

IMPLEMENTACIÓN Y CO-SIMULACIÓN DE UN CONTROLADOR PID EN TIEMPO REAL CON FPGA DE BAJO COSTO Y CÓDIGO LIBRE

IMPLEMENTATION AND CO-SIMULATION OF PID CONTROLLER RT BASED ON FPGA OF LOW COST AND OPEN SOURCE

Edson Cruz Miguel

Universidad Autónoma de Querétaro
ecruz30@alumnos.uaq.mx

Juvenal Rodríguez Reséndiz

Universidad Autónoma de Querétaro
juvenal@uaq.edu.mx

Carlos A. González Gutiérrez

Universidad del Valle de México
calberto.gonzalez@uvmnet.edu

Resumen

Los FPGA's (Field Programmable Gate Array) son dispositivos que tienen un gran número de elementos lógicos programables, por medio de lenguaje descripción de hardware (HDL) es posible desarrollar aplicaciones de nivel industrial. La disponibilidad de éstas en el mercado y la tendencia de herramientas libres han hecho que su uso prolifere en distintas áreas de la industria, aeronáutica, robótica, automatización. Los FPGA a través de HDL se convierten en verdadero hardware y esto se transforma en alta velocidad de procesamiento. La inclusión de dispositivos FPGA a procesos de automatización tiene sus razones en la capacidad del dispositivo en atender de forma paralela distintos procesos. El artículo presenta el diseño, implementación y validación de una plataforma de bajo costo para control de un motor CD (corriente directa) basado en FPGA. La plataforma permite desarrollar de control de velocidad y posición con controladores PID (Proporcional Integral Derivativo) industriales. En este trabajo se abordan la identificación y diseño controlador a través de un código libre

desarrollado en Python. La co-simulación se realiza por medio del software MATLAB®.

Palabras Clave: Código libre, Co-simulación, FPGA, Modelación PID.

Abstract

The FPGA (Field Programmable Gate Array) are devices with a large number of programmable logic elements, using language hardware description (HDL) it is possible to develop applications of industrial level. The availability of these in the market have made their use proliferate in different areas of industry, aeronautics, robotics, automation. FPGA devices using HDL are now hardware with high processing speed. The inclusion of FPGA devices to automation processes has its reasons in the ability of the device to attend different processes in parallel. This paper presents the design, implementation and validation of a low cost platform for the control of a dc (direct current) motor based on FPGA. The platform allows the development of speed and position control with industrial PID (Proportional Integral Derivative) controllers. In this work the identification, control design and simulation were programming into MATLAB® software.

Keywords: Co-simulation, FPGA, Modelling Open source, PID.

1. Introducción

La selección del tipo de dispositivo para la implementación de un servocontrolador depende de los requerimientos del sistema, características de la aplicación y una relación entre costo-rendimiento (Prado, 2017), (Rodriguez, 2010), (Rivera, 2012) y (Patrascoiu, 2017). Los microcontroladores estándar son los más adecuados para aplicaciones con requerimientos relativamente bajos de control en tiempo real y que además requieren desarrollar otras tareas como el control de interfaces de usuario. Los DSP son empleados cuando los requerimientos del algoritmo de control en tiempo real son mayores. Sin embargo, existen algunos problemas para los cuales el empleo de microprocesadores, microcontroladores y DSP no es suficiente, donde la solución requiere una gran cantidad de entradas y salidas a alta velocidad, cuando se requiere hacer

