

# Simulación Mediante “*Hardware In the Loop*” de un Convertidor *Buck*

F. Casellas<sup>1</sup>, J. Esteban, F. Guinjoan<sup>2</sup>, R. Piqué<sup>2</sup>, H. Martínez<sup>2</sup> y G. Velasco<sup>1</sup>

Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona

<sup>1</sup> *Consorci Escola Industrial de Barcelona (CEIB)*

<sup>2</sup> *Universitat Politècnica de Catalunya - BarcelonaTECH (UPC)*

Barcelona, Spain

[francisco.casellas@upc.edu](mailto:francisco.casellas@upc.edu)

**Abstract**— El presente artículo describe la implementación un convertidor reductor mediante su simulación por un ordenador estándar trabajando en tiempo real.

El objetivo es determinar las limitaciones operativas y verificar la viabilidad como equipo HIL (*Hardware In The Loop*) en una aplicación de electrónica de potencia de un ordenador estándar trabajando con un sistema operativo en tiempo real. Para verificar comportamiento del convertidor reductor simulado en tiempo real mediante el equipo HIL se utiliza como referencia el mismo convertidor reductor con modelado basado en *SPICE*. Finalmente se somete al modelo a diferentes ensayos para determinar el límite en los parámetros de la simulación que provoca el procesador trabajando en tiempo real.

**Index Terms**—*Hardware In The Loop*, Simulación en Tiempo Real.

## I. INTRODUCCIÓN

DEBIDO al incremento en la complejidad de los sistemas y con el objetivo de reducir el tiempo para introducir el producto al mercado, aparecen nuevas técnicas para el desarrollo y test de sistemas como es el caso del *Hardware In the Loop* (HIL).

Para los ensayos y desarrollo de sistemas existían dos posibilidades si el sistema a ensayar es de una potencia específica elevada. Primero, realizar un experimento en el hardware real y segundo, realizar una simulación de software. Actualmente existe una tercera opción, que es un sistema híbrido de los dos anteriores, denominado HIL.

Esta técnica se basa en la interconexión de hardware externo con un equipo informático, trabajando en tiempo real, que simula el circuito o sistema a ensayar [1]. Mediante esta técnica se dispone de una plataforma simulada, HIL, que es el equivalente del sistema físico a ensayar, en la plataforma se introduce el equivalente al sistema físico mediante modelos matemáticos [2].

Las ventajas que proporciona una simulación mediante la técnica HIL son [3]:

- Permite realizar simulación previa y en paralelo con el desarrollo de la planta, de forma que se pueden ir sustituyendo las partes simuladas por las ya implementadas físicamente conforme estén disponibles.
- Permite someter a la planta, basada en HIL, a situaciones extremas evitando dañar la planta real.
- Permiten reducir el coste de desarrollo, se evitan los prototipos parciales de los elementos del sistema y el

coste de verificación, se evitan las averías de un sistema real.

Los principales componentes de un equipo HIL son:

- Ordenador con sistema operativo en tiempo real.
- Interfaces de entrada/salida, para el ordenador.
- Interfaz con el operador, para el control.

El ordenador es el elemento más importante del HIL ya que es el encargado de proporcionar la ejecución determinista del proceso a simular con los diferentes componentes hardware del sistema informático [4]. Una aplicación es determinista o en tiempo real, si completa su proceso de cálculo o simulación, en un intervalo de tiempo fijo y definido. En este caso el tiempo de simulación, del sistema, se corresponde con el tiempo transcurrido en la realidad.

Por otra parte, para asegurar la correcta implementación del sistema HIL, es necesario que los elementos de entrada/salida realicen las adquisiciones de señal y por lo tanto el muestreo a altas velocidades de manera eficiente.

El objetivo de este artículo es estudiar las características de la simulación en tiempo real con un ordenador estándar, mediante técnicas HIL, en una aplicación de electrónica de potencia. Como planta a simular se utiliza el modelo de un troceador reductor, convertidor *Buck*.

## II. DESCRIPCIÓN DEL SISTEMA

La implementación del equipo HIL a utilizar se representa en la Fig. 1, incluye los elementos utilizados que se describen posteriormente [5].

### A. Ordenador personal

Tiene la función de proporcionar la interfaz con el



Fig. 1. Sistema propuesto para implementar el HIL

usuario, básicamente la monitorización y control del sistema. Permitiendo ajustar los valores de los componentes y las características del convertidor *Buck*, tal y como se observa en la Fig. 2. Los valores de los componentes son los parámetros adquiridos y enviados al ordenador remoto, real time, cada 100 ms mediante una conexión por intranet.

