

## Librería de Simulación por Eventos Discretos de Fuentes Conmutadas

Mario Bortolotto<sup>†‡</sup>, Fernando Fontenla<sup>†‡</sup>, Ernesto Kofman<sup>†‡</sup> y Mónica Romero<sup>†</sup>

<sup>†</sup>Laboratorio de Sistemas Dinámicos FCEIA - UNR. Riobamba 245 bis - (2000) Rosario. <sup>‡</sup>CONICET

**Abstract**— En este trabajo se presenta una librería de simulación de fuentes conmutadas de continua utilizando el simulador PowerDEVS y métodos de integración por cuantificación, y diferentes ejemplos de aplicación de la misma en sistemas complejos de control. Los resultados de simulación obtenidos muestran una mejora muy significativa, tanto en precisión como en tiempo de ejecución, respecto de lo que puede obtenerse utilizando herramientas clásicas como Matlab/Simulink.

**Keywords**— Fuentes Conmutadas, Simulación, Métodos de Integración Numérica, DEVS.

### I. INTRODUCCIÓN

Las fuentes de potencias son dispositivos pertenecientes al campo de la electrónica de potencia, utilizadas para la conversión de la potencia disponible. Dicha potencia disponible, que posee un determinado conjunto de características, es procesada por la fuente y transformada de modo de satisfacer los requerimientos especificados. En particular las fuentes conmutadas proveen la potencia necesaria para una carga utilizando componentes de bajo costo como capacitores, inductores y transformadores y el uso de llaves (dispositivos semiconductores) que conmutan entre corte y saturación. La ventaja de esta configuración es que las llaves disipan poca potencia cuando trabajan en estos dos estados y en consecuencia el proceso de conversión se hace con pocas pérdidas lo que se traduce en alta eficiencia. La desventaja asociada a esta forma de trabajo es el ripple de corriente y el stress de los componentes. [2][3][4]

El énfasis en el diseño de una fuente conmutada está puesto en la eficiencia, por lo cual no se utilizan elementos disipadores como las resistencias. El proceso de diseño de una fuente conmutada se considera exitoso cuando la topología y el control resultantes superan los requerimientos de desempeño. El mismo criterio se aplica a las protecciones. Debido a la naturaleza crítica de las fuentes de potencia en todos los equipos, este enfoque mejora tanto el desempeño como la confiabilidad del mismo.

Por todo lo dicho anteriormente se hace necesario contar con herramientas de modelado análisis y simulación de este tipo de fuentes. En la etapa de diseño del control se utilizan modelos promediados (modelos continuos) pero a la hora de evaluarlo se hace evidente la importancia de contar con una herramienta de simulación que permita

implementar modelos de fuentes más realistas.

Estos modelos, al igual que los de la mayor parte de los sistemas de electrónica de conmutación, tienen una representación en forma de ecuaciones diferenciales ordinarias (EDOs) con discontinuidades. La simulación de estos sistemas, por lo tanto, requiere la utilización de métodos numéricos de integración que resuelvan dichas EDOs discontinuas.

La presencia de discontinuidades presenta un problema para dichos algoritmos, ya que las hipótesis de continuidad que dan sustento a los métodos clásicos de integración numérica no son satisfechas [13].

Por este motivo, para obtener resultados aceptables, se deben integrar secciones continuas de la solución y cada vez que tiene lugar una discontinuidad se debe reiniciar el proceso con la siguiente sección continua. Esto requiere detectar exactamente el punto donde se produce la discontinuidad (para lo cual debe utilizarse normalmente un proceso iterativo) y adaptar el paso de manera acorde. Si bien hay distintas técnicas para detectar y tratar las discontinuidades, las mismas coinciden en la necesidad de iterar y adaptar el paso [1, Cap.9].

El problema de las discontinuidades adquiere un matiz particular en los sistemas de electrónica de conmutación, ya que aquí se suma el hecho de que la frecuencia a la que ocurren las discontinuidades es muchas veces más rápida que la dinámica de la parte continua del sistema [12]. Debido a esto, el tratamiento adecuado de las discontinuidades es crucial ya que pequeños errores de detección resultan en cambios cualitativos en las formas de onda simuladas.

