

Metodología de implementación de un modelo tiempo real en LabVIEW

A. Matallana¹, I. Kortabarria¹, J. Andreu¹, N. Arandia¹, J. Gutierrez²

¹Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU)

²National Instruments Spain

e-mail: asier.matallana@ehu.eus

http://det.bi.ehu.es/~apert/index_cast.html

Resumen—Este artículo propone una metodología para implementar modelos de simulación *Model In the Loop* complejos sobre plataformas de tiempo real, concretamente, sobre una plataforma diseñada con el entorno de programación gráfica LabVIEW. Para desarrollar dicha metodología se ha propuesto una aplicación ejemplo, compuesta por una máquina síncrona de imanes permanentes conectada a un inversor en modo fuente de tensión. La implementación en la plataforma de tiempo real, que dispone de *hardware* específico para la simulación, evita el empleo de técnicas de promediado para el desarrollo de algoritmos que requieren rápidas velocidades de ejecución. Por último, se realiza una comparación de los resultados del modelo, obtenido con la metodología presentada, con la misma aplicación desarrollada en Matlab/Simulink.

Palabras Clave—Electrónica de Potencia, Simulación, Metodología, LabVIEW, PXI, Real Time, MIL, RCP, HIL, Procesadores Deterministas, FPGA, Convertidor, Inversor, Máquina PMSM.

I. INTRODUCCIÓN

La electrónica de potencia es ampliamente usada hoy en día en una gran variedad de aplicaciones [1].

El proceso de diseño de un convertidor de potencia y de su correspondiente control consta de varias fases. Durante las etapas iniciales de ese proceso es muy común recurrir a las simulaciones. Mediante el empleo de la simulación, es posible analizar el comportamiento del convertidor antes de que éste se haya fabricado. De esta forma, es posible analizar algunos fenómenos relacionados con el funcionamiento del convertidor, tales como la interacción del convertidor con la red eléctrica [2], la tolerancia ante fallos en los dispositivos [3], las pérdidas en los semiconductores [4], etc. Por otra parte, la simulación es una herramienta excelente en el diseño de nuevas técnicas de control y modulación.

El uso de plataformas de *Real Time* (RT) en el proceso de simulación permite el diseño de modelos con los que se puede validar el comportamiento completo del convertidor y de su control. Dichos modelos reciben el nombre de *Model In the Loop* (MIL), a través de los cuales es posible obtener los modelos de *Rapid Control Prototyping* (RCP) y el *Hardware In the Loop* (HIL). En el RCP se simula el algoritmo de control siendo el resto de las partes que componen el sistema elementos reales. Por el contrario, en el HIL la plataforma simula el comportamiento de los dispositivos físicos mientras que el algoritmo de control se implementa en el *hardware* (HW) definitivo [5].

Sin embargo, la simulación de los convertidores de potencia dentro de una plataforma RT es compleja. Hay que

tener en cuenta que los convertidores están constituidos por interruptores que conmutan a frecuencias elevadas (5-20 kHz). Por lo tanto, se trata de circuitos no lineales que requieren altas capacidades de procesamiento para ser simulados en un periodo de tiempo razonable. En éste sentido, se han planteado una serie de técnicas de interpolación que permiten acelerar notablemente la simulación del modelo a la vez que se mantiene una buena precisión en los resultados. Dichas técnicas ha recibido el nombre de técnicas de promediado (TAM, *Time Average Method*) [6], [7]. Sin embargo, el uso de estos métodos promediados no permite visualizar y calcular el comportamiento total del sistema real como puede ser el estudio del sistema a altas frecuencias.

En la actualidad, el desarrollo de HW específico para simulación incorporado en las plataformas de RT, como los DSPs y FPGAs, permite incrementar la velocidad de ejecución de los algoritmos implementados. Este avance se debe sobre todo a las FPGAs que permiten trabajar a frecuencias de entorno a los 100 MHz con las cuales se pueden implementar modelos complejos compuestos por la combinación de diversas arquitecturas de circuitos de potencia como inversores, máquinas síncronas o asíncronas y el sistema de control [8] sin necesidad de tener que aplicar métodos de promediado.

En este documento, se quiere presentar la metodología para el desarrollo de un modelo MIL sobre una plataforma de RT de National Instruments con su entorno de desarrollo LabVIEW. Para el desarrollo de la metodología se ha implementado una aplicación compuesta por una máquina síncrona de imanes permanentes PMSM, el modelo ideal de un inversor de potencia y todo el sistema de control encargado de gestionar el sistema [9], [10]. Destacar que se ha empleado la FPGA de la plataforma RT para realizar el proceso de resolución de las ecuaciones de la máquina PMSM y del inversor [11] sin necesidad de emplear un método promediado.

