

DISEÑO DE UN CONTROLADOR BASADO EN FPGA PARA UN ROBOT CARTESIANO

González Villagómez, Jacob (1), Rodríguez Doñate, Carlos (2), Cabal Yépez, Eduardo (2).

1 [Licenciatura en Comunicaciones y Electrónica, Universidad de Guanajuato, DICIS Irapuato-Salamanca Sede Yuriria] | [j.gonzalezvillagomez@ugto.mx]

2 [Departamento de Estudios Multidisciplinarios, DICIS, Campus Irapuato-Salamanca, Universidad de Guanajuato] | [c.rodriguezdonate@ugto.mx, educabal@ugto.mx]

Resumen

En la actualidad la construcción de diversos robots basados en las leyes de control convencional como es el controlador Proporcional-Integral-Derivativo (PID), predominan en la mayoría de las aplicaciones. Recientemente, la implementación del PID se ha basado en Arreglos de Compuertas Programables en Campo (*field programmable gate array*, FPGA), debido a que la estimación de la señal de control se puede realizar en tiempo real, con bajo consumo de recursos computacionales, bajo consumo potencia, además de lograr la portabilidad del diseño. En este trabajo se presenta el diseño de un sistema embebido en FPGA para controlar un robot cartesiano, donde la ley de control que rige al sistema es un PID. Además, se propone la optimización de la arquitectura del PID para reducir el consumo de recursos del FPGA. En base a los resultados experimentales se logró obtener un tiempo de procesamiento de $0.27\mu s$ y un consumo de recursos del FPGA utilizado menor del 8%. Las pruebas experimentales se realizaron con un robot cartesiano de dos eslabones, donde cada eslabón cuenta con un servomotor DCM50, servoamplificador digital DCS303 y encoder incremental de 1000 pulsos por revolución, la implementación del sistema embebido se realizó en el kit de desarrollo DE1 de altera que tiene el FPGA Cyclone II EP2C20F484.

Abstract

Nowadays the construction of several robots based on conventional control laws such as Proportional-Integral-Derivative (PID) controller, predominate in most applications. The implementation of PID can be based on field programmable gate arrays (FPGA), since the control signal can be estimated in real time, with low computational resources, and low power consumption beside providing a portable design. This paper describes the design of an embedded system based on FPGA to control a cartesian robot using a PID control law. Furthermore, optimization of the PID architecture is proposed to reduce the resource consumption in the FPGA. Experimental results show that the proposed hardware implementation achieves a high processing performance of around $0.27\mu s$ on estimating the control signal, with low resource consumption of less than 8% of logic elements. The test was performed with a tow-link cartesian robot, and each link has a DCM50 servomotor, a DCS303 digital servo amplifier and an incremental encoder of 1000 pulses per revolution. The embedded system is implemented in the DE1 development kit from Altera into the Cyclone II EP2C20F484 FPGA.

Palabras Clave

Leyes de control; PID; Sistema Embebido; VHDL

INTRODUCCIÓN

Anteriormente los procesos en la industria eran hechos por trabajadores, pero al paso del tiempo el hombre comenzó a industrializar estos procesos mediante la fabricación de máquinas que hicieran este trabajo [1]. En las últimas décadas, se han propuesto diferentes técnicas de control, algunas se basan en leyes de control adaptivos, difusos, neuronales, proporcionales, integrales, derivativos y sus variantes [2-3]. Sin embargo, en la actualidad los controladores basados en la ley de control proporcional-integral-derivativo (PID), predominan en la mayoría de aplicaciones, sobre todo en el sector industrial [4]. Entre las principales características que se pueden destacar de los PID son su robustez ante perturbaciones, bajo grado de complejidad al implementarlo y su estabilidad.

