Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Posgrado y Educación Continua

Tesis Doctoral

Simulación eficiente de Inversores PWM Monofásicos en Modo Diferencial

Ing. Mario L. Bortolotto

Director: Dr. Ing. Gustavo Migoni

Miembros del Jurado:

Dr. Ing. Rodrigo Daniel Castro
Dr. Ing. Matías Nacusse
Dr. Ing. Luis Ignacio Silva

Tesis presentada en la Facultad de Ciencias Exactas, Ingeniería y Agrimensura, en cumplimiento de los requisitos para optar al título de
Doctor en Ingeniería

11 de noviembre de 2025

Certifico que el trabajo incluído en esta tesis es el resultado de tareas de investigación originales y que no ha sido presentado para optar a un título de postgrado en ninguna otra Universidad o Institución.

Mario Luciano Bortolotto

RESUMEN

Esta tesis se centra en el desarrollo de estrategias novedosas para abordar los desafíos que presenta la simulación de inversores PWM, fundamentales en el diseño y análisis de la electrónica de potencia. La presencia de conmutaciones de alta frecuencia introduce discontinuidades frecuentes en los modelos matemáticos, lo que dificulta su integración numérica cuando se emplean modelos realistas (modelos conmutados). Como resultado, el costo computacional y los requerimientos de procesamiento pueden volverse excesivos.

Existen estrategias de modelado que permiten simular largos períodos de tiempo con menor carga computacional, como los denominados modelos promediados. Sin embargo, estas estrategias sacrifican detalles en la simulación y/o precisión para lograr mejoras en los tiempos de cómputo.

Con el fin de mejorar la eficiencia sin sacrificar detalles en los resultados, esta tesis propone dos enfoques complementarios. Primero, se estudian los beneficios de emplear métodos de integración numérica basados en la cuantificación de estados (QSS, por Quantized State Systems) para simular modelos conmutados de inversores PWM. Segundo, se presenta una estrategia de modelado original denominada Modelado Mixto de Inversores PWM Monofásicos en Modo Diferencial, que combina el uso de modelos promediados y conmutados en un único esquema.

Los modelos desarrollados fueron implementados en Modelica y μ-Modelica, y las simulaciones se realizaron con OpenModelica, Wolfram System Modeler y la herramienta Stand-Alone QSS Solver, que permite comparar métodos QSS y métodos clásicos de integración numérica.

Los estudios comparativos demuestran que las estrategias propuestas reducen significativamente los tiempos de simulación sin comprometer la precisión, aportando soluciones innovadoras para la simulación eficiente de inversores PWM.

Palabras claves: inversores PWM, Modelado mixto, Métodos QSS, simulación eficiente.

ABSTRACT

This thesis focuses on the development of novel strategies to address the challenges posed by the simulation of PWM inverters, which are fundamental in the design and analysis of power electronics. The presence of high-frequency switching introduces frequent discontinuities in the mathematical models, which hinders their numerical integration when using realistic (switched) models. As a result, computational cost and processing requirements can become excessively high.

There are modeling strategies that allow the simulation of long time intervals with lower computational load, such as the so-called averaged models. However, these strategies often sacrifice simulation detail and/or accuracy in order to improve computation time.

To improve efficiency without sacrificing detail in the results, this thesis proposes two complementary approaches. First, it examines the benefits of using numerical integration methods based on quantized state systems (QSS) to simulate switched models of PWM inverters. Second, it introduces an original modeling strategy called Mixed Modeling of Single-Phase PWM Inverters in Differential Mode, which combines the use of averaged and switched models within a single scheme.

The developed models were implemented in Modelica and μ-Modelica, and the simulations were performed using OpenModelica, Wolfram System Modeler, and the Stand-Alone QSS Solver, a tool that allows comparison between QSS methods and classical numerical integration techniques.

Comparative studies show that the proposed strategies significantly reduce simulation times without compromising accuracy, providing innovative solutions for the efficient simulation of PWM inverters.

Keywords: PWM inverters, mixed modeling, QSS methods, efficient simulation.

AGRADECIMIENTOS

Este trabajo fue posible gracias al apoyo, motivación y la colaboración de muchas personas.

En primer lugar, agradezco profundamente la paciencia, la dedicación y la perseverancia de mi director, Gustavo Migoni, quien me acompañó a lo largo de todo el proceso de realización de esta tesis, ayudándome a atravesar cada una de las dificultades que se presentaron en el camino. También al director del grupo de investigación, Ernesto Kofman, por volver a abrirme las puertas para continuar con mi desarrollo como investigador.

A Joaquín Fernández, Mariana Bergonzi, Franco Sansone, Mónica Romero, Noelia Pizzi y otres compañeres del CIFASIS, que de una u otra manera hicieron posible llegar a esta instancia. A mis compañeres de cátedra de Física II y al Departamento de Fisicoquímica de la Escuela de Formación Básica de la FCEIA, por el acompañamiento y el respaldo, especialmente en el tramo final de este trabajo. Agradezco también a la Escuela de Posgrado de la FCEIA por su apoyo institucional.

Un especial agradecimiento a los jurados de esta tesis, cuyos comentarios y sugerencias fueron fundamentales para mejorar este trabajo y permitir que alcanzara el nivel académico y científico necesario.

En lo personal, agradezco profundamente a mi familia y seres queridos, fuerza vital para llegar hasta aquí. A mi madre, María Cristina López de Bortolotto, y a la memoria de mi padre, Aldo Mario Bortolotto, por su amor incondicional y por ser siempre sostén y ejemplo. A mis hermanos, Leonardo y Diego. A mi compañera y amor de mi vida, Ariadna Guerrero, y a mis adorables hijos, Camilo Ernesto e Inti Fidel, quienes son mi motor, mi alegría y el fuego que me impulsa a seguir adelante. A mis suegros, Rafael y Nancy, por su cercanía y acompañamiento. A mis querides amiges Laura Cadars, Natacha Scherbovsky, Andrés Carminati, María Celina Añaños, Gustavo Brufman y Juan Manuel Monti por su apoyo emocional constante.

Por último, a todes les compañeres de lucha que me alentaron para concluir este trabajo. Con su compromiso me siguen dando esperanza de que otro futuro es posible. A quienes defienden cada día la Universidad Pública, gratuita, libre e irrestricta, que me brindó la oportunidad de formarme, trabajar y llegar hasta aquí.

ÍNDICE GENERAL

ÍNDICE GENERAL

LISTA DE ACRÓNIMOS

AVM Modo Promediado (*Averaged Mode*)

CA corriente alterna

CC corriente continua

CC–CA corriente continua a corriente alterna

CC–CC corriente continua a corriente continua

CPU Central Processing Unit (Unidad Central de Procesamiento)

CVODE–BDF *C-variable Ordinary Differential Equation solver with Backward Differentiation Formula*

DASSL *Differential Algebraic System Solver*

DMSI Inversor Monofásico en Modo Diferencial

DMSIs Inversores Monofásicos en Modo Diferencial

EDO Ecuaciones Diferenciales Ordinarias

LIQSS *Linearly Implicit Quantized State System*

MCC Modo de Conducción Continua

MCD Modo de Conducción Discontinua

MIC Métodos de Integración Cuantificada

PSS Régimen Permanente Periódico (*Periodic Steady-State*)

PWM Modulación por Ancho de Pulsos

QBI *Quantization-Based Integration*

QSS Sistemas de Estado Cuantificado

QSS Solver Simulador Autónomo de QSS (*Stand-Alone QSS Solver*)

SPDT Interruptor unipolar de dos posiciones (*Single Pole Double Throw*)

SWM Modo Conmutado (*Switched Mode*)

1. INTRODUCCIÓN

La Electrónica de Potencia cumple un rol clave en el desarrollo de sistemas eléctricos sostenibles y eficientes, especialmente en contextos de transición energética [25, 22]. En un mundo donde el uso de combustibles fósiles sigue siendo predominante, contribuyendo significativamente a la crisis climática y sus consecuencias devastadoras [39], la transición hacia fuentes renovables y sistemas energéticos inteligentes es ineludible [78]. Este desafío requiere soluciones innovadoras que permitan integrar tecnologías avanzadas, como la electrónica de potencia, en la generación, distribución y consumo de electricidad.

En Argentina, la matriz de generación eléctrica refleja una fuerte dependencia de fuentes térmicas basadas en combustibles fósiles ($59\,\%$), seguidas por hidroeléctricas ($26{,}5\,\%$) y energías renovables ($10{,}7\,\%$, principalmente eólica)[12]. Esta composición evidencia la necesidad urgente de incrementar la participación de energías limpias en la matriz energética nacional.

Debido al creciente uso de tecnologías dependientes de energía eléctrica, resulta necesario contar con infraestructuras capaces de integrar fuentes renovables de manera eficiente [61]. Aquí radica la importancia de la electrónica de potencia, que permite optimizar la conversión, control y distribución de la energía en redes eléctricas modernas.

En este contexto, la integración de fuentes renovables (como las fotovoltaicas) a la red de distribución en corriente alterna requiere el uso de inversores de potencia [35]. En particular, para sistemas de energía renovable de baja potencia, es común utilizar Inversores Monofásicos en Modo Diferencial (DMSIs) con el objetivo de convertir la energía eléctrica de manera eficiente [1]. Estos inversores son controlados mediante Modulación por Ancho de Pulsos (PWM), en adelante denominados DMSIs PWM, se utilizan para convertir señales de continua en una señal que aproxima a una señal de alterna sinusoidal. El nivel de contenido armónico de esta aproximación, y por ende la calidad de la misma, depende fundamentalmente de la frecuencia de la señal moduladora del PWM. A mayor frecuencia de esta señal, menor es el contenido armónico.

El desarrollo de estos sistemas de electrónica de potencia depende, en gran medida, de la simulación computacional, que permite analizar y diseñar equipos de manera eficiente y segura. Las simulaciones no solo facilitan el estudio de condiciones normales y anormales sin riesgos físicos, sino que también reducen costos y acortan los tiempos de desarrollo

[63].

En este contexto, el proceso de modelado juega un papel central, ya que consiste en extraer, organizar y representar, de forma precisa, el conocimiento del sistema físico [15]. Un modelo bien estructurado permite realizar simulaciones confiables, capaces de predecir el comportamiento del sistema real bajo diversas condiciones [81, 37].

En particular, el proceso de análisis, diseño y resolución de DMSIs PWM permite a los profesionales de la ingeniería comprender profundamente el comportamiento de los circuitos, seleccionar topologías adecuadas, dimensionar componentes y estimar el rendimiento bajo diferentes condiciones operativas[88]. Además, permiten anticiparse a variaciones en los valores de los componentes y de las condiciones de operación, asegurando un diseño robusto incluso en casos extremos.

La principal dificultad en el modelado y simulación de modelos conmutados de DMSIs PWM, los cuales contemplan toda la dinámica generada por los cambios de estado de los elementos de conmutación tales como transistores y diodos, radica en cómo modelar estos componentes y poder simular los modelos obtenidos en tiempos aceptables.

Por un lado, si se consideran los elementos de conmutación como llaves ideales, los modelos resultantes presentan cambios de estructura y no pueden ser simulados con las herramientas estándar de simulación. Por otro lado, si estos componentes se modelan como impedancias que cambian entre valores muy altos y muy bajos según el estado de conducción que corresponda, los modelos resultantes presentan saltos en los valores de las derivadas de las variables de estado del modelo. Esto se produce debido a cambios en los parámetros de las ecuaciones diferenciales que describen el sistema [26].

Esta última forma de modelar los elementos de conmutación presenta una dificultad para los métodos de integración basados en la discretización temporal ya que requieren continuidad en las variables de estado entre pasos de integración [32, 74, 50, 23]. De no cumplirse esta condición, para garantizar el correcto funcionamiento de los métodos se deben utilizar algoritmos costosos que detecten los instantes de ocurrencia de discontinuidades, den un paso siempre en el instante de la discontinuidad y reiniciar la simulación a partir de ese punto [15]. A esta dificultad se suma que los modelos resultantes con este enfoque suelen ser rígidos, lo que exige el uso de métodos de integración implícitos [33]. Estos métodos requieren inversiones matriciales lo cual incrementa significativamente la demanda de Unidad Central de Procesamiento (CPU).

El elevado costo de los algoritmos de detección de discontinuidades, junto con la necesidad de reinicializar los métodos de integración y emplear métodos implícitos, provoca que la simulación de modelos de DMSIs PWM —donde la frecuencia de discontinuidades es muy alta— requiera tiempos de CPU excesivamente largos. En muchos casos, el tiempo de ejecución resulta inaceptable incluso con los algoritmos más eficientes [59]. Esto se debe a que, además, los tiempos de simulación utilizados normalmente son mucho mayores que el

tiempo entre discontinuidades por lo que la simulación de estos modelos requiere procesar un número muy alto de discontinuidades.

Los problemas mencionados para la simulación de DMSIs escalan cuando estos convertidores forman parte de sistemas más complejo que conforman sistemas de gran dimensión. En este caso, el uso de modelos conmutados es prácticamente prohibitivo por la excesivamente alta demanda de CPU.

Si bien los avances en capacidad computacional han hecho que la simulación de los modelos de DMSIs sea más rápida y permite manejar modelos más complejos, la simulación de estos convertidores o de sistemas que los integren sigue siendo un desafío. Esto resalta la necesidad de enfoques alternativos que permitan mejorar la eficiencia en la simulación.

En este sentido, en los últimos años se ha observado un creciente desarrollo de arquitecturas de *co-simulación híbrida* orientadas a la integración de modelos continuos y discretos en entornos de ejecución distribuidos [40, 6, 27, 70].

Para lograr integrar las distintas herramientas para la co-simulación híbrida cobra vital importancia la denominada *Functional Mock-up Interface* (FMI) [42]. La FMI es un estándar abierto desarrollado originalmente por el proyecto MODELISAR (iniciado por Daimler y Modelica Association) y hoy lo mantiene la Modelica Association Project FMI. La misma permite que modelos creados en distintas plataformas puedan comunicarse o simularse juntos sin depender de un software específico.

Actualmente existen trabajos que analizan en profundidad las limitaciones del estándar FMI para representar adecuadamente el tiempo y los eventos discretos en co-simulaciones híbridas [19, 34]. Una de las propuestas consiste en una extensión basada en modelos de "tiempo superdenso" que facilita la sincronización precisa entre dominios [65].

Estas limitaciones resultan especialmente relevantes en electrónica de potencia, donde los inversores PWM y los sistemas de control embebido generan eventos de conmutación de alta frecuencia, por lo que se requiere una co-simulación híbrida que maneje eficazmente tanto dinámicas continuas como eventos discretos.

Entre las estrategias actuales para reducir los tiempos de simulación de sistemas eléctricos de gran escala con alta integración de convertidores, se destaca también la simulación paralela. Este enfoque distribuye el cálculo entre múltiples núcleos o instancias de software, resolviendo sub-redes desacopladas mediante estándares de co-simulación con FMI, y logra importantes mejoras de desempeño sin comprometer la precisión de los resultados [66].

Aún utilizando estas estrategias, uno de los procesos más demandantes radica en la simulación propiamente de los convertidores e inversores de potencia [52].

Para poder realizar simulaciones más rápidas de DMSIs PWM existen estrategias alternativas a la de los modelos conmutados. Estas estrategias mejoran los tiempos de CPU sacrificando detalle o precisión y, en algunos casos, enmascaran fenómenos transitorios. El

enfoque más ampliamente utilizado por esta razón es el uso de modelos promediados en lugar de modelos conmutados [83, 64, 44, 77, 89].La idea básica detrás de esta estrategia consiste en obtener un modelo continuo (eliminando las discontinuidades) que represente el comportamiento promedio de las variables de estado del sistema [3, 25, 87]. Si bien estos modelos reducen significativamente el costo computacional, pierden detalle de las dinámicas rápidas introducidas por las conmutaciones y de otras posibles condiciones de operación que se alcancen debido a estas dinámicas no representadas. Además, se puede perder información valiosa en los estados transitorios o fenómenos generados debido a la interacción de las dinámicas rápidas no representadas con otras partes de modelo cuando los convertidores forman parte de sistemas mas complejos.

Para compensar la falta de información sobre el ripple en los modelos promediados, algunas estrategias de modelado proponen superponerlo a la dinámica promediada [60]. Sin embargo, esta estrategia resulta poco práctica cuando el objetivo es simplemente construir y simular un modelo de circuito de manera ágil, especialmente en situaciones donde se requiere evaluar cambios en la topología y la amplitud del ripple es un factor relevante. Esto se debe a que para poder realizar modelos promediados y superponerle a los mismos la señal de ripple se debe realizar mucho trabajo matemático previo [3] y, cualquier alteración en la topología implica volver a realizar todo ese trabajo. Además, este tipo de modelos introduce fuentes de error relevantes, ya que se basa en suposiciones simplificadas sobre la forma del ripple, omite su acoplamiento dinámico con las variables promediadas y no resulta adecuado para simular transitorios rápidos. Asimismo, al no modelar explícitamente los elementos conmutados, estos enfoques no permiten capturar con fidelidad fenómenos como la interacción no lineal entre armónicos, la modulación por saturación de componentes magnéticos o la sensibilidad del sistema ante pequeñas perturbaciones de alta frecuencia, lo cual puede ser crítico en aplicaciones con control digital o en contextos donde se evalúa la compatibilidad electromagnética (EMC) [51].

Otra estrategia aplicada recientemente para simular, de manera eficiente, convertidores de potencia integrados en sistemas de mayor dimensión es la denominada reducción de modelos [84]. Esta técnica busca disminuir el orden matemático y la complejidad del sistema, preservando sus principales comportamientos y características dinámicas ante pequeñas perturbaciones. Esta técnica resulta especialmente útil en sistemas con múltiples convertidores, donde se requieren simulaciones extensas. No obstante, su implementación exige un trabajo previo significativo antes de poder realizar la simulación numérica. Esto limita su aplicabilidad en etapas tempranas de diseño o en contextos donde se requiere flexibilidad para modificar el sistema.

Si bien todas estas estrategias permiten obtener soluciones rápidas que son útiles en muchas situaciones, existen muchos casos donde se necesita obtener resultados que reflejen los fenómenos asociados a las conmutaciones. Por ejemplo, dado que el ripple genera

voltajes y corrientes no deseados en los componentes del sistema, cuando se desea estudiar la confiabilidad de los componentes y predecir posibles fallas es necesario realizar simulaciones precisas que contemplen las conmutaciones y permitan análisis a largo plazo [63].

Además, cabe destacar que, aunque estos dispositivos suelen operar en régimen periódico estable, cuando se integran en sistemas más complejos pueden verse afectados por variaciones y perturbaciones a largo plazo [17], lo que podría modificar su régimen de operación. Por ejemplo, en plantas de generación fotovoltaica, la cobertura intermitente de nubes provoca fluctuaciones transitorias en la irradiancia solar, afectando el rendimiento del sistema. Por este motivo, se deben realizar simulaciones a largo plazo que incorporan estas perturbaciones y permiten evaluar de manera integral la respuesta del sistema ante condiciones reales. Esto facilita el análisis de su robustez, confiabilidad y precisión en la predicción del rendimiento energético [79].

Debido a los problemas antes mencionados para simular modelos conmutados de DMSIs, cuando se requieren resultados que reflejen tanto los fenómenos de régimen periódico estable como los transitorios, es común emplear distintos modelos para analizar el comportamiento transitorio y a largo plazo de los DMSIs [18]. Los modelos conmutados se utilizan principalmente para estudiar respuestas transitorias, mientras que los modelos promediados se prefieren para el análisis a largo plazo [55, 41]. Sin embargo, esta metodología implica un doble esfuerzo y la necesidad de realizar múltiples simulaciones para capturar los diversos transitorios que pueden ocurrir durante la evolución a largo plazo. Lo que reduce la eficiencia del proceso de diseño y análisis, especialmente en aplicaciones que requieren evaluación continua del comportamiento dinámico.

Por todos estos motivos, en la simulación de estos sistemas de electrónica de potencia se debe lograr un equilibrio óptimo entre tres factores fundamentales: velocidad de simulación, bajo costo computacional y precisión en los resultados [41]. Por lo tanto, en la simulación eficiente de DMSIs PWM es esencial la adecuada elección de la estrategia de modelado utilizada y del método de integración numérica seleccionado.

En este trabajo se proponen estrategias alternativas que abordan las limitaciones y objetivos descriptos que permiten obtener simulaciones más eficientes y con mayor detalle para los DMSIs PWM. Esta contribución, no solo permite mejorar el diseño, desempeño y optimización de los DMSIs PWM, sino que también aporta a la simulación eficiente de sistemas de gran escala que integran estos dispositivos.

1.1 Objetivos de la Tesis

El objetivo central de esta tesis es contribuir a la simulación eficiente de DMSIs PWM, especialmente en escenarios de largo plazo. En estos casos existen dos situaciones clara-

mente diferenciables que pueden darse dependiendo de la estrategia de modelado utilizada. Por un lado, si se utilizan estrategias de modelado simplificadas, la presencia de perturbaciones en el sistema original puede no manifestarse en los resultados de simulación o generar errores significativos durante los transitorios, ocultando fenómenos característicos como el ripple. El uso de estrategias conmutadas realistas, si bien más precisas, incrementa significativamente la carga computacional cuando se requieren períodos de simulación extensos.

En este trabajo se propone una nueva estrategia de modelado mixto de DMSIs PWM que combina modelos conmutados realistas durante los transitorios y modelos promediados en régimen periódico estable. Este enfoque busca mejorar la eficiencia de las simulaciones de larga duración (Long-Term Simulation) de DMSIs PWM sin pérdida de precisión y detalle en los resultados. Esta alternativa combina las ventajas de ambos enfoques, manteniendo el detalle dinámico donde es relevante.

Otro de los objetivos es analizar el uso de métodos numéricos alternativos para la simulación de modelos conmutados de DMSIs PWM. Dado que el uso de métodos clásicos de integración numérica basados en la discretización temporal resulta muy costoso para simular modelos conmutados de este tipo, en esta tesis se presenta el estudio de la simulación de DMSIs PWM utilizando métodos de integración basados en la cuantificación de las variables de estado (QSS).

1.2 Organización de la Tesis

La tesis se estructura en siete capítulos que guían progresivamente al lector desde los fundamentos teóricos hasta las contribuciones originales. El Capítulo 1 introduce el trabajo, estableciendo los objetivos, motivación y alcance de la investigación, junto con una revisión crítica del estado del arte en modelado y simulación de convertidores conmutados CC–CA PWM.

