

Low Cost and Optimized FPGA-HIL Real Time Simulation of a Boost Converter Powered by a Photovoltaic Panel

Rodolfo Orosco Guerrero , Elías Rodríguez Segura , Juan Martínez Nolasco , and Fany Rodríguez García 

Abstract— This paper presents the development, analysis and testing of a dynamic model for a boost converter powered by a photovoltaic panel. The mathematical model of the panel is presented, and a strategy is proposed to integrate its nonlinear dynamics into the dynamic model of the boost converter. Emphasis is placed on the dynamic behavior, leaving the modeling of the effects of parasitic losses inherent to electronic devices for a future study. A discrete model that is feasible to implement in an FPGA to achieve HIL simulation in real time is shown. An optimized strategy of implementation of nonlinear functions implemented in lookup tables is presented. That strategy allows optimizing the resolution without affecting the calculation latency. A latency analysis of the calculation of the model states is also included where a time of 100ns is established for each simulation step. This latency time is sufficient for real-time simulations. Finally, comparisons of results with a software-based simulator to validates the proposed model are shown.

Link to graphical and video abstracts, and to code: <https://latam.ieceer9.org/index.php/transactions/article/view/8678>

Index Terms— Hardware in the loop (HIL), Modelling & Simulation, Boost Converter, Photovoltaic Panel, Low-Cost.

I. INTRODUCCIÓN

La simulación HIL es ampliamente usada en diversas industrias como son: automotriz, aeronáutica, robótica y militar [1]. El uso de esta simulación ha sido un gran avance en el desarrollo de pruebas de nuevos productos. La electrónica de potencia no ha sido ajena al uso de este tipo de simulación [2]-[6]. Algunas de las ventajas del HIL son: permite realizar pruebas en sistemas mixtos, flexibilidad y capacidad de depuración, reducir notablemente el tiempo y costo de la fase de pruebas y ahorro en equipamiento e instalaciones de pruebas.

En la implementación de simulaciones HIL de convertidores de potencia aplicados en sistemas fotovoltaicos

se ha utilizado la plataforma de Xilinx® System Generator en conjunto con Simulink® de Matlab® obteniendo buenos resultados de optimización [7]. También se ha utilizado LabVIEW® de National Instruments® en la implementación de simulaciones HIL de convertidores de potencia con aplicación de vehículos híbridos [8] y [9]. De acuerdo con [10] utilizar LabVIEW® para implementar simulaciones HIL tiene las siguientes ventajas para ayudar a describir hardware:

- Sistema de desarrollo gráfico: El desarrollo tradicional para sistemas basados en FPGA requiere el uso de herramientas de software de bajo nivel y lenguajes de descripción de hardware (HDLs, por sus siglas en inglés). Aprender y usar de manera efectiva un HDL puede ser un proceso tedioso y tardado. LabVIEW® FPGA ofrece un enfoque de programación gráfica que simplifica la tarea de conectar a E/S y comunicar datos, mejorando enormemente la productividad del diseño y reduciendo el tiempo al mercado.
- Reutilización de código HDL y bibliotecas IPs: Para impulsar su productividad de desarrollo de software al diseñar con FPGAs, es imprescindible la reutilización eficiente de código. LabVIEW® FPGA ofrece IP desarrollado por NI® y Xilinx® para funciones básicas como contadores o algoritmos más avanzados como decodificación de video.
- Rápido desarrollo de aplicaciones: LabVIEW® FPGA está completamente equipado con habilidades integradas de simulación y herramientas de depuración, así se puede detectar el mayor número de errores de implementación como sea posible antes de la compilación. Dentro de la simulación, se puede depurar el código con funciones especiales de LabVIEW® básicas como ejecución resaltada, puntos de interrupción y puntas de prueba.

En [11] se presenta una alternativa para hacer más eficiente el cálculo de la corriente y voltaje de un panel fotovoltaico utilizando redes de Petri y el Lenguaje de Modelado de Sistemas (SysML) como lenguaje para programar el FPGA. Los autores aplican el método de Newton-Rapson para resolver el modelo del panel junto con el residuo de Lagrange logrando reducir el número de iteraciones. Además, con el objetivo de reducir la cantidad de recursos utilizados en el FPGA propone utilizar una aproximación basada en la serie de Taylor para el cálculo de los términos exponenciales. Los autores reportan un tiempo de desempeño de 0.96 microsegundos contra 19.2 microsegundo que requiere el método estándar. Cabe resaltar que este trabajo contempla

The associate editor coordinating the review of this manuscript and approving it for publication was Julio C. Rosas-Caro (*Corresponding author: Rodolfo Orosco Guerrero*).

Este trabajo fue financiado por Tecnológico Nacional de México.

R. O. Guerrero, E. R. Segura, J. J. M. Nolasco y F. R. García están con el Instituto Tecnológico de Celaya, Celaya, Guanajuato, México (e-mails: rodolfo.orosco@itcelaya.edu.mx, elias.rodriguez@itcelaya.edu.mx, juan.martinez@itcelaya.edu.mx, y fany.rodriguez@itcelaya.edu.mx).

como entradas del panel la irradiancia y la temperatura, pero no considera su interacción con un convertidor. En [12] se presenta el estudio de los efectos de la irradiancia, la temperatura, la resistencia serie y la resistencia paralela en las características de salida del panel fotovoltaico. La simulación se realizó en Simulink® de MatLab®. Cabe destacar que las simulaciones son muy exhaustivas, sin embargo, no se ejecutan en un FPGA y su objetivo no fue ejecutarlas en tiempo real. En [13] presenta un método para acoplar una simulación de Simulink® de MatLab® con la implementación en el FPGA de una plataforma MyRIO de National Instruments®. Los autores obtuvieron resultados muy buenos en cuanto a tiempo de ejecución, sin embargo, solo contemplan la simulación del panel fotovoltaico enfocándose en resolver rápidamente las ecuaciones no lineales inherentes al modelo. Cabe destacar que este trabajo obtiene buenos resultados en cuanto a eficiencia en tiempo, pero no incluye su acoplamiento con un convertidor. En [14] se presenta la simulación HIL de un panel conectado a un convertidor elevador en una plataforma MyRIO de National Instruments®. Los autores reportan un buen desempeño de la simulación comparado con datos experimentales. Reportan un tiempo de ejecución de 10 microsegundos lo cual fue suficiente para comprobar el desempeño de tres controladores, modificando la irradiancia solar. Cabe mencionar que la simulación incluye un algoritmo para obtener los resultados de la función exponencial inherente al modelo del panel, además contempla la temperatura constante. En [15] se presenta la simulación en una FPGA de un panel conectado a un convertidor elevador en una plataforma NI PCIe 7841R. Para obtener la relación entre el voltaje y la corriente del panel aplican el método de Newton Rapson. Los autores reportan buenos resultados en cuanto a la precisión de la simulación y la cantidad de recursos del FPGA requeridos en la implementación, sin embargo, consideran la temperatura constante y no reportan el tiempo de ejecución.