Los paquetes de simulación más eficientes y difundidos, tales como Matlab/Simulink [16] y Dymola [2], cuentan con herramientas relativamente eficientes para la detección y el tratamiento de discontinuidades, pero al utilizar métodos de integración clásicos no pueden escapar de los problemas antes mencionados. Por este motivo, las simulaciones de sistemas de electrónica conmutada se tornan muy lentas y los resultados suelen ser poco confiables.

Un enfoque esencialmente distinto al de los métodos clásicos de integración reemplaza la discretización temporal por la cuantificación de las variables de estado, dando lugar a los llamados *Métodos de Integración por Cuantificación* (MIC).

Los MIC se basan en una idea original de Bernard Zeigler, quien propuso que los sistemas continuos podían aproximarse por sistemas de eventos discretos bajo el for-

malismo DEVS [18, 17]. Zeigler mostró que un modelo DEVS podía simular exactamente un integrador con un cuantificador en la salida.

Posteriormente, en [6] se demostró que la cuantificación debía realizarse con el agregado de histéresis (para evitar la aparición de oscilaciones infinitamente rápidas) y con esta idea se formalizó el primer método de integración por cuantificación llamado QSS (por Quantized State System).

Actualmente existen varios métodos de integración por cuantificación: QSS, QSS2 [5], QSS3 [9], BQSS [7], LIQSS y LIQSS2 [11], siendo los últimos tres aptos para sistemas stiff.

Los MIC tienen propiedades teóricas muy fuertes: convergencia, estabilidad [6] y una propiedad notable de existencia de cota de error global calculable [5].

Más allá de las propiedades teóricas, los MIC funcionan de una forma muy particular. Los *pasos* se producen sólo cuando alguna componente del estado cruza un valor predeterminado, o cuando hay discontinuidades en alguna entrada. Además, los pasos son locales a cada variable de estado. Es decir, en cada paso se actualiza una variable de estado y eventualmente se recalcula el tiempo para el próximo cambio en las variables directamente afectadas por dicha actualización. Es decir, el funcionamiento de los MIC es completamente asíncrono y descentralizado.

Además, las trayectorias son conocidas en todo instante de tiempo (son seccionalmente constantes, lineales o parabólicas) por lo cual se pueden predecir exactamente los cruces por cero (o cruces por cualquier valor que implique una discontinuidad del modelo) sin necesidad de iteraciones. Por estos motivos, los MIC son extremadamente eficientes para la integración de sistemas discontinuos [8].

En lo que refiere específicamente a sistemas de electrónica de potencia, hay algunos resultados preliminares en la simulación de fuentes conmutadas [3], y de rectificadores y sistemas de control con inversores [1, Cap.12]. En todos estos casos se observa una disminución en los tiempos de simulación de entre 10 y 20 veces con respecto a lo mejor que puede obtenerse con Matlab/Simulink.

La herramienta de software que implementa los MIC se denomina PowerDEVS [14]. PowerDEVS es un simulador de modelos DEVS que cuenta con una interfaz gráfica similar a la de Simulink (permite editar Diagrama de Bloques, crear nuevos bloques, encapsular, etc.), y funciona generando y compilando código C++. Entre sus librerías, cuenta con todos los bloques necesarios para simular EDOs con los distintos MIC.

En este trabajo, se presenta una librería de modelos de PowerDEVS que incluye las principales topologías de fuentes conmutadas de continua (elevador, reductor, elevador-reductor y Cuk), con dos hipótesis alternativas de modelado: modelos con diodos *ideales* y modelos con diodos con características seccionalmente lineales.

Además de describir los modelos y su implementación en PowerDEVS, se verifica a través de diferentes

ejemplos complejos de aplicación el correcto desempeño de los mismos y se compara la performance con la de los métodos de simulación de Matlab/Simulink. Como se verá, al utilizar PowerDEVS con los MIC, las simulaciones son mucho más robustas y confiables, reduciendo además hasta en 10 veces los tiempos de cálculo (y este factor generalmente se agranda al simular modelos más complejos).