El Capítulo 2 desarrolla los fundamentos conceptuales de los convertidores conmutados PWM. El capítulo inicia con una introducción y una revisión de los antecedentes conceptuales necesarios para comprender la tesis (Sección 2.1), se describe la arquitectura básica de estos convertidores (2.2) y se profundiza en los principios de la Modulación por Ancho de Pulso (PWM, 2.3). El capítulo avanza con una breve descripción del funcionamiento de las topologías CC–CC, cubriendo configuraciones Buck, Boost, Buck-Boost y Cúk (2.4), para luego abordar los convertidores CC–CA, presentando específicamente las topologías DMSIs PWM que constituyen el foco principal de esta Tesis (2.5).

El Capítulo 3 presenta los conceptos básicos sobre modelado de convertidores conmutados. Comienza identificando los aspectos específicos en este dominio (3.1), para luego contrastar en detalle las dos estrategias clásicas de modelado: modelos conmutados (3.2.1)

y modelos promediados (3.2.2). La Sección 3.3 presenta los modelos mixtos para convertidores PWM CC–CC.

Los métodos de integración basados en cuantificación de estados (Sistemas de Estado Cuantificado (QSS)) utilizados posteriormente para simular los modelos obtenidos de DMSIs PWM se explican en el Capítulo 4. Partiendo de sus fundamentos teóricos (4.1), se describen las variantes de primer, segundo y tercer orden (4.2), así como también las versiones implícitas lineales *Linearly Implicit Quantized State System* (LIQSS) (4.2.3) de los mismos. Finalmente, en la Sección 4.3 se describe el entorno de simulación en el cual están implementados los métodos utilizados en esta tesis.

Los Capítulos 5 y 6 presentan las principales contribuciones de esta tesis.

El Capítulo 5 presenta la aplicación concreta de los métodos QSS para mejorar la eficiencia en la simulación de modelos conmutados de DMSIs PWM. Se presentan los problemas que tienen los métodos clásicos de integración para simular modelos conmutados de este tipo de convertidores (5.1). Con este objetivo, se construyen modelos realistas para las topologías Buck, Buck-Boost y Cúk y se incluye un riguroso análisis de rigidez (5.2.4). Luego de presentar las dificultades de los métodos clásicos, se propone utilizar los métodos QSS enfatizando el motivo por el cual estos métodos pueden sortear muchas de las dificultades de los otros métodos Sección 5.3. En esta sección se presenta también resultados comparativos que ponen en evidencia las ventajas del enfoque propuesto.

El Capítulo 6 presenta una nueva estrategia de modelado mixto para la simulación eficiente de DMSIs PWM. Tras un breve introducción(6.1), se presenta la propuesta innovadora (6.2). Esta nueva estrategia se implementa en lenguaje Modélica (6.3). Finalmente, en la Sección 6.4 se muestran los resultados de simulación obtenidos al utilizar esta nueva estrategia para obtener modelos de DMSIs PWM y se muestran sus ventajas frente a otras estrategias de modelado a partir del análisis de los resultados.

Finalmente, el Capítulo 7 sintetiza las conclusiones más relevantes (7.1) y traza líneas de trabajo futuro (7.2) que podrían extender esta investigación, cerrando así el ciclo que va desde los fundamentos teóricos hasta las aplicaciones prácticas y perspectivas de desarrollo.

1.3 Publicaciones de Apoyo y Contribuciones Originales

En esta sección se presenta una revisión de las publicaciones que han respaldado el desarrollo de esta tesis, destacando sus principales contribuciones y su influencia en el enfoque adoptado.

Se distinguen, por un lado, trabajos previos que aportaron fundamentos conceptuales, metodológicos y tecnológicos para la investigación desarrollada, y por otro, las contribuciones originales realizadas en el marco de esta tesis, muchas de las cuales han sido validadas

a través de publicaciones en congresos y revistas científicas especializadas.

1.3.1 Antecedentes relevantes

Como primer antecedente de este trabajo, se destaca el desarrollo de una librería de convertidores conmutadas corriente continua a corriente continua (CC–CC) utilizando QSS cuyos resultados fueron publicados en las actas de la XII Reunión de Trabajo en Procesamiento de la Información y Control [8]. Esta implementación sentó un precedente clave para la presente tesis, al introducir la utilización de Métodos QSS como una estrategia orientada a la simulación eficiente de convertidores de electrónica de potencia.

Los principales métodos QSS fueron desarrollados en los siguientes trabajos: Kofman, E., junto a Lee, J.S. y Zeigler, B., en *DEVS Representation of Differential Equation Systems* [49]; Kofman, E. y Junco, S., en *Quantized State Systems. A DEVS Approach for Continuous System Simulation* [48]; Kofman, E., en *A Second Order Approximation for DEVS Simulation of Continuous Systems* [45]; y Kofman, E., en *A Third Order Discrete Event Simulation Method for Continuous System Simulation* [47].

Dado que los modelos de convertidores conmutados suelen se rígidos, los *Linearly Implicit Quantized State System* permiten la simulación eficiente de convertidores conmutados. En este sentido, una de las contribuciones previas relevantes para esta tesis fue la colaboración con Migoni, G. y otros autores en el desarrollo de la nueva familia de métodos basados en la cuantificación de estados para la integración de Ecuaciones Diferenciales Ordinarias rígidas denominados *Linearly Implicit Quantized State System*. Los resultados de este trabajo fueron publicados en la revista *Simulation Modelling Practice and Theory* [57].

Los desarrollos relacionados con los métodos QSS también se encuentran recopilados en el libro de Cellier y Kofman [15].

Estos métodos, junto con los principales basados en discretización temporal, están implementados en el software de simulación Simulador Autónomo de QSS (*Stand-Alone QSS Solver*) [28]. Esta herramienta ha demostrado grandes ventajas en eficiencia respecto a otras herramientas comerciales de simulación.

A partir del desarrollo de estos métodos y mediante la herramienta de simulación mencionada, se publicó el trabajo *Quantization-Based Simulation of Switched Mode Power Supplies* [56]. Esta publicación constituye uno de los primeros antecedentes en los que se evidencian las ventajas del uso de los métodos LIQSS, en comparación con los métodos clásicos de integración, para la simulación de modelos de convertidores conmutados CC–CC. Por lo tanto, representa un antecedente relevante para los desarrollos presentados en esta tesis.

Por otro lado, debido a que la simulación de modelos de convertidores conmutados —incluidos los inversores PWM— presentan muchas dificultades para los métodos de in-

tegración clásicos, en la literatura especializada puede encontrarse muchas estrategias de modelado de estos convertidores que buscan sortear las dificultades encontradas en la instancia de simulación. Dentro de esta literatura se pueden destacar libros clásicos tales como[53, 72, 62, 2] y publicaciones más recientes tales como [61, 3, 82, 31, 25, 87, 84]. La mayoría de esta bibliografía se basa en el uso de la estrategias de modelado promediado que utilizan distintas técnicas de aproximación [83, 80, 11, 20, 44, 71, 55, 90].

Todas estas estrategias tienen como característica común la omisión del contenido de alta frecuencia. Sin embargo, diversos trabajos han planteado la necesidad de avanzar en el uso de modelos conmutados integrados en sistemas híbridos de energía renovable de gran escala [59, 13, 69]. Estos estudios abordan las dificultades inherentes a compatibilizar precisión, costo computacional y velocidad de simulación en el ámbito de la electrónica de potencia en sistemas complejos.

Dado que una de las principales dificultades en la simulación de convertidores conmutados radica en las conmutaciones que introducen componentes de alta frecuencia en las señales, un antecedente particularmente inspirador para la técnica de modelado propuesta en esta tesis fue el artículo *Mixed Phasor and Time Domain Modelling of AC Networks with Changeover Management* [68]. Si bien este artículo no trata de modelado de convertidores conmutados, en el mismo se propone un enfoque mixto (dominio temporal-fasorial) para el modelado de redes eléctricas con gestión de cambios entre dominios. Este trabajo sembró la idea de generar modelos que combinen distintas estrategias de modelado.

En tal sentido, la idea principal en la que se inspira parte de los aportes de esta tesis es el desarrollo de modelos mixtos específicos para convertidores PWM de CC–CC, presentado en la publicación *A Mixed Modeling Approach for Efficient Simulation of PWM Switching Mode Power Supplies* [58]. Este enfoque de modelado busca mejorar la eficiencia en la simulación de convertidores conmutados CC–CC, integrando la necesidad de realizar simulaciones rápidas y detalladas que consideren tanto el régimen transitorio como el régimen permanente. Esta propuesta se inscribe en el campo de investigación que se orienta al desarrollo de nuevas estrategias de modelado para sistemas con estructura dinámicamente cambiante [94].

1.3.2 Contribuciones originales de esta tesis

Esta tesis aborda el problema de la simulación eficiente de modelos de DMSIs PWM desde dos enfoques complementarios, sin pérdida de información en los resultados de simulación. Por un lado, busca abordar el problema desde la perspectiva de los métodos numéricos empleados; por otro, desarrollar estrategias de modelado que no presenten tantas dificultades para los métodos de integración, pero que mantengan el nivel de detalle requerido.

Desde el punto de vista de los métodos numéricos, uno de los aportes de esta tesis

consiste en un estudio detallado sobre la simulación eficiente de DMSIs PWM utilizando métodos numéricos QSS[57]. En particular, este estudio se centra en el análisis del costo computacional y los tiempos de procesamiento de CPU requeridos para la simulación de los modelos conmutados de DMSIs PWM. Además, se realiza un análisis detallado de las estructuras de estos dispositivos, con el fin de anticipar las ventajas y desventajas de cada método de integración, y así poder explotar al máximo las características de los métodos numéricos utilizados.

En consecuencia, se estudiaron las distintas estructuras y características dinámicas de los modelos conmutados realistas y se justificó la conveniencia de utilizar Métodos de Integración Cuantificada (MIC). Luego, se desarrollaron modelos de simulación utilizando la herramienta Simulador Autónomo de QSS (*Stand-Alone QSS Solver*) con el objetivo de verificar el desempeño y las ventajas de los MIC en la simulación eficiente de DMSIs PWM, en comparación con los métodos clásicos de integración basados en la discretización temporal. Este trabajo fue recientemente aceptado para publicación con el título *Simulación de Inversores PWM Monofásicos en Modo Diferencial con métodos basados en cuantificación*, en el simposio de modelado y simulación de las 54as Jornadas Argentinas de Informática e Investigación Operativa (JAIIO). En prensa.

A pesar de las ventajas evidenciadas mediante el uso de MIC, también se identificaron limitaciones en simulaciones de larga duración, debido a que el costo computacional continúa siendo elevado en este tipo de escenarios.

Ante esta situación, en esta tesis se propone una nueva estrategia de modelado de DMSIs PWM, orientada a reducir aún más la carga computacional sin comprometer el nivel de detalle en los resultados de simulación. Esta estrategia, denominada "*Estrategia de modelado mixto para DMSIs PWM*", constituye uno de los principales aportes del trabajo, junto con la implementación de modelos y ejemplos utilizando el lenguaje *Modélica* [29]. De este modo, se presenta una alternativa de modelado capaz de abordar simulaciones de larga duración en estudios complejos, manteniendo tanto la eficiencia como el detalle del resultado simulado.

Los resultados de este trabajo fueron publicados recientemente en la revista *SIMULATION*, bajo el título *Mixed Modeling Approach for Efficient Simulation of Single-Phase PWM Inverters* [10].

2. CONVERTIDORES CONMUTADOS PWM

2.1 Introducción

La electrónica de potencia es un campo fundamental de la ingeniería eléctrica, dedicado al control y a la conversión eficiente de la energía mediante el uso de dispositivos electrónicos. Los convertidores conmutados de electrónica de potencia transforman la energía, adaptando formas de onda, tensión y corriente para diversas aplicaciones. Por ejemplo, se utilizan para transformar una tensión de entrada de corriente continua (V_{in}) en una tensión de salida de corriente continua ($v_{\text{out}} = V_{\text{out}}$) (Convertidores CC-CC) o de corriente alterna ($v_{\text{out}}(t)$) (Convertidores CC-CA o inversores). Además, permiten regular dinámicamente la tensión de salida mediante técnicas de control por modulación de ancho de pulso (PWM), garantizando estabilidad frente a variaciones de carga o de la fuente de alimentación [85]. Por estos motivos, estos convertidores son esenciales en los sistemas modernos debido a su capacidad para manejar grandes cantidades de energía con alta eficiencia y fiabilidad [61].

Los convertidores conmutados son un subsistema de los sistemas de electrónica de potencia, como por ejemplo el mostrado en la Figura 2.1. En la misma, pueden observarse tanto un convertidor CC–CC como uno de corriente continua a corriente alterna (CC–CA), también conocido como *inversor*, dispositivo que reviste especial interés en esta tesis.

Los convertidores emplean dispositivos semiconductores de potencia controlados mediante señales (generalmente a través de circuitos integrados) e incluyen elementos de almacenamiento de energía, como inductores y capacitores [63].

Este capítulo revisa los conceptos fundamentales de estos dispositivos, enfocándose en sus topologías y funciones, para luego abordar, en los capítulos siguientes, su modelado y simulación.

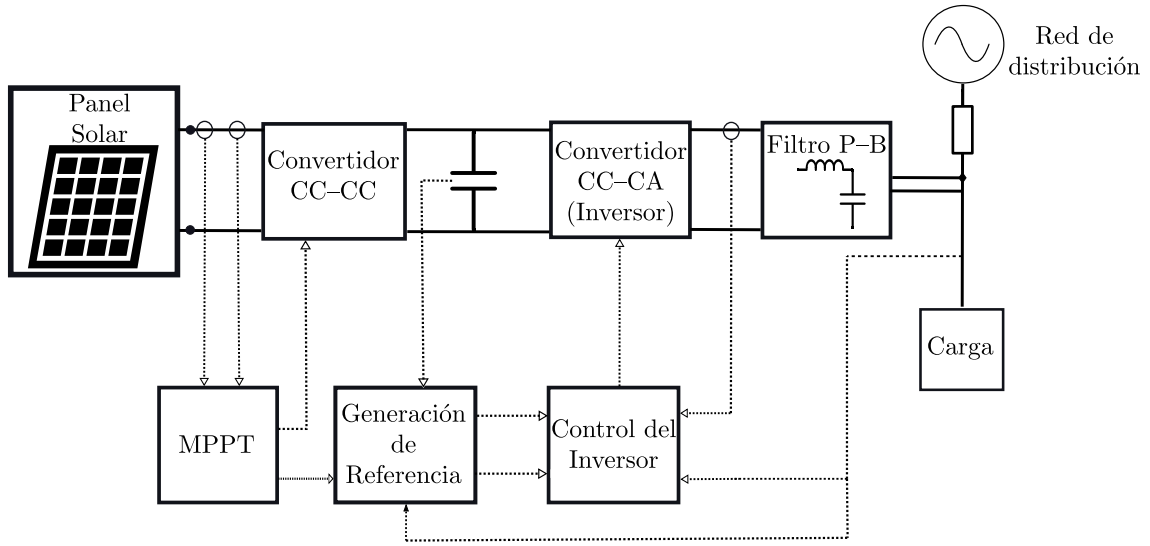


Figura 2.1: Esquema de un sistema fotovoltaico con convertidores CC–CC (MPPT) y CC–CA, filtrado y conexión a red.

2.2 Arquitectura básica de un convertidor conmutado

Un convertidor conmutado transforma la energía eléctrica de entrada en una forma diferente a la de su salida, modificando parámetros como el nivel de tensión, la corriente o la forma de onda. Para lograrlo, emplea componentes electrónicos de conmutación —como transistores, diodos o tiristores— que operan a alta frecuencia, así como elementos de almacenamiento de energía como inductores y capacitores. El proceso de conversión es gestionado por un controlador electrónico que regula la señal de conmutación, asegurando que la salida cumpla con los requisitos del sistema.

Una característica distintiva de los convertidores conmutados es su alta eficiencia, ya que los dispositivos de conmutación idealmente no disipan una cantidad significativa de potencia: operan alternando entre los estados de corte y saturación, donde la disipación es mínima. No obstante, la conmutación rápida introduce armónicos de alta frecuencia, conocidos como ripple, y puede generar interferencia electromagnética (EMI), así como estrés eléctrico en los componentes [43, 25, 75].

Para ilustrar el funcionamiento de los convertidores conmutados, se puede considerar una arquitectura básica de conversión de CC–CC como el de la Figura 2.2 que utiliza un interruptor electrónico conmutado que modifica periódicamente la tensión de salida. Este circuito utiliza un Interruptor unipolar de dos posiciones (*Single Pole Double Throw*) (SPDT) de manera que la tensión de salida del interruptor, $v_s(t)$, coincida con la tensión de entrada del convertidor, V_{in}, cuando el interruptor se encuentre en la posición 1, y sea cero cuando esté en la posición 2.

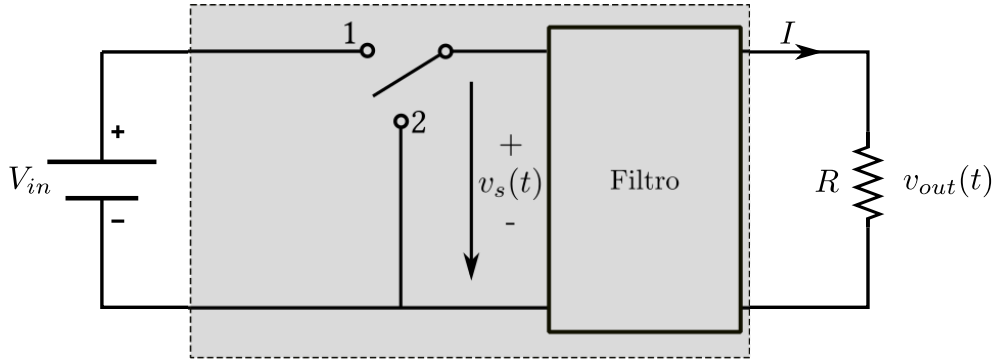


Figura 2.2: Convertidor conmutado con interruptor SPDT que modifica el componente de CC de la tensión.

La posición del interruptor cambia de forma periódica, como se ilustra en la Figura 2.3, generando una onda rectangular de período T_{sw}. El ciclo de trabajo $d_c(t)$ (Duty Cycle, en inglés) de esta onda cuadrada se define como la fracción del período durante la cual el interruptor permanece en la posición 1. Su valor puede oscilar entre 0 y 1.

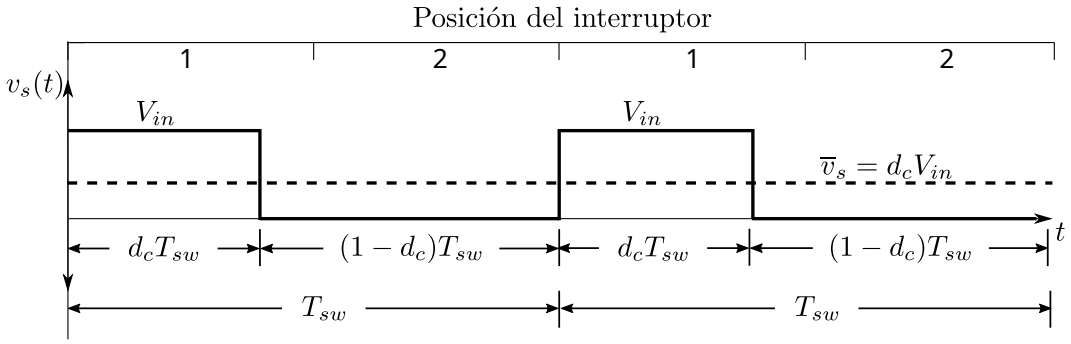


Figura 2.3: Forma de onda de la tensión de salida del interruptor $v_s(t)$.

Al controlar el ciclo de trabajo de la señal de conmutación, es posible ajustar el valor medio $\overline{v}_s$ de la tensión de salida del interruptor.

$$\overline{v}_s = \frac{1}{T_{sw}} \int_0^{T_{sw}} v_s(t)dt = \frac{1}{T_{sw}} \int_0^{d_c T_{sw}} V_{\text{in}} dt = d_c V_{\text{in}} \tag{2.1}$$

Finalmente, para obtener una señal continua a partir señal cuadrada $v_s(t)$, se incorpora un filtro pasabajos. Una de las topologías mas simples implementan un filtro pasabajos L–C (inductor–capacitor), como se muestra en la Figura 2.4. Este filtro atenúa los armónicos de alta frecuencia y permite que la tensión de salida se aproxime al componente continuo deseado.

El interruptor SPDT ideal de la Figura 2.4 se puede implementar mediante dispositivos semiconductores que operan en modo conmutado como se muestra en la Figura 2.5. En es-

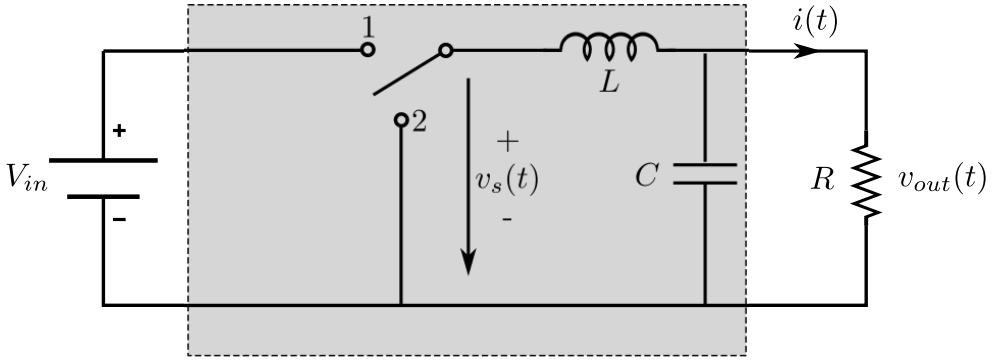


Figura 2.4: Incorporación de un filtro pasabajos L–C para la atenuación de los armónicos generados por la conmutación.

te esquema, el transistor opera en corte y saturación. Su estado de conducción se controla con una señal cuadrada $\delta(t)$ del mismo período T_{sw} y d_c que $v_s(t)$. El convertidor desarrollado hasta este punto se conoce como convertidor *buck*, ya que permite reducir el nivel de tensión continua (corriente continua (CC)) . La arquitectura completa se complementa con un lazo de control que ajusta dinámicamente el ciclo de trabajo del interruptor en base a una referencia externa. Este control se implementa comúnmente mediante PWM, lo que permite regular la salida en presencia de perturbaciones o variaciones de carga.