En esta publicación se muestra el modelado y simulación HIL de una de un convertidor CD-CD elevador alimentado con un panel fotovoltaico enfocándose en lograr un tiempo de simulación pequeño con el menor número de cálculos posible.

II. MODELO DEL PANEL FOTOVOLTAICO

En la Fig. 1 se muestra el circuito equivalente de un panel fotovoltaico que incluye las pérdidas debidas a la resistencia interna del material ante el flujo de corriente (R_s) y las pérdidas por la corriente de fuga (R_{sh}) [16].

Del circuito de la Fig. 1 se desprenden las ecuaciones de nodo (1) y de malla (2):

$$I_G = I_{pvp} + I_d + I_{sh}, \tag{1}$$

donde I_G es la corriente generada por la irradiancia solar sobre el panel, I_{pvp} es la corriente que proporciona el panel, I_d es la corriente de derivación de cada celda del panel e I_{sh} es la corriente de derivación en la resistencia R_{sh} de cada celda.

$$N_s V_D = V_{pvp} + I_{pvp} N_s R_s, \tag{2}$$

donde N_s es el número de celdas del panel conectadas en serie, V_D es el voltaje del diodo por celda, V_{pvp} es el voltaje de salida

en las terminales del panel y R_s es la resistencia serie de cada celda del panel,

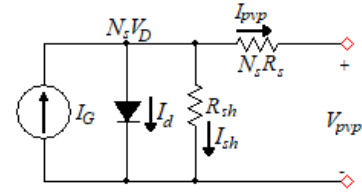


Fig. 1. Circuito equivalente de panel fotovoltaico con N_s celdas en serie.

Las corrientes I_d e I_{sh} dependen del voltaje del diodo V_D como se muestra en (3) y (4):

$$I_d = I_o \left(e^{\frac{N_s V_D}{V_t}} - 1 \right), \tag{3}$$

$$I_{sh} = \frac{N_s V_D}{R_{sh}}, \tag{4}$$

donde I_o es la corriente de saturación inversa de del diodo en cada celda, V_t es el voltaje térmico en los diodos de cada celda del panel y R_{sh} es la resistencia paralela de derivación del diodo en cada celda del panel.

Se definirá la corriente de derivación del panel (I_{dsh}) como:

$$I_{dsh} = I_d + I_{sh} = I_o \left(e^{\frac{N_s V_D}{V_t}} - 1 \right) + \frac{N_s V_D}{R_{sh}}. \tag{5}$$

Para extender la utilidad de este modelo se incluye la variable externa G que representa la radiación solar sobre el panel. Sin embargo, para simplificar el modelo se considerará el efecto de la temperatura como una constante del panel dejando su integración para un trabajo futuro. Así, el efecto de la radiación solar sobre la corriente I_G está definida por (6):

$$I_G = \frac{I_{Gn}}{G_{nom}} G + K_t, \tag{6}$$

donde G_{nom} es la irradiancia bajo condiciones de prueba estándar del panel, I_{Gn} es la corriente generada bajo condiciones de prueba estándar, K_t es el termino de desplazamiento de la corriente I_G debido a la temperatura y G es la irradiancia sobre el panel. Nótese que al considerar la temperatura constante K_t es constante.

III: MODELO DINÁMICO DEL CONVERTIDOR ELEVADOR ALIMENTADO CON UN PANEL FOTOVOLTAICO

Para obtener el modelo dinámico de un convertidor elevador alimentado con un panel fotovoltaico considérese el circuito de la Fig. 2.

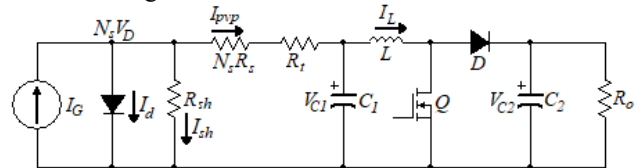


Fig. 2. Circuito de un convertidor elevador alimentado con un Panel fotovoltaico.

El objetivo del modelo es simular el comportamiento dinámico en tiempo real dentro de una FPGA. Por lo tanto, solo se consideran los elementos que dominan su dinámica. Por esta razón, no se incluyen los elementos que representan las pérdidas del sistema como son: las resistencias dinámicas

del transistor Q y el diodo D. Dicho lo anterior, para el análisis se considerarán los subcircuitos de las Figs. 3 y 4 que representan los posibles casos de conducción del transistor Q y el diodo D.

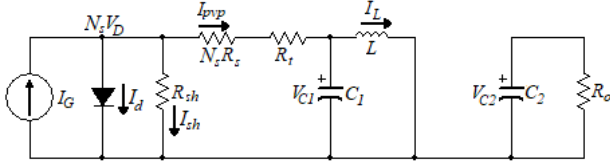


Fig. 3. Subcircuito con transistor Q conduciendo.

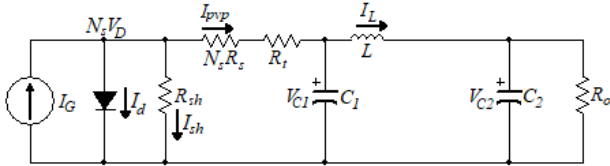


Fig. 4. Subcircuito con transistor Q abierto.