II. APLICACIÓN EJEMPLO: MÁQUINA PMSM + INVERSOR

Con el objeto de mostrar la metodología propuesta en este trabajo, se ha desarrollado una aplicación ejemplo que consiste en una máquina de imanes permanentes PMSM conectada a un inversor en modo fuente de tensión.

A. Características de la plataforma PCI eXtensions for Instrumentation (PXI)

El modelo MIL se ha desarrollado en un equipo PXI de National Instruments. Dicho equipo se ha programado a

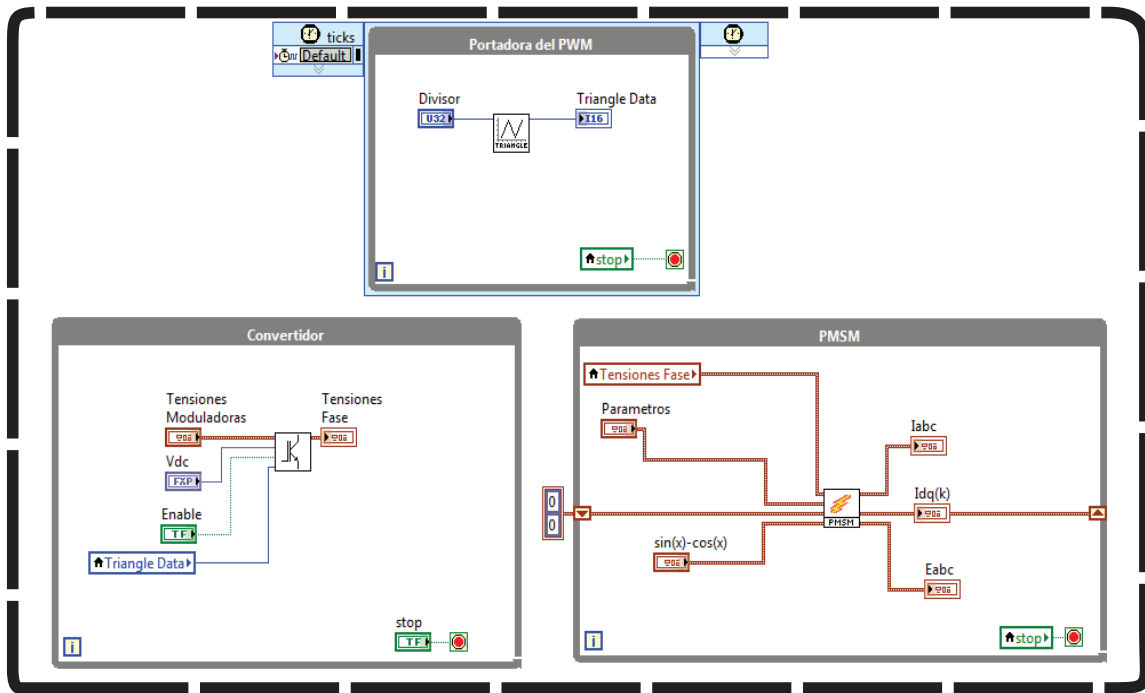


Fig. 1. Implementación con LabVIEW de los algoritmos FPGA.

través del entorno de programación gráfica LabVIEW. La composición de dicha plataforma se puede ver en la tabla I, donde se aprecia el HW que compone la plataforma PXI y los distintos paquetes de *software* (SW) LabVIEW empleados.

TABLA I
PLATAFORMA HW Y SW DE NATIONAL INSTRUMENTS.

Tipo	Dispositivo	Descripción
HW	NI PXIe-1071	Chasis
	NI PXIe-8108	Controlador doble núcleo
	NI PXI-7854R	FPGA
SW	LabVIEW RT	Programación RT
	Control Design & Simulation	Programación control en RT
	LabVIEW FPGA	Programación FPGA

B. Diseño de la aplicación

La ejecución del modelo se ha dividido en las distintas partes que componen el HW. Esta división se ha realizado de forma que el modelo pueda ser fácilmente convertible a RCP o HIL. A continuación se describe la división del modelo en las partes que componen el HW.