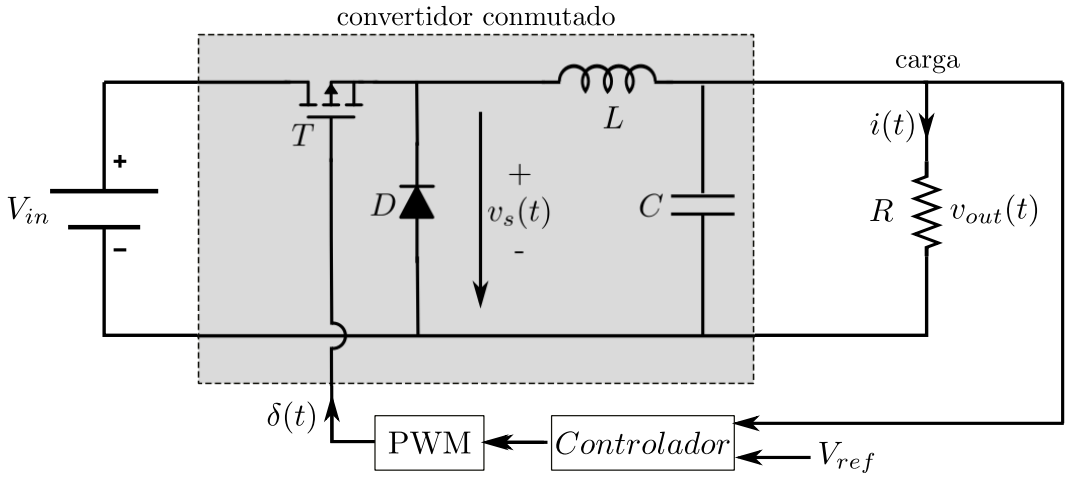


Figura 2.5: Incorporación de un sistema de control para regular el tensión de salida.

Esta misma lógica de funcionamiento se puede extender para formar otras topologías de convertidores CC–CC y como se verá más adelante para las topologías de los convertidores CC–CA.

2.3 Modulación por Ancho de Pulso (PWM)

La modulación por ancho de pulso (PWM) es una técnica ampliamente utilizada para generar señales de control destinadas a la activación y desactivación de los dispositivos de conmutación, regulando así el flujo de potencia en los convertidores.

La señal PWM genera pulsos que alternan entre *encendido* (on) y *apagado* (off), con un ancho ajustado según la diferencia entre la señal portadora y la referencia, lo que determina el ciclo de trabajo $d_c(t)$ que indica la fracción del período de conmutación durante la cual el interruptor permanece encendido.

Si bien existen varias formas de generar señales PWM uno de los métodos más comunes en el ámbito de las convertidores conmutados es el uso de una señal portadora de alta frecuencia que se compara con una señal de referencia, onda en diente de sierra. (Figura 2.6) . La onda diente de sierra se utiliza como señal de referencia debido a su capacidad para proporcionar una variación lineal en el tiempo, lo que permite una modulación precisa. El resultado de esta comparación es una señal cuadrada de frecuencia constante e igual a la frecuencia f_{sw} de la diente de sierra y cuyo ancho de pulso varía según la diferencia entre la señal portadora y la diente de sierra.

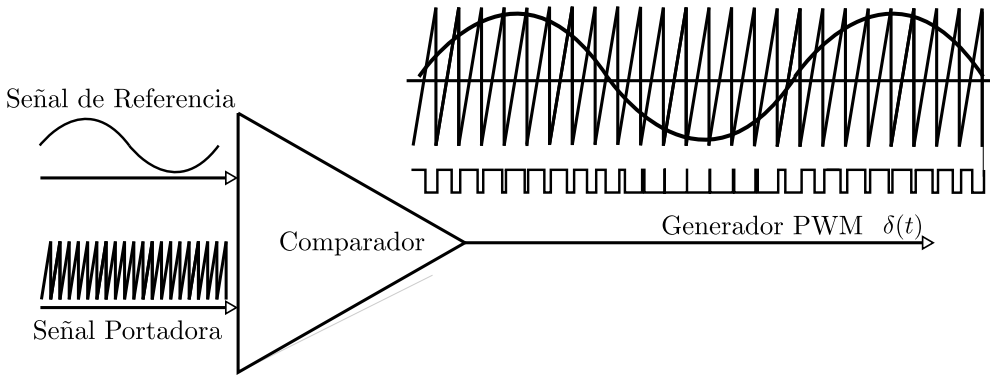


Figura 2.6: Esquema de Modulación por Ancho de Pulso (PWM) senoidal.

Matemáticamente, el ciclo de trabajo se puede expresar como:

$$d_c(t) = \frac{T_{\text{on}}}{T_{\text{sw}}} \tag{2.2}$$

donde $T_{\text{sw}} = T_{\text{on}} + T_{\text{off}}$ es el período de la señal portadora.

2.4 Convertidores conmutados CC–CC

A continuación, se presenta una breve descripción de las topologías más comunes de convertidores conmutados CC–CC, junto con los fundamentos de funcionamiento de cada

una de ellas.

2.4.1 Convertidor conmutado CC–CC tipo buck

El convertidor conmutado CC–CC tipo *buck* es un reductor de tensión. En términos generales, realiza la conversión de una tensión CC de alto valor a una de menor magnitud, manteniendo la misma polaridad.

La Figura 2.7 a) muestra un esquema circuital de este convertidor. El interruptor de conmutación se controla mediante una señal PWM. Esta señal permite regular el flujo de potencia y alcanzar el nivel de tensión deseado en la salida $v_{\text{out}}(t)$. El diodo (*D*) permite la circulación continua de corriente en el inductor durante el estado apagado del interruptor *Sw*.

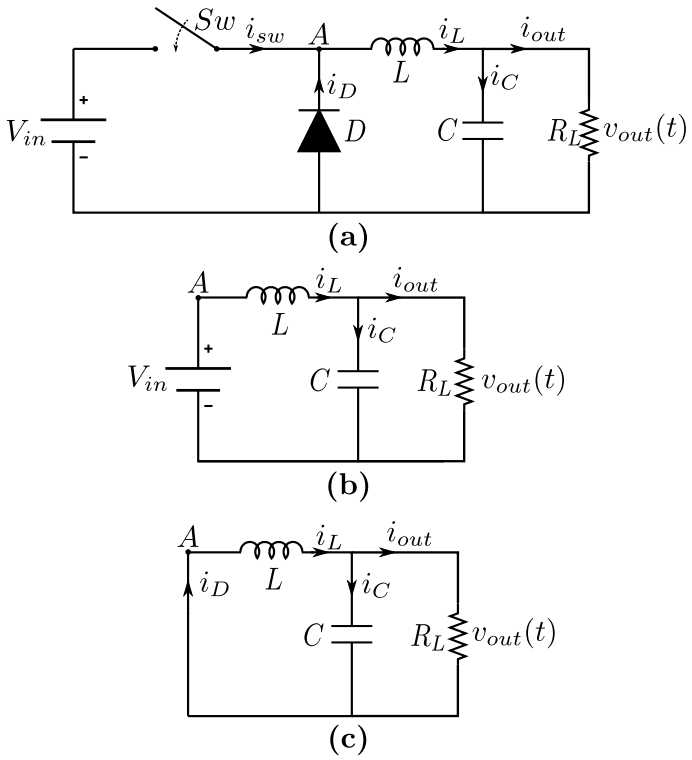


Figura 2.7: Esquema circuital de un convertidor CC–CC buck.

Para comprender el funcionamiento de este circuito y se analiza el estado estacionario del convertidor *buck* durante cada uno de los estados del interruptor de conmutación (*Sw*). Con el objetivo de simplificar la explicación se denominará T_{on} al intervalo de tiempo en el cual el interruptor se encuentra conduciendo y T_{off} al que no conduce. De manera equivalente, se denominará T_{up} al intervalo de tiempo durante el cual la corriente en la inductancia aumenta, y T_{down} al intervalo en que dicha corriente disminuye.

De este modo, durante T_{on}, cuando el interruptor Sw conduce, el nodo de conmutación

'A' se conecta directamente a la fuente de tensión de entrada, provocando la polarización inversa del diodo, dado que $(v_A = V_{\text{in}})$. Esta situación se representa mediante el circuito equivalente mostrado en la Figura 2.8 (a). En esta condición, la corriente del inductor i_L aumenta, ya que $v_L = V_{\text{in}} - v_{\text{out}} > 0$, y por tanto $\frac{di_L}{dt} = \frac{v_L}{L} = \frac{V_{\text{in}} - v_{\text{out}}}{L} > 0$.

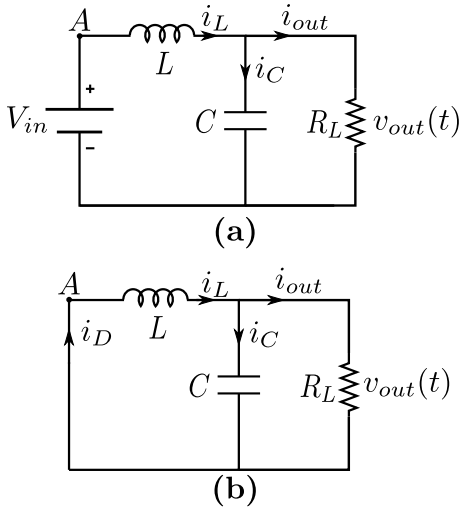


Figura 2.8: Esquema circuital de un convertidor CC–CC buck. (a) Circuito equivalente Sw $= On$ y $D = Off$. (b) Circuito equivalente con Sw $= Off$ y $D = On$.

Luego, en el intervalo de tiempo T_{off}, cuando el interruptor Sw esta abierto, la corriente del inductor i_L provoca la conducción del diodo, que se polariza en directo debido al efecto de la auto-inducción $v_L = L\frac{di_L}{dt}$. El circuito equivalente para este estado se muestra en la Figura 2.8 b). En esta condición, la corriente en el inductor disminuye, dado que $v_L = -v_{\text{out}} < 0$, y por lo tanto $\frac{di_L}{dt} = \frac{v_L}{L} = \frac{-v_{\text{out}}}{L} < 0$. La energía magnética previamente almacenada en el inductor se transfiere hacia la carga R. De esta manera contribuye a mantener la tensión en el capacitor y por ende la v_{out}. Para este primer análisis vamos a considerar que la corriente en la bobina nunca llega a ser nula, lo que se denominara Modo de Conducción Continua (MCC) de operación del convertidor.

La evolución descripta de las señales v_A, v_L e i_L durante los dos estados de conducción del interruptor puede verse en Figura 2.9.

Considerando que en régimen estacionario la variación ΔI_L (ripple de corriente) durante T_{on} es la misma que durante T_{off}, si denominamos $\overline{v}_{\text{out}}$ al valor medio en un período de conmutación T_{sw} de la salida v_{out} entonces según lo explicado anteriormente resulta:

$$L\frac{\Delta I_L}{T_{\text{on}}} = V_{\text{in}} - \overline{v}_{\text{out}} \quad \Rightarrow \quad |\Delta I_L| = \frac{V_{\text{in}} - \overline{v}_{\text{out}}}{L} T_{\text{on}} \tag{2.3}$$

$$L\frac{\Delta I_L}{T_{\text{off}}} = -\overline{v}_{\text{out}} \quad \Rightarrow \quad |\Delta I_L| = \frac{\overline{v}_{\text{out}}}{L} T_{\text{off}} \tag{2.4}$$

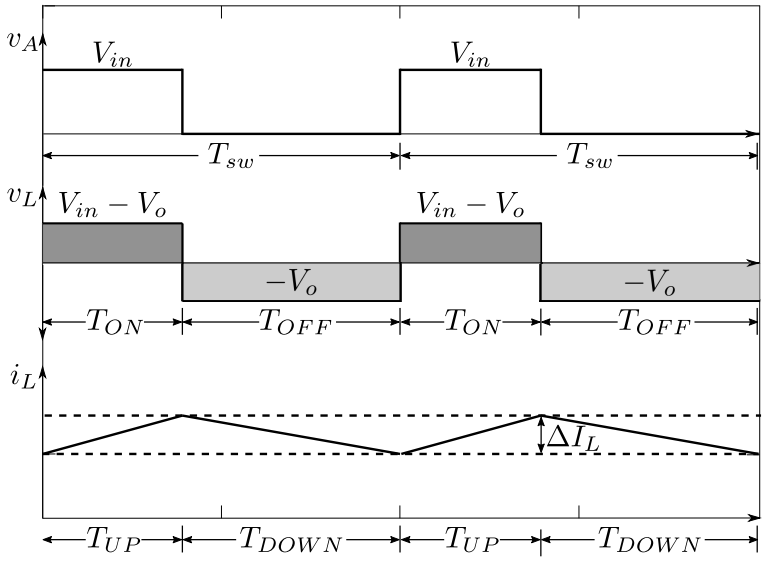


Figura 2.9: Formas de onda en estado estacionario en modo de conducción continua: tensión v_A en el nodo del interruptor, tensión v_L en el inductor, y corriente i_L a través del inductor.

Combinando las Ecuaciones (2.3) y (2.4) se obtiene la relación de conversión de tensión para el convertidor *buck* en régimen estacionario:

$$\frac{\overline{v}_{\text{out}}}{V_{\text{in}}} = \frac{T_{\text{on}}}{T_{\text{on}} + T_{\text{off}}} \quad \text{o} \quad \overline{v}_{\text{out}} = V_o = V_{\text{in}} \cdot \frac{T_{\text{on}}}{T_{\text{sw}}} \tag{2.5}$$

En la Ecuación (2.5), vemos que el valor de continua V_o a la salida del convertidor buck es igual al valor de la tensión de entrada V_{in} multiplicado por la relación entre el tiempo T_{on} y el período total de conmutación ($T_{\text{sw}} = T_{\text{on}} + T_{\text{off}}$).

Definiendo el ciclo de trabajo como:

$$d_c = \frac{T_{\text{on}}}{T_{\text{on}} + T_{\text{off}}} \tag{2.6}$$

se puede reescribir la relación de conversión de tensión del convertidor *buck* en MCC en función del ciclo de trabajo según:

$$V_o = V_{\text{in}} \cdot d_c \tag{2.7}$$

En el caso de que el convertidor no estar operando en MCC, es decir si la corriente i_L alcanza el valor cero durante el intervalo de apagado T_{off}, se dice que el convertidor opera en Modo de Conducción Discontinua (MCD). En este caso, el diodo deja de conducir, y el inductor queda desacoplado del resto del circuito. La Figura 2.10 muestra las formas de onda en estado estacionario de v_A, v_L, i_L y la corriente de entrada i_s, correspondientes a

la operación del convertidor *buck* en MCD.

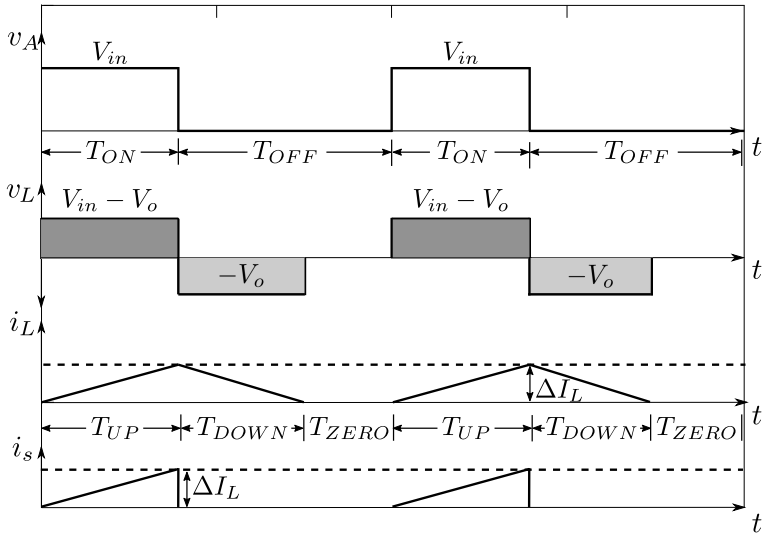


Figura 2.10: Formas de onda en estado estacionario para el convertidor buck operando en MCD.

Como puede verse en la Figura 2.8 b), cuando la corriente está bajando de valor durante T_{off} no puede nunca hacerse negativa por el diodo D. En el caso de MCD, la corriente i_L alcanza el cero antes de que vuelva a conducir el interruptor (nuevo T_{on}) tal como se muestra en la Figura 2.10. De este modo, T_{off} no coincidirá con T_{down}. En este caso se define un nuevo intervalo de tiempo denominado T_{zero} que representa el tiempo en el cual la corriente en el inductor es nula. Notar que $T_{\text{off}} = T_{\text{zero}} + T_{\text{down}}$

De forma similar a lo que se hizo en MCC, la relación entre la tensión de entrada y de salida en el convertidor buck al estar en MCD es:

$$\frac{\overline{v}_{\text{out}}}{V_{\text{in}}} = \frac{T_{\text{on}}}{T_{\text{on}} + T_{\text{down}}} > d_c \tag{2.8}$$

Por lo tanto, en el modo de conducción discontinua (MCD), la tensión de salida V_o no se puede calcular exclusivamente a partir del d_c. El tiempo T_{down} y valor medio de i_L dependen del valor de carga R_L, por lo que en MCD el nivel de tensión de salida V_o dependerá del d_c y de la carga. Esto contrasta con lo que ocurre en MCC donde V_o solo depende del d_c.

Se puede ver además a partir de las expresiones de V_o obtenidas tanto para MCC como para MCD que como era de esperar V_o resulta siempre menor que V_{in}.

2.4.2 Convertidor conmutado CC–CC tipo boost

El convertidor *boost* es un regulador que incrementa la tensión de una señal de entrada de corriente continua (CC), elevándola a un nivel superior sin invertir su polaridad.

La topología básica del convertidor *boost* emplea los mismos componentes que la del convertidor *buck*, pero organizados en una configuración distinta, como se muestra en la Figura 2.11.

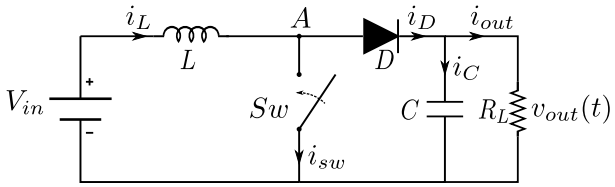


Figura 2.11: Circuito del convertidor *boost*: (a) topología general; (b) estado de conducción del interruptor S_w; (c) estado de apagado del interruptor S_w.

En este circuito la tensión de salida v_{out} tendería a igualarse con V_{in} en ausencia de conmutación, debido a la conexión directa a través del diodo. Sin embargo, el objetivo del convertidor *boost* es incrementar v_{out} por encima de V_{in}. Para ello, se introduce una conmutación controlada mediante el interruptor S_w, que interrumpe temporalmente el flujo directo de corriente, permitiendo la acumulación y posterior transferencia de energía a través del inductor de manera eficiente.

Cuando el interruptor S_w está conduciendo, el nodo de conmutación 'A' se conecta a tierra. En esta condición, el diodo D se polariza en inversa debido a que $v_{\text{out}} > v_A$, impidiendo la conexión directa entre el inductor L y el capacitor C. La Figura 2.12 (a) muestra como queda configurado el circuito de la Figura 2.11 en estas condiciones.

Durante el intervalo de tiempo que el interruptor conduce, la tensión en la inductancia resulta $v_L = V_{\text{in}}$ y la corriente a través del inductor i_L aumenta y almacena energía en forma de campo magnético. De esta manera, la variación de la corriente i_L mientras el interruptor conduce (T_{on}) resulta

$$|\Delta i_L| = \frac{1}{L} V_{\text{in}} T_{\text{on}} \tag{2.9}$$

La salida durante este tiempo permanece aislada de la fuente de alimentación, y el capacitor se encarga de alimentar la carga.

La Figura 2.12 (b) muestra como resulta el circuito de la la Figura 2.11 cuando el interruptor S_w deja de conducir. En este circuito, la energía almacenada en el inductor L, junto con la energía suministrada por la fuente V_{in}, se transfiere al capacitor de salida C y a la carga R_L a través del diodo (que queda polarizado en directa) haciendo que la tensión de salida se eleve.

La tensión sobre el inductor en esta condición es $v_L = V_{\text{in}} - v_{\text{out}}$, la cual resulta negativa, ya que en estado estacionario se cumple $v_{\text{out}} > V_{\text{in}}$. Como consecuencia, la corriente i_L disminuye durante este intervalo (T_{off}) y la variación Δi_L resulta:

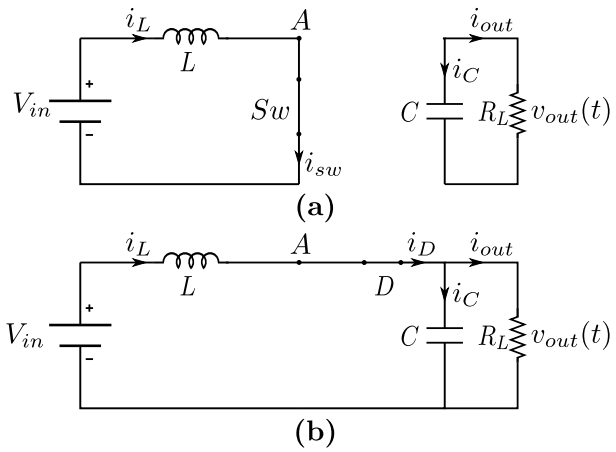


Figura 2.12: Circuito del convertidor *boost*: (a) $S_w = on$ y $D = off$; (b) $S_w = off$ y $D = on$.

$$|\Delta i_L| = -\frac{1}{L}(V_{\text{in}} - \overline{v}_{\text{out}})T_{\text{off}} \tag{2.10}$$

De manera similar a lo realizado con el convertidor tipo *buck*, se tiene a partir de las Ecuaciones 2.9 y 2.10 que en régimen estacionario la relación entre la tensión de entrada y la tensión de salida del convertidor *boost* operando en MCC resulta:

$$\frac{V_o}{V_{\text{in}}} = 1 + \frac{T_{\text{on}}}{T_{\text{off}}} = \frac{1}{1 - d_C} \tag{2.11}$$

Esta expresión muestra que la tensión media de salida V_o puede superar la tensión de entrada V_{in} a medida que el ciclo de trabajo d_c se aproxima a 1.

Asimismo, debe considerarse la existencia de un valor crítico de carga, $R_{\text{crítica}}$, a partir del cual la corriente del inductor L se reduce a cero antes de finalizar el período de conmutación. Cuando $R_L > R_{\text{crítica}}$, el convertidor opera en modo de conducción discontinua (MCD). En estas condiciones, al igual que ocurre con el convertidor *buck*, la relación de conversión de tensiones dada por la ecuación 2.11, no dependerá solamente del d_c sino también de la carga.