### B. Ordenador remoto

Se trata de un ordenador personal, tipo estación de trabajo, con procesador Intel Core i7 de segunda generación de 6 núcleos ejecutando un sistema operativo en tiempo real

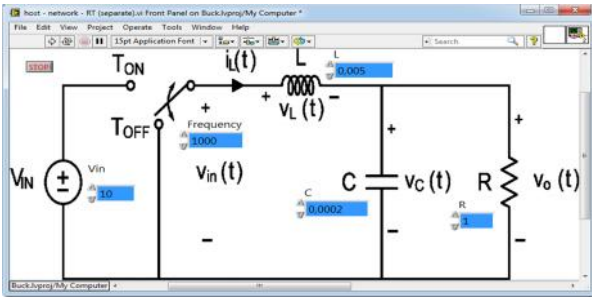


Fig. 2. Interfaz de usuario en el ordenador personal

(RTOS: *Real Time Operating System*).

Su principal función es ejecutar dos procesos o tareas en paralelo con diferente prioridad de ejecución. El primer proceso es el encargado de la simulación en tiempo real del convertidor *Buck*. Es por ello que se le asigna mayor prioridad de manera que su ejecución no se vea interrumpida por otras tareas. Este proceso determinará los límites de la simulación en tiempo real del sistema.

El segundo proceso, de menor prioridad y por lo tanto no determinista, se ejecuta cada 100 ms y adquiere los parámetros enviados por el ordenador personal, estos datos los comparte con el proceso de mayor prioridad.

### C. Chasis PXIe-1082

Se trata de un chasis de National Instruments [6] con los siguientes módulos:

- Módulo PXIe-PCIe8388 [7], es un puente que permite establecer la interconexión entre el chasis PXIe y el ordenador remoto.
- Tarjeta de adquisición de datos PXIe-6358 [8].
- Tarjeta de adquisición de datos PXIe-6251 [9].
- Tarjeta RIO (Reconfigurable I/O) multifunción con una FPGA Virtex-5 LX85 incorporada [10].

De estos módulos disponibles se utiliza la tarjeta de adquisición PXIe-6358, que es la encargada de proporcionar de manera externa los valores de las variables simuladas para monitorizarlas, y que permite visualizarlas en el osciloscopio. En una fase inicial se usan dos de las cuatro salidas analógicas disponibles en la planta, una para representar la tensión del condensador o tensión de salida ( $v_C$ ) y la otra para la corriente en el inductor ( $i_L$ ).

## III. IMPLEMENTACIÓN DEL CONVERTIDOR *BUCK*

Para simplificar la implementación del convertidor estático, se utiliza un modelo del convertidor *Buck* con componentes ideales, como el de la Fig. 3.

En los siguientes modelos, para las simulaciones HIL, se define como variable de control  $u(t)$ , la cual toma el valor 1 cuando el interruptor se encuentra en  $T_{ON}$  y 0 en la posición  $T_{OFF}$ . También se fuerza a que el convertidor siempre trabaje en modo de conducción continua de manera que no se produzca anulación de la corriente en el inductor.

Su implementación se realiza en el proceso de mayor prioridad del ordenador remoto a partir del módulo *Control*

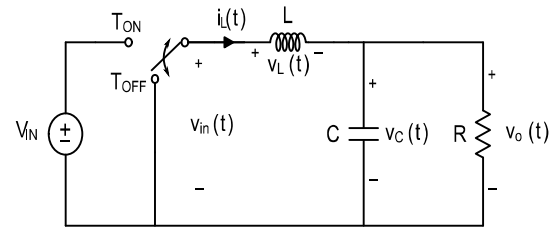


Fig. 3. Convertidor *Buck* ideal

*Design and Simulation*. Algunos de sus bloques se pueden visualizar en las Fig. 4 y Fig. 5. Entre ellos, el más importante es el lazo de simulación, *Control & Simulation Loop*, ya que determina la temporización del proceso de simulación. Los parámetros de configuración del módulo, con mayor interés en esta aplicación, se indican a continuación:

- **Reloj de referencia:** permite sincronizar el lazo con un reloj interno de 1 kHz o de 1 MHz. Ésta es la primera limitación temporal del sistema ya que por el teorema de Nyquist la frecuencia de la señal a muestrear debe ser menor a 500 kHz para el reloj de 1 MHz.
- **Step size:** especifica el intervalo de tiempo entre el momento que el software evalúa el modelo a simular y después el momento que actualiza los resultados de la simulación, este intervalo se puede denominar *time step*.
- **Periodo:** determina el periodo de ejecución del lazo. Al tratarse de una simulación en tiempo real, este valor debe coincidir con el *time step*. Estos dos parámetros determinan la máxima capacidad de implementación del sistema. Un *time step* demasiado pequeño causa la incapacidad de ejecutar todas las instrucciones del lazo de simulación en un solo periodo de ejecución, obteniendo así resultados erróneos, ya que dejaría de tratarse de una ejecución determinista. Mientras que un *time step* demasiado elevado provocaría que las señales obtenidas presenten insuficientes muestras por periodo, produciendo formas de onda incorrectas o irreales.
- **ODE Solver:** indica el método usado para resolver el modelo implementado. La condición que debe cumplir el *solver* para una simulación en tiempo real es que sea de paso fijo. En este caso se ha seleccionado el método de Runge-Kutta.

Para determinar la eficiencia de cálculo, el convertidor *Buck* se ha implementado en el módulo simulador utilizando dos metodologías diferentes de simulación. Mediante el modelo integral y mediante el modelo de variables de estado.

### A. Modelo integral

Se ha implementado el modelo directamente a partir de las ecuaciones integrales siguientes (1) [11], correspondientes a las variables de interés del convertidor ( $v_o$  e  $i_L$ ). Se muestra el código correspondiente al modelo integral dentro del rectángulo negro en la Fig. 4.

$$v_o(t) = \frac{1}{LC} \iint (v_{in}(t) - v_o(t)) dt - \frac{1}{RC} \int v_o(t) dt$$

$$i_L = \frac{1}{L} \int (v_{in}(t) - v_o(t)) dt \quad (1)$$

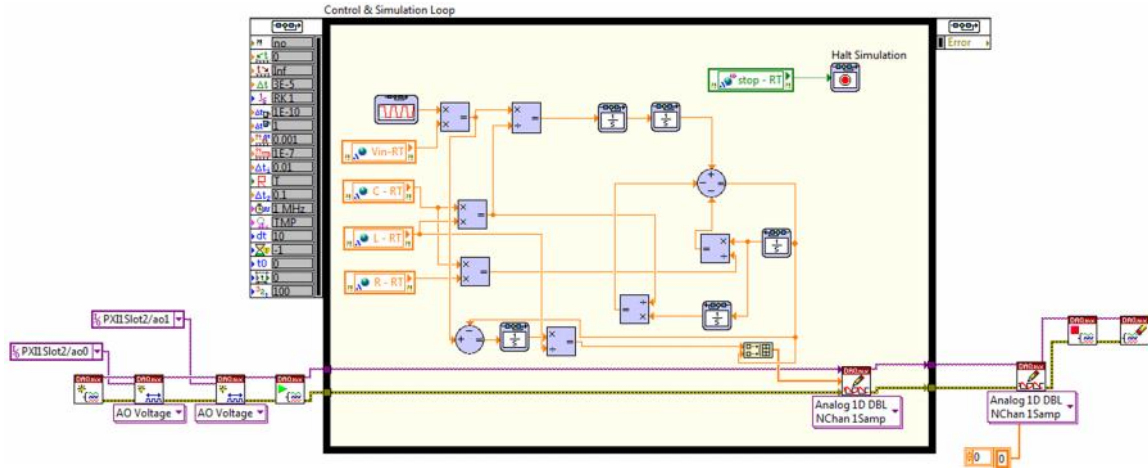


Fig. 4. Implementación del modelo integral del convertidor Buck

**B. Variables de estado**

Se ha implementado el convertidor reductor usando su modelo dinámico de variables de estado expresado matricialmente (2).

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/(RC) \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} V_{in}/L \\ 0 \end{bmatrix} u(t) \quad (2)$$

Se muestra el código correspondiente al modelo de variables de estado dentro del rectángulo negro en la Fig. 5.

El módulo *Control Design and Simulation* incluye un bloque que permite trabajar directamente con variables de estado. Para ello necesita como datos unos parámetros de tipo *cluster*, marcado en la figura como *StateCluster-RT*, con las matrices correspondientes al modelo y la variable de control de los interruptores  $u(t)$ , marcado en la figura como *Pulse Signal*.