Para modelar la dinámica de los circuitos mostrados en las Figs. 3 y 4 se proponen las siguientes variables de estado: $x_1 = \frac{N_s}{a} V_D$, $x_2 = I_L$ y $x_3 = V_{C2}$; donde a es un factor de normalización, I_L es la corriente que circula por el inductor del convertidor y V_{C2} es el voltaje en el capacitor de salida del convertidor.

Para obtener la dinámica del estado x_1 se consideran: la ecuación de la malla que contiene a R_{sh} , R_s , R_t y C_1 en la Fig. 2, (1), (5) y (6). Despejando para V_{C1} se obtiene:

$$V_{C1} = ax_1 - (N_s R_s + R_t) \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh} \right). \quad (7)$$

donde V_{C1} es el voltaje del capacitor de entrada del convertidor y R_t es la resistencia del cable que conecta el panel con el convertidor.

Derivando (7) se obtiene (La derivada la irradiancia solar G se considera despreciable comparada con la derivada de I_{dsh} debido a que la irradiancia presenta cambios apreciables en cientos de milisegundos e incluso en segundos mientras que los cambios apreciables de I_{dsh} pueden presentarse en microsegundos):

$$\dot{V}_{C1} = \left(a + (N_s R_s + R_t) \frac{\partial I_{dsh}}{\partial x_1} \right) \dot{x}_1. \quad (8)$$

Aplicando Ley de corrientes de Kirchoff en el nodo donde se conectan R_t , V_{C1} e I_L se obtiene:

$$C_1 \dot{V}_{C1} = I_{pvp} - x_2. \quad (9)$$

Sustituyendo (8) en (9) y despejando para \dot{x}_1 se obtiene:

$$\dot{x}_1 = \frac{1}{C_1 (a + (N_s R_s + R_t) \frac{\partial I_{dsh}}{\partial x_1})} \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh} - x_2 \right). \quad (10)$$

Reescribiendo (10) para incluir el comportamiento no lineal de I_{dsh} y su derivada parcial en función x_1 , la dinámica de x_1 estará definida como:

$$\dot{x}_1 = f(x_1) \left[\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1) - x_2 \right], \quad (11)$$

donde:

$$f(x_1) = \frac{1}{a C_1 \left(1 + (N_s R_s + R_t) \left(\frac{I_0}{V_t} e^{\frac{ax_1}{V_t}} + \frac{1}{R_{sh}} \right) \right)}, \quad (12)$$

$$I_{dsh}(x_1) = I_0 \left(e^{\frac{ax_1}{V_t}} - 1 \right) + \frac{ax_1}{R_{sh}}. \quad (13)$$

Nótese que la dinámica para el estado x_1 no está influido directamente por la conmutación de Q y D. Por lo tanto, se puede concluir que la dinámica expresada por (11) es válida para los subcircuitos de las Figs. 3 y 4. En contraste la dinámica de los estados x_2 y x_3 si está influida por la conmutación de Q y D por lo que se hace necesario definir la variable lógica de control u como:

$$u = \begin{cases} 0 & \text{cuando Q conduce y D está abierto.} \\ 1 & \text{cuando Q está abierto y D conduce.} \end{cases}$$

Analizando las mallas que contienen al inductor en los subcircuitos de las Figs. 3 y 4, considerando la variable u se obtiene la siguiente ecuación que expresa la dinámica de la corriente del inductor:

$$\dot{x}_2 = \frac{1}{L} (V_{C1} - ux_3). \quad (14)$$

Sustituyendo (7) en (14) se obtiene:

$$\dot{x}_2 = \frac{1}{L} \left(ax_1 - (N_s R_s + R_t) \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1) \right) - ux_3 \right). \quad (15)$$

Analizando los nodos a los que está conectado el capacitor C_2 en los subcircuitos de las Figs. 3 y 4, considerando la variable u se obtiene la siguiente ecuación que expresa la dinámica del voltaje del capacitor:

$$\dot{x}_3 = \frac{1}{C_2} ux_2 - \frac{1}{R_o C_2} x_3. \quad (16)$$

Agrupando términos la dinámica de x_3 se puede escribir como:

$$\dot{x}_3 = \frac{1}{C_2} (x_2 u - \frac{1}{R_o} x_3), \quad (17)$$

Finalmente, el modelo dinámico de espacio de estado para el circuito de la Fig. 2 queda definido por (11), (15) y (17):

$$\left. \begin{aligned} \dot{x}_1 &= f(x_1) \left[\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1) - x_2 \right] \\ \dot{x}_2 &= \frac{1}{L} \left(ax_1 - (N_s R_s + R_t) \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1) \right) - ux_3 \right) \\ \dot{x}_3 &= \frac{1}{C_2} (x_2 u - \frac{1}{R_o} x_3) \end{aligned} \right\}. \quad (18)$$

Para la implementación del modelo dinámico (18) es necesario considerar algunas restricciones debidas a la naturaleza misma del sistema modelado.

Primero se analizarán las restricciones del voltaje del diodo escalado x_1 . El voltaje del diodo siempre es positivo y además no puede exceder el valor del voltaje de circuito abierto del panel (V_{oc}). Estas restricciones se pueden obtener por análisis directo del circuito mostrado en la Fig. 2. Por lo tanto, se puede definir la siguiente restricción para x_1 :

$$0 \leq x_1 \leq \frac{V_{oc}}{a}. \quad (19)$$

La corriente del inductor I_L y el voltaje V_{C2} también deben ser positivos, esto es:

$$\begin{aligned} 0 &\leq x_2, \\ 0 &\leq x_3. \end{aligned}$$

Aplicando la desigualdad (19) a las funciones $f(x_1)$ e $I_{dsh}(x_1)$ podemos concluir que:

$$\begin{aligned} 0 &< \frac{1}{aC_1 \left(N_s + (N_s R_s + R_t) \left(\frac{I_o}{V_t} e^{\frac{V_{oc}}{N_s V_t} + \frac{1}{R_{sh}}} \right) \right)} \leq f(x_1), \\ f(x_1) &\leq \frac{1}{aC_1 \left(N_s + (N_s R_s + R_t) \left(\frac{I_o}{V_t} + \frac{1}{R_{sh}} \right) \right)}, \\ 0 \leq I_{dsh}(x_1) &\leq I_o \left(e^{\frac{V_{oc}}{N_s V_t}} - 1 \right) + \frac{V_{oc}}{N_s R_{sh}} \approx I_{Gn}. \end{aligned}$$

Otra restricción importante que considerar es que el valor de la radiación solar G es positiva y no puede superar la radiación nominal G_{nom} , es decir,

$$0 \leq G \leq G_{nom}.$$

Es importante considerar todas estas restricciones al momento de implementar el algoritmo de cálculo en el FPGA de manera que genere las saturaciones correspondientes.