2.4.3 Convertidor conmutado CC–CC tipo buck-boost

La topología *buck-boost* integra las funcionalidades de los convertidores *buck* y *boost*, permitiendo reducir o aumentar la tensión de salida respecto de la entrada. Esta configuración es especialmente útil en aplicaciones que requieren una relación de conversión de tensión adaptable utilizando un único convertidor. El esquema circuital se representa en la Figura 2.13.

En esta configuración, la tensión de salida tiene la polaridad invertida respecto de la tensión de entrada, y su magnitud puede ser mayor o menor que V_{in}, dependiendo del ciclo

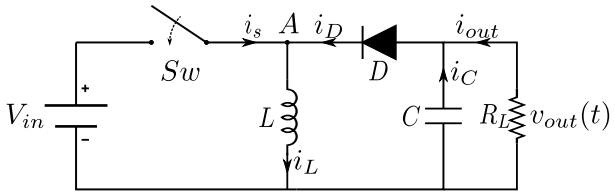


Figura 2.13: Circuito del convertidor CC–CC *buck-boost*.

de trabajo aplicado.

La Figura 2.14 (a) muestra el circuito resultante cuando el interruptor conduce ($S_w = on$). En este caso, el nodo $'A'$ tiene una tensión $v_A = V_{\text{in}}$ y el diodo queda polarizado en inversa. Entonces, durante el intervalo de tiempo (T_{on}), el inductor almacena energía magnética, y la variación Δi_L resulta:

$$|\Delta i_L| = \frac{1}{L} V_{\text{in}} T_{\text{on}} \tag{2.12}$$

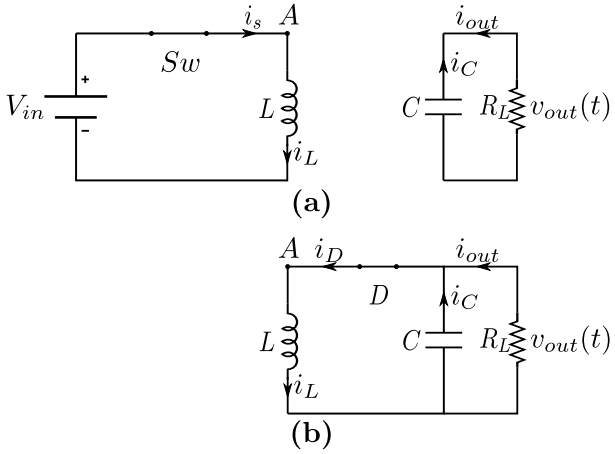


Figura 2.14: Circuito del convertidor CC–CC *buck-boost*: (a) $S_w = on$ y $D = off$; (b) $S_w = off$ y $D = on$.

Cuando el interruptor se abre ($S_w = off$), se tiene el circuito de la Figura 2.14 (b). En este caso, el diodo se polariza en directa y comienza a conducir de modo que la energía previamente almacenada en el inductor L durante T_{on} se transfiere hacia el capacitor de salida y la carga por lo que la polaridad de la tensión en la salida v_{out} es negativa. De este modo, la variación de corriente Δi_L durante el intervalo de tiempo T_{off} el interruptor este abierto ($Sw = off$) resulta:

$$|\Delta i_L| = -\frac{1}{L} \overline{v}_{\text{out}} T_{\text{off}} \tag{2.13}$$

Finalmente, a partir de las Ecuaciones 2.12 y 2.12 la relación en estado estacionario

entre la tensión de entrada y de salida de este convertidor operando en MCC resulta:

$$\frac{V_o}{V_{\text{in}}} = -\frac{T_{\text{on}}}{T_{\text{off}}} = -\frac{d_c}{1-d_c} \tag{2.14}$$

Esta expresión muestra que la magnitud de la tensión de salida puede ser en módulo menor o mayor que la tensión de entrada del convertidor y que la misma tiene polaridad inversa respecto a la entrada.

Tal como ocurre en los otros convertidores que se mostraron, el MCD ocurre cuando la corriente del inductor se anula antes de finalizar el intervalo T_{off}.

2.4.4 Convertidor conmutado CC–CC tipo Ćuk

Al igual que el convertidor *buck-boost*, el convertidor *Ćuk* puede generar una tensión de salida cuya magnitud puede ser mayor o menor que la de entrada, y con polaridad inversa. Sin embargo, presenta ventajas como un menor ripple en la corriente de entrada y salida, y un desacoplamiento más efectivo entre ambos lados del circuito. Para lograr estas características, el convertidor *Ćuk* emplea un mayor número de elementos almacenadores de energía, como se muestra en el esquema circuital de la Figura 2.15.

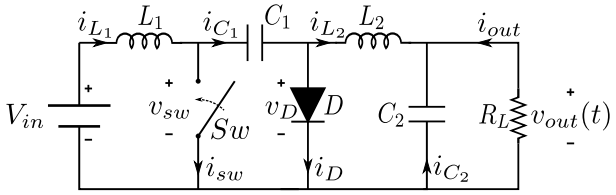


Figura 2.15: Circuito del convertidor CC–CC Ćuk.

Mientras el interruptor está en conducción (S_w = on) durante el intervalo de tiempo T_{on}, el diodo permanece en corte (D = off), y el circuito representado en la Figura 2.15 se comporta como el mostrado en la Figura 2.16(a). En este intervalo, la energía de la fuente se almacena en el inductor L_1. En el lado derecho del circuito, el capacitor C_1 transfiere la energía almacenada hacia el capacitor de salida y la carga. La variación de tensión ΔV_{C_1} durante el intervalo T_{on} resulta:

$$|\Delta V_{C_1}| = -\frac{\bar{i}_{L_2}}{C_1} T_{\text{on}} \tag{2.15}$$

En el intervalo T_{off}, cuando el interruptor está abierto (S_w = off), la corriente continua del inductor obliga al diodo a entrar en conducción (D = on). La energía almacenada en L_1 se transfiere a C_1, lo que provoca un aumento en V_{C_1} y una disminución en i_{L_1} debido a la descarga del inductor. La variación de tensión ΔV_{C_1} durante el intervalo T_{off} resulta:

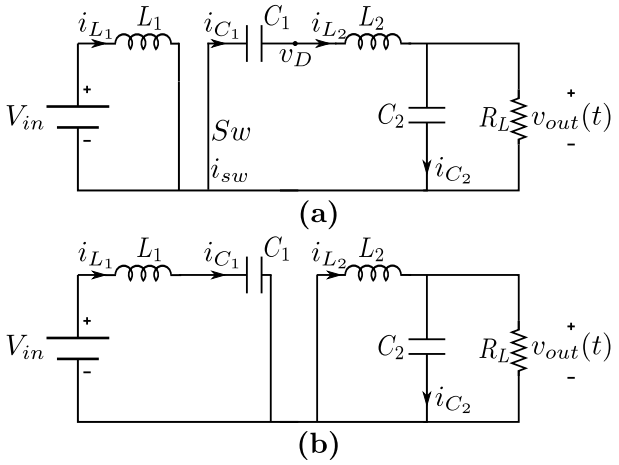


Figura 2.16: Circuito del convertidor Ćuk: (a) $S_w = on$ y $D = off$; (b) $S_w = off$ y $D = on$.

$$|\Delta V_{C_1}| = \frac{\bar{i}_{L_1}}{C_1} T_{\text{off}} \tag{2.16}$$

Combinando las Ecuaciones 2.15 y 2.16 se obtiene el siguiente balance de corriente:

$$-\bar{i}_{L_2} T_{\text{on}} = \bar{i}_{L_1} T_{\text{off}} \tag{2.17}$$

Además, dado que i_{L_1} es la corriente de entrada desde la fuente de energía e i_{L_2} es equivalente a la corriente de salida hacia la carga, el balance de potencia se puede expresar como:

$$V_{\text{in}} \bar{i}_{L_1} = \overline{v}_{\text{out}} \bar{i}_{L_2} \tag{2.18}$$

Finalmente, a partir de las Ecuaciones 2.17 y 2.18 se obtiene la relación de conversión de tensión, en MCC:

$$\frac{V_o}{V_{\text{in}}} = \frac{-T_{\text{on}}}{T_{\text{off}}} = -\frac{d_c}{1 - d_c} \tag{2.19}$$

Comparando la relación de conversión del convertidor buck-boost (Ecuación 2.14) y la del convertidor Cuk. Vemos que el comportamiento del valor medio de sus salidas de tensión son exactamente iguales. Como se menciono la diferencia radica en el ripple superpuesto sobre estos niveles medios.

2.4.5 Convertidores conmutados CC–CC bidireccionales

Todos los convertidores CC–CC presentados anteriormente en esta sección permiten únicamente la transferencia de energía desde la entrada (fuente) hacia la salida (carga).

Sin embargo, en muchas aplicaciones es necesario que este intercambio pueda producirse en ambos sentidos, como en el caso de la recuperación de energía en vehículos con frenado regenerativo, donde se devuelve energía a las baterías. Para estos casos, todos estos convertidores tienen su versión bidireccional. Para obtener las versiones bidireccionales de los convertidores se reemplaza los diodos en las topologías anteriores por interruptores que conducen en el intervalo de tiempo T_{off} (cuanto el transistor original de las topologías deja de conducir).

2.5 Convertidores conmutados CC–CA

Los convertidores conmutados CC–CA, comúnmente conocidos como *inversores*, desempeñan un papel central en los sistemas de electrónica de potencia. Su función principal consiste en convertir una señal de tensión continua en una señal alterna con características controladas de frecuencia, forma de onda y amplitud.

Estos convertidores resultan indispensables en aplicaciones que requieren interconectar fuentes de energía continua (sistemas fotovoltaicos, bancos de baterías, etc) con cargas de corriente alterna (CA) o redes de alterna. La calidad de la señal de salida, así como la eficiencia y robustez del proceso de conversión, dependen en gran medida de la técnica de modulación adoptada y de la topología del inversor utilizada.

A continuación, se describe una familia de inversores ampliamente utilizadas en sistemas monofásicos de baja potencia: los inversores PWM en modo diferencial.

2.5.1 Inversores monofásicos PWM en modo diferencial

El principio de funcionamiento de un inversor monofásico en modo diferencial (Inversor Monofásico en Modo Diferencial (DMSI)) se basa en la interconexión diferencial de dos convertidores conmutados CC–CC bidireccionales como se muestra en la Figura 2.17 [1]. Cada uno de estos convertidores es modulado mediante una señal PWM cuyo ciclo de trabajo $d_c(t)$ se obtiene a partir de una señal de referencia alterna senoidal. Además, la señal de referencia para el $d_c(t)$ en cada convertidor deben estar desfasados $180°$ (entre ambas ramas).

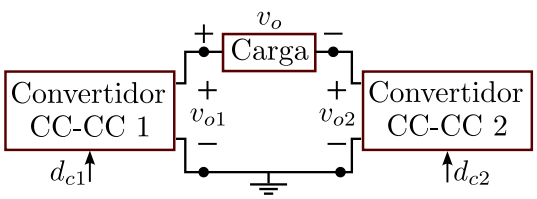


Figura 2.17: Esquema de conexión de la topología DMSI PWM.

Al utilizar una señal sinusoidal como referencia para el ciclo de trabajo

$$d_{c_i}(t) = 0{,}5 + K_i \sin(\omega t + \phi_i) \quad \text{con } K_i \leq 0{,}5, \tag{2.20}$$

se obtiene una tensión de salida $v_{o_i}(t)$ alterna, con una componente continua superpuesta:

$$v_{o_i}(t) \simeq A \sin(\omega t + \phi_i) + V_{\text{offset}} = d_{c_i}(t) \cdot V_{\text{in}}. \tag{2.21}$$

Si se conectan dos de estos convertidores conmutados como en el esquema de la Figura 2.17, y se aplican ciclos de trabajo desfasados 180 grados, entonces ambas tensiones de salida $v_{o1}(t)$ y $v_{o2}(t)$ presentan un desfase de $\phi_1 = 0$ y $\phi_2 = \pi$, respectivamente, como se ilustra en la Figura 2.18 (a).

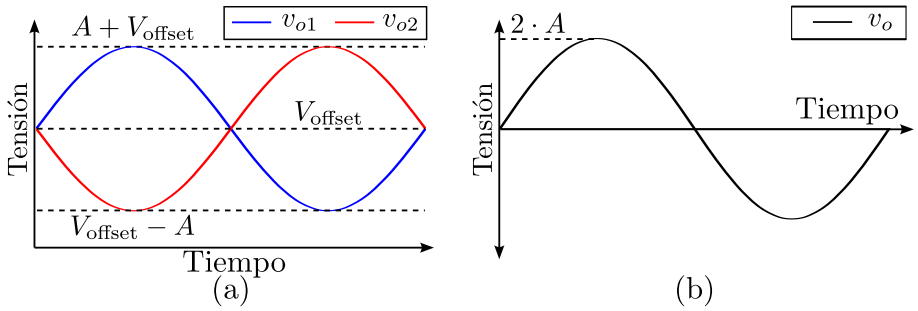


Figura 2.18: DMSI PWM: (a) tensiones en convertidores 1 y 2; (b) tensión de carga.

Como se observa en la Figura 2.18 (b), con esta topología se obtiene una tensión de salida CA sin componente de CC dado que :

$$v_{\text{out}}(t) = v_{\text{out}}(t) = v_{o1}(t) - v_{o2}(t) \simeq 2A \sin(\omega t) \tag{2.22}$$

Si bien el análisis realizado permite comprender de forma cualitativa el funcionamiento de los DMSIs PWM, para conocer con precisión el comportamiento de sus variables y diseñar estrategias de control adecuadas, es necesario recurrir a modelos matemáticos.

2.5.2 Topologías básicas de DMSIs PWM

A partir del análisis desarrollado en la Sección 2.4 sobre las distintas topologías clásicas de convertidores PWM CC–CC, es posible adaptar dichos esquemas para implementar DMSIs controlados mediante PWM.

En estos convertidores, los diodos y transistores unidireccionales se reemplazan por transistores con diodos en antiparalelo (par transistor diodo), capaces de conducir corriente en ambos sentidos. A diferencia de los casos previamente explicados, las señales de referencia utilizadas para generar el ciclo de trabajo no son constantes, sino funciones senoidales, lo cual permite sintetizar una tensión alterna modulada.

En función de las topologías básicas definidas para los convertidores CC–CC, a continuación se presentan las configuraciones equivalentes de inversores en modo diferencial.

DMSIs PWM buck

La Figura 2.19 muestra el esquema circuital básico de un inversor DMSI tipo *buck*, compuesto por dos convertidores conmutados buck. Puede observarse la conexión en modo diferencial de la carga (*Load*), ubicada a la salida de los filtros *LC* de cada rama.

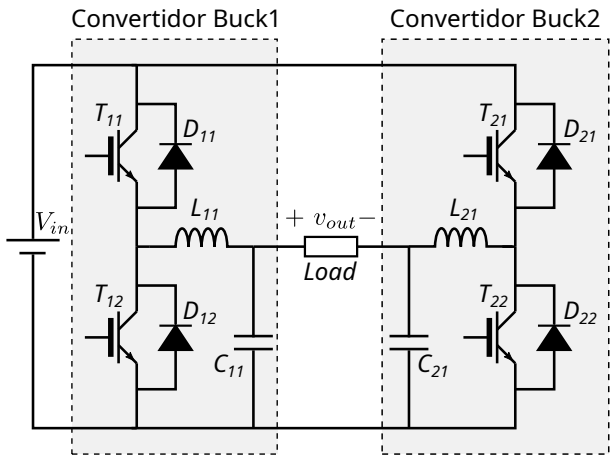


Figura 2.19: Esquema circuital del DMSI PWM buck.

Al igual que en los convertidores CC–CC, esta configuración opera como un reductor de tensión de manera que la tensión de salida es una señal de alterna $v_{\text{out}}(t)$ cuya amplitud está acotada según $-V_{\text{in}} \leq v_{\text{out}}(t) \leq V_{\text{in}}$.

El control de los transistores debe garantizar que al activarse un transistor en una de las ramas el correspondiente transistor de la misma rama no esté conduciendo. Esta lógica de funcionamiento es permite evitar cortocircuitos en la etapa de conmutación. Para garantizar que esta condición se cumpla, se agregan siempre tiempos muertos T_{delay} entre el apagado y encendido de los transistores de cada uno de los siguientes pares de transistores $(T_{11} - T_{12})$ y $(T_{21} - T_{22})$.

Además, para obtener una señal de salida con bajo contenido armónico, se utiliza en los PWM una señal portadora de alta frecuencia. Esta estrategia permite realizar la conversión de CC a CA, generando una forma de onda que se aproxima a una señal sinusoidal.

La operación de modulación por ancho de pulso (PWM) fue introducida en la Sección 2.3 y representada gráficamente en la Figura 2.6.

DMSIs PWM boost

La Figura 2.20 muestra un inversor DMSI tipo *boost*, cuya denominación se debe a la configuración particular de los elementos reactivos, que permite una operación elevadora de la tensión de salida, es decir $|v_{\text{out}}(t)| \geq V_{\text{in}}$.

Este diseño corresponde a uno de los primeros desarrollos de inversores en modo diferencial documentados en la bibliografía especializada [11, 20].

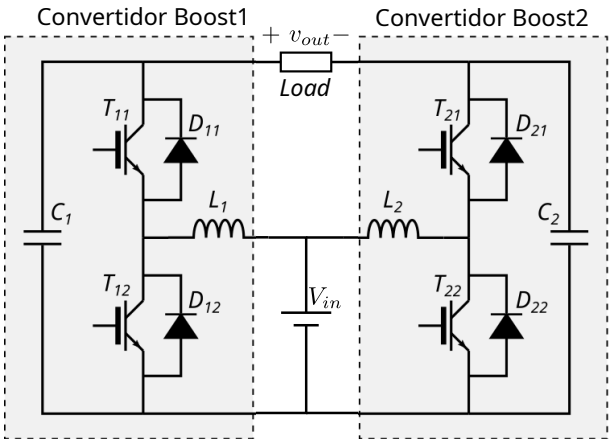


Figura 2.20: Esquema circuital del DMSI PWM tipo boost.

El encendido y apagado de los transistores se realiza de manera análoga a lo explicado en el caso del inversor buck.

DMSI PWM buck–boost

Entre las topologías DMSI más difundidas se encuentran aquellas de tipo reductor–elevador ya que permiten una mayor flexibilidad en los niveles de tensión de salida. El esquema más sencillo dentro de los inversores que cumplen con esta característica es el DMSI PWM buck–Boost. En la Figura 2.21 se muestra el esquema circuital básico del inversor PWM tipo buck–Boost en configuración diferencial.

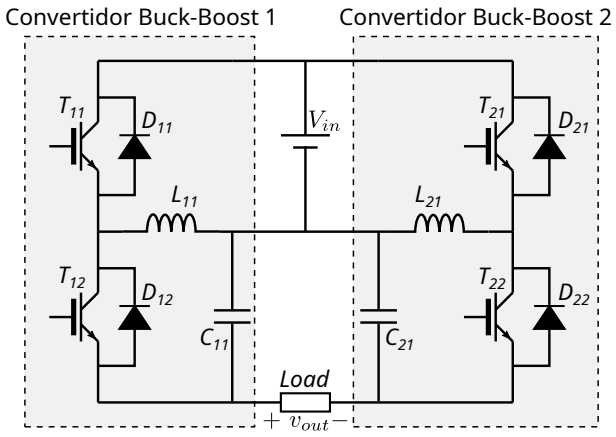


Figura 2.21: Esquema circuital del DMSI PWM buck–Boost.

DMSI PWM Ćuk

Por último, también puede implementarse el inversor DMSI PWM Ćuk [76], cuyo esquema circuital básico se muestra en la Figura 2.22.

Este inversor, al igual que el DMSI PWM buck–boost, tiene la característica de ser un inversor elevador reductor y es uno de los más utilizados ya que permite implementar un aislamiento galvánico entre la entrada y la salida. Esta topología suele ser una de las más utilizadas en aplicaciones de conversión dentro de sistemas de energía renovable.

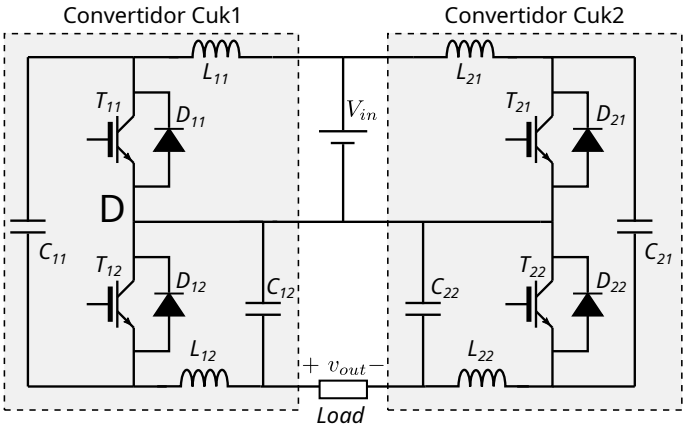


Figura 2.22: Esquema circuital del DMSI PWM Ćuk.

3. MODELADO DE CONVERTIDORES CONMUTADOS

3.1 Introducción

El modelado de convertidores conmutados PWM orientado a la simulación ha sido un campo de desarrollo activo desde fines del siglo XX, impulsado por el avance sostenido de las capacidades tecnológicas de procesamiento digital [63, 82, 87].

El modelado de estos dispositivos constituye una etapa fundamental en el diseño y análisis de sistemas electrónicos de potencia, ya que el nivel de detalle de los modelos permite predecir diversos aspectos del comportamiento del sistema bajo distintas condiciones de operación [3].

Este capítulo presenta las principales estrategias de modelado de convertidores PWM descritas en la bibliografía especializada. Se analizan tanto los modelos conmutados como los promediados, destacando las ventajas y limitaciones de cada enfoque. Los modelos conmutados representan con gran detalle la dinámica real, pero requieren un alto costo computacional. Por otro lado, los modelos promediados reducen esa complejidad mediante suposiciones simplificadoras, aunque a costa de enmascarar las dinámicas rápidas de las conmutaciones [81, 90].

3.2 Estrategias de modelado de convertidores PWM

Estos sistemas presentan desafíos para el modelado debido a la presencia de elementos como diodos y transistores, cuyos estados de corte y conducción generan discontinuidades muy frecuentes en las señales de tensión y corriente. Esta característica no lineal y la estructura variable complica tanto el análisis como la simulación, ya que las ecuaciones dinámicas asociadas cambian de forma a intervalos muy breves y sus parámetros y condiciones dependen de variables que conmutan a frecuencias elevadas.