procesamiento paralelo. La lógica reconfigurable ha ganado relevancia en implementación de aplicaciones en los campos de robótica, automatización, comunicaciones, biomédica, por mencionar algunas. Debido a que tienen ventajas sobre los microprocesadores y DSP, su arquitectura abierta los hace adecuados para trabajar con procesos donde la necesidad de los sistemas es el control en tiempo real, el alto rendimiento o procesamiento paralelo. Los FPGA's permiten un diseño flexible que puede ejecutar un mayor número de operaciones por ciclo de reloj gracias a su paralelismo y a que no posee una arquitectura específica (Prado, 2017), (Rodríguez, 2010), (Rivera, 2012), (Patrascoiu, 2017) y (Rodríguez, 2011). El uso de la lógica reconfigurable no es novedoso en el estado del arte del control de movimiento. Una de las primeras obras en este campo describe la implementación en un dispositivo FPGA de un contador de cuadratura para el control de posición de un servomotor (Prado, 2017). Otro trabajo de investigación trata sobre la implementación del control del vector espacial PWM para un inversor (Rodríguez, 2010); los autores también describen una arquitectura digital mixta compuesta por un DSP y un FPGA. En la actualidad se han incrementado los diferentes enfoques en el control de movimiento digital, como los que están enfocados en lograr un menor consumo de lógica con un ancho de banda mayor (Prado, 2017).

En este trabajo se presenta el diseño, implementación y validación de una plataforma para el control de motores cd implementado en FPGA. El diseño es realizado con el uso de Verilog y un FPGA de la familia Lattice con el uso de la herramienta APIO, que es una herramienta multiplataforma open source. El sistema es validado experimentalmente a través de la implementación del control de posición de un motor CD. Se realiza una descripción de los métodos y materiales utilizados para el desarrollo del sistema. Se presentan resultados experimentales del sistema y finalmente las conclusiones.

2. Métodos

Modelo dinámico del motor CD

El modelo dinámico de un motor de CD depende de las características

eléctricas y mecánicas, como resistencia R , inductancia L e inercia J de armadura, la fuerza contra-electromotriz e_a presente en la parte eléctrica y mecánica y la fricción. Mencionadas las características anteriores se considera la figura 1 para la obtención del modelo (Hernández, 2013).

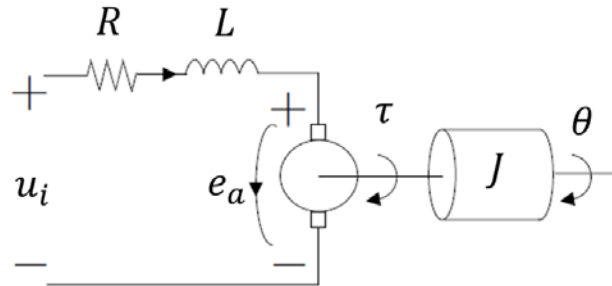


Figura 1 Diagrama Eléctrico-mecánico de un motor de CD.

La dinámica eléctrica y mecánica se muestra en las ecuaciones 1, 2, 3 y 4.

$$u_i = L \frac{d}{dt} i + Ri + e_a \quad (1)$$

$$e_a = k_b \frac{d}{dt} \theta \quad (2)$$

$$\tau = k_m i \quad (3)$$

$$\tau = B \frac{d}{dt} \theta + J \frac{d^2}{dt^2} \theta \quad (4)$$

Aplicando Transformada de Laplace a las ecuaciones 1, 2, 3 y 4 se obtienen las ecuaciones 5, 6, 7 y 8.

$$U_i = LIs + RI + E_a \quad (5)$$

$$E_a = k_b \Theta s \quad (6)$$

$$\tau = k_m I \quad (7)$$

$$\tau = B\Theta s + J\Theta s^2 \quad (8)$$

Resolviendo la función de transferencia de las ecuaciones 5, 6, 7 y 8 se tiene la relación eléctrica de la ecuación 9, la relación electromecánica en las ecuaciones

10 y 11, y la relación mecánica en la ecuación 12.

$$\frac{I}{U_i - E_a} = \frac{1}{Ls + R} \quad (9)$$

$$\frac{E_a}{\Theta} = k_b s \quad (10)$$

$$\frac{\tau}{I} = k_m \quad (11)$$

$$\frac{\Theta}{\tau} = \frac{1}{s(Js + B)} \quad (12)$$

Se construye a partir de las ecuaciones 10, 11 y 12 el diagrama a bloques de la figura 2.