IV. DISCRETIZACIÓN DEL MODELO DINÁMICO PARA SU IMPLEMENTACIÓN EN FPGA

Uno de los objetivos de la simulación HIL es lograr un desempeño en tiempo de ejecución mayor que el de los simuladores basados en software y de ser posible lograr que se desempeñen en tiempo real. Con este objetivo en mente se aplicará el método de discretización de Euler hacia adelante para optimizar el uso de recursos del hardware que ejecutará los cálculos. Aplicando esta técnica de discretización al modelo dinámico (18), resulta:

$$\left. \begin{aligned} x_1(k+1) &= x_1(k) \\ &+ T f(x_1(k)) \left[\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1(k)) - x_2(k) \right] \\ x_2(k+1) &= x_2(k) + \frac{T}{L} a x_1(k) - \frac{T}{L} x_3(k) u(k) \\ &- \frac{T}{L} (N_s R_s + R_t) \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1(k)) \right) \\ x_3(k+1) &= x_3(k) + \frac{T}{C_2} \left(x_2(k) u(k) - \frac{1}{R_o} x_3(k) \right). \end{aligned} \right\} (20)$$

Con el propósito de disminuir la latencia de cálculo de cada uno de los estados es necesario reducir el número de sumas y multiplicaciones involucradas en su cálculo. El dispositivo que se utilizará para la implementación de este modelo es una tarjeta de desarrollo MyRIO 1900 de National Instruments. En la documentación del fabricante se establece que es necesario considerar un pulso de reloj por cada suma y/o multiplicación conectada en cascada. Por esta razón se propone modificar el modelo discreto (20) de la siguiente manera:

$$\left. \begin{aligned} x_1(k+1) &= \left(x_1(k) - f_T(x_1(k)) x_2(k) \right) \\ &+ f_T(x_1(k)) \left[\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1(k)) \right] \\ x_2(k+1) &= x_2(k) + \frac{aT}{L} x_1(k) - \frac{T}{L} x_{3u}(k) \\ &- \frac{T(N_s R_s + R_t)}{L} \left(\frac{I_{Gn}}{G_{nom}} G + K_t - I_{dsh}(x_1(k)) \right) \\ x_3(k+1) &= \left(1 - \frac{T}{R_o C_2} \right) x_3(k) + \frac{T}{C_2} x_{2u}(k) \end{aligned} \right\} (21)$$

donde:

$$\begin{aligned} x_{2u}(k) &= \begin{cases} 0 & \text{si } u = 0, \\ x_2 & \text{si } u = 1, \end{cases} \\ x_{3u}(k) &= \begin{cases} 0 & \text{si } u = 0, \\ x_3 & \text{si } u = 1, \end{cases} \\ f_T(x_1(k)) &= T f(x_1(k)). \end{aligned}$$

La implementación en hardware de las funciones $x_{2u}(k)$ y $x_{3u}(k)$ no requiere de pulsos de reloj adicionales ya que es posible implementarlos con un circuito combinatorial haciendo pasar el valor $x_2(k)$ y $x_3(k)$ con el valor de u por una máscara AND.

A. Optimización Espacio-resolución de las Tablas de Búsqueda

Para la implementación en hardware de las funciones no lineales $f_T(x_1(k))$ e $I_{dsh}(x_1(k))$ se utilizarán tablas de búsqueda unidimensionales. En la documentación del fabricante de la tarjeta MyRIO se establece que las Tablas de búsqueda unidimensionales requieren de un pulso de reloj para ejecutarse y que utilizan bloques de memoria del FPGA para alojarse. Tomando esto en consideración, a continuación, se describirán las estrategias utilizadas para optimizar la latencia y la resolución de la tabla de búsqueda.

La constante de escalamiento a , propuesta en la definición del estado x_1 , tiene el propósito de optimizar la resolución de la tabla de búsqueda donde se implementarán las funciones no lineales. Para lograrlo, se debe considerar que el acceso a los bloques de memoria, que son utilizados para alojar los resultados de las funciones no lineales para cada valor de x_1 , solo pueden definirse en capacidades establecidas por potencias de 2. En este caso, se desea alojar las funciones no lineales en bloques de memoria de 2^{14} y que entregue el resultado codificado en un valor de 32 bits, esto significa que se utilizará un bloque de memoria de 64KB. Como se puede observar la resolución máxima equivale a la capacidad del bloque de memoria. Por lo tanto, si se considera una codificación IQ(14,9) para x_1 , para optimizar la resolución de la tabla de búsqueda se debe lograr que:

$$0 \leq x_1 \leq \frac{2^{14}-1}{2^5}. \quad (22)$$

Aplicando la desigualdad (19), para lograr la restricción de la desigualdad (22) entonces:

$$a = \frac{2^5 V_{oc}}{(2^{14}-1)}. \quad (23)$$

B. Análisis de Latencia del Modelo Propuesto

En la Tabla 1 se muestra el análisis de latencia para el cálculo de las variables de estado considerando el modelo

discreto (20).