Debido a las dificultades encontradas tanto en el modelado como en la simulación, se han desarrollado diversas estrategias de modelado. Dentro de estas estrategias se distinguen dos enfoques claramente diferenciados. En uno de ellos se busca representar con detalle todas las características de estos dispositivos, incluyendo tanto las dinámicas lentas

como las rápidas asociadas a las conmutaciones. El gran inconveniente de este enfoque es que requiere un muy alto costo computacional, llegando a ser el mismo en muchos casos inadmisible. Por este motivo, en el enfoque alternativo, se introducen suposiciones simplificadoras con el objetivo de reducir el costo computacional. Estas hipótesis enmascaran usualmente las dinámicas rápidas asociadas a las conmutaciones.

En esta sección, en primer lugar se analizan los modelos conmutados, los cuales representan con gran detalle la dinámica real de los convertidores al incorporar explícitamente el comportamiento de los elementos de conmutación.

Posteriormente, se desarrollan los modelos promediados, que simplifican la dinámica al aproximar el comportamiento dinámico en un marco continuo logrando así reducir sustancialmente la demanda computacional para su simulación. Esta estrategia resulta especialmente útil para el diseño y la interpretación de la respuesta global del sistema. Es particularmente ventajosa en simulaciones de larga duración [25, 81].

A continuación, se discutirán las ventajas y limitaciones de ambas estrategias, destacando los compromisos entre nivel de detalle y eficiencia computacional. Además se establecerán criterios que permiten discernir en que casos son más adecuados cada uno de estos métodos.

Finalmente, se introducirá un enfoque más reciente específico para convertidores PWM de CC–CC basado en modelos mixtos. Este enfoque, que combina las ventajas de los modelos conmutados y promediados, fue la base para uno de los principales aportes originales esta tesis: una nueva estrategia de modelado mixto para DMSIs PWM.

3.2.1 Modelos Conmutados

Los modelos conmutados de convertidores se construyen a partir de representaciones matemáticas que contemplan los distintos estados de corte y conducción de los elementos discontinuos, como diodos y transistores. En este enfoque, el modelo del circuito equivalente presenta cambios de estructura y/o de los valores de sus parámetros el estado de conducción de estos dispositivos, determinado tanto por la señal de control como por las condiciones eléctricas del propio circuito.

Por lo tanto, el modelo conmutado es la representación más detallada del convertidor al contemplar todas sus dinámicas. Sin embargo, para representar las dinámicas asociadas a las conmutaciones rápidas de los dispositivos de conmutación se generan familias de Ecuaciones Diferenciales Ordinarias (EDO) con cambios en sus parámetros o en su estructura. Los casos con cambios estructurales no pueden simularse —o resultan muy difíciles de simular— con herramientas de simulación estándar. En cambio, cuando los cambios son únicamente paramétricos, sí es posible simularlos, aunque se requiere el uso de algoritmos numéricos capaces de detectar y procesar adecuadamente los instantes de ocurrencia. Por este motivo, en entornos de simulación digital, la ejecución de estos modelos resulta compu-

tacionalmente muy costosa o incluso inviable cuando se pretende simular largos tiempos de simulación.

En este enfoque se debe entonces definir cómo se representan las características eléctricas de los dispositivos de conmutación. A continuación se describen las hipótesis de modelado más habituales para estos elementos cuando se desea reflejar explícitamente su comportamiento conmutado.

Modelado de Elementos Conmutados

El comportamiento de los elementos de conmutación puede modelarse a grandes rasgos de dos maneras: como interruptores ideales binarios (que presentan impedancia nula o infinita según el estado de conducción) o a través aproximación de su característica v-i mediante funciones continuas. En este último caso se pueden utilizar complejas funciones matemáticas no lineales o aproximaciones seccionalmente lineales.

Si se modelan los elementos de conmutación como interruptores binarios se obtienen modelos que no son tan realistas (por ejemplo no contemplan la disipación de potencia del dispositivo de conmutación) y suele conducir a modelos con cambios de estructura que no pueden ser simulados directamente. Si en cambio se opta por modelos realistas de los elementos de conmutación que representan su característica v-i mediante aproximaciones seccionalmente lineales, se obtienen modelos que no presentan cambios de estructura sino solo en sus parámetros.

Modelo de Diodos

Los diodos son elementos fundamentales en la electrónica de potencia y presentan un comportamiento no lineal, determinado por su capacidad de conmutar entre los estados de conducción (on) y corte (off). Esta dinámica puede analizarse a partir de su curva característica tensión-corriente, mostrada en la Figura 3.1 (a).

Modelar con precisión la característica de estos componentes resulta crucial para predecir adecuadamente el comportamiento dinámico de los circuitos que los incorporan. La principal diferencia entre un modelo ideal (Figura 3.1 (b)) y uno realista (Figura 3.1 (c)) radica en el grado de simplificación con que se representa la respuesta eléctrica del dispositivo. Mientras que el modelo ideal asume un cambio instantáneo y sin pérdidas entre los estados de conducción, el modelo realista incorpora una aproximación lineal por tramos que refleja mejor la conducta física del dispositivo en cada región de operación.

El modelo ideal se emplea en análisis teóricos y estudios preliminares donde se prioriza la comprensión del funcionamiento general del circuito por sobre la precisión cuantitativa. Además, si se usan modelos ideales de los diodos al modelar circuitos que incorporan estos componentes se obtienen modelos con cambios de estructura.

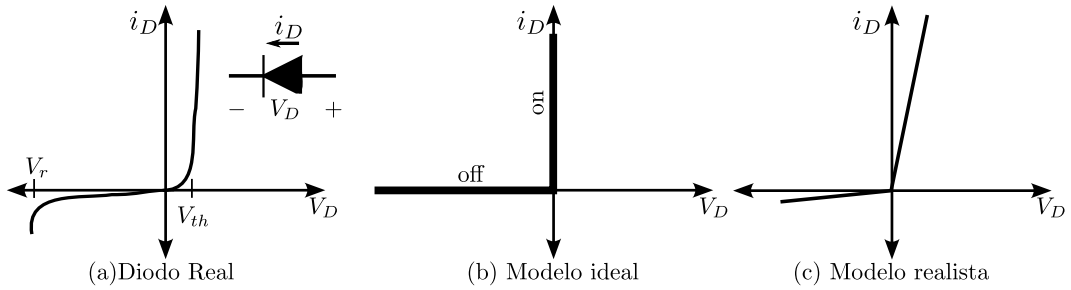


Figura 3.1: Curvas tensión-corriente para: a) un diodo real, b) un modelo ideal, y c) un modelo realista basado en linealización por tramos.

Para evitar los cambios de estructura y tener una mejor representación del comportamiento de estos elementos, los modelos de circuitos que integran diodos suelen implementar modelos realistas de los mismos.

El modelo realista utilizado para representar el comportamiento del diodo se basa en una **aproximación lineal por tramos**. Este enfoque simplifica la característica corriente–tensión (I–V) del diodo real, sustituyéndola por un modelo resistivo cuya conductancia varía en función del estado de polarización del componente.

- **Estado off:** Cuando el diodo está polarizado inversamente ($V_D < 0$), se representa mediante una resistencia elevada, R_{Off}.
- **Estado on:** Cuando el diodo está polarizado directamente ($V_D > 0$), se modela mediante una resistencia baja, R_{On}, permitiendo la conducción con una caída de tensión mínima.

La ley que define el valor de la resistencia efectiva del diodo, R_D, puede expresarse como:

$$R_D = \begin{cases} R_{\text{on}} & \text{si } V_D > 0, \\ R_{\text{off}} & \text{si } V_D < 0. \end{cases} \tag{3.1}$$

Observar que esta representación matemática de la característica del diodo se corresponde con la mostrada en la Figura 3.1 (c).

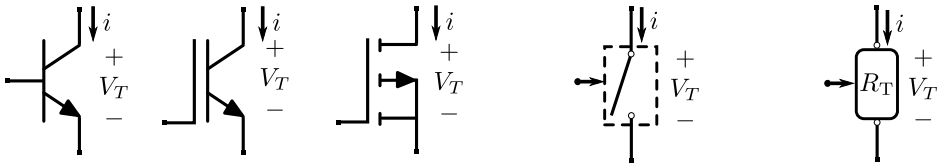
Esta forma de modelar el diodo permite evitar los problemas asociados a los cambios estructurales que presentan los modelos conmutados con diodos ideales. Aun así, persisten las discontinuidades en las EDOs que dependen de la impedancia del diodo, ya que esta varía según su estado de conducción.

Además de evitar los problemas asociados a los cambios de estructura, si se seleccionan adecuadamente los valores de R_{on} y R_{off}, de modo que representen las impedancias reales del elemento de conmutación en los estados de conducción y corte, los modelos

resultantes serán más precisos y permitirán, por ejemplo, realizar análisis detallados de la disipación de potencia en dichos elementos. Esta consideración es extensiva a todos los elementos de conmutación modelados mediante cambios de impedancia.

Modelado de Transistores

Los transistores operando en corte y saturación actúan como interruptores controlables fundamentales en los convertidores conmutados. En este modo de operación, pueden representarse como se muestra en la Figura 3.2. Su modelado puede abordarse a partir de dos hipótesis.



(a)Transitores BJT, IGBT y MOSFET (b) Modelo ideal (c) Modelo realista

Figura 3.2: (a) Símbolos de los transistores; (b) característica idealizada de conmutación; (c) comportamiento realista del dispositivo.

El modelo ideal representa al transistor como un elemento binario (circuito abierto o cerrado) en función del estado de su compuerta de comando (Figura 3.2 (b)). Esta aproximación es la más apropiada para analizar cualitativamente el funcionamiento de circuitos que integran transistores operando como interruptores.

El modelo realista de estos dispositivos, de manera similar a lo que ocurre con el modelo realista del diodo, aproxima el comportamiento del transistor mediante una característica seccionalmente lineal. Sin embargo, a diferencia del diodo, en el caso del transistor el estado de conducción no está determinado únicamente por la tensión en bornes, sino también por la señal de control aplicada a la compuerta (Figura 3.2 (c)). De este modo, la característica de los transistores operando como interruptores puede escribirse como:

$$R_T = \begin{cases} R_{\text{on}} & \text{si } \textit{State} = \text{on}, \\ R_{\text{off}} & \text{si } \textit{State} = \text{off}. \end{cases} \tag{3.2}$$

donde *State* indica si el transistor se encuentra en conducción o no. El mecanismo mediante el cual el transistor alcanza dicho estado depende del tipo específico de dispositivo utilizado, y no resulta relevante para el presente análisis. Esta estrategia de modelado permite capturar las pérdidas asociadas al estado de conducción.

Por este motivo, es común representar los transistores mediante un símbolo genérico o un interruptor controlado en los circuitos donde operan en corte y saturación, independientemente del tipo específico de dispositivo. Además, suele omitirse el modelado detallado

del circuito de control de la compuerta, salvo en situaciones particulares en las que dicho nivel de detalle sea relevante.

Modelo Realista de transistor con diodo en antiparalelo

Como se presentó en el Capítulo 2, los DMSIs utilizan transistores con diodo antiparalelo, cuyo símbolo eléctrico se muestra en la Figura 3.3 (a).

(a)Transistor con diodo en antiparalelo (b) Modelo realista

Figura 3.3: (a) Símbolo eléctrico de un transistor con diodo antiparalelo; (b) modelo realista del dispositivo.

En este caso, utilizando las mismas hipótesis de modelado realista aplicadas a diodos y transistores, el dispositivo compuesto por un transistor con diodo antiparalelo puede modelarse como una resistencia controlada R_{T_P}, cuyo valor depende del estado de activación del transistor y el diodo. Para este modelo se utilizará el símbolo mostrado en la Figura 3.3 (b).

La expresión correspondiente para R_{T_P} es:

$$R_{T_P} = \begin{cases} R_{\text{on}} & \text{si T está activado } (\textit{State} = \text{on}) \\ R_{\text{on}} & \text{si T está desactivado } (\textit{State} = \text{off}) \text{ y D polarizado directo } (v_{T_P} < 0) \\ R_{\text{off}} & \text{si T está desactivado } (\textit{State} = \text{off}) \text{ y D polarizado inverso } (v_{T_P} \geq 0) \end{cases} \quad (3.3)$$

A continuación se presentan algunos ejemplos de aplicación del proceso de modelado en convertidores conmutados típicos.

Ejemplo de convertidor conmutado CC–CC PWM tipo *boost*

En el convertidor elevador (*boost*) de la Figura 3.4, se observa el control de activación del transistor T, mediante el cual se obtiene una tensión promedio de salida V_{out} mayor que la tensión de entrada V_{in}.

La operación del circuito, al modelar los elementos de conmutación como interruptores o resistencias controladas, puede dividirse en tres modos de funcionamiento diferenciables, según el estado de conducción del transistor y del diodo.

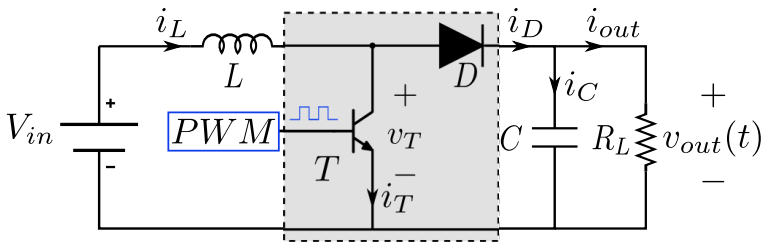


Figura 3.4: Convertidor conmutado CC–CC *boost* con control de transistor.

Al realizar un modelo de este convertidor utilizando *modelos ideales* de los elementos de conmutación —es decir, al reemplazar el transistor y el diodo por interruptores controlados, Sw y D— se distinguen tres condiciones de operación claramente identificables, según el estado en que se encuentren el transistor y el diodo.

La primera condición corresponde al transistor (interruptor) conduciendo y el diodo en corte (durante T_{on}). La segunda ocurre cuando el transistor está en corte y el diodo en conducción (durante T_{off}). La tercera condición se presenta cuando tanto el transistor como el diodo están en corte. Esta última situación solo ocurre cuando el convertidor opera en MCD.

$$\begin{cases} Sw = \text{on} \quad D = \text{off} & \text{Condición 1} \\ Sw = \text{off} \quad D = \text{on} & \text{Condición 2} \\ Sw = \text{off} \quad D = \text{off} & \text{Condición 3 (MCD)} \end{cases} \tag{3.4}$$

Estas tres condiciones están unívocamente determinadas por el estado de activación del transistor y por la tensión V_D en el diodo:

$$\begin{cases} \text{Condición 1} & T \text{ activado y } V_D \leq 0 \\ \text{Condición 2} & T \text{ desactivado y } V_D > 0 \\ \text{Condición 3} & T \text{ desactivado y } V_D \leq 0 \end{cases} \tag{3.5}$$

Las EDOs que rigen el comportamiento del circuito para las tres condiciones son:

Condición 1	**Condición 2**	**Condición 3**
$L\frac{di_L}{dt} = V_{\text{in}}$ $C\frac{dv_C}{dt} = -\frac{v_C}{R_L}$	$L\frac{di_L}{dt} = V_{\text{in}} - V_C$ $C\frac{dV_C}{dt} = i_L - \frac{V_C}{R_L}$	$C\frac{dV_C}{dt} = -\frac{V_C}{R_L}$

La ecuación que vincula la tensión de salida es:

$$v_{\text{out}} = v_C \tag{3.6}$$

Obsérvese que en la *Condición 3*, al estar la inductancia desconectada del circuito, el modelo en EDOs resulta de orden inferior.

Como se puede observar, este enfoque genera modelos que presentan cambios de estructura, donde distintos conjuntos de ecuaciones de estado describen cada configuración posible. Para un circuito con N componentes de conmutación, se requieren 2^N conjuntos de ecuaciones, lo cual implica una carga significativa de trabajo para el usuario al implementar estos modelos en herramientas de simulación. Además, en caso de incluir condiciones de conducción discontinua, el número de configuraciones posibles puede aumentar.

Además, este tipo de modelado ideal enmascara todas las pérdidas de energía asociadas a los elementos de conmutación, como las caídas de tensión en conducción.

Si en cambio se utilizan *modelos realistas* de los elementos de conmutación para construir el modelo de este convertidor —es decir, si se reemplazan el interruptor y el diodo por resistencias controladas, R_T y R_D, que conmutan entre un valor muy bajo R_{on} (cuando están en conducción) y un valor muy alto R_{off} (cuando están en corte)—, el modelo en EDOs que describe el comportamiento del convertidor puede expresarse de forma unificada, independientemente del estado de conducción del diodo o del transistor, según:

$$\begin{aligned} L\frac{di_L}{dt} &= -\frac{R_D R_T}{R_D+R_T}\, i_L + \frac{R_T}{R_D+R_T}\, v_C + V_{\text{in}} \\ C\frac{dv_C}{dt} &= \frac{R_T}{R_D+R_T}\, i_L + \left(\frac{1}{R_D+R_T} - \frac{1}{R_L}\right) v_C \end{aligned} \tag{3.7}$$

A diferencia de lo que ocurre al modelar los elementos de conmutación de manera ideal, en este caso las tres situaciones mostradas anteriormente no implican cambios estructurales en el modelo. En cambio, se traducen únicamente en una variación de los valores de los parámetros R_T y R_D:

$$\begin{cases} R_T = R_{\text{on}},\ R_D = R_{\text{off}} & \text{Condición 1} \\ R_T = R_{\text{off}},\ R_D = R_{\text{on}} & \text{Condición 2} \\ R_T = R_{\text{off}},\ R_D = R_{\text{off}} & \text{Condición 3} \end{cases} \tag{3.8}$$

La variable V_D, utilizada para controlar el estado de conducción del diodo, puede calcularse como:

$$V_D = R_D \left(\frac{V_{\text{in}} - V_D}{R_T} - i_L\right) \tag{3.9}$$

A partir de las EDOs de la Ecuación3.7 se puede tener la matriz Jacobiana:

$$J_{\text{Boost}} = \begin{bmatrix} -\dfrac{R_D R_T}{L(R_D+R_T)} & \dfrac{R_T}{L(R_D+R_T)} \\ \dfrac{R_T}{C(R_D+R_T)} & \dfrac{1}{C}\left(\dfrac{1}{R_D+R_T} - \dfrac{1}{R_L}\right) \end{bmatrix} \tag{3.10}$$

Debe notarse que modelar el interruptor y el diodo mediante elementos no ideales, con el objetivo de evitar el problema de la estructura variable, tiene un costo asociado. El modelo resultante puede volverse rígido en determinadas condiciones de conducción del interruptor y del diodo. Por ejemplo, en el caso del convertidor Boost, la matriz Jacobiana correspondiente a la condición 1 de conmutación (Ecuación 3.8) resulta:

$$J_{\text{Boost}_{\text{cond1}}} = \begin{bmatrix} -\dfrac{R_{\text{off}} R_{\text{on}}}{L(R_{\text{off}}+R_{\text{on}})} & \dfrac{R_{\text{on}}}{L(R_{\text{off}}+R_{\text{on}})} \\ \dfrac{R_{\text{on}}}{C(R_{\text{off}}+R_{\text{on}})} & \dfrac{1}{C}\left(\dfrac{1}{R_{\text{off}}+R_{\text{on}}} - \dfrac{1}{R_L}\right) \end{bmatrix} \approx \begin{bmatrix} -\dfrac{1}{LR_{\text{off}}} & \dfrac{R_{\text{on}}}{LR_{\text{off}}} \\ \dfrac{R_{\text{on}}}{CR_{\text{off}}} & -\dfrac{1}{CR_L} \end{bmatrix}. \tag{3.11}$$

En esta matriz puede observarse que los términos fuera de la diagonal principal son prácticamente nulos cuando $R_{\text{on}} \ll R_{\text{off}}$. Además, si $R_L \ll R_{\text{off}}$, los autovalores de la matriz difieren en varios órdenes de magnitud, por lo que el sistema se vuelve rígido. En consecuencia, la simulación de estos modelos requiere pasos de integración más pequeños, incrementando el número de eventos y la demanda computacional.

A medida que la relación $R_{\text{on}}/R_{\text{off}}$ deja de ser tan extrema, la rigidez del sistema disminuye y, con ello, los requerimientos de cómputo. Aunque este análisis se presenta para un caso particular, el mismo comportamiento se extiende de manera análoga a otras topologías de convertidores conmutados.

Debido a que esta estrategia evita los cambios estructurales y no pierde información de las perdidas energéticas en los transistores, será la estrategia usada como modelo conmutado en el resto de la tesis.

Ejemplo de un DMSI PWM *boost*.

Como se presentó en el capítulo anterior, el DMSI *boost* es un convertidor CC–CA cuya principal ventaja es su capacidad para generar una tensión de pico CA de salida mayor que la tensión continua de entrada.

El DMSI *boost* está compuesta por dos convertidores conmutados *boost* CC–CC bidireccionales, entre los cuales se conecta la carga de modo diferencial.

Del mismo modo en que se realizó el procedimiento de modelado para el convertidor *boost* CC–CC, para obtener el modelo conmutado del inversor se reemplaza los transistores T_{Pij} por resistencias controladas $R_{T_{Pij}}$ ($i = 1, 2$ representan el convertidor o rama a la que pertenece el transistor). Luego, para construir el modelo matemático, se toman como variables de estado del inversor *boost* las corrientes $i_{L_{11}}$, $i_{L_{21}}$ y las tensiones $v_{C_{11}}$, $v_{C_{21}}$. Las

tensiones V_{in} y v_{out} son como variables de entrada y de salida, respectivamente.

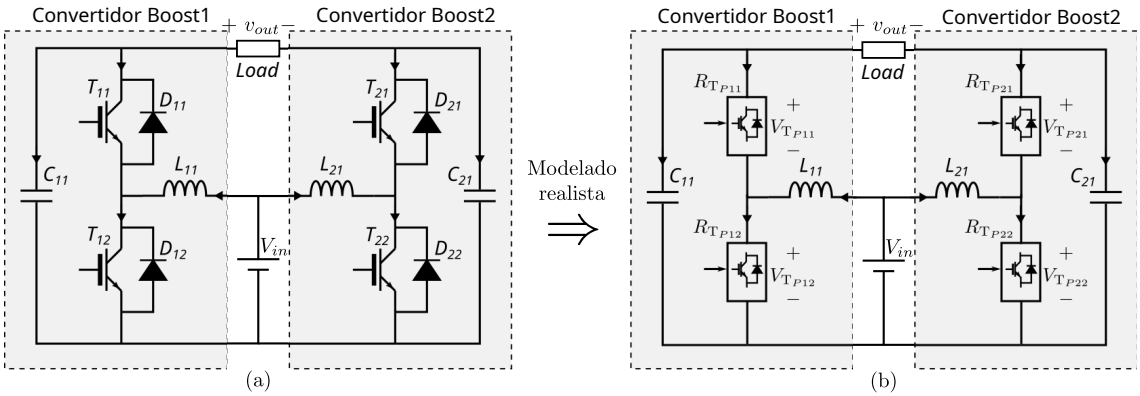


Figura 3.5: DMSI PWM *boost*: a) Modelo conmutado b) modelo conmutado realista.