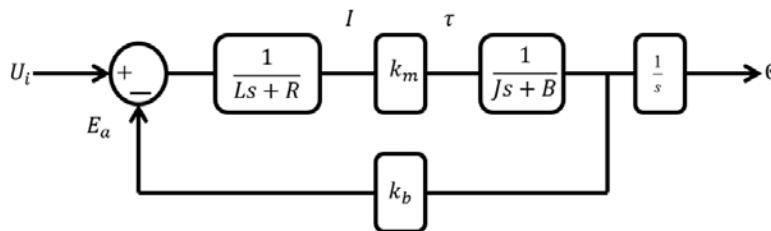


Figura 2 Diagrama a bloques del motor de CD.

La respuesta transitoria y de estado estable puede ser obtenida mediante la aplicación de la señal escalón unitario se muestra en la figura 3.

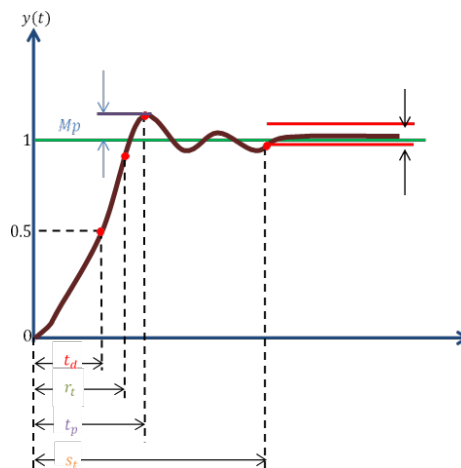


Figura 3 Respuesta transitoria de un sistema de segundo orden ante una entrada escalón.

El sobrepaso máximo M_P es un parámetro a limitar pues se relaciona con la estabilidad del sistema, un sistema con un sobrepaso alto es indeseable, con fines de diseño el sobrepaso suele darse como una especificación en el dominio del tiempo.

Tiempo de retardo t_d se define como el tiempo requerido para que la respuesta escalón alcance el 50% de su valor final.

Tiempo de levantamiento t_r se define como el tiempo requerido para que la respuesta al escalón avance del 10 al 90% de su valor final. Una medida alternativa es representar el tiempo de levantamiento como recíproco de la pendiente de la respuesta al escalón en el instante que la respuesta es igual al 50% de su valor final.

Tiempo de asentamiento t_s es el tiempo requerido para que la respuesta al escalón disminuya y permanezca dentro de un porcentaje específico de su valor final. Una cifra de uso frecuente es 5%.

Identificación de sistemas mediante el Método de Mínimos Cuadrados (MMC)

El MMC es uno de los enfoques básicos y los métodos más utilizados en la reciente teoría y práctica de la identificación de procesos. El MMC tiene una serie de propiedades estadísticas muy convenientes, se puede encontrar una forma recursiva de este método bastante simple. En el esquema de la figura 4, una planta se presenta en forma simplificada con una entrada $U[k]$, una salida $Y[k]$, con ruido aditivo $V[k]$ en el dominio discreto.

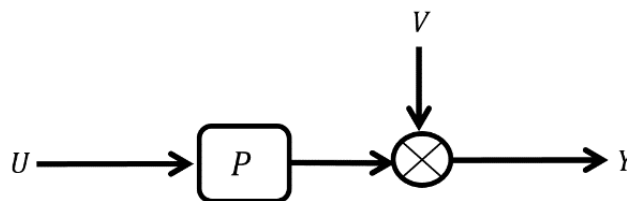


Figura 4 Diagrama a bloques de la representación de una planta en el modelo MMC.

La entrada u y la salida y deben ser accesibles, es decir, pueden medirse con un período de muestreo de tiempo T_s . Independientemente del hecho de que el

sistema original P puede ser continuo, es posible describir la dinámica del comportamiento de un conjunto de mediciones de su entrada y su salida, tomando durante un cierto número de intervalos de tiempo (Johnson, 2006), (Lomeli, 2017), (Abdullah, 2015) y (Tang, 2017).