Como se puede observar en la Tabla I para calcular los estados siguientes se utilizan tres operaciones aritméticas y dos evaluaciones de funciones no lineales. Cada operación aritmética requiere de un pulso de reloj para ejecutarse. También la evaluación de las funciones no lineales implementadas en tablas de búsqueda requiere de un pulso de

reloj para entregar el resultado. De acuerdo con la Tabla 1 se requiere de 4 pulsos de reloj para calcular los estados x_1 y x_2 , y 2 para el cálculo de x_3 . Si consideramos que la tarjeta MyRIO tiene un reloj de 40 MHz en su FPGA se puede concluir que la latencia del cálculo será de 100ns.

TABLA I
ANÁLISIS DE LATENCIA EN EL CÁLCULO DE LOS ESTADOS DEL MODELO

Estado calculado	Pulsos de reloj necesarios para ejecutar el cálculo			
	1	2	3	4
$x_1(k+1)$	$I_G = \frac{I_{Gn}}{G_{nom}}G + K_t$ $a_1 = I_{dsh}(x_1(k))$ $a_2 = f_T(x_1(k))$ $b_1 = \frac{aT}{L}x_1(k)$	$I_{pvp} = I_G - a_1$ $a_3 = a_2 * x_2(k)$ $b_3 = b_1 - b_2$	$a_4 = a_2 * I_{pvp}$ $a_5 = x_1(k) - a_3$ $b_4 = b_3 + x_2(k)$	$a_4 + a_5$
$x_2(k+1)$	$b_2 = \frac{T}{L}x_{3u}(k)$		$b_5 = \frac{T(N_s R_s + R_t)}{L}I_{pvp}$	$b_4 - b_5$
$x_3(k+1)$	$c_1 = \left(1 - \frac{T}{R_o C_2}\right) * x_3(k)$ $c_2 = \frac{T}{C_2} * x_{2u}$	$c_1 + c_2$		

VI. RESULTADOS

Para validar el modelo propuesto se aplicaron tres pruebas en lazo abierto comparando los resultados de la simulación HIL en tiempo real con una simulación clásica de este convertidor en Simulink® de MatLab®. Las pruebas consisten en aplicar una señal PWM en la variable u con un ciclo de trabajo de 50%, 75% y 100%. Nótese que estas pruebas serían equivalentes a un PWM aplicado directamente al transistor Q con ciclo de trabajo del 50%, 25% y 0% respectivamente. Se considera el caso de $u = 100\%$ como caso de control que permite observar el transitorio y estado estable esperado del modelo. La simulación de Simulink se realizó en un PC con Windows 10, procesador AMD A6-7310 a 2 GHz. La simulación HIL en tiempo real se ejecutó en una tarjeta de desarrollo MyRIO 1900. Se ejecutaron simulaciones con duración de 100ms. Se obtuvieron los resultados de la corriente del inductor y los voltajes de los capacitores uno y dos. La duración de la simulación en Simulink® se estableció entre 10 y 13 segundos, siendo la de mayor duración cuando se simuló el ciclo de trabajo de “u” en 50%.

El panel fotovoltaico que se consideró para la simulación fue un arreglo de 10 módulos conectados en serie de la marca ReneSola® modelo Virtus II de 250 W. Este panel está compuesto por 60 celdas policristalinas de 156x156 mm conectadas en serie en un arreglo de 6x10. En la Tabla II se muestran las características eléctricas que proporciona el fabricante considerando una irradiancia de 1000 W/m² a una temperatura de 25 °C.

Ingresando los valores de la Tabla 2 en la ventana de

configuración del “Arreglo PV” (PV Array) que se utilizó para las simulaciones en Simulink® MatLab®, se obtienen los parámetros definidos en la Tabla III.

Por otra parte, los parámetros del convertidor utilizados en estas simulaciones se muestran en la Tabla IV.

Finalmente, la carga considerada (R_o) fue de 366 Ω, la irradiancia G de 500 W/m², la temperatura de 25 °C y la frecuencia de la señal PWM aplicada fue de 50 KHz.

TABLA II
CARACTERÍSTICAS ELÉCTRICAS DEL PANEL FOTOVOLTAICO DE 250 W VIRTUS II DE RENESOLA®

P_{max}	I_{mp}	V_{mp}	I_{sc}	V_{oc}
250 W	8.45 A	30 V	9.04 A	37.5 V

TABLA III
PARÁMETROS DEL MODELO MATEMÁTICO DEL PANEL FOTOVOLTAICO

R_{sh}	R_s	V_t	I_0	N_s	I_{Gn}
200.32 Ω	0.3832 Ω	1.42 V	30.295 pA	60	9.06 A

TABLA IV
PARÁMETROS DEL CONVERTIDOR ELEVADOR

R_t	C_1	C_2	I_L
0.22 Ω	82 μF	150 μF	16 mH

En la gráfica superior de las Figs. 5 a la 13 se muestra la comparación entre los resultados de las simulaciones de Simulink® (rojo) y el HIL en tiempo real (azul). Se puede observar que los valores obtenidos en ambas simulaciones son

muy cercanos por lo que se muestran dos acercamientos en diferentes tiempos donde se puede apreciar la diferencia. En la gráfica inferior de las Figuras se muestra el error entre ambas simulaciones.

Las Figs. 5 a la 7 muestran los resultados de la corriente del inductor para un ciclo de trabajo de “u” en 100%, 75% y 50%. Conforme el ciclo de trabajo disminuye se puede observar un aumento en el error absoluto de corriente llegando a ser de casi +/- 15 mA con 50% de ciclo de trabajo. Sin embargo, en los acercamientos se puede observar que ese error es producido por el desfaseamiento entre los picos del rizo de corriente y que en ningún momento se hace mayor a dicho rizo. En los acercamientos también se puede observar que los picos de rizo son ligeramente mayores en la simulación HIL en tiempo real, esto se debe a la menor precisión en los cálculos de punto fijo realizados en el FPGA que la que se obtiene en la simulación de Simulink® con punto flotante.

El error absoluto promedio en estado estable de la simulación al 100% es de -76.1 μ A, al 75% es de 204.62 μ A y al 50% es de 669.93 μ A; si consideramos que la corriente promedio en estado estable es de 681.6 mA al 100%, de 1.2 A al 75% y de 2.58 A al 50% encontramos que los errores relativos correspondientes son de -0.011%, 0.017% y 0.026%. Todos los errores relativos son prácticamente nulos.