El resto del artículo está organizado de la siguiente forma: La Sección II presenta los modelos de las fuentes y su implementación como librerías de PowerDEVS. Luego, la Sección III introduce dos ejemplos complejos de aplicación, con sus resultados de simulación y la comparación (en términos de precisión, robustez y tiempo de ejecución) con Matlab/Simulink. Finalmente, en la Sección IV se presentan las conclusiones y las futuras direcciones para este trabajo.

## II. MODELOS y LIBRERIAS

Como mencionamos anteriormente, las librerías desarrolladas contienen modelos para cada una de las cuatro topologías bajo dos hipótesis diferentes: diodo *realista* (seccionalmente lineal) y diodo conmutado.

El motivo de utilizar ambas hipótesis es esencialmente numérico. La utilización de una característica *realista* en el diodo produce normalmente que el sistema sea *stiff* [1], lo que a su vez requiere utilizar métodos implícitos que agregan carga computacional a la simulación.

Por otro lado, la utilización de una característica de llave ideal además de restarle cierto realismo al modelo, complica la obtención del mismo, ya que al conmutar de un estado al otro cambia la topología del sistema.

Como veremos más adelante, dependiendo de las características del sistema a simular y de los métodos numéricos empleados, será conveniente utilizar uno u otro modelo.

Por razones de espacio, explicaremos sólo uno de los modelos, correspondiente al circuito buck (ver Fig.1).

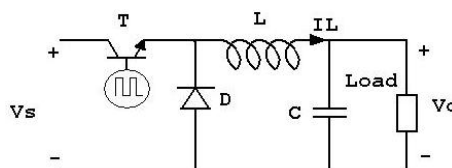


Figura 1: Circuito Buck

### A. Diodo Realista

Asumiendo que el diodo tiene una característica continua y seccionalmente lineal como muestra la Fig.2, las ecuaciones de estado de la fuente resultan las siguientes.

Cuando el transistor está en saturación se tiene

$$\begin{aligned} \dot{I}_L &= 1/L(V_s - V_C) \\ \dot{V}_C &= 1/C(I_L - I_{load}) \end{aligned} \quad (1)$$

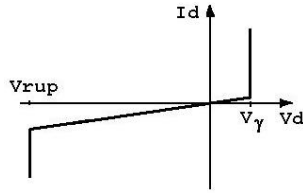


Figura 2: Diodo *real*

mientras que cuando el transistor está cortado, se verifica:

$$\begin{aligned} \dot{I}_L &= 1/L(-V_d(I_L) - V_C) \\ \dot{V}_C &= 1/C(I_L - I_{load}) \end{aligned}$$

En estas ecuaciones,  $I_L$  es la corriente en el inductor,  $V_C$  es la tensión en el capacitor,  $i_{load}$  es la corriente en la carga,  $V_s$  es la tensión de alimentación y  $V_d(I_L)$  es la tensión dada por la característica no lineal del diodo (ver Fig.2).

El Diagrama de Bloques de PowerDEVS correspondiente al sistema se muestra en la Fig.3.

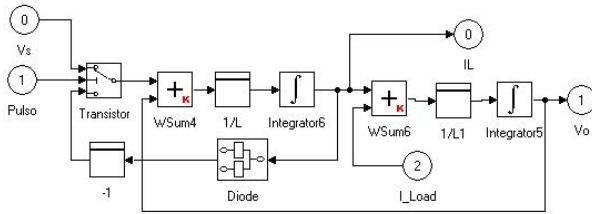


Figura 3: Diagrama de Bloques del Buck

### B. Diodo conmutado

En este caso hay tres situaciones posibles, según el estado de la llave y del diodo (la única situación imposible es que la llave esté cerrada y el diodo conduzca). Dichas situaciones llevan a las siguientes ecuaciones.