La gran variedad de aplicaciones que puede tener un ese tipo de máquinas (robots) ha llevado a que varios autores aborden este tema, ya sea para mejorar el desempeño mecánico o el sistema de control. Recientemente, en la literatura se han propuesto diferentes arquitecturas para un PID basadas en un FPGA, como es el caso en la propuesta [5], donde se diseña un sistema embebido para controlar un robot de cuatro eslabones, la arquitectura del PID se diseñó en base de cómputo paralelo y es replicada para cada uno de los eslabones, lo que aumenta drásticamente el consumo de recursos del FPGA. En el trabajo presentado en [6] se muestra un diseño del controlador PID en FPGA con Look-Up-Table (LUT), en el artículo se menciona que se reduce el consumo de los elementos de la FPGA, pero este diseño no es reconfigurable, ya que las constantes de control son guardadas en la memoria ROM del FPGA. Al igual que los trabajos anteriores, se pueden encontrar propuestas donde la arquitectura diseñada para PID se basa en una estructura paralela que consumen altos recursos del FPGA, pero con un excelente desempeño de la estimación de señal de control, que en la mayoría de las aplicaciones prácticas basta con un desempeño de 1 ms.

A diferencia de las propuestas mencionadas, en este trabajo se presenta el diseño de un sistema embebido en FPGA para controlar un robot cartesiano de dos eslabones, optimizando la arquitectura de los dos PID que se requieren para

el robot, así como también se realiza un sistema de control con arquitectura abierta, tanto para fines didácticos como de investigación, ya que como trabajos futuros se explorarán diferentes leyes y esquemas de control, además de que se pretende integrar sistemas de visión.

Marco teórico

Un sistema de control es la combinación de distintos componentes físicos o digitales, entrelazados entre sí con el fin de regir el comportamiento de un dispositivo o mecanismo. Los componentes de un sistema pueden ser representados como modelos matemáticos nombrados funciones de transferencia. Entre estos componentes se pueden encontrar los distintos controladores. En este trabajo se utilizará la estructura paralela de un controlador PID, la cual se muestra en la Imagen 1. En este lazo de control, la señal de error $e(t)$ se define mediante la diferencia de la señal de referencia $r(t)$ y la señal que sale de la planta $c(t)$. Por otro lado, la acción del controlador PID o señal de control $u(t)$ queda determinada por la ganancia proporcional (k_p) que actúa directamente con la señal de error, la ganancia integral (k_i) que actúa con la integral del error y finalmente la ganancia derivativa (k_d) que actúa con la derivada del error.

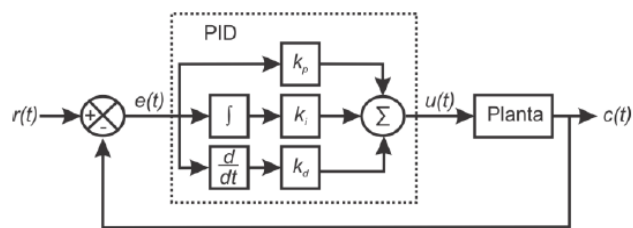


Imagen 1.- Lazo de control, mediante PID con topología paralela.

MATERIALES Y MÉTODOS

El método que se siguió para el proyecto se basó en tres etapas: Identificación de la planta, sintonización y diseño del sistema embebido.

Identificación

El proceso que se utilizó para la identificación del sistema a controlar fue mediante el método

experimental, éste consiste en enviar una señal de entrada conocida al sistema de control (imagen 2 a) y obtener la señal de respuesta (imagen 2 b). A partir de estas dos señales se utilizó el toolbox Ident de Matlab para estimar la función de transferencia (imagen 2 c) que representará la dinámica del sistema de control.

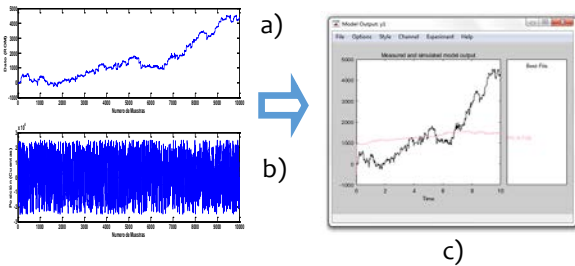


Imagen 2.- Identificación de la planta, a) Señal conocida, b) Respuesta (Cuentas de encoder), c) Función de Transferencia.