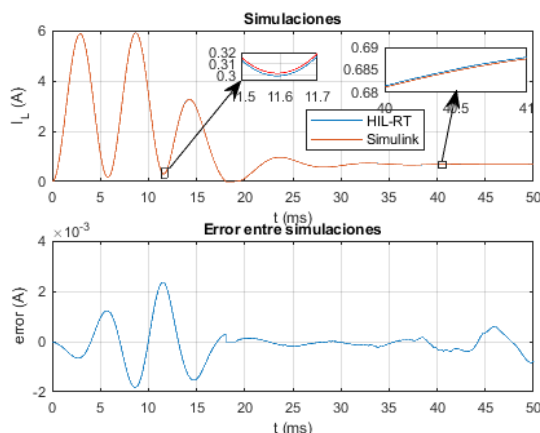


Fig. 5. Resultados de la simulación de I_L en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 100%.

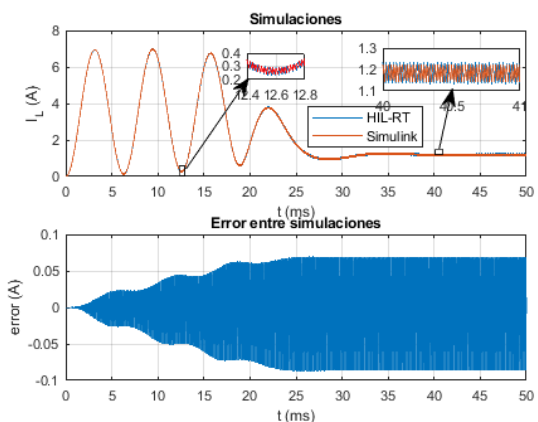


Fig. 6. Resultados de la simulación de I_L en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 75%.

Las Figs. 8 a la 10 muestran los resultados del voltaje del capacitor uno para un ciclo de trabajo de 100%, 75% y 50%.

Conforme el ciclo de trabajo disminuye se puede observar un aumento en el error absoluto de voltaje llegando a ser de casi +380 mV con 50% de ciclo de trabajo. También se observa un error de estado estable que es mayor conforme disminuye el ciclo de trabajo llegando a 98.8 mV con 50% de ciclo de trabajo. Estos errores de estado estable se deben a la comparativamente baja resolución y precisión del valor que entregan las tablas de búsqueda que simulan al panel fotovoltaico. Además, aunque los errores absolutos promedio en estado estable son de 32.9 mV a 100%, 46.5 mV a 75% y 98.8 mV a 50% los errores relativos son de 0.013%, 0.018% y 0.042% respectivamente.

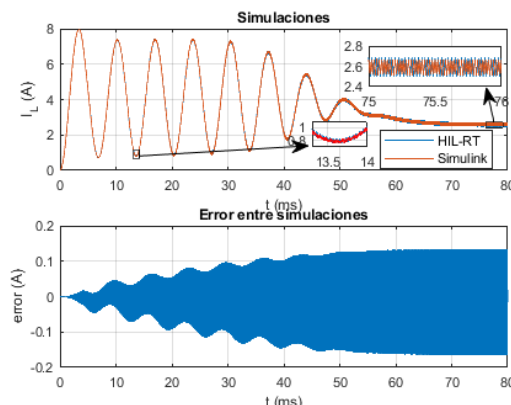


Fig. 7. Resultados de la simulación de I_L en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 50%.

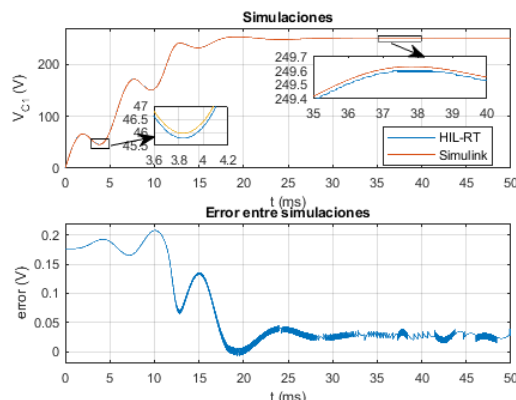


Fig. 8. Resultados de la simulación de V_{C1} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 100%.

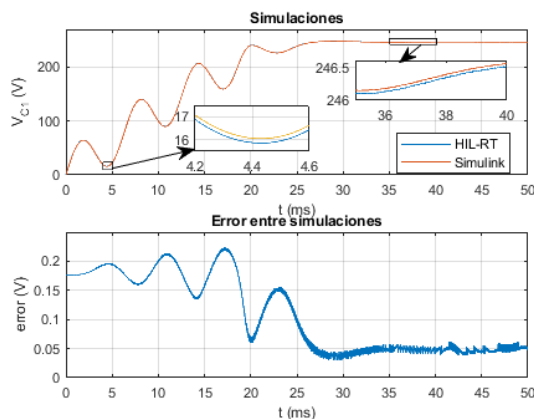


Fig. 9. Resultados de la simulación de V_{C1} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 75%.

Las Figs. 11 a la 13 muestran los resultados del voltaje del

capacitor dos para un ciclo de trabajo de 100%, 75% y 50%. Conforme el ciclo de trabajo disminuye se puede observar un aumento en el error absoluto de voltaje llegando a ser de casi ± 250 mV con 50% de ciclo de trabajo. Sin embargo, en los acercamientos se puede observar que, al igual que en la corriente del inductor, ese error es producido por el desfaseamiento entre los picos del rizo de voltaje y que en ningún momento se hace mayor a dicho rizo. En los acercamientos también se puede observar que los picos de rizo son ligeramente mayores en la simulación HIL en tiempo real, esto se debe a la menor precisión en los cálculos que la que se obtiene en la simulación de Simulink®.