A partir de la Figura 3.5 y considerando un *modelo realista* de los transistores con diodo antiparalelo, se obtiene a partir de las siguientes EDOs, correspondientes:

$$\frac{di_{L11}}{dt} = -\alpha_1 i_{L11} - \beta_1 v_{C11} + \frac{1}{L_1 1} V_{\text{in}} \tag{3.12}$$

$$\frac{di_{L21}}{dt} = -\alpha_2 i_{L21} - \beta_2 v_{C21} + \frac{1}{L_2 1} V_{\text{in}} \tag{3.13}$$

$$\frac{dv_{C11}}{dt} = \gamma_1 i_{L11} - \delta_1 v_{C11} + \varepsilon_1 v_{C21} \tag{3.14}$$

$$\frac{dv_{C21}}{dt} = \gamma_2 i_{L21} + \varepsilon_2 v_{C11} - \delta_2 v_{C21} \tag{3.15}$$

Los coeficientes que aparecen en las ecuaciones anteriores agrupan parámetros del circuito (resistencias, inductancias y capacidades) y se definen del siguiente modo:

$$\alpha_1 = \frac{R_{T_{P11}} R_{T_{P12}}}{(R_{T_{P11}} + R_{T_{P12}}) L_1}, \qquad \beta_1 = \frac{R_{T_{P11}}}{L_1 (R_{T_{P11}} + R_{T_{P12}})}$$

$$\alpha_2 = \frac{R_{T_{P21}} R_{T_{P22}}}{(R_{T_{P21}} + R_{T_{P22}}) L_2}, \qquad \beta_2 = \frac{R_{T_{P21}}}{L_2 (R_{T_{P21}} + R_{T_{P22}})}$$

$$\gamma_1 = \frac{R_{T_{P11}}}{(R_{T_{P11}} + R_{T_{P12}}) C_1}, \qquad \delta_1 = \frac{1}{C_1}\left(\frac{1}{R_{\text{Load}}} + \frac{1}{R_{T_{P11}} + R_{T_{P12}}}\right), \qquad \varepsilon_1 = \frac{1}{R_{\text{Load}} C_1}$$

$$\gamma_2 = \frac{R_{T_{P21}}}{(R_{T_{P21}} + R_{T_{P22}}) C_2}, \qquad \delta_2 = \frac{1}{C_2}\left(\frac{1}{R_{\text{Load}}} + \frac{1}{R_{T_{P21}} + R_{T_{P22}}}\right), \qquad \varepsilon_2 = \frac{1}{R_{\text{Load}} C_2}$$

Estos coeficientes permiten representar las pérdidas resistivas en los interruptores y el efecto de la carga resistiva de salida, condensando múltiples relaciones no lineales en una formulación linealizada adecuada.

3.2.2 Modelos Promediados

La simulación puede volverse compleja en sistemas de suministro de energía, como las micro-redes compuestas por numerosos convertidores conmutados. Esta complejidad resulta particularmente crítica en estudios a largo plazo, tales como la generación diaria de energía a partir de fuentes solares y eólicas, o la variación lenta del estado de carga en sistemas de almacenamiento.

Las conmutaciones de alta frecuencia presentes en los convertidores limitan el paso de integración a intervalos extremadamente pequeños —con relación al tiempo total de simulación— y obligan a emplear costosos algoritmos de detección de discontinuidades para evitar integrar a través de ellas. Ello puede ralentizar considerablemente la simulación e, incluso, hacer que el tiempo de ejecución sea inaceptable [21].

En consecuencia, dada la creciente necesidad de contar con modelos de sistemas que incluyan convertidores —por ejemplo, para evaluar el funcionamiento a largo plazo de la generación distribuida con energías renovables y almacenamiento—, surge la conveniencia de utilizar modelos cuya demanda de *CPU* sea menor para este tipo de análisis [4].

En consecuencia, es habitual utilizar versiones simplificadas de los modelos de los convertidores. La estrategia más difundida es la de los *modelos promediados*. Su idea principal consiste en desarrollar representaciones que sólo consideran la evolución del valor medio de las variables del sistema a lo largo de los períodos de conmutación T_{sw} [83, 64, 44, 25, 77, 89]. Este estrategia permite realizar un modelo continuo del sistema discontinuo original cuyos resultados de simulación preservan la evolución promedio de las variables del sistema original.

A continuación, se presenta una breve descripción del uso de esta técnica para distintos tipos de convertidores.

Modelos promediados de convertidores conmutados

Los *modelos promediados* resultan adecuados para describir la evolución media de las tensiones y corrientes en los convertidores conmutados. Con este objetivo solo se consideran los valores promedio de las variables de estado $x(t)$, usualmente las corrientes en los inductores y las tensiones en los capacitores durante un período de conmutación T_{sw}. Este valor medio depende del ciclo de trabajo d_c como se muestra en la Figura 3.6. Entonces:

$$\begin{aligned} \bar{x}(t) &= \tfrac{1}{T_{\text{sw}}} \int_{t_0}^{t_0+T_{\text{sw}}} x(\tau)\, d\tau \\ &= \tfrac{1}{T_{\text{sw}}} \left(\int_{t_0}^{t_0+d_c T_{\text{sw}}} x(\tau)\, d\tau + \int_{t_0+d_c T_{\text{sw}}}^{t_0+T_{\text{sw}}} x(\tau)\, d\tau \right) \end{aligned} \tag{3.16}$$

Promediando todas las señales sobre un período de conmutación, se obtiene un conjunto de ecuaciones diferenciales continuas, Ecuación 3.17, que describen el comportamiento

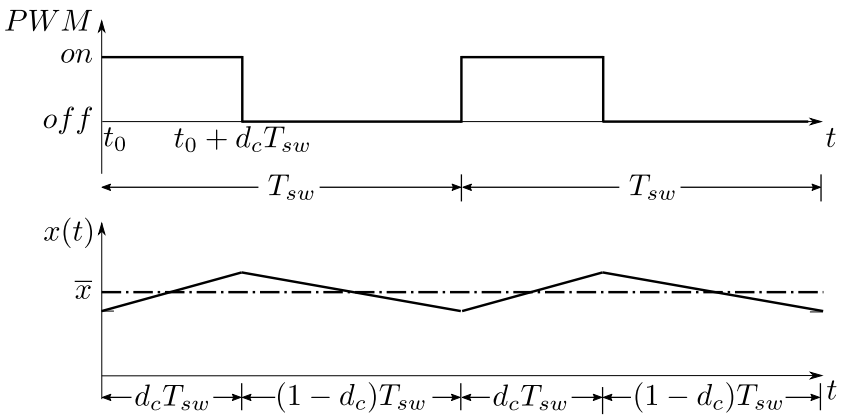


Figura 3.6: Promediado de la variable de estado $x(t)$ en un período de conmutación T_{sw}.

promedio de las variables del sistema es decir, no contempla los armónicos de alta frecuencia introducidos por la conmutación.

$$\dot{\bar{\mathbf{x}}} = f(\bar{\mathbf{x}}, \mathbf{u}) \tag{3.17}$$

De este modo se reduce la carga computacional para simular los modos a costa de sacrificar detalles en los resultados de simulación [25, 87]. Esta representación facilita el diseño y estudio de estabilidad de controladores y las simulaciones de largo plazo de sistemas que incluyan estos convertidores.

Modelos promediados de convertidores conmutados CC–CA

El *modelo promediado en el espacio de estados* de cualquier topología de DMSI PWM en el espacio de estado es:

$$\begin{aligned} \dot{\bar{\mathbf{x}}} &= \mathbf{A}_{\text{prom}}\, \bar{\mathbf{x}} + \mathbf{B}\, \mathbf{u}, \\ \mathbf{y} &= \mathbf{C}\, \mathbf{x} + \mathbf{D}\, \mathbf{u}, \end{aligned} \tag{3.18}$$

donde

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{i}_{L11} \\ \bar{i}_{L21} \\ \bar{v}_{C11} \\ \bar{v}_{C21} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} V_{\text{in}} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \bar{v}_o \end{bmatrix}. \tag{3.19}$$

En esta representación se tomó como variables de estado los valores medios de las corrientes en los inductores ($\bar{i}_{L11}$ y $\bar{i}_{L21}$) y las tensiones en los capacitores ($\bar{v}_{C11}$ y $\bar{v}_{C21}$). La matriz

$\mathbf{A}_{\text{prom}}$ se deriva de las distintas EDOs asociadas a todos los circuitos equivalentes que resultan de las distintas condiciones de conducción de los elementos de conmutación de los DMSI PWM. Para determinar la matriz $\mathbf{A}_{\text{prom}}$ se analizan los estados de conducción de los transistores dentro de un período de conmutación T_{sw}. Para este análisis se considera que los transistores tienen en todas las topologías de DMSI PWM la misma distribución que en el caso de la boost mostrado en la Figura 3.5 a) y se les asigna los mismos subíndices al nombrarlos.

Para calcular $\mathbf{A}_{\text{prom}}$ se considera entonces la secuencia de conmutación de los transistores mostrada en la Figura 3.7. Esta secuencia está gobernada por las señales PWM_1 y PWM_2. La señal PWM_1 controla el par T_{11}–T_{12}, mientras que la señal PWM_2 controla T_{21}–T_{22}. Los intervalo tiempo de conducción de cada par está determinado por los ciclos de trabajo d_{c_1} y d_{c_2}, respectivamente. En esta figura, los intervalos se definieron en función de d_{c_1} aprovechando que los ciclos de trabajo están relacionados por $d_{c_2} = 1 - d_{c_1}$ en los DMSI PWM. También se muestran las señales PWM_1 y PWM_2, los estados de conducción de todos los transistores en función de estas señales y se indicó con A_i con $i = 1, 2, 3$ a la matriz de estado asociada a los distintos estados de conducción de los transistores.

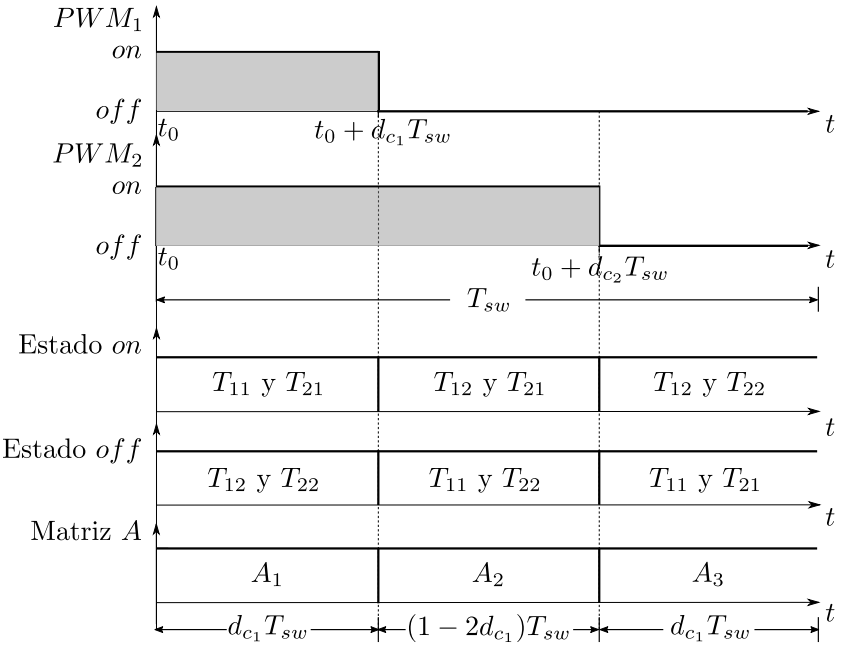


Figura 3.7: Diagrama secuencial de funcionamiento transistores en los DMSIs y las matrices resultantes en cada intervalo del período de conmutación T_{sw}.

La matriz promediada se obtiene entonces como

$$\mathbf{A}_{\text{prom}} = d_{c_1}\,\mathbf{A}_1 + (1 - 2d_{c_1})\,\mathbf{A}_2 + d_{c_1}\,\mathbf{A}_3. \tag{3.20}$$

Si bien con esta estrategia se obtiene un modelo continuo que puede simularse con muy bajo costo computacional en cualquier herramienta de simulación, calcular $\mathbf{A}_{\text{prom}}$ directamente mediante la expresión anterior suele implicar manipulaciones algebraicas extensas y propensas a errores. Además, al tratar de analizar pequeñas variaciones circuitales se debe realizar todo el proceso nuevamente. Por estos motivos, para esta tesis se adopta la estrategia el modelado promediado a nivel circuital. Esta otra estrategia de promediación es más fácil de aplicar y permite escribir el modelo en pequeña señal casi de forma directa.

Métodos de Promediado Circuital

La promediación circuital es otra técnica ampliamente conocida para la obtención de circuitos equivalentes de convertidores. En lugar de promediar las ecuaciones de estado del convertidor, con la técnica de promediación circuital se promedian directamente las formas de onda del convertidor [25]. En esta estrategia, las manipulaciones se realizan sobre el diagrama del circuito, en lugar de sobre sus ecuaciones. Por lo tanto, la promediación circuital brinda una interpretación más física del modelo.

El paso clave en la promediación circuital consiste en reemplazar los interruptores del convertidor por fuentes de tensión y corriente, para así obtener una topología de circuito invariante en el tiempo.

Por esta razón, para abordar la modelización promediada a nivel circuital, resulta útil considerar que un convertidor conmutado puede entenderse como un sistema dinámico híbrido. El mismo combina dinámicas continuas asociadas a la evolución de tensiones y corrientes en elementos almacenadores de energía con transiciones discretas asociadas a los cambios de estado de los dispositivos de conmutación [38]. Por lo tanto, podemos separar la estructura del sistema en subsistemas tal como se puede observar en la Figura 3.8. En la misma se observa que uno de los subsistemas contiene los elementos invariantes en el tiempo (elementos reactivos y resistivos), y el otro contiene los elementos de conmutación.

Los elementos de conmutación se agrupan de a pares de componentes que conducen y se cortan secuencialmente, es decir, cuando uno deja de conducir el otro comienza a conducir y viceversa. Cada uno de estos pares conforman lo que se denomina una celda de conmutación. En la Figura 3.9 a) puede verse como se conforma una celda de conmutación genérica a partir de dos interruptores que se abren y cierran secuencialmente. En las Figuras 3.9 b) y c) pueden verse los componentes electrónicos con los que se implementan típicamente estas celdas en los circuitos electrónicos tales como convertidores conmutados e inversores.

La idea fundamental de ésta metodología es entonces obtener un modelo promediado, denominado en adelante celda de dos puertos, de la celda de conmutación manteniendo sin cambios los componentes continuos (resistencias, inductancias y capacitores) y su interconexión. La celda de dos puertos resultante puede entonces ser insertado en el modelo

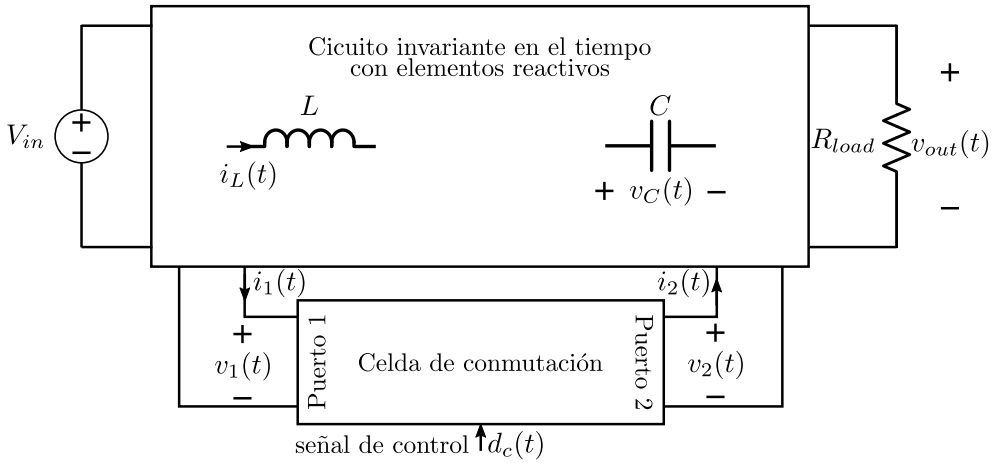


Figura 3.8: Convertidor conmutado interpretado como una celda de conmutación conectada a un subsistema invariante en el tiempo.

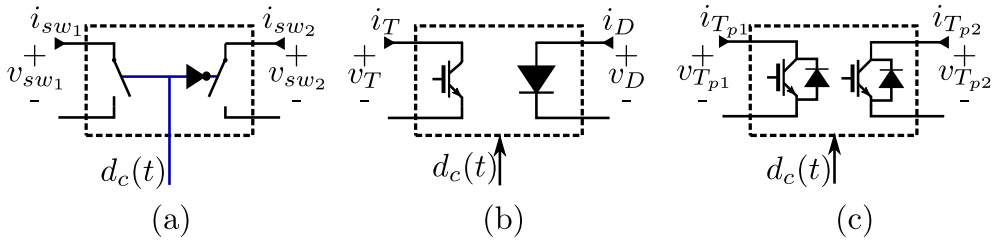


Figura 3.9: Celdas de conmutación.

del circuito del convertidor reemplazando la celda de conmutación correspondiente para obtener un modelo de circuito promediado completo del convertidor. Además, de este modo, el mismo modelo de celda de conmutación puede reutilizarse para múltiples topologías de convertidores conmutados en los que solo cambian los elementos invariantes en el tiempo y/o su distribución. Esta es en una de las principales ventajas de este enfoque ya que le otorga gran versatilidad.

En este sentido, como se muestra en la Figura 3.10, las celda de conmutación de la Figura 3.10 a)) son reemplazados por celdas de dos puertos integradas por fuentes controladas como se observa en la Figura 3.10 b)). En este caso, las tensiones de entrada y salida de la celda de conmutación son las tensiones en el interruptor y en el diodo ($v_{sw}(t)$ y $v_D(t)$) y las corrientes de entrada y salida de esta celda son las corrientes en el interruptor y en el diodo ($i_{sw}(t)$ e $i_D(t)$). Analogamente, las tensiones de entrada y salida de la celda de dos puertos son las tensiones medias en las fuentes controladas de tensión y corriente ($\overline{v}_{sw}(t)$ y $\overline{v}_D(t)$) y las corrientes de entrada y salida de esta celda son ($\overline{i}_{sw}(t)$ e $\overline{i}_D(t)$)

Para deducir como se relaciona la celda de conmutación con la celda de dos puertos matemáticamente, primero se debe tener en cuenta que el ciclo de trabajo $d_c(t)$ actúa como la entrada de control independiente en las celdas de conmutación. Luego, tomando la tensión

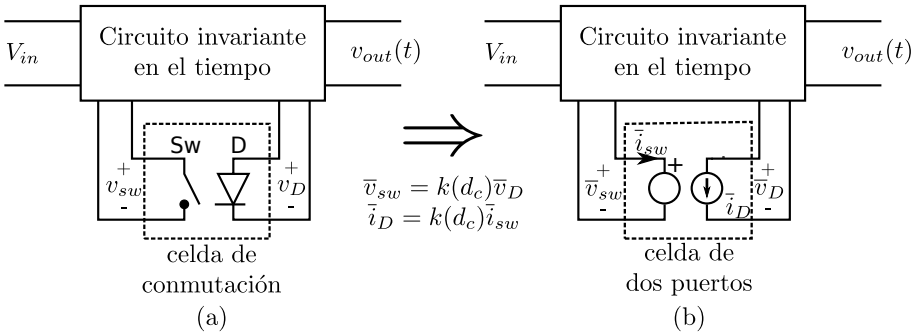


Figura 3.10: Aproximación del modelo circuital promediado de un convertidor conmutado: a) Modelo híbrido y b) Modelo promediado.

media $\overline{v}_D(t)$ y la corriente media $\overline{i}_{sw}(t)$) en un período $T_{\mathbf{sw}}$ como variables independientes, las salidas dependientes $v_{sw}(t)$ e $i_D(t)$ resultan:

$$\begin{aligned} \overline{v}_{sw}(t) &= K(d_c) \cdot \overline{v}_D(t) \\ \overline{i}_D(t) &= K(d_c) \cdot \overline{i}_{sw}(t) \end{aligned} \tag{3.21}$$

En estas ecuaciones, las variables controladas $\overline{v}_{sw}(t)$ e $\overline{i}_D(t)$ de las fuentes controladas son idénticas a los valores medios de las variables dependientes originales $v_{sw}(t)$ e $i_D(t)$. Por lo tanto los circuitos son eléctricamente equivalentes.

Para establecer la relación entre las tensiones y entre las corrientes de la celda de dos puertos se analiza las formas de onda típica de tensión y corriente de una celda de conmutación de un circuito conmutado operando en MCC (Figura 3.11).