Sintonización

El análisis de sistemas en el dominio de la frecuencia, es un método que se usa con la finalidad de conocer las características físicas de del sistema a controlar a partir de su respuesta. Para poder sintonizar un controlador con este método se debe de cumplir la condición de fase y magnitud, lo cual asegura que el sistema será estable. Estas condiciones se muestran en la ecuación (1) y (2):

$$|G_c(j\omega)||G(j\omega)| = 1 \quad (1)$$

$$\angle G_c(j\omega) = -\pi + MF - \angle G(j\omega) \quad (2)$$

donde $G_c(j\omega)$ es la función de transferencia del controlador y $G(j\omega)$ la función de transferencia de la planta, ambos en el dominio de la frecuencia. Si el PID se define mediante la función de transferencia presentada en (3), se asegura que el controlador agregará polos reales a la función de transferencia en lazo abierto.

$$G_c(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k(s+b)(s+a)}{s} \quad (3)$$

Al sustituir $s = j\omega$, se obtendrá la función de transferencia del controlador en el dominio de la frecuencia. Y como se presenta en [7], los parámetros de sintonización para el controlador PID quedarán definidos por (4) y (5)

$$b = \frac{\omega}{\tan\left[MF - \frac{\pi}{2} - \tan^{-1}\left(\frac{\omega}{a}\right) - \angle G(j\omega)\right]} \quad (4)$$

$$k = \frac{\omega}{\left(ab \sqrt{\left(\frac{\omega}{a}\right)^2 + 1}\right) \sqrt{\left(\frac{\omega}{b}\right)^2 + 1}} (|G(j\omega)|) \quad (5)$$

Donde b es la ubicación de uno de los polos del controlador, ω , MF y a son parámetros proporcionados por el diseñador que es la frecuencia de corte, margen de fase y la ubicación de un polo, respectivamente.

Aplicando las aproximaciones de integración y diferenciación presentadas en [7], la ecuación en diferencias discretizada del PID queda definida por (6).

$$Y(n) = b_1 Y(n-1) + a_0 e(n) + a_1 e(n-1) + a_2 e(n-2) \quad (6)$$

donde b_1 , a_0 , a_1 y a_2 son coeficientes definidos por (7-10)

$$b_1 = 1 \quad (7), \quad a_0 = k_p + \frac{k_i T}{2} + \frac{k_d}{T} \quad (8),$$

$$a_1 = \frac{k_i T}{2} - k_p - \frac{2k_d}{T} \quad (9), \quad a_2 = \frac{k_d}{T} \quad (10)$$

donde T es el periodo de muestreo.

Sistema embebido

El sistema embebido diseñado para el control del robot cartesiano se implementó en un FPGA el cual se muestra en la imagen 3, el cual está integrado por: la implementación de un PID, un decodificador de encoder, un decodificador de RS-232, un punto suma y un modulador de frecuencia.

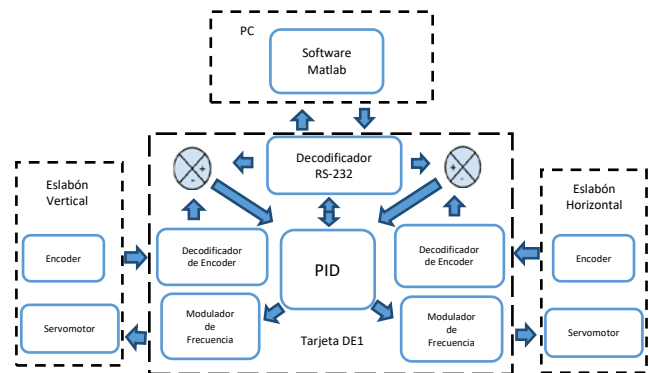


Imagen 3.- Sistema embebido general

Parte fundamental de este proyecto es la implementación del PID. A partir de la ecuación discreta del PID, se diseñó la arquitectura que se muestra en la imagen 4. Esta arquitectura se compone por dos bancos de registros, dos ROM,

una unidad de multiplicación y acumulación (MAC), un MUX para seleccionar que eslabón utilizará la MAC, un bloque de retroalimentación, dos registros de salida (OM1, OM2) y una unidad de control (UC) para sincronizar el procesamiento.