- PC: este es el encargado de recibir los datos y resultados generados por los bloques de los procesadores deterministas y FPGA en el ordenador desde el cual se lanza la ejecución de todo el sistema.
- Cluster de procesadores deterministas (figura 2(a)): compuesto por el algoritmo de control y la parte mecánica de la máquina PMSM (ejecutándose a 7,2 kHz). El algoritmo de control muestrea las señales provenientes de la FPGA y aplica las acciones pertinentes. De esta forma, se obtiene el modelo RCP de forma automática. La decisión de implementar la parte mecánica del PMSM dentro de procesadores deterministas, se debe a que la dinámica de ésta es mucho más lenta que la parte eléctrica. Además al programar dichos procesadores se disponen de más recursos para implementar el modelo que en FPGA.

- FPGA (figura 2(b)): implementación de los bloques del inversor y la máquina eléctrica PMSM. El inversor está compuesto de dispositivos ideales que conmutan a 7,2 kHz. Para implementar el modelo de la máquina eléctrica PMSM se ha optado por emplear el modelo DQ en valores por unidad para simplificar el modelo y hacerlo independiente de los parámetros de la máquina. Un ejemplo de implementación de dichos algoritmos en la FPGA mediante LabVIEW se observa en la figura I.

En resumen, dentro de la FPGA se han implementado las secciones que requieren una alta velocidad computacional y en los procesadores deterministas las secciones que requieren una menor velocidad de ejecución.

III. IMPLEMENTACIÓN DE UN MIL EN RT

A continuación se van a presentar los distintos HW que se han tenido en cuenta a la hora de plantear la metodología para el desarrollo de un modelo MIL. Considerando las ventajas y desventajas que presentan dichos HW a la hora de implementar la aplicación. Para ello, se han analizado las distintas restricciones que pueden presentar las acciones y funcionalidades que constituyen el modelo.

- PC convencional: la implementación del modelo se realiza empleando herramientas de alto nivel que facilitan y agilizan la obtención del algoritmo a simular. El principal inconveniente que presentan este tipo de HW es la presencia de un sistema operativo no determinista imposibilitando la realización de un modelo en RT. Se emplea como soporte adicional para realizar la descarga de los datos, realizar algún tipo de visualización, enviar/recibir algún tipo de mensaje, etc. Siempre y cuando no interfiera en la ejecución del modelo MIL determinista.
- Procesadores deterministas: dicho HW esta formado por uno o varios procesadores deterministas. Al ser proce-