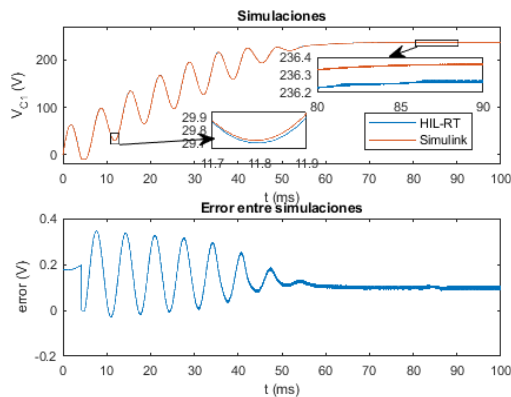


Fig. 10. Resultados de la simulación de V_{C1} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 50%.

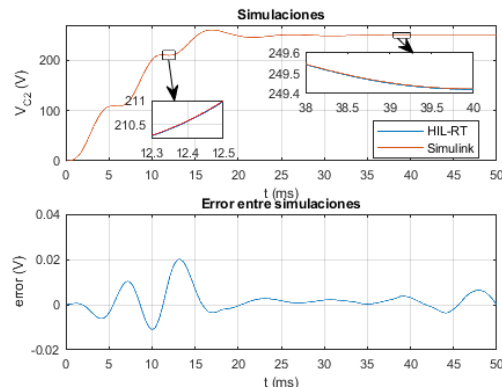


Fig. 11. Resultados de la simulación de V_{C2} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 100%.

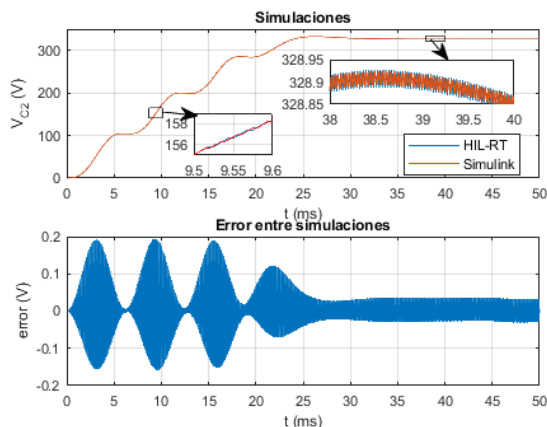


Fig. 12. Resultados de la simulación de V_{C2} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 75%.

Se puede observar en las Figs. 12 y 13 que el error de estado estable es pulsante y que alcanza hasta ± 100 mV

cuando el ciclo de trabajo es del 50%. Aunque este error puede parecer apreciable representa apenas el 0.021% de los 472.7 V que alcanza el voltaje del capacitor dos. Además, el error absoluto promedio en estado estable de la simulación es de -6.2 mV al 100%, de -8.3 mV al 75% y de 10.4 mV al 50%; los errores relativos correspondientes son prácticamente nulos.

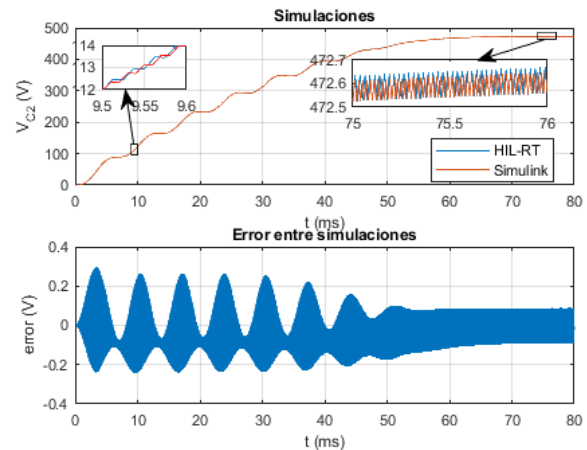


Fig. 13. Resultados de la simulación de V_{C2} en MatLab y HIL-RT, y del error entre las dos simulaciones con u al 50%.

VII. CONCLUSIONES

El modelo HIL propuesto tiene un desempeño muy bueno con diferencias muy pequeñas a una simulación computacional. El tiempo de ejecución de la simulación HIL es en tiempo real al tener un tiempo de latencia de cálculo de 100 ns. El modelo propuesto se puede implementar en cualquier plataforma FPGA. El tiempo de latencia se puede mejorar utilizando técnicas de pipe-lining. También se explora la posibilidad de simular hasta 4 convertidores similares de forma paralela con el mismo núcleo de cálculo del FPGA sin aumentar la latencia del cálculo.

REFERENCIAS

- [1] Sarhadi, P., Yousefpour, S. "State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software," *International Journal of Dynamics and Control*, vol. 3, pp. 470–479, 2015. <https://doi.org/10.1007/s40435-014-0108-3>
- [2] Dusan Majstorovic, Ivan Celanovic, Nikola Dj. Teslic, Nikola Celanovic, and Vladimir A. Katic, "Ultralow-Latency Hardware-in-the-Loop Platform for Rapid Validation of Power Electronics Designs," *IEEE Transactions On Industrial Electronics*, vol. 58, pp. 4078-4716, 2011. <https://doi.org/10.1109/TIE.2011.2112318>
- [3] Mohamed Dagbagi et al., "ADC-Based Embedded Real-Time Simulator of a Power Converter Implemented in a Low-Cost FPGA: Application to a Fault-Tolerant Control of a Grid-Connected Voltage-Source Rectifier," *IEEE Transactions On Industrial Electronics*, vol. 63, pp. 1179-1190, 2016. <https://doi.org/10.1109/TIE.2015.2491883>
- [4] Luis Herrera, Cong Li, Xiu Yao, and Jin Wang, "FPGA-Based Detailed Real-Time Simulation of Power Converters and Electric Machines for EV HIL Applications," *IEEE Transactions On Industry Applications*, vol. 51, pp. 1702-1712, 2015. <https://doi.org/10.1109/TIA.2014.2350074>
- [5] Wentao Wang, Zhuoxuan Shen, and Venkata Dinavahi, "Physics-Based Device-Level Power Electronic Circuit Hardware

Emulation on FPGA," *IEEE Transactions On Industrial Informatics*, vol. 10, pp. 2166-2179, 2014. <https://doi.org/10.1109/TII.2014.2361656>