El modelo discreto definido en la ecuación 13 se conoce como el modelo ARX (auto-regresivo y variable exógena) y consta de dos partes fundamentales: una autorregresión de los valores previos de la salida hasta $y[k - n]$, así como una suma de valores previos de la entrada hasta $u[k - n]$ y el proceso $e[k]$ puede asociarse al ruido presentado en el sistema, se lo denominará ruido modelo o ruido blanco y las propiedades estadísticas se atribuyen como media cero y varianza no correlacionada con la entrada o salida del proceso.

$$y[k] = \sum_{j=1}^n a_j y(k - j) + \sum_{j=0}^n b_j u(k - j) + e[k] \quad (13)$$

El modelo de la ecuación 13 también se puede escribir en la forma:

$$y[k] = P^T z[k] + e[k] \quad (14)$$

Donde P^T and $z[k]$ son, respectivamente, los vectores de los parámetros y de las medidas, ecuaciones 15 y 16.

$$P^T = [b_0 \quad a_1 \quad b_1 \quad \dots \quad a_n \quad b_n] \quad (15)$$

$$z^T = [u[k] \quad y[k - 1] \quad u[k - 1] \quad \dots \quad y[k - n] \quad u[k - n]] \quad (16)$$

P^T y z^T son vectores de dimensión $2n + 1$.

A continuación, se considera el modelo ARX de un sistema de una entrada y una salida para mediciones t , ecuaciones 17 y 18.

$$Y^T(t) = P^T Z^T(t) + E^T(t) \quad (17)$$

$$E^T(t) = Y^T(t) - P^T Z^T(t) \quad (18)$$

La estimación de los parámetros contenidos en el vector P^T , utilizando el MMC, consiste en hallar un estimador P tal que minimice la forma cuadrática J_c , ecuaciones 19, 20 y 21.

$$J_c = E^T(t)E(t) = \sum_{k=1}^t e^2 \quad (19)$$

$$J_c = Y^T Y(t) - 2P^T Z^T(t)Y(t) + P^T Z^T(t)Z(t)P \quad (20)$$

$$\frac{\delta}{\delta P^T} J_c = -2Z^T(t)Y(t) + 2Z^T(t)Z(t)P = 0 \quad (21)$$

La expresión del estimador cuadrático mínimo tiene la forma de la ecuación 22.

$$P = [Z^T(t)Z(t)]^{-1} + Z^T(t)Y(t) \quad (22)$$

La condición de excitación persistente garantiza que la matriz $Z(t)$ es de rango completo y asegura que la estimación de los parámetros del modelo por MMC es estadísticamente consistente.

Controlador PID

Su robustez y alto desempeño en la mayoría de las plantas sean o no lineales, es la principal razón por la que el controlador PID es ampliamente utilizado en la industria. Además, sus parámetros se pueden ajustar empíricamente y lograr un buen rendimiento. A continuación, se presenta la ecuación del PID (ecuación 23) en tiempo continuo, dominio de Laplace (ecuación 24) y dominio discreto respectivamente (ecuación 25).

$$u_c(t) = k_p e_r(t) + k_d \frac{d}{dt} e_r(t) + k_i \int_0^t e_r(\tau) d\tau \quad (23)$$

$$U_c(s) = \left(k_p + k_d s + \frac{k_i}{s} \right) E_r(s) \quad (24)$$

$$U_c[z] = (a_0 + a_1 z^{-1} + a_2 z^{-2}) E_r[z] + z^{-1} U_c[z] \quad (25)$$

Donde $e_r(t)$ es el error, la diferencia entre el valor deseado y la respuesta medida del sistema, $u_c(t)$ es la señal de control, utilizada para controlar la operación del actuador y así lograr el rendimiento deseado en lazo cerrado. Finalmente k_p es la ganancia proporcional, k_i es la ganancia integral y k_d es la ganancia derivativa, T_s es el tiempo de muestreo.