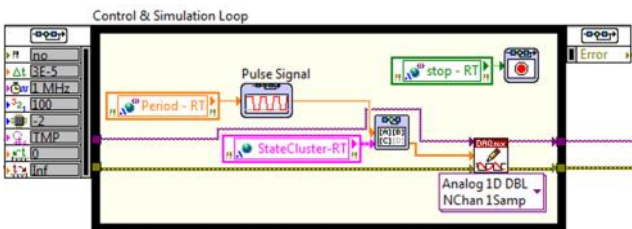


Fig. 5. Implementación del modelo mediante variables de estado del convertidor Buck

**C. Comparación de las implementaciones**

En la simulación del convertidor se han utilizado los valores de la Tabla I.

TABLA I  
VALORES PARA LA SIMULACIÓN DEL CONVERTIDOR

$V_{in}$	Frecuencia de conmutación	L	C	R	Relación de conducción
10 V	1 kHz	5 mH	200 $\mu$ F	1 $\Omega$	50%

El resultado numérico de la simulación obtenido el mismo en los dos métodos de simulación utilizados, Fig. 6a y 6b.

Para determinar el mejor método de implementación se utiliza la figura de mérito que representa el consumo de CPU necesario para la simulación.

En el caso del modelo integral se obtiene un consumo de CPU correspondiente al 56% y para el modelo con variables de estado se obtiene un consumo de CPU correspondiente al 28%.

También la implementación del modelo con variables de estado resulta más compacta y cómoda de realizar, como se puede comparar en las Fig. 4 y Fig. 5.

En las simulaciones finales se implementará solamente el modelo mediante variables de estado en forma de matriz.

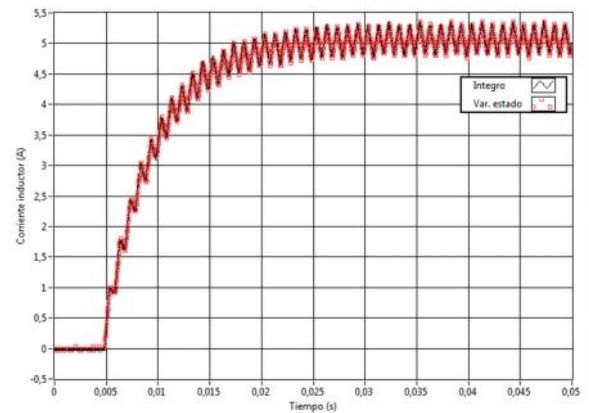


Fig. 6a. Comparación de la corriente en el inductor para los dos modelos simulados

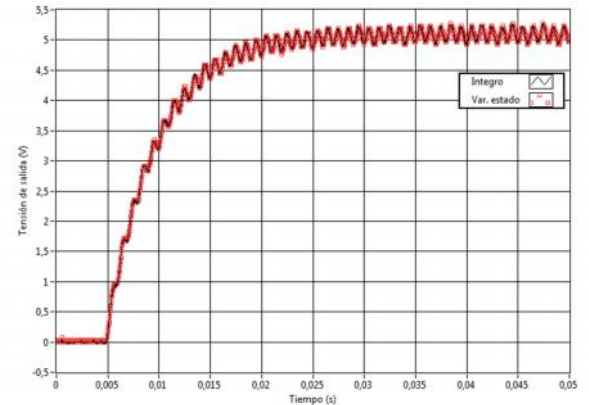


Fig. 6b. Comparación de la tensión de salida para los dos modelos simulados



#### IV. SIMULACIONES Y RESULTADOS

Los resultados en los cálculos de las simulaciones mediante el equipo HIL se han comparado, con el resultado obtenido de *LTSpice IV* este es un simulador basado en *SPICE* de *Linear Technologies*, orientado a simular sistemas electrónicos de potencia como son los convertidores conmutados [12].

Los resultados satisfactorios en el proceso de simulación HIL, con el modelo del convertidor *Buck* seleccionado, lo determinarán los límites temporales en los que este modelo es operativo, modificando el valor de *time step*.

##### A. Comparación con *LTSpice*

La Fig. 7 muestra la comparación entre los resultados obtenidos, para la corriente en el inductor, con el simulador *LTSpice IV* y el modelo implementado en *LabVIEW*. Se verifica que la simulación en *LabVIEW* es consistente al obtener el mismo resultado que con el simulador *SPICE*.