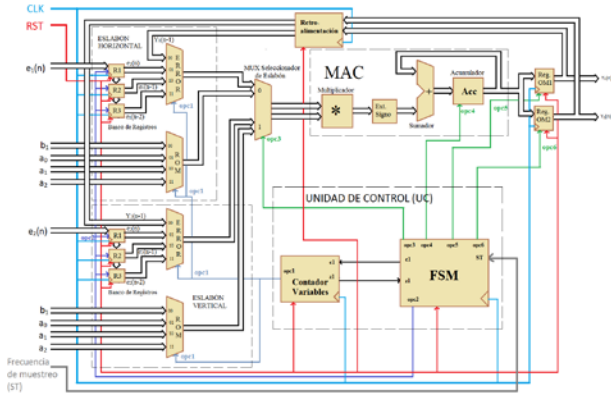


Imagen 4.- Arquitectura digital del PID

RESULTADOS Y DISCUSIÓN

Los resultados de la síntesis de la arquitectura utilizando el Software Quartus II, demuestran que se logró proponer una arquitectura eficiente (máxima frecuencia de operación de 64 MHz) y de bajo consumo de hardware para un FPGA, ya que se utiliza menos del 8% de las unidades lógicas y menos del 21 % de los multiplicadores embebidos del FPGA Cyclone II EP2C20F484.

Por otro lado, la etapa experimental se realizó con el robot que se muestra en la figura 5, donde a partir de los resultados de la identificación se obtuvo la función de transferencia descrita por (11).

$$G(s) = \frac{1.352}{3.856 \times 10^{-9} s^3 + 0.02507 s^2 + 14.75 s + 1} \quad (11)$$

Utilizando la función de transferencia (11) se realizó la sintonización del PID con un margen de fase elevado con la finalidad de evitar alterar la estabilidad del sistema (Grados = 81) y un tiempo de respuesta pequeño para reducir el ancho de banda del sistema ($t_r = 0.2$) y el cero de este controlador se ubicó en 7 ($a = 7$).

Utilizando los parámetros definidos anteriormente se obtuvo la función de transferencia del controlador (12).

$$G_c(s) = \frac{4.499 s^2 + 53.85 s + 156.3}{s} \quad (12)$$



Imagen 5.- Estación de Trabajo, Robot cartesiano de dos ejes.

En base a la simulación de este controlador se observó que la respuesta del sistema ante un escalón se comporta con un sobrepaso pequeño y llega rápidamente a su estado estacionario sin mantener oscilaciones constantes. Al discretizar y enviar las constantes de este controlador al sistema embebido, lo cual se realizó mediante el software de Matlab y la interfaz RS-232, se procedió a enviar diferentes referencias al eslabón del robot para verificar su comportamiento. Estos resultados se muestran las imágenes 7-9. La imagen 7 a) describe el comportamiento de uno de los eslabones del robot con las siguientes referencias 20000, 25300, -2100 y 50100 cuentas de encoder, las cuales se indican con las líneas punteadas de color rojo, y la señal de error se muestra en la imagen 7 b), en base a estas dos figuras se observa que el eslabón llega a la posición que se le indica y no genera oscilación en estado estacionario, además su respuesta es muy rápida. Para apreciar mejor el comportamiento de este controlador, la imagen 8 a) y b) presenta un zoom en el intervalo de tiempo de 15s a 30s de una de las pruebas. Finalmente, en la imagen 9 a) y b) se presentan los resultados del sistema embebido al controlar los dos eslabones del robot con una referencia de 100000 cuentas de encoder, en esta figura se puede observar que el comportamiento de ambos eslabones es muy similar. Cabe destacar que en ambos casos los errores son alrededor de 3 a 6 cuentas de encoder.

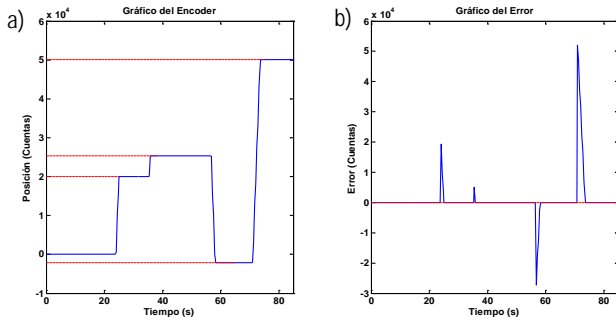


Imagen 7.- Prueba del sistema con distintas referencias (línea roja punteada), a) Gráfico del Encoder (posición-cuentas) b) Gráfico del Error (cuentas).