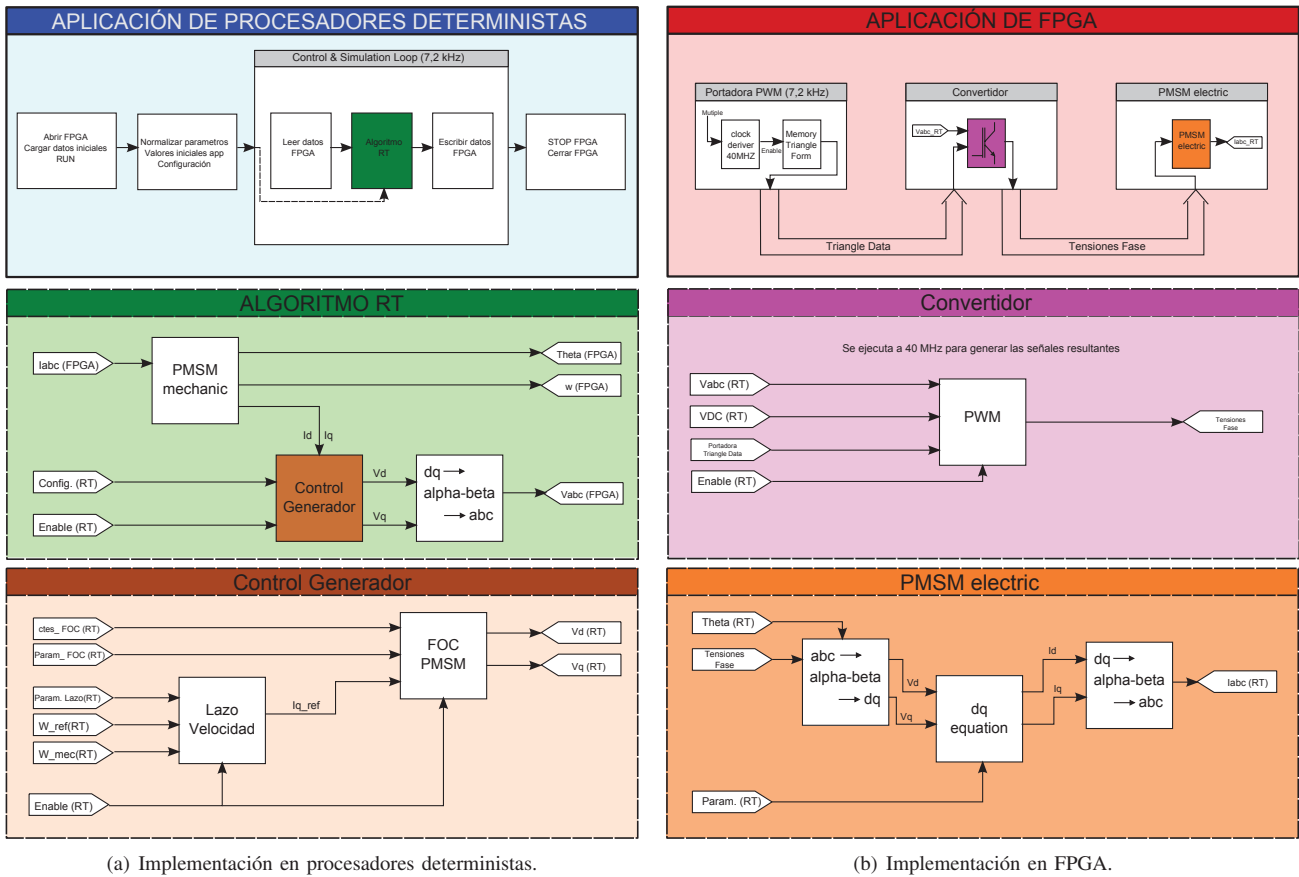


Fig. 2. Implementación del modelo MIL dentro de cada dispositivo de la plataforma RT.

sadores, el entorno de programación ofrece gran variedad de herramientas y funcionalidades que agilizan la obtención del modelo. Para aprovechar dicha ventaja, se deben incluir las operaciones que requieren procesamientos complejos. El problema de este tipo de HW es que presentan unas frecuencias de ejecución entorno a los 20 kHz, claramente insuficientes para simular ciertas partes del modelo como, por ejemplo, el convertidor.

Se deben implementar aquellas operaciones que no requieren una gran velocidad de procesamiento, debido a los límites físicos de frecuencia que presentan los procesadores.

- **FPGA:** sobre dicho HW se pueden implementar acciones que requieren velocidades por debajo del μs , dependiendo de la capacidad de las FPGAs empleadas. La desventaja que presenta este HW es que las herramientas de desarrollo en FPGA proporcionan funciones muy básicas, complicando la implementación de funcionalidades complejas. Además, el número de acciones o funcionalidades que se pueden implementar es menor que en las otras dos HW descritos, ya que en este tipo de dispositivos el código se implementa de forma física, de ahí las grandes velocidades que pueden soportar. Se emplea para aquellas funcionalidades que requieren una gran velocidad de ejecución.

Las dispositivos HW descritos en esta sección se utilizan en función de la complejidad del modelo a implementar. Así, tras

el análisis de las distintos HW con los que se puede contar para desarrollar el modelo MIL es necesario analizar que HW debe disponer la plataforma RT para poder cumplir con las necesidades del modelo.

IV. METODOLOGÍA DE DISEÑO DE UN MIL EN RT

Como se ha podido ver, en la actualidad existen tres tipos de HW para formar una plataforma de RT. Dicha plataforma puede tener multitud de configuraciones, pero para poder ser de RT es necesario que ésta disponga de al menos algún HW determinista, como es el caso del cluster de procesadores o de la FPGA. Hoy en día, la mayor parte de las plataformas de RT cuentan con los tres tipos de HW combinando los procesadores deterministas para las operaciones que requieren gran complejidad matemática y las FPGAs para aquellas operaciones que requieren velocidades de ejecución elevadas. También disponen de un PC para realizar operaciones de apoyo a los otros dos HW que realizan las operaciones deterministas.

Dicho tipo de plataformas RT que integran los tres tipos de HW son necesarias para la implementación de los modelos MIL, que integren tanto el modelo RCP como el HIL, ya que requieren una elevada carga computacional. Pero el proceso de depuración de los algoritmos que se implementan en este tipo de plataformas RT, compuestas por varios HW, es muy complicado debido a la distinta naturaleza de los dispositivos HW que forman la plataforma RT.