[6] Matar, M., Paradis, D. and Iravani, R., "Real-time simulation of modular multilevel converters for controller hardware-in-the-loop testing," *IET Power Electronics*, vol. 9, 42-50, 2016. <https://doi.org/10.1049/iet-pel.2015.0012>

[7] Selvamuthukumar, R. and Gupta, R., "Rapid prototyping of power electronics converters for photovoltaic system application using Xilinx System Generator," *IET Power Electronics*, vol. 7: 2269-2278, 2014. <https://doi.org/10.1049/iet-pel.2013.0736>

[8] Sorina-Maria Ciornei, Raul-Octavian Nemeş, Mircea Ruba, Horia Hedeşiu, and Claudia Marţiş, "Real-Time FPGA Simulator for Electric Vehicle power supply systems," *2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, pp. 983-988, 2017. <https://doi.org/10.1109/OPTIM.2017.7975098>

[9] Mircea Ruba, Sorina Ciornei, Horia Hedeşiu, and Claudia Martis, "Complete FPGA Based Real-Time Motor Drive Simulator With Bidirectional Battery And Ultracapacitor Power Supply," *International Symposium On Advanced Topics In Electrical Engineering*, pp. 186-191, 2017. <https://doi.org/10.1109/ATEE.2017.7905078>

[10] Marco Liserre, Thilo Sauter, and John Y. Hung, "Future energy systems: Integrating renewable energy sources into the smart power grid through industrial electronics," *IEEE Industrial Electronics Magazine*, vol. 4, no. 1, pp. 18-37, 2010. <https://doi.org/10.1109/MIE.2010.935861>

[11] A. Gutierrez, H. R. Chamorro, L. F. L. Villa, J. F. Jimenez and C. Alonso, "SysML methodology for HIL implementation of PV models," *2015 17th European Conference on Power Electronics and Applications (EPE'15 ECCE-Europe)*, Geneva, Switzerland, pp. 1-7, 2015. <https://dx.doi.org/10.1109/EPE.2015.7309196>

[12] Yağan, Yunus & Vardar, Kadir & Ebeoğlu, Mehmet, "Modeling and Simulation of PV Systems," *IOSR Journal of Electrical and Electronics Engineering*, vol. 13, pp. 1-11, 2018. 10.9790/1676-1302030111.

[13] H. Palahalli, Y. Huo and G. Gruosso, "Real Time Simulation of Photovoltaic System using FPGA," *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, Amalfi, Italy, pp. 865-870, 2018. doi: 10.1109/SPEEDAM.2018.8445288.

[14] Samano-Ortega, V.; Padilla-Medina, A.; Bravo-Sanchez, M.; Rodríguez-Segura, E.; Jimenez-Garibay, A.; Martínez-Nolasco, J., "Hardware in the Loop Platform for Testing Photovoltaic System Control," *Applied Sciences*, vol. 10, pp. 1-24, 2020. <https://doi.org/10.3390/app10238690>

[15] Bautista O., Bernal Á., Pantoja A., Revelo J., "FPGA Implementation of a PV Generator and a Boost Converter Controllers for a Virtual Laboratory Environment," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 48, pp. 533-551, 2024. <https://doi.org/10.1007/s40998-024-00704-2>

[16] De Soto, W., Klein, S. A., & Beckman, W. A., "Improvement and validation of a model for photovoltaic array performance," *Solar Energy*, Vol. 80, No.1, 78-88, 2006. <http://doi.org/10.1016/j.solener.2005.06.010>



Rodolfo Orosco Guerrero has been a professor at the Instituto Tecnológico de Celaya for 31 years, teaching engineering and postgraduate courses in electronics. He received his B.S. degree in Electronics Engineering from the Instituto Tecnológico de Celaya in 1992; M.S. degree in Electrical Engineering from the Universidad de Guanajuato in 2001; and Ph.D. in Electrical Engineering from the Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV), Zacatenco unit in Mexico City in 2003. He has taught PLC programming courses for the past 25 years, from basic ladder diagram programming to advanced GRAFCET-based sequential logic controller design. In 2012 he developed a standard for PLC programs for the automated processes of the company KOSTAL de México located in the city of Querétaro. Currently his area of interest is oriented to the fields of instrumentation and control of power systems, industrial automation, and control of renewable energy systems. Other interests in the teaching field of the development of professional competences stand out, such as: Developing innovative teaching and learning strategies, as well as evaluation tools that allow evaluating the scope of professional competences.



Elias Rodríguez was born in México in 1972. He received the B.Sc. degree in electronics engineering from the Universidad Autonoma Metropolitana, México City, México, in 1994, and the M.Sc. and Ph.D. degrees in electronics engineering from the Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Mexico, in 1996 and 1999, respectively. He is currently a Professor and Researcher in the Department of Electronics Engineering, Tecnológico Nacional de México, Celaya, México, where he teaches and conducts research in power electronics since 1999. His fields of interest are high-frequency power conversion, high-power-factor rectifiers, and DC microgrids.



Juan José Martínez Nolasco received his B.S. degree in Electronics Engineering from the Technological Institute of Guzman City, Jalisco, Mexico, in 2007; M.S. degree in Electronics Engineering from the Technological Institute of Celaya, Guanajuato, Mexico, in 2009; and a Ph.D. in Engineering Science from the Technological Institute of Celaya, Guanajuato, Mexico, in 2018. He is currently teacher and researcher in the Department of Mechatronics Engineering, Technological Institute of Celaya, his research interests include intelligent control, direct current microgrids and applications of fuzzy logic control.



Fany Rodríguez García has been a professor at the Instituto Tecnológico de Celaya for 30 years, teaching engineering courses in electronics. She received her B.S. degree in Electronics Engineering from the Instituto Tecnológico de Celaya in 1992; Studies in Electrical Engineering from the Universidad de Guanajuato; MED from the UTEL in 2019. Currently his area of interest is oriented to the fields of instrumentation, linear control and semiconductors. ther interests in the teaching field of the development of professional competences stand out, such as: Developing innovative teaching and learning strategies, as well as evaluation tools that allow evaluating the scope of professional competences.