Donde, los coeficientes están definidos por ecuaciones 26, 27 y 28.

$$a_0 = k_p + T_s k_i + \frac{k_d}{T_s} \quad (26)$$

$$a = -k_p - 2 \frac{k_d}{T_s} \quad (27)$$

$$a_2 = \frac{k_d}{T_s} \quad (28)$$

Implementación en FPGA

El FPGA se encarga de realizar la medición de la posición de la flecha del motor, calcular el error, realizar los cálculos del módulo PID y aplicar la señal de control a través de un módulo PWM (Pulse-Width Modulation). La medición de la posición de la flecha del motor la realiza a través de un encoder incremental con señal de cuadratura, cuya resolución es de 500 PPR (Pulsos Por Revolución). La señal de cuadratura es monitoreada cada 20 ns por el FPGA y se almacena en un registro cada 5 ms. La lectura del encoder, se realizó por medio de un módulo desarrollado en el FPGA, en la figura 5 se muestra la lógica para obtener la dirección y la señal de control para el contador de la posición de la flecha del motor.

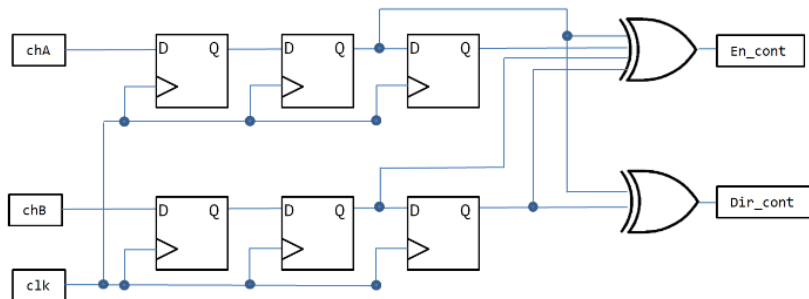


Figura 5 Módulo de lectura del encoder implementado en FPGA.

La implementación del módulo PID en FPGA está sujeta a la ecuación 25, por lo tanto, el número y las características de los multiplicadores incorporados o segmentos DSP que posee el dispositivo es un aspecto muy importante a considerar. La mayoría de los FPGA's disponibles en el mercado tienen

multiplicadores con un ancho de bus fijo, 18x18 bits. Si se utiliza un ancho de bus personalizado, la herramienta de síntesis debe usar recursos lógicos para construir un multiplicador personalizado en lugar de usar los disponibles en el hardware. Por lo tanto, la señal de error y las ganancias del controlador deben tener un ancho de bus que coincida con el ancho del bus de los multiplicadores incorporados disponibles. Para este caso se cuenta con multiplicadores de 16x16 bits. La señal del error tiene un formato en punto fijo de 20.16 y las ganancias tienen un formato en punto fijo de 16.16.

El módulo PID visto como una caja negra se representa en la figura 6. La señal *ER* representa la señal de error, que es la diferencia entre el punto de referencia y los datos de retroalimentación.

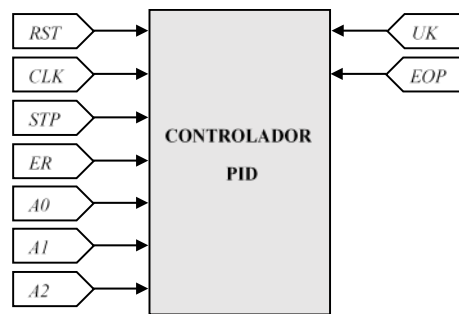


Figura 6 Módulo PID implementado en FPGA.

De manera similar, las señales *A0*, *A1* y *A2* son las ganancias del controlador, y su valor depende de k_p , k_i , k_d y T_s como se explica en las secciones anteriores. Por otro lado, la señal de *UK* es la salida del filtro, que se envía hacia delante al actuador. *CLK*, *RST*, *STP* y *EOP* son las señales maestras de reloj y reinicio maestro, inicio de proceso y final de proceso, respectivamente.