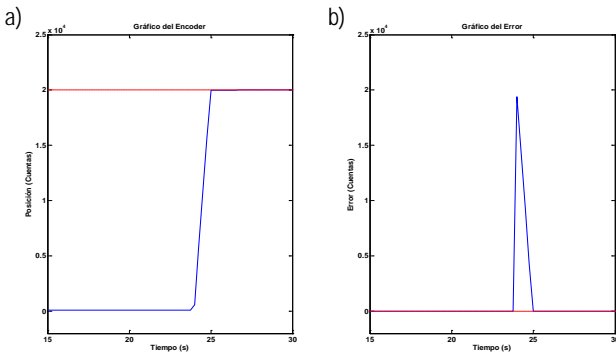


Imagen 8.- Zoom de la imagen 7, a) Gráfico del Encoder (posición-cuentas) b) Gráfico del Error (cuentas).

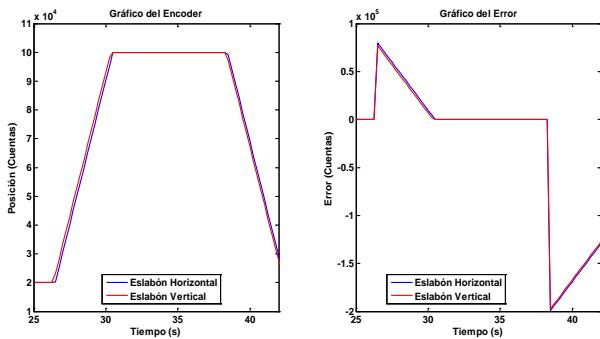


Imagen 9.- Comparación de la respuesta de los dos eslabones del robot cartesiano, a) Gráfico del Encoder (posición-cuentas) b) Gráfico del Error (cuentas).

CONCLUSIONES

En este trabajo se realizó la optimización de la arquitectura del controlador, logrando implementar

el sistema en un FPGA de bajo costo, con un tiempo de procesamiento de $0.27\mu s$, el cual supera al tiempo requerido en la mayoría de las aplicaciones industriales de control. Por otro lado, el error en estado estacionario que presentó el sistema fue entre 3 a 6 cuentas de encoder, que es muy poco considerando que el encoder tiene una resolución de 1000 pulsos por vuelta.

AGRADECIMIENTOS

Al DEM-Yuriria DICIS, Universidad de Guanajuato. Al Consejo de Ciencia y Tecnología del Estado de Guanajuato (CONCyTEG) por financiar el proyecto (convenio 120-UG2015 y 096-UG2016).

REFERENCIAS

- [1] Ollero Baturone, A. (2001). Robótica Manipuladores y Robots Móviles (1ra ed.), Marcombo S.A.
- [2] Hong, F., Ge, S. S., Pang, C. K., Lee, T. H. & Sun Z. (2010). Robust adaptive neuro-fuzzy control of uncertain no holonomic systems, In Proceedings of the 8th IEEE International Conference on Control and Automation (ICCA), pp. 2201-2206.
- [3] Aboelaze, M. & Shehata, M. G. (2015). Implementation of multiple PID controllers on FPGA, In Proceedings of IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 446-449.
- [4] Kocur, M., Kozak, S. & Dvorscak, B. (2014). Design and Implementation of FPGA-digital based PID controller, In Proceedings of 15th International Carpathian Control Conference (ICCC), pp. 233-236.
- [5] Akkaya, S., Akbati, O. & Gorgun, H. (2014). Multiple closed loop system control with digital PID controller using FPGA, In Proceedings of International Conference on Control, Decision and Information Technologies (CoDIT), pp. 764-769.
- [6] Chan, Y. F., Moallem, M. & Wang, W. (2004). Efficient implementation of PID control algorithm using FPGA technology, in proceedings of IEEE 43rd Conference on Decision and Control (CDC), p.p. 4885 – 4890.
- [7] González, J. (2016). Diseño de un controlador basado en FPGA para un robot cartesiano. Licenciatura. Universidad de Guanajuato.