En este documento se desarrolla una metodología que permite simplificar el proceso de depuración de las plataformas

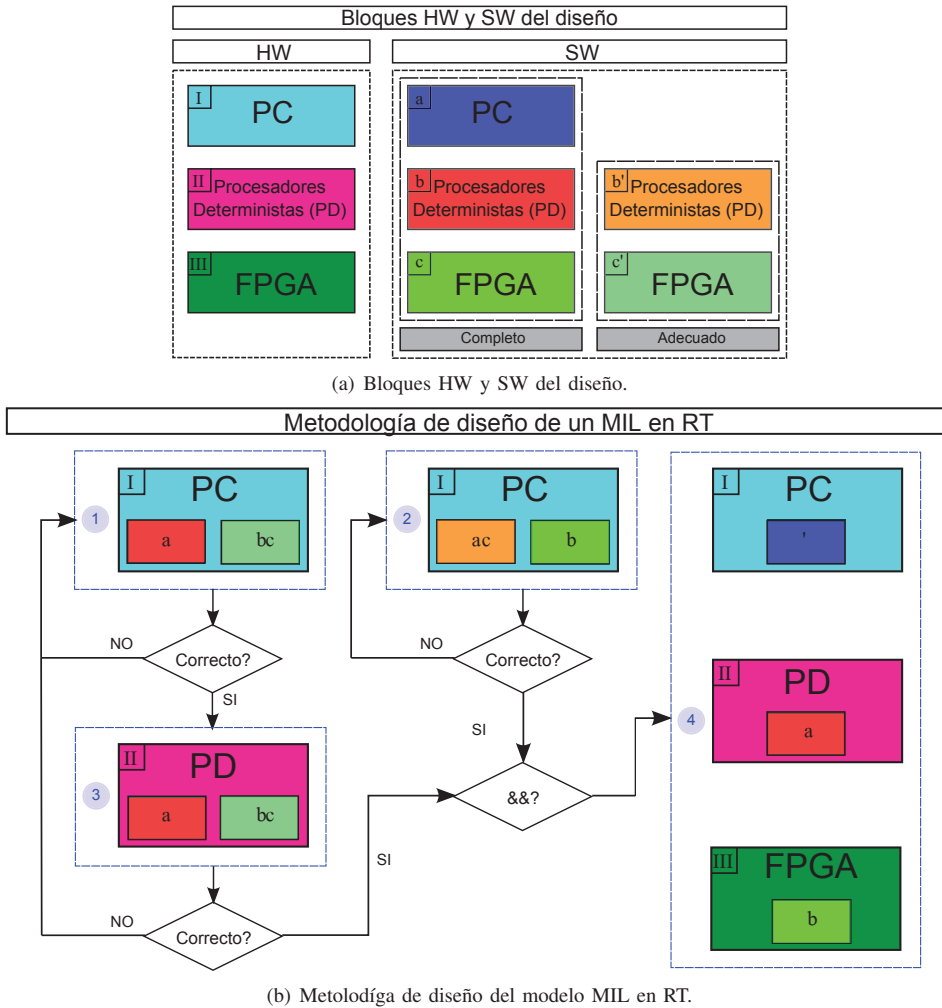


Fig. 3. Metodología propuesta para el diseño de modelos MIL sobre LabVIEW.

RT a la hora de implementar modelos complejos como un modelo MIL determinista. Mediante el uso de una plataforma flexible, como es la plataforma PXI de National Instruments, con el entorno de programación gráfica LabVIEW se realiza la programación de los distintos HW de la plataforma RT, empleando el mismo lenguaje. En la figura 3(a) se pueden ver los distintos dispositivos HW que componen la plataforma empleada para desarrollar la metodología, así como, los distintos bloques SW necesarios desarrollar para conseguir depurar y validar el modelo MIL que se desee implementar. Para ello, es interesante seguir los siguientes pasos que se corresponden con el esquema de la figura 3(b):

- 1) Realizar y simular los algoritmos de control en el PC. Sobre los procesadores deterministas de la plataforma se debe implementar el control de la aplicación (paso 3). El paso previo debe ser la implementación de dichos algoritmos de control en el PC para facilitar el proceso de corrección de errores y detectar problemáticas en los algoritmos, ya que el PC permite detectar y ver la problemática a nivel computacional de forma más sencilla que en los procesadores deterministas y en la FPGA. Para poder realizar este paso es necesario adecuar los algoritmos que se van a implementar dentro

de la FPGA al PC. En este contexto, se usa un modelo promediado del convertidor. Ésto se debe a que los PCs convencionales no tienen la potencia suficiente para simular los algoritmos que se van a implementar dentro de los procesadores deterministas y en la FPGA a la vez, así que es necesario adecuar la parte del modelo que no se está validando. El siguiente paso para validar los algoritmos de control de la aplicación se realiza en el paso 3.