Sea $\hat{\overline{i}}_D$ el valor medio de la corriente del diodo durante el estado *off* del interruptor (con duración $(1 - d_c) \cdot T_{\mathbf{sw}}$), y sea $\hat{\overline{i}}_{sw}$ el valor medio de la corriente del interruptor durante su estado *on* (con duración $d_c \cdot T_{\mathbf{sw}}$). Entonces, los valores medios de estas señales durante un período completo $T_{\mathbf{sw}}$ se calculan como:

$$\overline{i}_D = \hat{\overline{i}}_D \cdot (1 - d_c) \tag{3.22}$$

$$\overline{i}_{sw} = \hat{\overline{i}}_{sw} \cdot d_c \tag{3.23}$$

Teniendo en cuenta que la celda de dos puertos no consume potencia, debe cumplirse la siguiente ecuación:

$$P_D = \overline{i}_D \cdot \overline{v}_D = \overline{i}_{sw} \cdot \overline{v}_{sw} = P_{sw} \tag{3.24}$$

Por lo tanto, la ganancia de la fuente controlada $K(d_c)$ verifica:

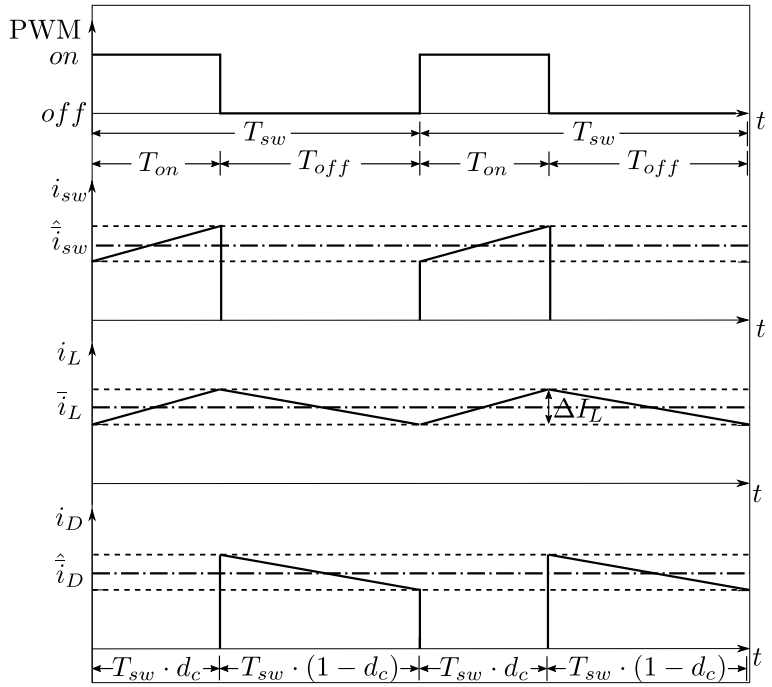


Figura 3.11: Formas de ondas de un convertidor en estado estacionario en MCC.

$$K(d_c) = \frac{\overline{v}_{sw}}{\overline{v}_D} = \frac{\bar{i}_D}{\hat{\bar{i}}_{sw}} = \frac{\hat{\bar{i}}_D(1-d_c)}{\hat{\bar{i}}_{sw} d_c} = k_1 \cdot \frac{1-d_c}{d_c} \tag{3.25}$$

Luego, en conducción continua $\hat{\bar{i}}_D = \hat{\bar{i}}_{sw}$ y $k_1 = 1$. En este caso la función de las fuentes controladas se puede escribir como:

$$\begin{aligned} \overline{v}_{sw} &= \overline{v}_D \tfrac{1-d_c}{d_c} \\ \bar{i}_D &= \bar{i}_{sw} \tfrac{1-d_c}{d_c} \end{aligned} \tag{3.26}$$

El modelado promediado de convertidores PWM constituye una herramienta fundamental para el análisis y diseño en electrónica de potencia. Presenta varias ventajas, entre las que se destacan la reducción de la complejidad computacional de sus modelos, lo que permite que las simulaciones muestren rápidamente la evolución de los valores medios de sus variables, y la simplificación del diseño de controladores basados en la respuesta dinámica del sistema, ya que permiten realizar análisis en el dominio frecuencial.

Además, es muy sencillo construir modelos de convertidores conmutados o inversores utilizando la estrategia de modelado promediado circuital, como se muestra en el siguiente ejemplo, donde se aplica esta técnica para obtener el modelo de uno de los DMSIs PWM tratados.

Ejemplo promediado circuital del DMSI *boost*

Para aplicar la estrategia de promediado circuital al ejemplo del DMSI PWM *boost*, se mostrara primero como se se aplica esta estrategia a una de las ramas del inversor (Figura 3.12 a)) ya que el proceso en la otra rama es igual [77]. Si bien esta rama corresponde en este ejemplo a uno de los convertidores *boost* bidireccionales CC–CC que conforman el DMSI *boost*, dado que la distribución de los transistores en las ramas de una gran familia de inversores es igual, el análisis será extensivo a otras topologías.

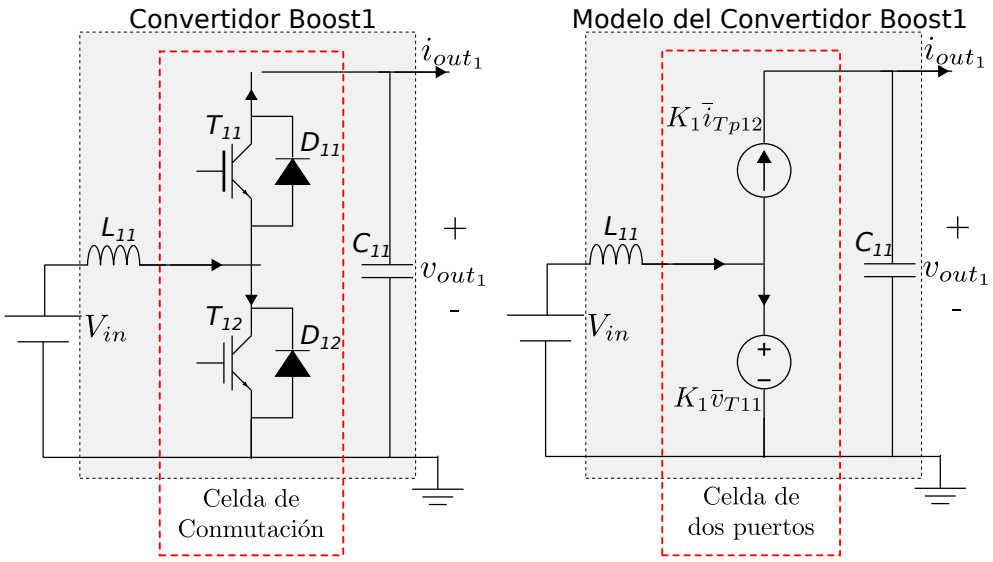


Figura 3.12: Convertidor boost que integra el DMSI PWM *boost* a) Esquema circuital b) Modelo promediado circuital.

En la Figura 3.12 a), los transistores T_{11} y T_{12} se disparan de manera alternada, por lo que conforman una celda de conmutación que puede ser reemplazada por una celda de dos puertos, tal como se muestra en la Figura 3.12 b).

En esta celda de dos puertos, dado que la señal PWM que comanda el encendido y apagado de los transistores posee un ciclo de trabajo d_{c_i} (donde i indica el convertidor al que pertenece la rama), la ganancia $K_i(d_{c_i})$ de las fuentes controladas está dada por:

$$K_i(d_{c_i}) = \frac{1 - d_c}{d_c} \tag{3.27}$$

En el caso de los DMSI PWM, se tienen dos convertidores diferenciales, por lo tanto, al aplicar esta estrategia se obtiene el modelo promediado circuital mostrado en la Figura 3.13.

Además, teniendo en cuenta que en esta familia de inversores las señales de alterna de referencia de los ciclos de trabajo deben estar desfasadas 180^o resulta entonces:

$$d_{c_2}(t) = 1 - d_{c_1}(t) \tag{3.28}$$

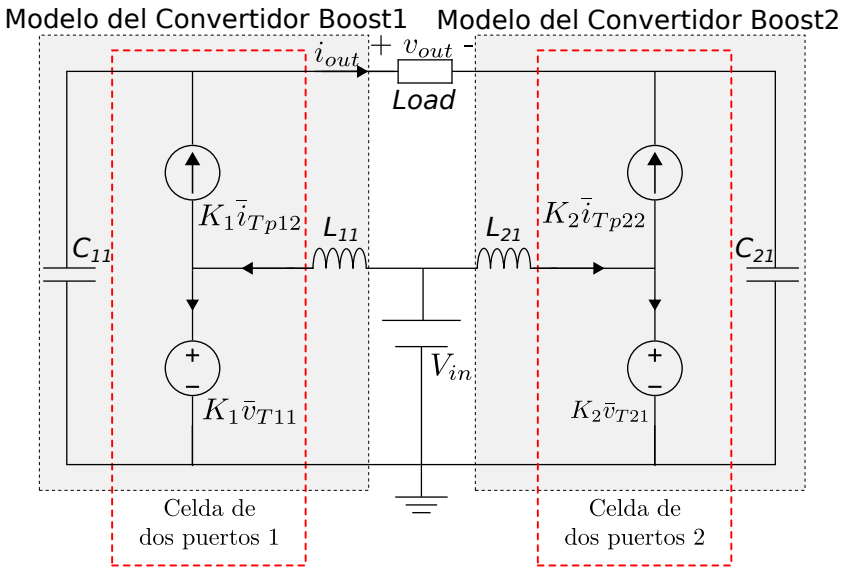


Figura 3.13: Modelo promediado circuital del DMSI PWM *boost*

y de este modo queda completamente definido el modelo promediado circuital del DMSI PWM.

3.3 Modelos mixtos de convertidores conmutados CC–CC PWM

Como ya fue mencionado, los *modelos conmutados* son precisos, pero tienen un costo computacional muy alto, en especial cuando se aplican a sistemas complejos o a simulaciones de larga duración. La estrategia alternativa de usar modelos promediados también tiene limitaciones, ya que los resultados de simulación no contienen toda la información, al enmascarar los fenómenos de alta frecuencia asociados a las conmutaciones. Además, pueden tener mucho error aun en la evolución de los transitorios de los valores medios de las variables, en presencia de perturbaciones transitorias pronunciadas o cuando la interacción de las oscilaciones producidas por las conmutaciones con saturaciones u otras partes del sistema modifica los resultados.

La elección entre estas dos estrategias de modelado depende entonces de los objetivos específicos de la simulación y el equilibrio entre detalle y recursos computacionales.

Recientemente se ha desarrollado una nueva estrategia de modelado mixta para convertidores PWM CC–CC que combina ambas estrategias de modelado, aprovechando las ventajas de cada una de ellas. Esto permite lograr mejoras significativas en cuanto al uso de recursos computacionales sin perder información de la dinámica rápida asociada a las conmutaciones.

Esta nueva estrategia de modelado desarrollada para convertidores CC–CC PWM consiste en combinar submodelos promediados y conmutados que se emplean de manera

alternada dentro de un único esquema denominado *modelo mixto* [58]. La selección entre ambos submodelos se realiza a través de un algoritmo que supervisa las variables fundamentales para seleccionar el submodelo adecuado según la condición de régimen en la que se encuentra. La idea básica de funcionamiento de este algoritmo es detectar si las están en régimen permanente o en una evolución transitoria y en función de es usar el submodelo promediado o conmutado. De este modo se tiene que por un lado se logra un resultado de simulación muy detallado cuando hay cambios en las variables porque se usa el submodelo conmutado. Por el otro, en las condiciones de régimen permanente se usa el submodelo promediado, lo que permite una disminución significativa del costo computacional y por lo tanto del tiempo de ejecución de CPU.

La idea central de ésta estrategia de modelado mixto se muestra el esquema general mostrado en la Figura 3.14. Al igual que en la estrategia de promediación circuital, en esta estrategia también se deben identificar los componentes continuos (capacitores, inductores y resistencias) del circuito y los elementos de conmutación. La principal diferencia radica que en lugar de reemplazar las celdas de conmutación por las celdas de dos puertos, en esta estrategia se deja ambos tipos de celdas y se las conecta a un conmutador. Este conmutador es el que conecta una u otro tipo de celdas con los componentes continuos, en función de la decisión tomada por el selector de modo. Este selector de modo, toma la decisión observando la evolución de las variables en los componentes continuos.

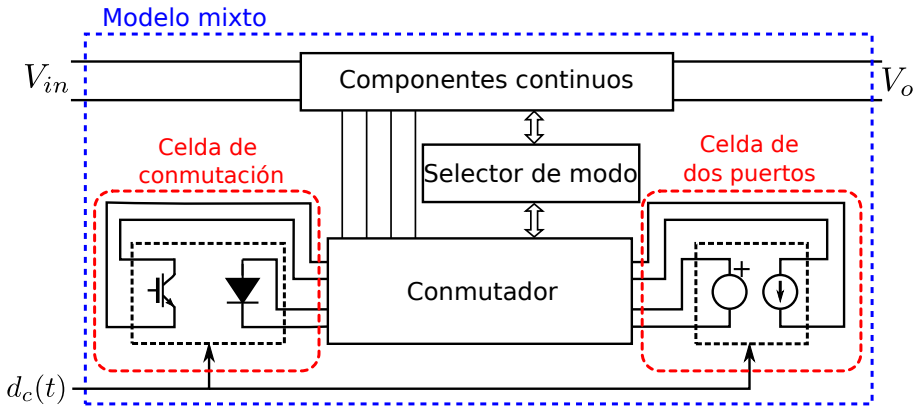


Figura 3.14: Esquema de modelo mixto de un convertidor CC–CC PWM.

El selector de modo distingue entre estados transitorios y de régimen permanente (o de variación lenta), seleccionando el modo operativo correspondiente y permitiendo que el conmutador elija el submodelo correspondiente. Para comprender esta lógica, es importante entender el comportamiento del algoritmo supervisor en el selector de modo. Con este fin, se analiza las evoluciones que se muestran en la Figura 3.15. En esta figura se muestra la evolución de una variable de estado $x_i(t)$ genérica y su correspondiente valor medio $\overline{x}_i(t)$. Además, sobre esta misma gráfica se indico la conmutación del Modo Conmutado

(*Switched Mode*) (SWM) al Modo Promediado (*Averaged Mode*) (AVM) en el instante t_k, para un convertidor CC–CC PWM.

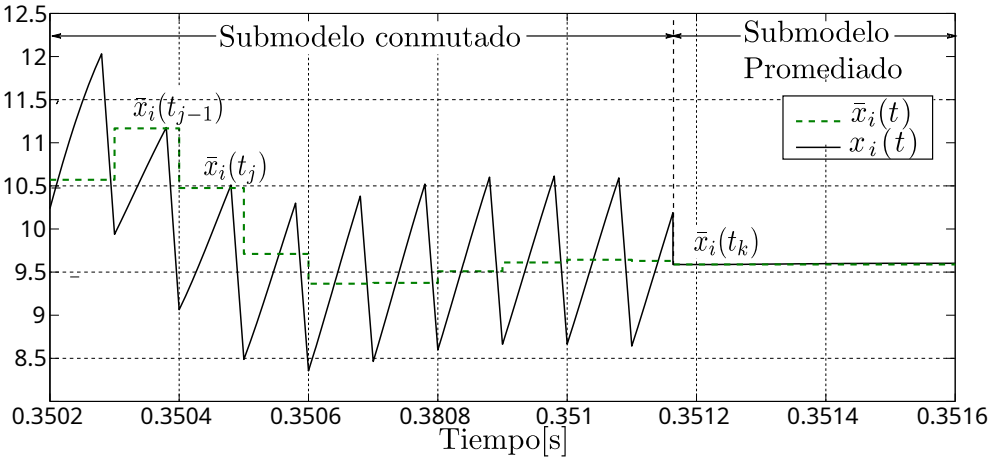


Figura 3.15: Evolución de una variable de estado en un modelo mixto de una SMPS PWM CC–CC.

En el caso de los convertidores conmutados CC–CC PWM, es fácil detectar el instante en el que se debe comenzar a utilizar el submodelo promediado, ya que solo requiere identificar una condición de variación lenta. Esta condición puede encontrarse comparando sucesivos valores del valor medio de las variables en un período de conmutación T_{sw} y verificando si se mantienen aproximadamente constantes. Es decir, verificando la condición:

$$|\bar{x}_i(t_j) - \bar{x}_i(t_{j-1})| \leq \alpha_i \tag{3.29}$$

donde α_i es un parámetro elegido por el usuario que permite seleccionar qué se considera dinámica lenta en esa variable. Si se cumple esta condición para todas las variable de estado, se detecta una variación lenta y debe seleccionarse el AVM. Como se puede observar en la Figura 3.15, en el instante t_k se selecciona el submodelo conmutado ya que se verifica esta condición.

De manera similar, como puede verse en la Figura 3.16, siempre que una variable de estado de convertidor cambie más allá de ciertos límites preestablecidos ($\psi_i r_i(k)$), el selector de modo detecta que debe utilizarse el submodelo conmutado.

Notar que $\bar{x}_i(t_k)$ es el valor medio de la i-ésima variables de estado calculada en el último instante de conmutación, cuando estaba activo el submodelo conmutado. En consecuencia, incluso una deriva arbitrariamente lenta en la evolución de una variable de estado eventualmente resultará en una diferencia mayor al umbral de detección. Esta condición de cambio del modelo promediado al conmutado se puede escribir entonces como:

$$|x_i(t) - \bar{x}_i(t_k)| \geq \sigma_i r_i(k) \tag{3.30}$$

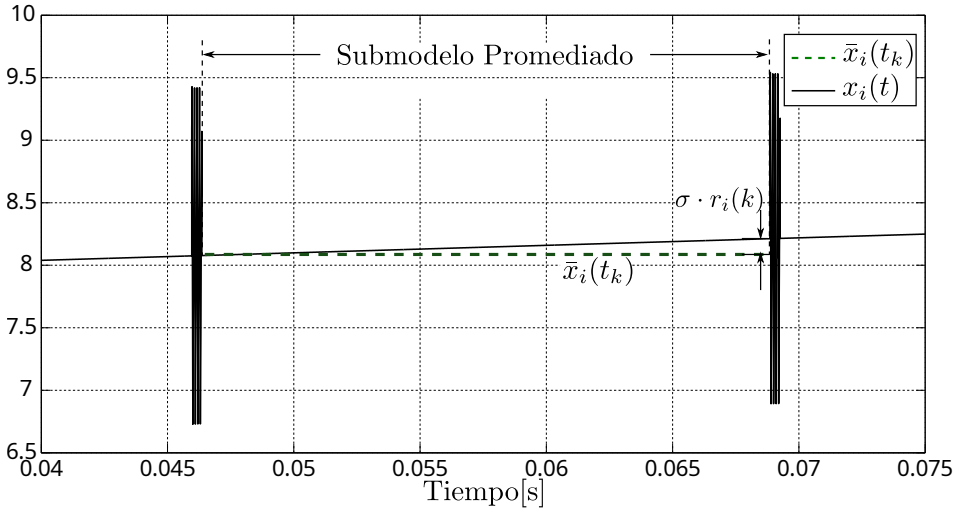


Figura 3.16: Evolución de una variable de estado en un modelo mixto de una SMPS PWM CC–CC: Paso de AVM a SWM.

donde σ_i es un parámetro equivalente a α_i, que define la variación máxima permitida en la variable de estado —relativa al ripple $r_i(k)$— sin volver al modo conmutado. El valor de ripple utilizado para ese cálculo es el obtenido en el último instante de conmutación (t_k) en que se estaba utilizando el submodelo conmutado.

4. MÉTODOS QSS PARA SIMULACIÓN DE CONVERTIDORES CONMUTADOS

4.1 Introducción

Los convertidores conmutados son sistemas híbridos ya que en los mismos coexisten comportamientos dinámicos continuos y discretos. Los modelos conmutados de estos sistemas se representan mediante modelos matemáticos definidos por EDOs discontinuas.

Estos modelos, si bien tienen la ventaja de modelar todas las dinámicas presentes en los convertidores, tienen el inconveniente de que su simulación mediante métodos clásicos de discretización temporal es ineficiente. Esto se debe a que estos métodos requieren un muy alto tiempo de ejecución de CPU respecto al tiempo de simulación ([82]).

Por este motivo, al simular modelos conmutados de convertidores e inversores, es necesario tener en cuenta las dificultades que presentan las conmutaciones de alta frecuencia para los métodos de integración numérica con el fin de lograr simulaciones eficientes.

4.1.1 Simulación de sistemas híbridos con discontinuidades frecuentes

Las discontinuidades inherentes a los sistemas híbridos deben ser manejadas adecuadamente por los algoritmos de integración numérica. En este contexto, se las denomina eventos, los cuales pueden clasificarse en dos categorías según su naturaleza. Por un lado, están los eventos de tiempo, que ocurren en un instante de tiempo específico predeterminado, independientemente del comportamiento de las variables continuas. Por otro lado, se encuentran los eventos de estado, que se disparan cuando se cumple una condición particular relacionada con los estados del sistema.

Es bien sabido que la integración numérica a través de discontinuidades puede producir resultados erróneos en la simulación global, ya que estas violan los supuestos teóricos sobre los que se basan los métodos numéricos clásicos. Para evitar este problema, es fundamental detectar y procesar los eventos correctamente. Cuando se identifica una discontinuidad, el tiempo de simulación debe avanzar hasta el instante exacto en que ocurre el evento y una vez procesado, la simulación debe reiniciarse bajo las nuevas condiciones resultantes [15].

La detección de eventos de tiempo es relativamente sencilla, dado que su momento de ocurrencia se conoce de antemano. En cambio, los eventos de estado requieren el uso de rutinas iterativas para determinar con precisión el instante en que se satisface la condición que los activa. Todo este proceso implica un costo computacional adicional. En sistemas con alta frecuencia de discontinuidades —donde los eventos ocurren tan rápidamente como la propia dinámica del sistema—, el problema se vuelve crítico. En tales casos, los algoritmos pueden dedicar más tiempo a la gestión de eventos que a la integración numérica propiamente dicha, afectando significativamente la eficiencia de la simulación.

Los modelos conmutados de convertidores e inversores son sistemas híbridos con alta frecuencia de discontinuidades. Si bien los mismos permiten representar en forma completa la dinámica de los convertidores, presentan el inconveniente de que su simulación mediante métodos clásicos de discretización temporal resulta ineficiente. Esto se debe a que dichos métodos requieren un tiempo de CPU muy alto respecto al tiempo de simulación real [82] ya que deben realizar múltiples cálculos por paso (detectar el instante exacto de las discontinuidades y reinicializar los métodos luego de cada evento).

Además, si el modelado de los convertidores conmutados considera características realistas de los elementos discontinuos, los modelos pueden presentar dinámicas simultáneamente lentas y rápidas (rígidos). Como consecuencia, los métodos numéricos clásicos deben recurrir a esquemas implícitos, que requieren iteraciones costosas e inversión de matrices en cada paso.

Uno de los métodos de integración numérica clásicos más eficientes para simular sistemas rígidos es *Differential Algebraic System Solver* (DASSL) [73]. Este se encuentra disponible en herramientas de simulación avanzadas, como Matlab/Simulink [54] y Dymola [24]. Aunque estas plataformas incorporan algoritmos muy eficientes para detectar y manejar discontinuidades, al basarse en discretización temporal mantienen un alto costo computacional cuando las discontinuidades son frecuentes. Como resultado, las simulaciones de sistemas de electrónica conmutada suelen ser muy lentas.

Una alternativa a los métodos clásicos de discretización temporal son los MIC basados en QSS, que discretizan los estados en lugar del tiempo. Estos métodos reemplazan la tradicional división del tiempo por una cuantificación de las variables de estado, lo que conduce a un modelo asincrónico de eventos discretos, en lugar de un sistema de ecuaciones en diferencias de tiempo discreto [93]. Esta característica les confiere una ventaja significativa para tratar sistemas con discontinuidades frecuentes.