Cuando la llave conduce, tenemos nuevamente la Eq.(1). Cuando la llave está abierta y el diodo conduce, resulta

$$\begin{aligned} \dot{I}_L &= 1/L(-V_\gamma - V_C) \\ \dot{V}_C &= 1/C(I_L - I_{load}) \end{aligned}$$

y finalmente con ambos dispositivos en corte tendremos

$$\begin{aligned} \dot{I}_L &= 0 \\ \dot{V}_C &= 1/C(I_L - I_{load}) \end{aligned}$$

El paso a esta última situación se da al verificarse la condición  $I_L = 0$ .

El Diagrama de Bloques de PowerDEVS correspondiente a este nuevo sistema se muestra en la Fig.4.

### C. Librerías en PowerDEVS

Los modelos desarrollados para la fuente buck (que se muestran en las Figs.3 y 4), junto a los modelos análogos de las otras tres topologías (boost, buck-boost y Cuk)

fueron incorporados a una nueva librería de PowerDEVS, que se muestra en la Fig. 5.

Cada uno de los modelos de la librería tiene 3 entradas. La primera es la tensión de alimentación  $V_s$ . La segunda es la señal de comando de la llave, que funciona con la siguiente lógica: cuando  $v > 0.5$  la llave conduce. En otro caso, está cortada. La última entrada en tanto, es la corriente de carga.

Los modelos a su vez tienen una salida principal, que es la tensión de salida  $V_C$ . Además, hay salidas adicionales para obtener las restantes variables de estado ( $I_L$ , y en el caso del Cuk dos salidas más para el inductor y el capacitor adicional).

Por otro lado, los modelos tienen varios parámetros que pueden modificarse haciendo doble click sobre los mismos: valor de las inductancias y capacitores, tensión de conducción en el diodo  $V_\gamma$ , quantum utilizado para la simulación en los almacenadores (inductores y capacitores) y método de integración (QSS, QSS2, etc.).

## III. RESULTADOS DE APLICACIÓN

En esta sección presentamos dos ejemplos de aplicación de los modelos de la librería para observar su desempeño en distintas condiciones y comparar los resultados de simulación con los de los métodos de integración numéricos de Matlab/Simulink.

Todas las simulaciones reportadas fueron realizadas con una computadora con un microprocesador Intel Celeron de 1.4Ghz, y 512Mb de memoria RAM.

Ejemplos adicionales de aplicación pueden encontrarse en [4].

### A. Control PI de una Fuente Boost

Utilizando el modelo Boost con diodo conmutado, se aplicó un control PI seguido de un PWM para regular la tensión de salida de la fuente. Los parámetros utilizados para la fuente son:  $L = 150\mu H$ ,  $C = 220\mu F$ ,  $V_s = 5V$ ,  $f = 25kHz$ . Como set point de tensión fue utilizada una rampa de valor final 20V y tiempo de crecimiento de 0.2seg. La Fig.6 muestra el Diagrama de Bloques de PowerDEVS con el que se realizó la simulación.

Las trayectorias de salida  $V_o$  obtenida con ambos simuladores se muestran en las Figs.7-8. Puede notarse en-