- 2) Emplear la plataforma PC para simular los algoritmos FPGA. LabVIEW permite simular computacionalmente en un PC todas las operaciones que se ejecutan en una FPGA, pero con un coste temporal y computacional muy elevado. Al igual que en el paso 1, se deben adecuar los algoritmos de control para poder probar el funcionamiento de los algoritmos que se implementan dentro de la FPGA. Asimismo, las herramientas de desarrollo en PC permiten visualizar cualquier señal del circuito descrito en la FPGA, facilitando la detección de posibles fallos. Conviene indicar que realizar la implementación de la FPGA requiere un proceso de larga duración, en el ejemplo propuesto en este trabajo entorno a los 30 min. Además, cualquier pequeña modificación

en el código significa la reconfiguración de toda la FPGA. Aunque la simulación de dicho código lleve mucho tiempo, el empleo este método para depurar el código de la FPGA permite ir visualizando las señales de la FPGA según se va simulando, lo que sirve de gran ayuda en el proceso de depuración.

- 3) Implementación de los algoritmos de control en los procesadores deterministas. Una vez simulados y depurados los bloques que constituyen el control de la aplicación se deben implementar sobre los procesadores deterministas para comprobar que los algoritmos cumplen las condiciones de tiempo que se hayan especificado. Para poder comprobar el funcionamiento de dichos algoritmos se debe implementar el modelo promediado del convertidor de la FPGA, así como el resto de funcionalidades de la FPGA, adecuándolas para poder ser ejecutadas sobre los procesadores deterministas. El modelo adecuado de la FPGA, tanto el convertidor promediado como el resto de funcionalidades, se deben implementar sobre procesadores deterministas distintos de los empleados para implementar los algoritmos de control. De esta forma se puede comprobar que el control cumple los tiempos de ejecución necesarios, de lo contrario se debe realizar un rediseño de dichos algoritmos volviendo al paso 1.

El empleo de LabVIEW permite una rápida adaptación de los bloques diseñados para ser implementados en FPGA a los procesadores deterministas (sustituir los bucles que determinan el número de clocks de FPGA por μs). De esta forma, los procesadores deterministas también permiten comprobar de forma sencilla los tiempos de ejecución de algunos algoritmos de la FPGA y realizar las acciones correctoras para subsanar los problemas, ya que el proceso para depurar dentro de la FPGA es mucho más complejo.

- 4) Implementación final de los algoritmos en cada parte del HW. Una vez realizados y validados los pasos 2 y 3 se realiza la implementación de cada uno de los bloques que constituyen el modelo MIL en su HW correspondiente, procesadores deterministas o FPGA. Es conveniente introducir variables patrón y señales básicas para detectar posibles problemas. Por último, para comprobar el correcto funcionamiento de la FPGA, es conveniente implementar una transmisión de las señales clave de la FPGA a los procesadores deterministas para ver su funcionamiento, así como el uso de algún instrumento externo, como osciloscopio o analizador lógico para comprobar la entrada y salida de valores de la FPGA. Asimismo, se debe realizar la implementación sobre el PC de las acciones de apoyo tanto a los procesadores como a la FPGA para liberar dichos dispositivos de elevadas cargas computacionales, datos, visualización de resultados, etc.

El empleo de una plataforma RT de National Instruments, como la plataforma PXI, permite el uso de la tecnología propuesta para agilizar el proceso de diseño de un modelo MIL, ya que la arquitectura de dicha plataforma permite el

empleo del mismo lenguaje de programación para los distintos HW que componen la plataforma y una fácil adecuación del código de un HW a otro utilizando LabVIEW.

V. RESULTADOS

En este apartado se presentan algunos resultados obtenidos en la aplicación ejemplo, realizada para desarrollar la metodología de diseño de un modelo MIL sobre una plataforma RT. Concretamente, sobre la plataforma PXI de National Instruments, empleando el entorno de programación LabVIEW.

Para comprobar el correcto funcionamiento de la aplicación realizada sobre la plataforma PXI con LabVIEW, los resultados de dicha plataforma han sido comparados con los resultados de la misma aplicación simulada en un PC e implementada con Matlab/Simulink.

La principal diferencia que presentan las dos aplicaciones en los distintos entornos, Matlab/Simulink y LabVIEW, son los tiempos de ejecución. Tal y como se puede ver en la tabla II.

TABLA II
TIEMPOS DE SIMULACIÓN Y EJECUCIÓN DE LOS MODELOS.

