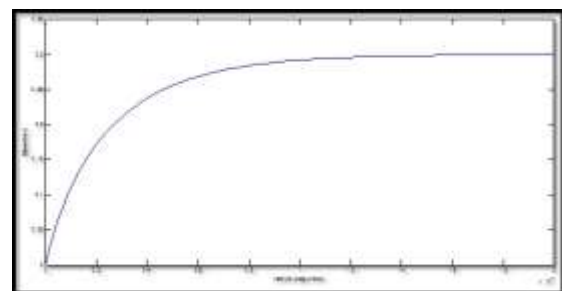


Fig. 3. Respuesta a un escalón unitario para el tanque superior del sistema acoplado.

3. CONTROLADOR PI DISCRETO

Un controlador PID puede ser aproximado a su forma discreta usando la aproximación trapezoidal para la parte integral y la aproximación de diferencias hacia atrás para la parte derivativa. De esta forma la función de transferencia resulta como sigue:

$$\frac{U(z)}{E(z)} = K_p \left[1 + \frac{T}{T_i(1-z^{-1})} + T_d \frac{(1-z^{-1})}{T} \right] \quad (3)$$

A partir de la ecuación 3 se puede representar un controlador PID discreto como una estructura paralela, sumando los términos proporcional, derivativo e integral, como se muestra en la figura 4.

Con base en la figura 4 se pueden plantear las siguientes ecuaciones en diferencias:

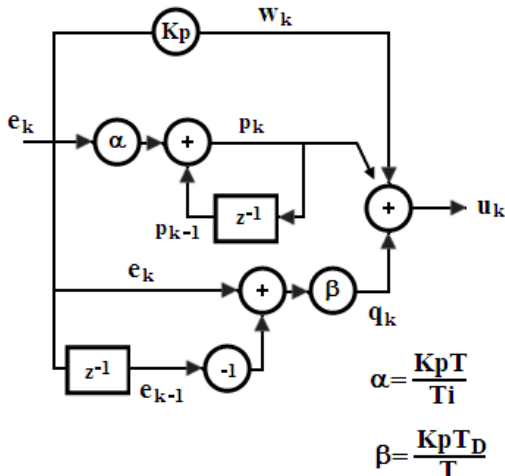


Fig.4. Controlador PID discreto.

$$w_k = K_p e_k \quad (4)$$

$$p_k = \alpha e_k + p_{k-1} \quad (5)$$

$$q_k = \beta(e_k - e_{k-1}) \quad (6)$$

$$u_k = w_k + p_k + q_k \quad (7)$$

Las ecuaciones 4, 5, 6 y 7 fueron incluidas en la programación del controlador digital en el microprocesador embebido; la función de transferencia de un controlador PI discreto no incluye la función derivativa del PID, por lo tanto la función de transferencia que representa al controlador PI discreto se muestra a continuación:

$$\frac{U(z)}{E(z)} = K_p \left[1 + \frac{T}{T_i(1-z^{-1})} \right] \quad (8)$$

Los controladores de tipo PI son usualmente encontrados en procesos de control de temperatura, posición y velocidad de motores monofásicos (PhD. Omar Pinzón Ardila, 2011), sistemas de regulación de voltaje como en circuitos RC (Ing. Yamir Hernando Bolaños, 2012), entre otras aplicaciones.

4. IMPLEMENTACIÓN DEL CONTROLADOR PI DIGITAL EN MICROBLAZE

4.1 Conversor Análogo-Digital

El controlador Digital es implementado en una tarjeta de desarrollo Nexys 2, que incluye una FPGA Spartan 3E de Xilinx. (Fig. 5).



Fig.5. Tarjeta de desarrollo Nexys2.

Como dispositivo de adquisición de datos se utilizó un ADC0808 de National Semiconductors, que puede operar con una señal de reloj desde 500 Khz hasta 1.2 Mhz y muestrear hasta 8 canales análogos a una resolución de 8 bits. Figura 6.