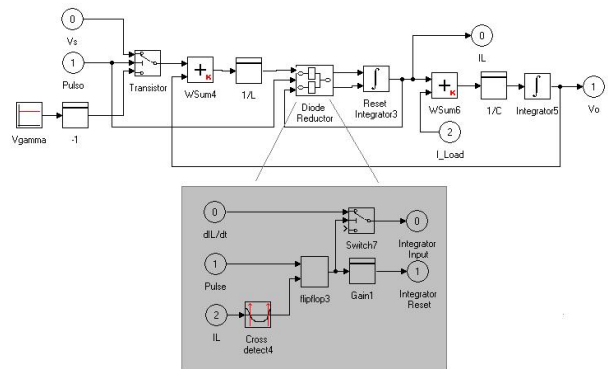


Figura 4: Diagrama de Bloques del Buck

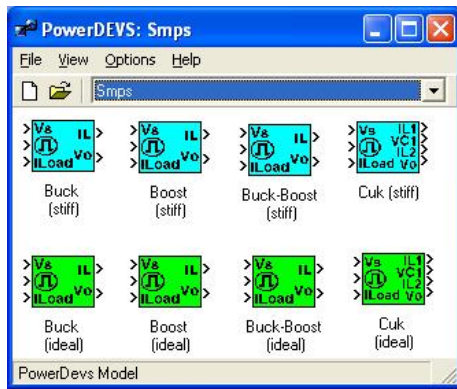


Figura 5: Librería de Fuentes Conmutadas

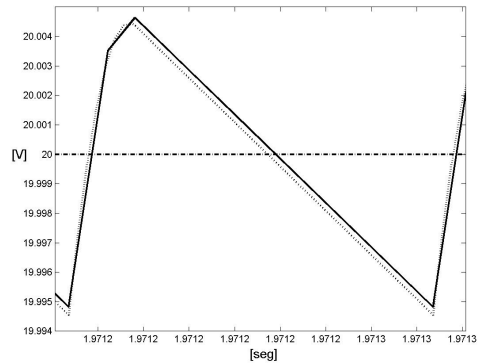


Figura 8: Tensión en el Boost (detalle del ripple)

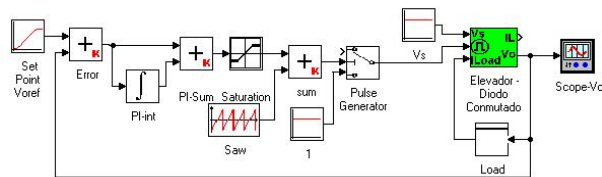


Figura 6: Control PI de una Fuentes Boost

tre ambos una muy pequeña diferencia en el ripple.

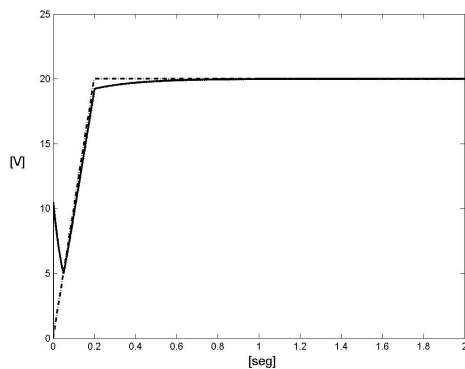


Figura 7: Tensiones de referencia y de salida en el Boost

La simulación en PowerDEVS se realizó con el método de QSS2, utilizando quantum de 0.3, 0.15 y 0.01 en los integradores que calculan  $V_C$ ,  $I_L$  y el término integral del error respectivamente. Para un tiempo final de simulación  $t_f = 3\text{seg}$ . PowerDEVS realizó un total de 520714 pasos y utilizó 7.87seg.

En cuanto a Matlab/Simulink, para conseguir que la simulación arroje resultados aceptables se debió limitar el máximo paso de integración tal que  $h_{max} = 1 \times 10^{-6}$ . De otra manera, se producen errores muy grandes debido a que la simulación se *saltea* algunas conmutaciones. En este caso, el método que mejor funcionó fue el ODE23, que realizó 2815619 pasos y (en modo acelerado) completó la simulación en 32.3seg.

En este ejemplo, si bien la reducción del tiempo de simulación de PowerDEVS no es tan grande, hay una im-

portante ventaja de robustez. Más allá del quantum elegido en este caso, cualquier elección del mismo permite visualizar resultados cualitativamente correctos con una resolución del orden dictado por el quantum. Al simular con los métodos de Matlab/Simulink, en cambio, los resultados suelen ser cualitativamente incorrectos y hay que recurrir a ciertos *trucos* tales como limitar el paso de integración.

## B. Control de Velocidad de un Motor de Continua

En la Figura 9 se muestra el Diagrama de Bloques de PowerDEVS con esquema de control de velocidad de un motor de corriente continua (MCC) utilizado para esta simulación. En este sistema se observan dos lazos de control anidados que regulan la velocidad de un MCC de imán permanente alimentado a través de una fuente conmutada tipo buck. Esta fuente a su vez posee un control PWM con lazo de tensión.