	Matlab/Simulink*	LabVIEW
Tiempo de simulación (s)	20 s	20 s
Tiempo de ejecución (s)	355 s	20 s

*tiempo de paso $2 \mu s$

El emplear una plataforma RT no sólo agiliza la ejecución, sino que también al emplear HW específico para simulación, como es el caso de la FPGA, permite realizar operaciones a muy alta velocidad prescindiendo de modelos promediados para simular el comportamiento de la aplicación. En el caso de la aplicación desarrollada, la FPGA ha permitido implementar el modelo del convertidor y de la parte eléctrica de la máquina PMSM sin tener que recurrir a otro tipo de métodos. De esta forma, en la simulación se pueden obtener resultados que se acercan más a la realidad que usando métodos promediados. Viendo los resultados de la figura 4 se demuestra que los resultados de la simulación del modelo LabVIEW son iguales que los del modelo Matlab/Simulink utilizado como referencia, lo que valida el modelo RT. Indicar que en las figuras el subíndice M hace referencia al modelo en Matlab/Simulink y el subíndice L al modelo LabVIEW.

En la figura 4(a) se puede observar como el modelo sigue el patrón de velocidad de referencia, y que, tanto el modelo de LabVIEW como el modelo Matlab/Simulink, presentan la misma respuesta a dicho patrón. De esta forma, se comprueba que ambos modelos tienen el mismo funcionamiento.

Sin embargo, en la figura 4(b) donde se visualiza el par electromagnético que genera la máquina PMSM se puede apreciar que ambos modelos siguen la señal de referencia, pero se aprecia una diferencia entre los resultados obtenidos en ambos. Esta diferencia se debe a que los dos modelos se han ejecutado con distintos tiempos de paso.

Por último, en la figura 4(c) se realiza una comparativa de la corriente que se genera en la misma fase para la aplicación. Aparentemente presentan la misma forma, aunque en algunos

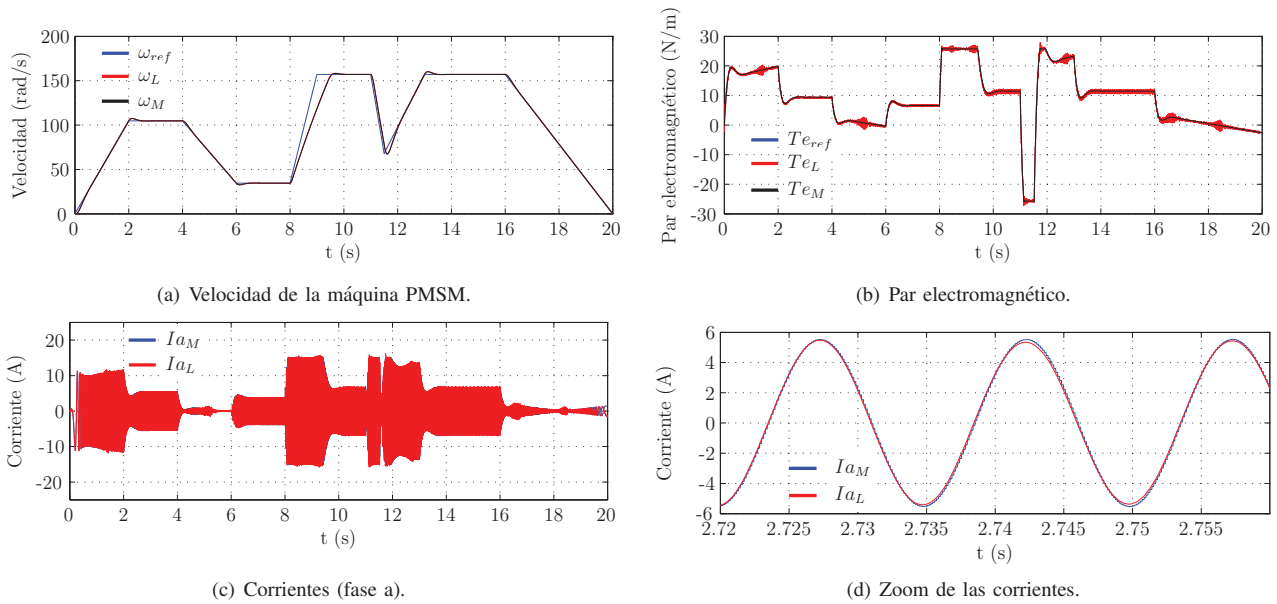


Fig. 4. Medidas realizadas sobre la aplicación de LabVIEW (L) y Matlab (M).

puntos la corriente del modelo LabVIEW es algo distinta del modelo Matlab/Simulink. Pero al ampliar la gráfica como en la figura 4(d) se puede ver que ambos modelos tienen el mismo comportamiento, las diferencias que pueden existir se deben al uso de tiempos de ejecución distintos.