3. Resultados

En la obtención del modelo del motor CD se puede despreciar la inductancia debido a que es un valor generalmente muy pequeño, es decir, su ausencia no afecta en gran medida el comportamiento de la planta. La función de transferencia de la figura 2 está dada por ecuaciones 29 y 30.

$$\frac{U_i}{\theta} = \frac{c_k}{s + c_p} \frac{1}{s} \quad (29)$$

$$\frac{U_i}{\Omega} = \frac{c_k}{s + c_p} \quad (30)$$

$$c_k = \frac{k_m}{JR}; \quad c_p = \frac{RB + k_m k_b}{JR}$$

Se utilizó el MMC con entrada en voltaje y salida en velocidad angular de acuerdo con la ecuación 30, con 3 entradas escalón de 16, 18 y 20 V que fueron cambiadas cada 500 ms. En la figura 7 se muestra en color azul la entrada y en color naranja se muestra la respuesta del sistema, para poder apreciar que el sistema sigue a la entrada se realizó un escalamiento a la salida de $\frac{20}{800}$.

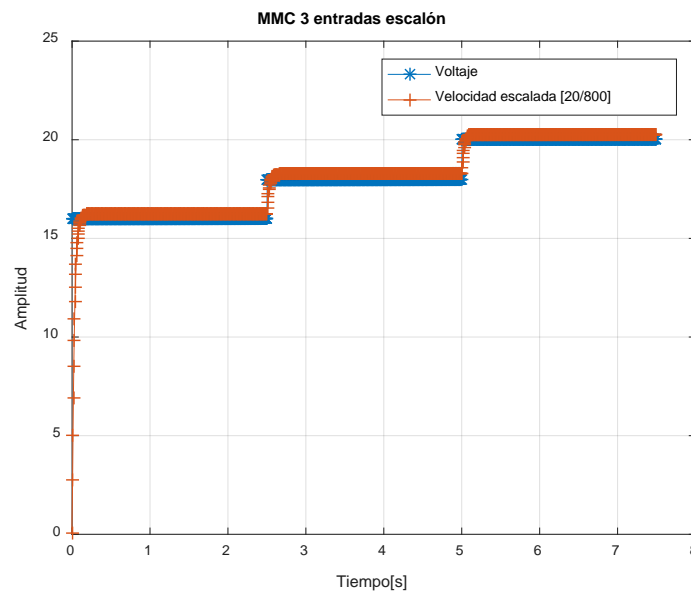


Figura 7 MMC aplicado al motor CD.

Los resultados arrojados por el método son:

$$P^T = [0 \quad 0.83066 \quad 6.85776]$$

Donde:

$$e^{-c_p T_s} = 0.8366$$

$$\frac{c_k}{c_p} (1 - e^{-c_p T_s}) = 6.85776$$

Se obtiene:

$$c_k = 1502.7; c_p = 37.1$$

En la figura 8 se muestra la co-simulación realizada entre el software Modelsim y la herramienta Simulink de MATLAB®. Donde a_0 , a_1 y a_2 son las ganancias del controlador PID discreto, los bloques de conversión de datos convierten los datos de punto decimal a binario en punto fijo con formato 16.16.