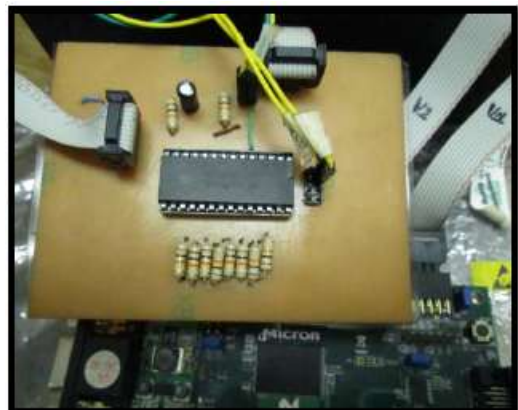


Fig.6. Tarjeta de Adquisición de datos ADC0808.

Para la distribución de datos se usaron dos conversores DAC0808 de 8 bits. Con el propósito de generar las señales de control del ADC se creó un periférico IP CORE¹ descrito en lenguaje VHDL en la herramienta XILINX EDK 10.1 y el cual se comunica con MicroBlaze por medio de interfaces de interconexión IPIF² para la transmisión y recepción de datos. Posee un registro de escritura para la selección del canal del conversor ADC0808 del cual se va a adquirir datos, asimismo líneas de comunicación para indicar al conversor ADC cuando iniciar el proceso de conversión, canal de lectura de datos y 8 líneas para recepción de datos muestreados por el conversor. El periférico incorpora un proceso que hace posible la generación de la señal de reloj del conversor ADC, realizando una división de frecuencia a partir de la señal de reloj del oscilador de la tarjeta Nexys 2 que es de 50 MHz a un valor inferior o igual a 500 KHz, que es la frecuencia con la cual trabaja el conversor ADC0808; la figura 7 muestra la estructura y las señales características del IP CORE que representa al conversor análogo-digital.



Fig. 7. Tarjeta de transmisión de datos con DAC0808.

4.2 Display 7 Segmentos

El display 7 segmentos de la Nexys 2 hace posible la visualización del nivel de cada uno de los tanques. El control de este periférico de salida se logra por medio de un IP CORE que incluye 2 registros accesibles por software desde el procesador. En cada uno de estos registros se escribe, desde el procesador, la altura de líquido correspondiente a cada tanque, de acuerdo a los datos leídos por el ADC.

Mediante un *dip-switch* de la tarjeta de desarrollo, se multiplexa el registro del cual se toma la información para enviarla como parámetro a una función que hace la conversión de formato binario a BCD, en forma concurrente un proceso hace la conversión de cada dígito BCD a lógica 7 segmentos y finalmente otro proceso multiplexa cada dígito para así mostrar adecuadamente la altura de líquido en el display 7 segmentos, medida que se entrega en centímetros con una cifra decimal; la figura 8 representa la estructura y señales que participan en la creación del IP CORE que modela al display 7 segmentos.

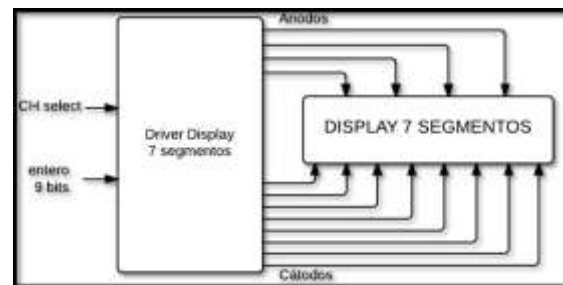


Fig. 8. IP CORE del display 7 segmentos.

4.3 Arquitectura y estructura hardware

La figura 9 permite visualizar todos y cada una de las instancias periféricas que forman parte de la aplicación realizada en la plataforma EDK de XILINX, aquí se pueden identificar el microprocesador MicroBlaze, registros de memoria, IP CORE's elaborados a medida en base a periféricos de propósito general GPIO, controlador de interrupciones XINTC y el contador XTIMER, así como también las IP's agregadas al sistema por defecto pertenecientes al generador de reloj y el estado reset.

Pueden identificarse dos clases de buses en la figura y que corresponden a los tipos PLB y LMB:

PLB De sus siglas en inglés "*Processor Local Bus*", surge como mejora a los buses OPB y consiste en una interfaz de 32 bits con el procesador capaz de soportar conexiones con periféricos de 8, 16 y 32 bits.

LMB De sus siglas en inglés "*Local Memory Bus*", se trata de un bus síncrono de baja latencia utilizado principalmente para acceder a la BRAM; el procesador MicroBlaze contiene tanto LMB de datos (DLMB) como también de instrucciones (ILMB).