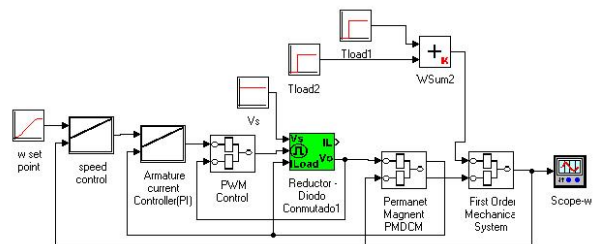


Figura 9: Control de Velocidad de un MCC

Cada uno de los lazos mencionados implementa un controlador tipo PI cuyas salidas son valores de referencia (ciclo de trabajo del pulso,  $I_{a\_ref}$  y  $U_{a\_ref}$ ) en función de la referencia de velocidad que se desea seguir. Los controles PI de velocidad y corriente fueron diseñados siguiendo el método de [15] de modo tal que el MCC pueda seguir la referencia de velocidad y rechazar perturbaciones de carga, tales como un escalón de torque. Los valores de los parámetros del sistema son los siguientes:

- Motor:  $P = 20\text{HP}$ ,  $R_a = 180\text{m}\Omega$ ,  $L_a = 2.93\text{mHy}$ ,  $J_m = 0.156\text{Kgm}$ ,

$b = 0.0044 Nm \cdot sec / rad$ ,  $Wf = 200 Wb$ ,  
 $U_{a_{nom}} = 230 V$ ,  $n_{nom} = 1750 rpm$  ( $\omega_{nom} = 183.26 rad / sec$ ).

- Fuente Buck:  $L = 0.5 mH$ ,  $C = 2 mF$ ,  $V_s = 240 V$ ,  $f = 10 kHz$ .
- PI lazo tensión:  $K_{pi} = 0.1$ ,  $T_i = 0.67$ .
- PI lazo de corriente:  $K_{pi} = 0.0148$ ,  $T_i = 0.321$ .
- PI lazo de velocidad:  $K_{pi} = 1.22$ ,  $T_i = 2.22$ .

Se implementaron dos modelos de simulación, uno en PowerDEVS utilizando el modelo de la fuente Buck de la librería desarrollada (con la hipótesis de diodo como llave ideal) y otro en Matlab/Simulink.

El ensayo de simulación consiste en aplicar una referencia de velocidad en forma de rampa que al cabo de tres segundos alcanza un valor final de  $135 rad / sec$  ( $0.7 \cdot \omega_{nom}$ ). El sistema arranca además cargado con un torque de  $5 Nm$ , y a los 5 segundos se suma un escalón de  $30 Nm$  ( $0.3 T_{nom}$ ). El tiempo total de simulación se fijó en  $t_f = 10$  seg de modo de poder observar todos los fenómenos transitorios hasta llegar al régimen permanente estacionario.

La Figura 10 muestra la velocidad del motor y la velocidad de referencia para el ensayo simulado en PowerDEvs, mientras la Figura 11 muestra la tensión de referencia que fija el lazo de corriente y la tensión de salida del convertidor. Por último la Figura 12 muestra el detalle del ripple en la tensión de salida del convertidor, y compara los resultados obtenidos con PowerDEVS y con Matlab/Simulink.

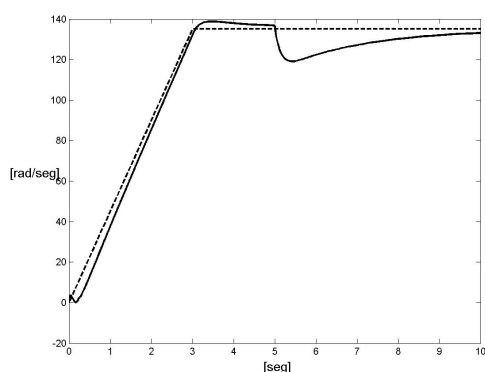


Figura 10: Referencia y velocidad del MCC

Como puede verse en el detalle de la Fig.12, los resultados son de muy alta calidad. En comparación, los resultados con PowerDEVS y Simulink son muy similares, sin embargo los costos computacionales difieren mucho en este caso.