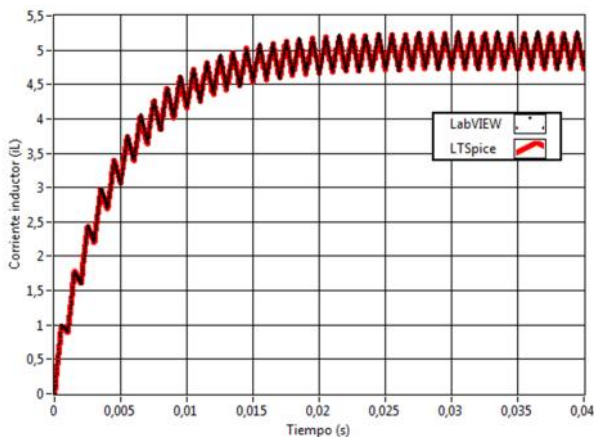


Fig. 7. Comparación entre la simulación con *LTSpice* y *LabVIEW*

##### B. Mínimo *time step* de simulación

El parámetro que más afecta al determinismo de la simulación del convertidor es el *time step*, ya que éste tiene además relación directa sobre el periodo de ejecución en los lazos de simulación mediante HIL.

A partir de los ensayos, simulaciones realizadas, se determina que el mínimo *time step* que puede utilizarse, que para este modelo de convertidor es de 10  $\mu$ s.

Se capturan las señales obtenidas del simulador con el osciloscopio, en la Fig. 8a, para las corrientes en el inductor, y en la Fig. 8b para las tensiones en la carga.

Resultando con un *time step* de 30  $\mu$ s una señal que coincide con el simulador de referencia.

Mientras que la simulación con un *time step* de 8  $\mu$ s aparece con un retraso de la señal respecto a la simulación de referencia, debido a utilizar un *time step* demasiado reducido.

Estos resultados se deben a que el equipo HIL no es capaz de proporcionar el resultado de la simulación dentro del periodo especificado por el *time step*, es decir, el *Control and Simulation Loop* no puede realizar todas sus operaciones dentro del periodo especificado de 8  $\mu$ s y devuelve los resultados en ciclos posteriores de simulación, la razón del retraso en la señal. Este fenómeno desaparece para *time step* mayores de 10  $\mu$ s.

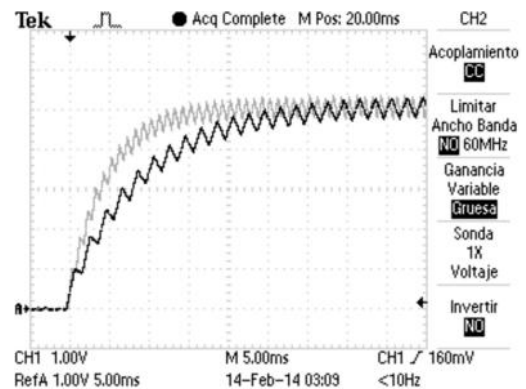


Fig. 8a. Corriente del inductor para un *time step* de 30  $\mu$ s (respuesta más rápida) y un *time step* de 8  $\mu$ s (respuesta más lenta)

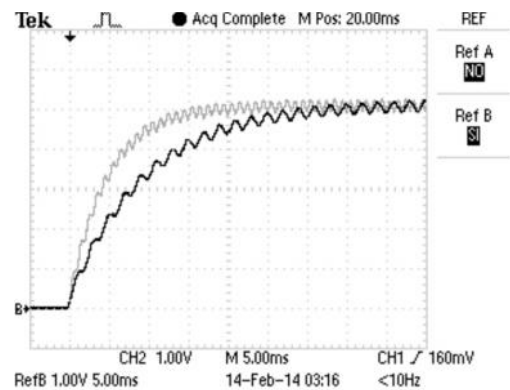


Fig. 8b. Tensión de salida para un *time step* de 30  $\mu$ s (respuesta más rápida) y un *time step* de 8  $\mu$ s (respuesta más lenta)

Al aumentar la complejidad del sistema simulado, por ejemplo añadiendo un control PID al convertidor *Buck* y la adquisición de variables externas, aumenta el límite del *time step* por lo que se adopta un *time step* y periodo de lazo de 30  $\mu$ s, que permitirá mantener un margen de seguridad suficiente.