Con los resultados obtenidos, se puede decir que el empleo de una plataforma RT con HW específico para simulación permite obtener resultados válidos que simulan el comportamiento real de los dispositivos, además de permite agilizar el proceso de obtención de resultados. La complejidad de dicha plataforma radica en el proceso de depuración, pero empleando la metodología propuesta en este documento para las plataformas diseñadas con LabVIEW se agiliza el proceso de validación de los modelos MIL a implementar.

VI. CONCLUSIONES

La metodología propuesta en el presente documento permite sintetizar los pasos fundamentales para el diseño de un modelo MIL, del cuál se quieren obtener tanto los modelos RCP como HIL, en una plataforma RT que consta de tres tipos de HW: PC, cluster de procesadores deterministas y FPGA. Donde el uso de HW específico para la simulación evita tener que emplear modelos promediados de los algoritmos que requieren una alta velocidad de ejecución. De esta forma, los resultados que se obtienen se acercan mucho más al comportamiento real de la aplicación. Además, el empleo del entorno LabVIEW es idóneo para el desarrollo de la metodología propuesta, ya que permite la adecuación de los códigos de un HW a otro de la plataforma RT sin tener que realizar grandes modificaciones y sin tener que emplear lenguajes de programación distintos.

VII. AGRADECIMIENTOS

Este trabajo se ha realizado dentro de la Unidad de Investigación y Educación UFI11/16 de la UPV/EHU y con el apoyo del Gobierno Vasco a través de: las ayudas para los grupos de investigación del sistema universitario vasco IT394-10, del proyecto de investigación FUTUREGRIDS-2020 del programa ETORTEK (IE14-389), la financiación del Ministerio de Economía

y Competitividad a través del proyecto de investigación DPI2014-53685-C2-2-R, el apoyo del programa a la formación de investigadores del Gobierno Vasco PRE.2014.1.13 y a National Instruments Spain.

REFERENCIAS

- [1] D. Maksimovic, A. Stankovic, V. Thottuvelil, and G. C. Verghese, "Modeling and simulation of power electronic converters," *Proceedings of the IEEE*, vol. 89, no. 6, pp. 898–912, June 2001.
- [2] A. Sannino, M. Bollen, and J. Svensson, "Voltage tolerance testing of three-phase voltage source converters," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1633–1639, April 2005.
- [3] V. Choudhary, E. Ledezma, R. Ayyanar, and R. Button, "Fault tolerant circuit topology and control method for input-series and output-parallel modular dc-dc converters," *IEEE Transactions on Power Electronics*, vol. 23, no. 1, pp. 402–411, January 2008.
- [4] G. Vazquez, P. Rodriguez, R. Ordonez, T. Kerekes, and R. Teodorescu, "Adaptive hysteresis band current control for transformerless single-phase pv inverters," in *Conference of IEEE Industrial Electronics (IECON)*, 2009, pp. 173–177.
- [5] B. De Kelper, L. Dessaint, V. Do, and J. C. Soumagne, "An algorithm for accurate switching representation in fixed-step simulation of power electronics," in *IEEE Power Engineering Society Winter Meeting*, 2000, pp. 762–767.
- [6] K. Lian and P. Lehn, "Real-time simulation of voltage source converters based on time average method," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 110–118, February 2005.
- [7] E. Ibarra, I. Kortabarria, J. Andreu, I. de Alegria, J. Martin, and P. Ibañez, "Improvement of the design process of matrix converter platforms using the switching state matrix averaging simulation method," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 220–234, January 2012.
- [8] R. Champagne, L. Dessaint, H. Fortin-Blanchette, and G. Sybille, "Analysis and validation of a real-time AC drive simulator," *IEEE Transactions on Power Electronics*, vol. 19, no. 2, pp. 336–345, March 2004.
- [9] C. Dufour, J. Belanger, V. Lapointe, and S. Abourida, "Real-time simulation on FPGA of a permanent magnet synchronous machine drive using a finite-element based model," in *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, 2008, pp. 19–25.
- [10] L. Vesely and L. Pohl, "Modeling and real-time control of permanent magnet synchronous motor on cRIO," in *International Conference on Control, Automation and Systems (ICCAS)*, 2012, pp. 1217–1220.
- [11] C. Dufour, S. Cense, T. Yamada, R. Imamura, and J. Belanger, "FPGA permanent magnet synchronous motor floating-point models with variable-DQ and spatial harmonic finite-element analysis solvers," in *International Power Electronics and Motion Control Conference (EPE/PEMC)*, 2012, pp. LS6b.2–1–LS6b.2–10.