¹ Núcleo de Propiedad Intelectual

² Interfaz de Propiedad Intelectual

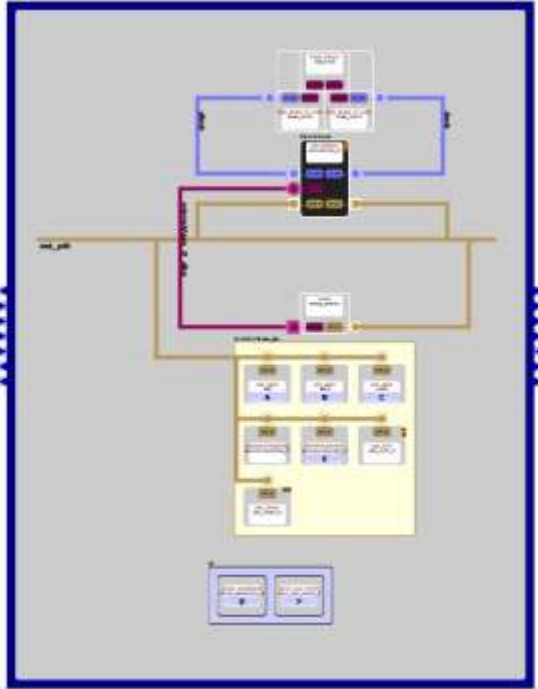


Fig. 9. Arquitectura hardware del controlador.

4.4 Algoritmo del controlador PI

La herramienta EDK de XILINX provee dos formas diferentes para poder implementar un archivo software en el soft-processor MicroBlaze, una de estas es mediante la herramienta SDK³ de XILINX que nos brinda una plataforma de desarrollo que soporta los lenguajes de programación C/C++ y que permite el trabajo de forma paralela a la inclusión de hardware que se lleva a cabo en el XPS; y el otro camino que se brinda es a través del mismo XPS, en donde se elige e inserta un proyecto software de tipo C/C++ almacenado en el equipo previamente realizado en alguna plataforma que ofrezca los beneficios de la programación en estos lenguajes de programación ya mencionados.

La figura 10 refleja el esquema de programación que se llevó a cabo para implementar el controlador PI que permita el manejo y manipulación de los actuadores del sistema de forma adecuada, para esto se hizo necesaria la inclusión de algunos IP CORE's que corresponden a los ya mencionados periféricos DAC y display de 7 segmentos, así como también de TIMER's y un controlador de interrupciones INTC.

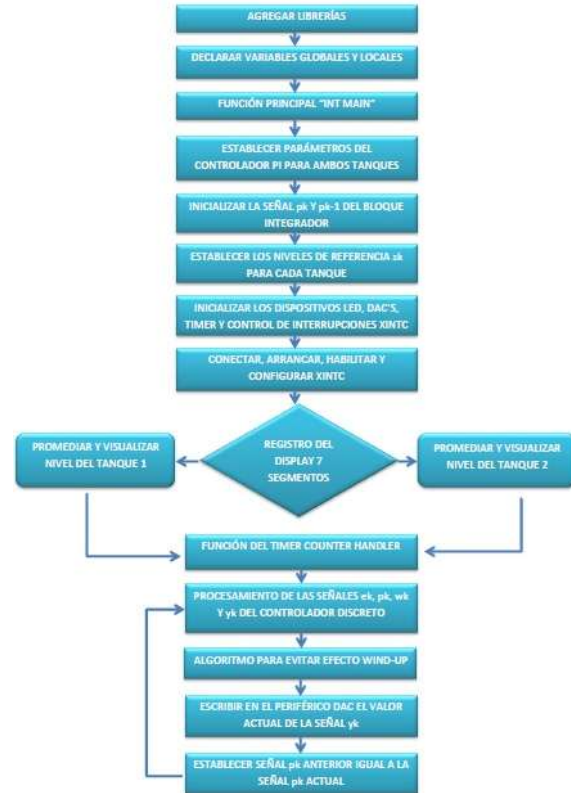


Fig. 10. Carta ASM del algoritmo controlador PI.

En la aplicación software se definen:

- La subrutina de atención de la interrupción, que es generada por un *timer* configurado para generar interrupciones periódicas cada segundo de acuerdo al periodo de muestreo.
- Se definen punteros con los cuales se manipulan los datos que se leerán o escribirán en los registros de los periféricos IP.
- Se inicializan y configuran periféricos: Controlador de interrupciones XINTC, *timer* XTIMER y periféricos de salida de propósito general XGPIO para escribir datos digitales en los DACs.