En PowerDEVS se utilizó el método de QSS2 con las siguientes cuantificaciones: 0.05 en el PI de velocidad, 0.1 en el de corriente, 0.01 en el de tensión, 0.3 en  $V_C$ , 0.15 en  $I_L$ , 0.05 en  $\omega$  y 0.1 en  $I_a$ .

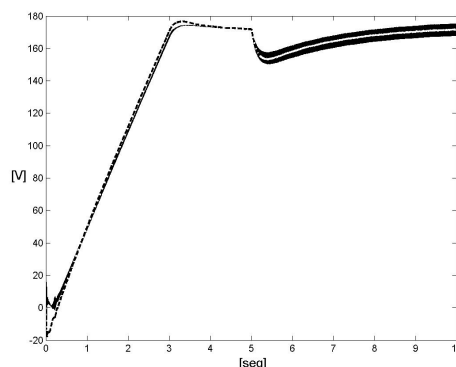


Figura 11: Referencia y tensión de salida del Buck en el MCC

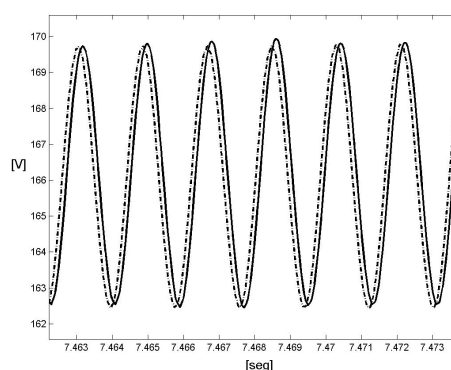


Figura 12: Tensión de salida del Buck (detalle del ripple)

Para los 10 segundos de tiempo final simulado, PowerDEVS realizó 520884 pasos, lo que llevó 6.36 segundos (en este caso la velocidad de simulación fue superior al tiempo real).

La simulación del modelo implementado en Matlab/Simulink sólo funcionaba bien fijando ciertos puntos de simulación de manera externa, esto es, diciéndole al simulador en que momentos debe realizar cálculos de las variables de estado (lo que no es una solución aceptable en general), o bien limitando el paso máximo de integración (esto es lo que se decidió hacer).

Para obtener resultados como los mostrados en la Figura 12 el máximo paso de integración debió limitarse a  $h_{max} = 1 \times 10^{-6}$ , obteniéndose ahora 10595608 pasos que tomaron 88.18 segundos con el método ODE23 (nuevamente el que mejor funcionó de los de Matlab).

Como en el case anterior, vale destacar que la simulación con QSS2 es mucho más robusta que la de los métodos de Simulink. Aunque los quantum utilizados para la comparación están optimizados, con cualquier elección se obtienen resultados cualitativamente correctos y sin un aumento significativo de los costos computacionales.

Es importante mencionar que este último modelo es de una complejidad importante. En un caso como este, es casi imposible diseñar el sistema de control sin pasar

por una etapa de verificación mediante simulación. En tal caso resulta crítico contar con una herramienta confiable para analizar los resultados. En este sentido, los métodos de QSS ofrecen garantía de la calidad de los resultados con una resolución del orden del quantum elegido.

#### IV. CONCLUSIONES

En este trabajo se presentó una nueva librería de PowerDEVS que contiene las 4 topologías básicas de Fuentes Conmutadas.

Se verificó la capacidad de PowerDEVS y de los métodos de QSS para simular este tipo de sistemas conmutados, observándose importantes ventajas respecto a los enfoques clásicos de tiempo discreto (particularmente respecto a Matlab/Simulink). Se observó además que, en general, al agregar realismo y complejidad a los modelos a simular, la diferencia a favor de PowerDEVS (tanto en tiempo como en precisión) se hace cada vez más notoria. Los métodos de QSS también mostraron mayor robustez que Simulink en cuanto al ajuste de los parámetros numéricos (quantum y tolerancia) para obtener una simulación coherente

Si bien el desarrollo de las librerías fue el resultado de un proyecto de fin de carrera [4], el trabajo está enmarcado en un proyecto más amplio para el desarrollo de herramientas de simulación en tiempo real de sistemas de electrónica de potencia. Por tal motivo, el trabajo futuro estará orientado a implementar en tiempo real las simulaciones (para lo cual se requiere optimizar muchos aspectos actuales de PowerDEVS) por un lado, y por el otro a ampliar las librerías desarrolladas para cubrir otras topologías y aplicaciones.