Con *time step* de 30  $\mu$ s la máxima frecuencia de ejecución del lazo de simulación es 33,3 kHz, de acuerdo con el criterio de Nyquist, y la máxima frecuencia de la señales a muestrear en la simulación por el equipo HIL es de 16,7 kHz.

#### V. CONCLUSIONES

El equipo HIL implementado, para la simulación en tiempo real de sistemas electrónicos de potencia, como es un convertidor conmutado básico, convertidor *Buck*. Presenta ciertos problemas en los tiempos de procesado trabajando a frecuencias de señal superiores a los 15 kHz, al utilizar un procesador estándar como soporte de cálculo.

Trabajar con frecuencias de muestreo superiores a 30 kHz implica paralelizar procesos de cálculo para el simulador HIL.

Una alternativa que permite este procesado paralelo es distribuir la carga de cálculo entre los distintos núcleos del sistema. Que implica un esfuerzo extra en la programación y la necesidad de añadir más elementos de sincronización entre los diferentes lazos del proceso.

Otra alternativa es la implementación del modelo en la FPGA que dispone el sistema PXIe. Este cambio en el soporte de cálculo implica resolver una serie de limitaciones:

- El módulo FPGA trabaja con coma fija y el modelo implementado en el procesador trabaja con variables en coma flotante. Esto implica que antes de realizar su implementación se debe determinar y validar el modelo en coma fija.
- El módulo *Control Design and Simulation* no se puede usar en el módulo FPGA por lo que, entre otros, se debe implementar manualmente el funcionamiento del *solver* lo que implica una mayor complejidad del modelo a diseñar.
- Cambios en el modelo pueden resultar más complicados de realizar en la FPGA que en el código para el procesador, la disponibilidad de módulos *software* para procesado es menor.
- La carga del código en la FPGA es mucho más lenta que en el procesador, requiere mayor tiempo de compilado.

Por otra parte, en comparación con la implementación en el procesador presenta las siguientes ventajas:

- Dispone de un reloj de referencia de 40 MHz, permitiendo que el uso de *time steps* más pequeños, por lo que consecuentemente, es posible trabajar con frecuencia de conmutación más altas.
- Evita retrasos debido a las latencias propias del procesador (que pueden ser del orden de los  $\mu$ s) y permite procesado paralelo.

#### REFERENCIAS

- [1] M. Dargahi, A. Ghosh, G. Ledwich, F. Zare, "Studies in Power Hardware in the Loop (PHIL) Simulation Using Real-Time Digital Simulator (RTDS)," *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems*, pp. 1-6
- [2] X. Wu, S. Lentijo, A. Deshmuk, A. Monti, F. Ponci, "Design and Implementation of a Power-Hardware-in -the-Loop Interface: a Non-Linear Load Case Study in *Applied Power Electronics Conference and Exposition (APEC), APEC 2005. Twentieth Annual IEEE*, vol.2, pp.1332-1338.
- [3] E. Ormaetxea, E. Ibarra, J. Andreu, I. Kortabarria, M. Santos, "FPGA Based Real Time Simulation of a Matrix Converter," in *Proc. of 14th International Power Electronics and Motion Control Conference*, 2010, pp. s2-14-s2-20
- [4] "Hardware in the Loop Testing," [En línea]. Disponible: <http://www.ni.com/hil-test/>
- [5] de Jong, E., de Graaff, R., Vaessen, P., Crolla, P., Roscoe, A., Lehfuß, F., ... & Gafaro, F. "European White Book on Real-Time Powerhardware-in-the-Loop testing." *DERlab Report No* (2011): 2011-12.
- [6] "Chasis NI PXIe-1082," [En línea]. Disponible: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/207346>
- [7] "NI PXIe-PCIe8388," [En línea]. Disponible: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/209710>
- [8] "NI PXIe-6358 Adquisición de Datos Simultánea Serie X," [En línea]. Disponible: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/209216>
- [9] "NI PCI-6251 Adquisición de Dator Serie M," [En línea]. Disponible: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/14124>
- [10] "NI PXI-7853R con FPGA Virtex-5 LX85," [En línea]. Disponible: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/206323>
- [11] Ballester Portillo, Eduard; Piqué, Robert. *Electrónica de potencia: principios fundamentales y estructuras básicas*. Barcelona: Marcombo, 2011. ISBN 9788426716699.
- [12] "Simulador LTSpice Iv de Linear Technology," [En línea]. Disponible: <http://www.linear.com/designtools/software/#LTSpice>