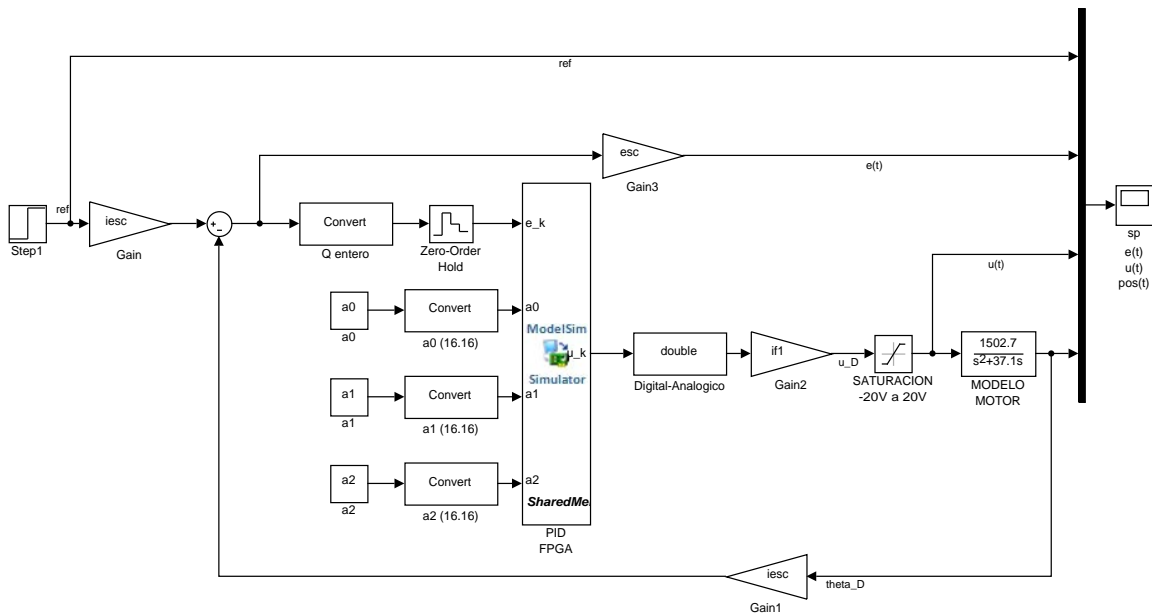


Figura 8 Diagrama a bloques de la co-simulación del PID implementado en FPGA.

En la figura 9 se muestra los resultados de la simulación del control PID aplicado al motor CD, del cual se obtuvieron los parámetros por medio de MMC.

En la tabla 1 se muestran los recursos utilizados del dispositivo FPGA ICE40UP5 de la familia Lattice, donde se implementó el controlador PID, módulo PWM, módulo de lectura del encoder y comunicación serial. Los recursos utilizados fueron mínimos y a partir de la co-simulación se observa que el controlador PID responde de manera estable y puede utilizarse para cualquier otra planta.

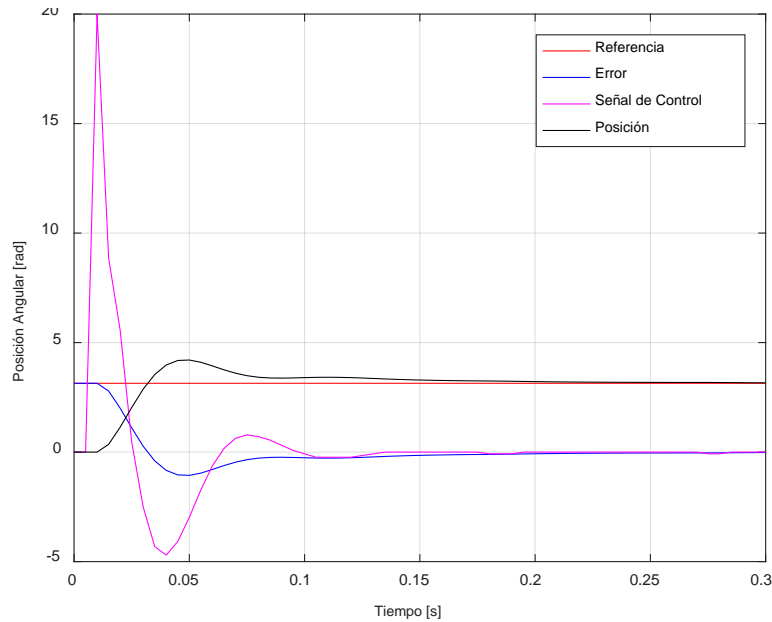


Figura 9 Respuesta de la co-simulación del PID implementado en FPGA.

Tabla 1 Recursos lógicos utilizados de ICE40UP5.