Desde el punto de vista estrictamente numérico, uno de los mayores inconvenientes de los métodos de QSS tiene que ver con la necesidad de elegir el quantum en cada variable. Una solución que debería implementarse en las librerías es la de utilizar cuantificación logarítmica, que realiza un control intrínseco del error relativo de la simulación [10].

#### Referencias

- [1] François Cellier and Ernesto Kofman. *Continuous System Simulation*. Springer, New York, 2006.
- [2] H. Elmqvist, D. Brueck, and M. Otter. *Dymola User's Manual*. Dynasim AB, Research Park Ideon, Lund, Sweden, 1995.
- [3] Flavia Felicioni and Ernesto Kofman. Simulación por Eventos Discretos de Sistemas de Electrónica de Potencia. In *Proceedings of RPIC'03*, volume 1, pages 134–139, San Nicolas, Argentina, 2003.
- [4] Fernando Fontenla and Mario Bortolotto. Fuentes Conmutadas DC-DC en PowerDEVS. Modelización y simulación. Proyecto Final de Ingeniería, 2007.
- [5] E. Kofman. A Second Order Approximation for DEVS Simulation of Continuous Systems. *Simulation*, 78(2):76–89, 2002.
- [6] E. Kofman and S. Junco. Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS*, 18(3):123–132, 2001.
- [7] E. Kofman, G. Migoni, and F.E. Cellier. Integración por Cuantificación de Sistemas Stiff. Parte I: Teoría. In *Proceedings of AADECA 2006*, Buenos Aires, Argentina, 2006.
- [8] Ernesto Kofman. Discrete Event Simulation of Hybrid Systems. *SIAM Journal on Scientific Computing*, 25(5):1771–1797, 2004.
- [9] Ernesto Kofman. A Third Order Discrete Event Simulation Method for Continuous System Simulation. *Latin American Applied Research*, 36(2):101–108, 2006.
- [10] Ernesto Kofman. Relative Error Control in Quantization Based Integration. Number LSD0307. LSD–UNR, 2007. Disponible en [www.fceia.unr.edu.ar/kofman](http://www.fceia.unr.edu.ar/kofman).
- [11] G. Migoni and E. Kofman. Linearly Implicit Discrete Event Methods for Stiff ODEs. Part I: Theory. Technical Report LSD0107, LSD–UNR, 2007. Enviado a RPIC 2007.
- [12] N. Mohan, W. Robbins, T. Undeland, R. Nilssen, and O. Mo. Simulation of Power Electronic and Motion Control Systems – An Overview. *Proceedings of the IEEE*, 82(8):1287–1302, 1994.
- [13] M. Otter and F.E. Cellier. *The Control Handbook*, chapter Software for Modeling and Simulating Control Systems, pages 415–428. CRC Press, Boca Raton, FL, 1996.
- [14] Esteban Pagliero, Marcelo Lapadula, and Ernesto Kofman. PowerDEVS. Una Herramienta Integrada de Simulación por Eventos Discretos. In *Proceedings of RPIC'03*, volume 1, pages 316–321, San Nicolas, Argentina, 2003.
- [15] Gerhard Pfaff. *Regelung elektrischer antriebe I*. R. Oldenbourg Verlag, 1990.
- [16] L. Shampine and M. Reichelt. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.
- [17] B. Zeigler, T.G. Kim, and H. Praehofer. *Theory of Modeling and Simulation. Second edition*. Academic Press, New York, 2000.
- [18] B. Zeigler and J.S. Lee. Theory of quantized systems: formal basis for DEVS/HLA distributed simulation environment. In *SPIE Proceedings*, volume 3369, pages 49–58, 1998.