5. RESULTADOS

Para obtener la curva característica del control de nivel en el tanque superior fue necesaria la elaboración de un nuevo VI⁴, gracias a LabVIEW⁵ de National Instruments, capaz de adquirir la señal de voltaje del sensor ultrasónico de nivel en tiempo real, compuesto de un diagrama de bloques, lugar en que

³ Software Development Kit

⁴ Proyecto de instrumentación virtual generado en LabVIEW

⁵ Laboratory Virtual Instrumentation Engineering Workbench

se incluye un asistente DAQ⁶ con sus respectivas conexiones a filtros de señal y operadores matemáticos que facilitan la lectura de la medición a través de un mapa de forma de onda, que almacena el nivel de tensión medido y lo tabula en una gráfica de nivel de agua en milímetros vs tiempo como lo muestra la figura 11, y un indicador de salida que muestra el valor a almacenar en la gráfica.

El comportamiento del primer tanque del sistema de tanques acoplados revela que se parte de una acumulación inicial de agua equivalente a 20 centímetros y se especifica un nivel de establecimiento de 30 centímetros a través del algoritmo software a lo largo de un tiempo de ejecución igual a 9000 segundos.

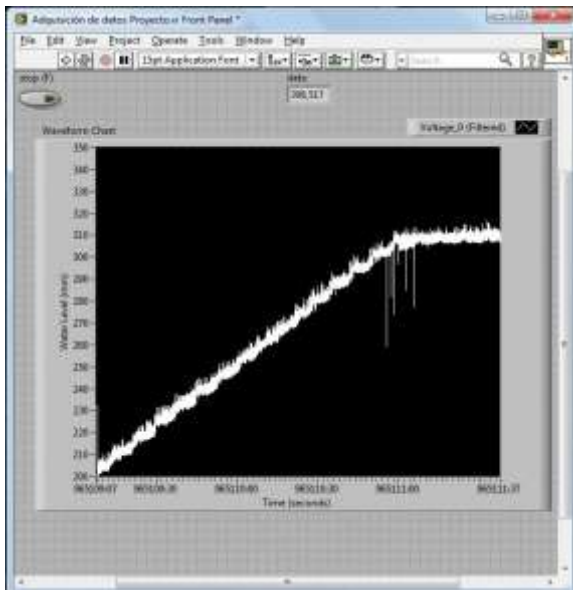


Fig. 11. Adquisición de datos para el nivel del tanque 1

El tabla 1 muestra un cuadro comparativo entre los resultados obtenidos en el proceso de simulación y los resultados anteriormente tabulados en la fig. 11, se evalúan algunos de los parámetros característicos de la forma de onda de un controlador PI tal como lo son el sobrepaso máximo y el tiempo de establecimiento.

La implementación del controlador se realizó en una FPGA Spartan 3E mediante la herramienta de diseño XPS de XILINX, utilizando la tarjeta de prueba Nexys2 de Digilent de la que se utilizaron los cuatro puertos de expansión PMod's.

⁶ Data Acquisition-Sistema de desarrollo que permite la adquisición de datos y señales en tiempo real

En la FPGA se consumieron los siguientes recursos internos para el diseño del procesador:

- 26 % del número de flip-flops dispuestos en la tarjeta (total de 9312 unidades).
- 40 % de la cantidad de LUT's con los que cuenta (total de 9312 LUT's).
- 23 % del número total en bloques de entrada/salida IOB's (total de 232 IOB's).
- 80 % de la cantidad total de bloques de memoria RAM.

Tabla 1: Simulación vs adquisición de datos en tiempo real del sistema

	SIMULACIÓN	EXPERIMENTAL
Sobrepaso Máximo (cm)	1.1	0.3
Tiempo de Establecimiento (seg)	8000	7320

6. CONCLUSIONES

El comportamiento observado a partir de las pruebas realizadas con la DAQ6009 refleja que el desempeño del sistema de control resulta acorde a los parámetros de diseño planteados y observados en las simulaciones con Matlab. El error del nivel en estado estacionario en el tanque superior es de 1% que representa una fluctuación de 3 mm en el nivel. Por otra parte el sobrepaso máximo nunca es mayor a 1 cm, lo que corresponde a un 3% del nivel de referencia y que al mismo tiempo cumple con la condición de diseño establecida por Ziegler-Nichols referida a un sobrepaso máximo menor o igual al 5%. Estos parámetros de desempeño, son óptimos para un controlador PI y se comprueba el correcto funcionamiento de la implementación.