Descripción	Utilizado	Disponibles	Porcentaje
Número de Slice Flip Flops	177	5280	3%
Número de LUTs	283	5280	5%
Número de IOs	8	39	21%
Número de MULT18X18s	3	8	38%
Número de GCLKs	1	6	16%

4. Discusión

La arquitectura propuesta y desarrollo modular ha sido implementada en un FPGA ICE40UP5, de la familia Lattice. El sistema puede adecuarse para la implementación de estrategias de control futuros, tales como los propuestos anteriormente o algoritmos más robustos, que se adecuen a sistemas de control mediante hardware reconfigurable e inclusive extrapolar a sistemas de más de una entrada y una salida, para el presente caso se ha empleado un algoritmo para la identificación de parámetros y control de posición de un motor de corriente directa, que puede ser adaptado fácilmente a otras interfaces para la comunicación y control de sistemas. Este sistema se convierte en una base que posibilita la implementación de diferentes topologías basadas en el esquema convencional

PID debido a la modularidad del diseño y la independencia de una arquitectura específica. Es una plataforma de control de bajo costo, y permite la implementación con el uso de herramientas de software libre como APIO desarrollada por el proyecto ICESTORM (González, 2018).

5. Bibliografía y Referencias

- [1] Prado, M. A. M., Reséndiz, J. R., Pérez, D. C. T., Hernández, C. M. T., & Ruiz, G. H. (2017). Motion Control with FPGA. In Field-Programmable Gate Array. InTech.
- [2] Rodriguez-Resendiz, J., Rivas-Araiza, E. A., Herrera-Ruiz, G., & Gonzalez-Gutierrez, C. A. (2010, December). Virtual instrumentation for analysis of an adjustable speed drive parameters based on DSC. In Education and Research Conference (EDERC), 2010 4th European (pp. 195-199). IEEE.
- [3] Rivera-Guillen, J. R., de Jesus Rangel-Magdaleno, J., de Jesus Romero-Troncoso, R., Osornio-Rios, R. A., & Guevara-Gonzalez, R. G. (2012). An open-access educational tool for teaching motion dynamics in multi-axis servomotor control. *IEEE Transactions on Education*, 55(2), 218-225.
- [4] Patrascioiu, N. (2017, June). Identification of parameters for dynamic regime of DC motor through virtual instrumentation. In Engineering of Modern Electric Systems (EMES), 2017 14th International Conference on (pp. 192-195). IEEE.
- [5] Rodriguez-Resendiz, J., Herrera-Ruiz, G., & Rivas-Araiza, E. A. (2011). Adjustable speed drive project for teaching a servo systems course laboratory. *IEEE Transactions on Education*, 54(4), 657-666.
- [6] Hernandez-Guzman, V. M., Silva-Ortigoza, R., & Carrillo-Serrano, R. V. Control Automatico: Teoria de Diseno, Construccion de Prototipos, Modelado, Identificacion y Pruebas Experimentales, Coleccion CIDETEC-IPN, Mexico City, Mexico, 2013.
- [7] Pati, N., & Swain, N. (2017, December). Design and study of speed control of DC motor using Youla parameterization and PID controller. In Calcutta Conference (CALCON), 2017 IEEE (pp. 433-437). IEEE.

- [8] Sahputro, S. D., Fadilah, F., Wicaksono, N. A., & Yusivar, F. (2017, July). Design and implementation of adaptive PID controller for speed control of DC motor. In *Quality in Research (QiR): International Symposium on Electrical and Computer Engineering, 2017 15th International Conference on* (pp. 179-183). IEEE.
- [9] Burzo, E. (2010). *Pid Control: New Identification and Design Methods*. Springer.
- [10] G. R. Lomeli, "Modelado e Identificación de Parámetros," 2017.
- [11] Abdullah, S. M., Yassin, I. M., & Tahir, N. M. (2015, April). Comparison between PSO, NE, QR, SVD methods for least squares DC motor identification. In *Computer Applications & Industrial Electronics (ISCAIE), 2015 IEEE Symposium on* (pp. 105-112). IEEE.
- [12] Tang, W. J., Liu, Z. T., & Wang, Q. (2017, July). Dc motor speed control based on system identification and pid auto tuning. In *Control Conference (CCC), 2017 36th Chinese* (pp. 6420-6423). IEEE.
- [13] J. González-Gómez. (2018, Sep). FPGAWARS: <http://FPGAwards.github.io/>.