La creación de periféricos IP CORE permite al procesador central relevar tareas específicas rutinarias con el fin de descargar funciones de procesamiento y hacer el sistema más eficiente y modular. Una ventaja de los IP CORE's es la capacidad de hacer las tareas por medio de hardware, lo que los convierte en una opción más veloz que establecer rutinas mediante software, y al estar el procesador central libre de realizar tareas repetitivas, puede disponer recursos para otras funciones; otras de las ventajas que acarrea la inclusión de IP CORE's radica en su flexibilidad y escalabilidad, "la flexibilidad se asocia a la diversidad de IP CORE's que pueden ser implementados en un sistema y la escalabilidad radica en la cantidad de IP CORE's que pueden coexistir sin alterar el sistema al ser añadidos" (Salazar, 2012).

Con la finalidad de ejercer control del sistema de manera adecuada y operar correctamente los actuadores de este, se hizo necesaria la utilización de un parámetro denominado BIAS que consiste en un valor digital comprendido entre 0 y 255, debido a la resolución que presenta el conversor análogo-digital ADC0808 de 8 bits. Este valor nos asegura que en casos en donde la señal de error se torne negativa la electroválvula tenga una mayor apertura, y en caso contrario, cuando la señal de error se muestre positiva la electroválvula tienda al cierre de su diafragma; todo esto se hace ineludible debido a que el control que presenta el sistema de tanques interactivos es de tipo inverso.

Dentro del proyecto software se tornó de carácter obligatorio agregar una serie de líneas de programación cuya función consiste en evitar el efecto WIND-UP del integrador mediante una comparación del máximo y mínimo valor de la acción de control que en el caso particular de los tanques interactivos incluye el desempeño del parámetro BIAS, esta consideración se es precisa con el propósito de impedir la saturación en los actuadores y al mismo tiempo evitar una sobre-oscilación enorme.

REFERENCIAS

- Bolaños, H., Pineda L., Mosquera V. (2012). *Didactic System For The Implementation Of Digital Controllers*. Revista Colombiana de Tecnologías de Avanzada, No 20, Vol 2, pp 127-135.
- Cadena, Julio., Mollocana, Juan. (2012). *Diseño de Hardware y Software de Systems-On-Chip empleando tecnología Xilinx EDK*. Repositorio. Esp. Sede Sangolquí. Ecuador
- Contreras J. (2010) *Identification And Fuzzy Control Of A Two Tank System*. Revista Colombiana de Tecnologías de Avanzada, No 16, Vol 2, pp 43-49.
- Ibrahim, D. (2006). *Microcontroller Based Applied Digital Control*. Editorial J. Wiley, & L. Sons, Ltda., England.
- Jesman R., Vallina F., y Saniie J. *MicroBlaze Tutorial Creating a Simple Embedded System and Adding Custom Peripherals Using Xilinx EDK Software Tools*. ECASP Research Lab., Illinois Institute of Technology. E.U.
- Laubwald, E. (2006). *Coupled tanks systems 1*, control-systems-principles. co. uk.
- Minev, P., y Kukenska, V. (2007, November). *Implementation of soft-core processors in FPGAs*. UNITECH'07 International Scientific Conference.
- Tong, J., & Khalid, M. (2008). *Profiling Tools for FPGA-Based Embedded Systems: Survey and Quantitative Comparison*. Journal Of Computers, No 3, Vol 6, pp 1-14
- Urriza I., Barragan L., Navarro D., Artigas J., Lucía O., y Jiménez O., (2009). *FPGA Embedded Soft-Core Processor Implementation of a Digital Controller for a DC-DC Converter*. XXIV Conference on Design of Circuits and Integrated Systems (DCIS'09). Zaragoza. España.
- Xilinx, Inc. (2012). *MicroBlaze Processor Reference Guide, Embedded Development Kit EDK 14.1.*, UG081. V14.1., E.U.
- Xilinx, Inc. (2009). *EDK Concepts, Tools and Techniques.*, UG683 EDK 11., Estados Unidos.