Existen versiones de los métodos QSS de hasta tercer orden [16], así como métodos LIQSS diseñados específicamente simular modelos de sistemas rígidos [57].

En este capítulo se presenta una revisión de los MIC que se aplicarán en capítulos posteriores para la simulación de modelos de DMSI PWM. Además, se introducen las herramientas en las que estos métodos están implementados.

4.2 Integración basada en cuantificación de estados

Los MIC se fundamentan en una idea original de Bernard Zeigler, quien propuso que los sistemas continuos podían aproximarse mediante sistemas de eventos discretos [92, 91]. Zeigler demostró que estos modelos eran capaces de simular con exactitud un integrador equipado con un cuantificador en la salida.

Posteriormente, se estableció que la cuantificación debía incorporar histéresis para evitar oscilaciones infinitamente rápidas. A partir de esta idea, se formalizó el primer método de integración por cuantificación de primer orden, denominado QSS1[48].

Con el tiempo se desarrollaron variantes de orden superior, como QSS2 [45] y QSS3[47], que mejoran la precisión sin aumentar significativamente el costo computacional [15] y métodos adecuados para sistemas rígidos denominados LIQSS [57].

Una de las principales ventajas de los métodos QSS es su naturaleza asíncrona, lo que los hace especialmente eficaces para la simulación de sistemas con discontinuidades [46, 59] tanto espaciales como temporales. Esto hace que estos métodos sean potencialmente adecuados para simular convertidores e inversores de potencia.

Además de su utilidad práctica, los MIC presentan importantes propiedades teóricas sólidas, tales como estabilidad, convergencia [48] y la capacidad de ofrecer una cota global de error calculable [45].

Los algoritmos QSS están implementados en *Stand-Alone QSS Solver* [28], una herramienta de simulación desarrollada por el grupo de investigación.

A continuación, se presenta una descripción general de la familia de métodos QSS y LIQSS.

4.2.1 Método de Cuantificación de Estados de Primer Orden (QSS1)

Los métodos de integración numérica parten normalmente de considerar que el comportamiento de un sistema puede describirse mediante un conjunto de EDOs invariantes en el tiempo:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.1}$$

donde $\mathbf{x}(t)$ y $\mathbf{u}(t)$, representan el vector de estado y de entrada en el instante t respectivamente.

La representación dada por la Ecuación (4.1), es particularmente útil para la integración numérica con métodos clásicos basados en la discretización temporal que resuelven las ecuaciones resultantes en puntos discretos.

Esta representación del sistema también es utilizada como punto de partida por los métodos QSS, aunque estos adoptan un enfoque diferente.

Los métodos QSS aproximan el sistema definido por la Ecuación 4.1 mediante un Sistema de Estados Cuantificados. El mismo se obtiene reemplazando el vector de estado $x(t)$ por su versión cuantificada $q(t)$:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{q}(t), \mathbf{u}(t)) \tag{4.2}$$

donde $\mathbf{q}(t)$ está relacionado con $\mathbf{x}(t)$ mediante una función de cuantificación con histéresis. Las componentes $q_j(t)$ de $\mathbf{q}(t)$ se denominan variables cuantificadas.

En el caso del método QSS1, $q_j(t)$ es una aproximación de primer orden de $x_j(t)$ que serealiza mediante la función de cuantificación de orden cero con histéresis:

$$q_j(t) = \begin{cases} x(t) \text{ si } |x_j(t) - q_j(t^-)| = \Delta Q_j \\ q_j(t^-) \text{ en otro caso} \end{cases} \tag{4.3}$$

donde ΔQ_j se denomina *Quantum*.

Como puede verse en la Figura 4.1, en el caso de QSS1 puede observarse que $q_j(t)$ sigue una trayectoria constante a tramos, cambiando solo cuando la diferencia entre $q_j(t)$ y $x_j(t)$ alcanza el Quantum. Después de cada cambio, se cumple que $q_j(t) = x_j(t)$.

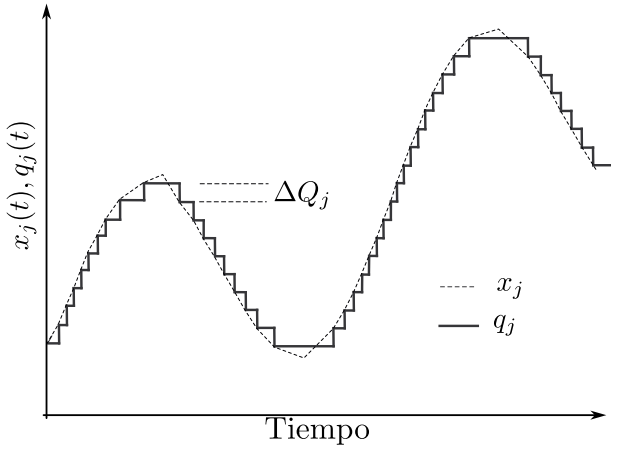


Figura 4.1: Trayectoria de una variable de estado $x_j(t)$ y su versión cuantificada $q_j(t)$ en QSS1.

Como el método QSS1 realiza aproximaciones de primer orden, a menudo requiere un gran número de pasos para alcanzar un nivel de precisión aceptable. Esta limitación fue superada con la introducción de los métodos de orden superior QSS2 y QSS3 que realizan aproximaciones de segundo y tercer orden respectivamente. Estos métodos permiten realizar simulaciones más exactas sin necesidad de aumentar excesivamente el número de pasos, logrando asi un equilibrio entre precisión y eficiencia computacional.

Si bien los métodos QSS son eficaces para la simulación de diversos tipos de sistemas, no son eficientes para simular sistemas rígidos. En los sistemas donde coexisten dinámicas

rápidas y lentas, los métodos QSS generan oscilaciones espurias de alta frecuencia, lo que incrementa la cantidad de pasos de integración y, en consecuencia, los costos computacionales. Este fenómeno puede observarse en la Figura 4.2 donde se muestra el resultado de simulación del sistema rígido:

$$\begin{aligned} \dot{x}_1(t) &= 0{,}01\, x_2(t) \\ \dot{x}_2(t) &= -100\, x_1(t) - 100\, x_2(t) + 2020 \end{aligned} \tag{4.4}$$

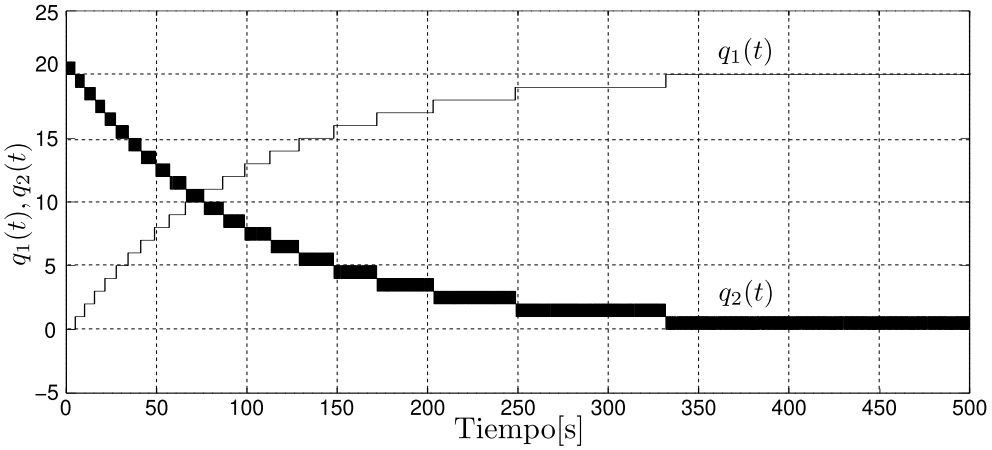


Figura 4.2: Resultado de simulación de un sistema rígido (Ecuación 4.4) con QSS.

4.2.2 QSS de Segundo y Tercer Orden (QSS2 y QSS3)

Los métodos QSS2 y QSS3 son versiones del método QSS1 que realizan aproximaciones de segundo y tercer orden diseñadas para aumentar la precisión en la simulación. La principal diferencia es que consideran información de derivadas de orden superior.

El método QSS2 se basa en los mismos principios que QSS, pero reemplaza la función de cuantificación de orden cero por una función de cuantificación de primer orden. El comportamiento entrada-salida de una función de cuantificación de primer orden se muestran en la Figura 4.3 a). En el caso de la trayectoria de salida de QSS2, la función cuantificación ya no es constante por tramos, sino que es seccionalmente lineal. Estos segmentos mantienen la misma pendiente que la entrada ($\dot{x}_j$) hasta que la diferencia entre la salida y la entrada supera un umbral determinado ΔQ_j. En ese momento, se genera un nuevo segmento con una pendiente ajustada al valor actual de la entrada$\dot{x}_j$(t). El uso de una función de cuantificación de primer orden permite reducir los errores de simulación sin necesidad de disminuir significativamente el tamaño del Quantum.

En el caso del método QSS3 no solo se considera la primera derivada de las variables

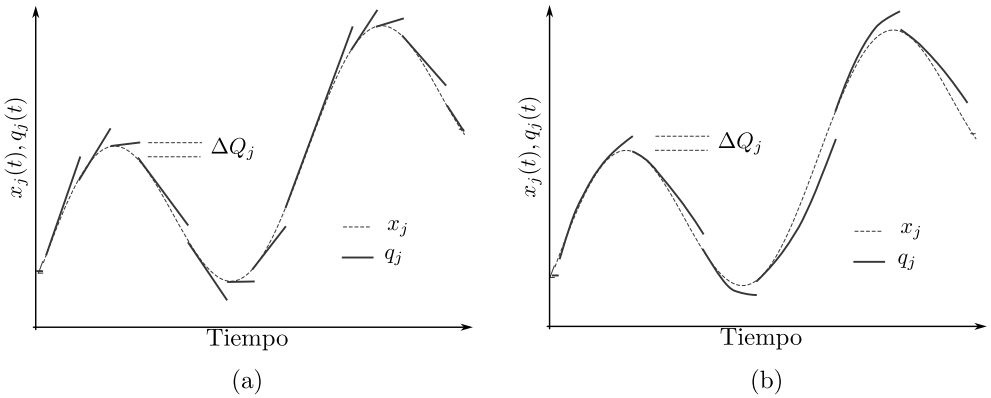


Figura 4.3: Trayectoria de una variable de estado $x_j(t)$ y su versión cuantificada $q_j(t)$ en a) QSS2 y b) QSS3.

de estado, sino también su segunda derivada. Esto se logra utilizando una función de cuantificación de segundo orden. En este caso entonces, la trayectoria de salida es parabólica por tramos, como se observa en la Figura 4.3 b). Esta mejora se traduce en una mayor precisión en los resultados de simulación y una reducción del error de aproximación en comparación con QSS2.

Estos métodos resultan eficientes para la simulación de EDOs discontinuas no rígidas. Sin embargo, al igual que QSS1, no pueden simular de manera eficiente sistemas rígidos ya que también generan oscilaciones de alta frecuencia.

4.2.3 Métodos LIQSS

La familia de métodos LIQSS es una extensión de los métodos QSS que permite simular de manera eficiente un gran variedad de sistemas rígidos[57]. Esta familia de métodos combina los principios de cuantificación de estados con ideas tomadas de los métodos implícitos clásicos.

En los métodos implícitos clásicos el valor de las variables de estado se calcula evaluando las derivadas del estado en instantes futuros. Por ejemplo en el método *Backward-Euler*:

$$x(t_{k+1}) = x(t_k) + h.f(x(t_{k+1}), u(t_k)) \tag{4.5}$$

La evaluación de las derivadas del estado en instantes futuros requiere en general resolver ecuaciones no lineales mediante iteraciones o inversión de matrices.

Los métodos LIQSS comparten con los QSS la definición de la Ecuación 4.2, pero las funciones de cuantificación que relacionan q_j con x_j son más complejas ya que no solo dependen de la magnitud de la variable de estado sino también de sus derivadas de orden superior. De este modo, los métodos LIQSS aprovechan una característica clave de los métodos QSS, que es el conocimiento explícito del valor futuro de las variables cuantificadas

y siempre buscan que $q_j(t)$ sea un valor futuro de $x_j(t)$. Esta propiedad permite calcular las derivadas utilizando una proyección futura del estado sin necesidad de resolver sistemas implícitos. De este manera la implementación de los algoritmos LIQSS es completamente explícita y computacionalmente eficiente.

Una de las ventajas de los métodos LIQSS es que además de ser adecuado para sistemas rígidos, posee todas las ventajas de los métodos QSS en lo que respecta a la simulación de sistemas híbridos. Una de las mayores desventajas de los métodos LIQSS, en los que solo se puede asegurar su eficiencia en la simulación de sistemas rígidos cuando la rigidez se debe a la presencia de valores muy grandes en la diagonal principal de la matriz Jacobiana.

El método LIQSS1 emplea una función de cuantificación de orden cero del estado $x_j(t)$. Esto implica que las trayectorias $q_j(t)$ son seccionalmente constantes, al igual que en QSS1.

A modo de ejemplo, en la Figura 4.4 a) se muestra la trayectorias de $x_j(t)$ y $q_j(t)$ en LIQSS1. Como puede apreciarse, en LIQSS1 $q_j(t)$ depende de $x_j(t)$ y de la primera derivada de $x_j(t)$ ya que $q_j(t)$ se elige de manera que siempre se cumpla que $\dot{x}_j(t)(q_j(t) - x_j(t)) \geq 0$.

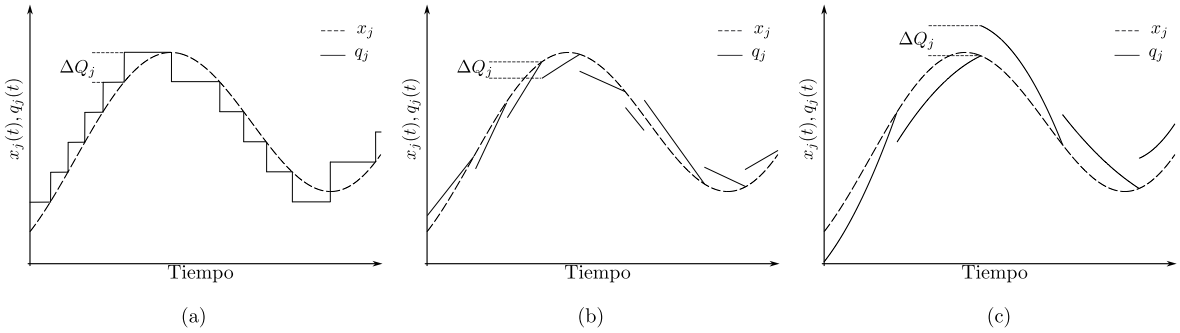


Figura 4.4: Trayectorias de $q(t)$ y $x(t)$ en los métodos a) LIQSS1 b) LIQSS2 c) LIQSS3

En la Figura 4.5 se muestran los resultados de simulación con LIQSS1 del ejemplo del sistema rígido simulado con QSS (Ecuación 4.4). En el mismo se puede ver la evolución de $q_1(t)$ y $q_2(t)$ y las variables de estado correspondientes, $x_1(t)$ y $x_2(t)$. Puede observarse que las oscilaciones espurias que se generaban en la variable $q_2(t)$ al simular con QSS1 no se generan al simular con LIQSS1. Esto redunda en una reducción significativa del requerimiento de CPU.

Existen algoritmos de segundo y tercer orden que se basan en la misma idea que LIQSS1, es decir, que las trayectoria de $q_j(t)$ sean valores futuro de $x_j(t)$.

El método LIQSS2 emplea una función de cuantificación de primer orden del estado $x_j(t)$, lo que significa que las trayectorias $q_j(t)$ son seccionalmente lineales como en QSS2. En este algoritmo en lugar de tenerse en cuenta la primera derivada de $x_j(t)$, como en LIQSS1, para verificar que $q(t)$ sea siempre un valor futuro de $x_j(t)$ se observa la segunda

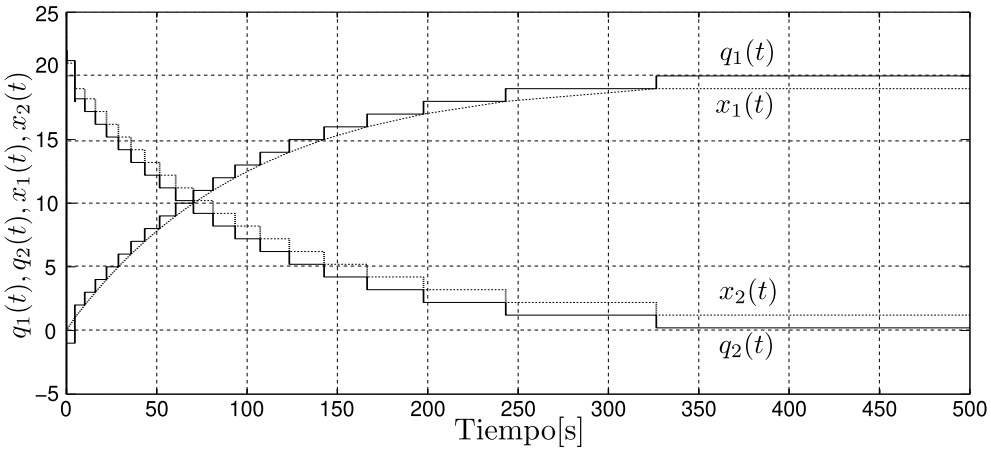


Figura 4.5: Resultado de simulación de un sistema rígido (Ecuación 4.4) con LIQSS1.

derivada de manera que siempre se cumpla que $\ddot{x}_j(t)(q_j(t) - x_j(t)) \geq 0$. De este modo, las trayectorias de de $x_j(t)$ y $q_j(t)$ en LIQSS2 resultan como las mostradas en las Figuras 4.4 b).

En el caso del método LIQSS3, la función de cuantificación es de segundo orden por los que las trayectorias $q_j(t)$ resultan seccionalmente parabólicas. En este caso, se selecciona se verifica que $q_j(t)$ es un valor futuro de $x_j(t)$ observando la tercera derivada de $x_j(t)$. De este modo, la trayectoria $q_j(t)$ se elige de manera tal que siempre se verifique $\dddot{x}_j(t)(q_j(t) - x_j(t)) \geq 0$. Las trayectorias de de $x_j(t)$ y $q_j(t)$ en LIQSS3 resultan entonces como las mostradas en las Figuras 4.4 c).

Si se realizan simulaciones con el mismo requerimiento de error usando LIQSS1, LIQSS2 y LIQSS3 el numero de pasos como es de esperar se reduce. Sin embargo no siempre ocurre lo mismo con respecto a requerimiento de tiempo de ejecución de CPU. Esto se debe a que el costo computacional del algoritmo LIQSS3 es significativamente mayor al de sus versiones de menor orden. Por consiguiente LIQSS2 suele ser el algoritmo mas adecuado para simular sistemas híbridos rígidos.

Limitaciones del método LIQSS

Para poder simular eficientemente una gran variedad de estructuras de modelos rígidos, los LIQSS aprovechan que la dinámica de cada variable x_j se puede escribir como

$$\dot{\hat{x}}_j = A_{jj}\,\hat{x}_j + v_j, \tag{4.6}$$

donde A_{jj} es el elemento diagonal correspondiente de la jacobiana $\partial f/\partial x$.

Esta aproximación permite elegir el nuevo valor cuantificado q_j (y sus derivadas en órdenes superiores) de forma tal que x_j avance hacia q_j. En particular, si $\dot{x}_j$ depende explíci-

tamente de q_j, entonces $A_{jj} \neq 0$ y el algoritmo LQSS puede seleccionar $q_j h$ de manera de evitar oscilaciones numéricas de alta frecuencia. Dado que A_{jj} y v_j es conocido q_j se puede seleccionar sin necesidad de resolver sistemas no lineales acoplados ni de realizar iteraciones tipo Newton. Esto evita el costo típico de un método implícito general. La principal limitación del método radica en que en principio solo resuelve el problema de rigidez en los casos en que la misma se deba a la presencia de elementos grandes en la diagonal principal (Términos con $A_{jj} \neq 0$)

Esta construcción funciona muy bien cuando la rigidez está asociada a términos locales fuertemente disipativos (por ejemplo, resistencias de conducción y de bloqueo extremadamente grandes o pequeñas en convertidores electrónicos de potencia), que se reflejan en grandes valores de A_{jj}. En esos casos LIQSS reduce drásticamente el número de eventos necesarios y mantiene estabilidad numérica aun frente a conmutaciones y discontinuidades.

4.3 Herramientas de simulación que implementan los métodos QSS

Los MIC están implementados en dos herramientas de simulación desarrolladas específicamente para la utilización de los algoritmos *Quantization-Based Integration* (QBI): PowerDEVS [67] y Simulador Autónomo de QSS (*Stand-Alone QSS Solver*) (QSS Solver) [28]. Sin embargo, la implementación de los MIC en PowerDEVS no es muy eficiente. Esto se debe a que esta herramienta es un simulador de DEVS y las implementaciones DEVS de los algoritmos QSS no son tan eficientes ya que los cálculos en cada paso de simulación son realizados por distintos modelos atómicos DEVS que deben intercambiar eventos. El problema radica en que la coordinación y el manejo de los mensajes pueden resultar computacionalmente más costosos que los propios cálculos asociados a la simulación [14].

Por lo tanto, en esta tesis se utiliza la herramienta llamada QSS Solver para la simulación de inversores PWM. Esta herramienta no está basada en DEVS, sino que implementa toda la familia de métodos QSS. Con ello logra una mejora de más de un orden de magnitud en los tiempos de simulación respecto a las versiones basadas en DEVS.

4.3.1 Stand–Alone QSS solver

La herramienta de simulación con métodos QSS más eficiente y completa es el QSS Solver. En esta herramienta, los modelos se describen en lenguaje $\mu-$Modelica (subconjunto del lenguaje Modelica)([7]). Esta descripción es posteriormente traducida de manera automática por el software a un fragmento de código en lenguaje C, que contiene el conjunto de EDOs, junto con las funciones correspondientes de cruce por cero y controladores de eventos para casos discontinuos. La herramienta también extrae la información de la estructura (matrices de incidencia) y genera el código para la evaluación simbólica de la matriz Jacobiana. El código en C producido luego se vincula a los diferentes algoritmos

QSS (QSS y $LIQSS$ de orden 1 a 3) o a métodos clásicos como DASSL, *C-variable Ordinary Differential Equation solver with Backward Differentiation Formula* (CVODE–BDF) , etc.

La Figura 4.6 muestra la interfaz gráfica del QSS Solver durante la simulación de un modelo, incluyendo las ventanas de configuración de la simulación y de visualización de las variables seleccionadas.

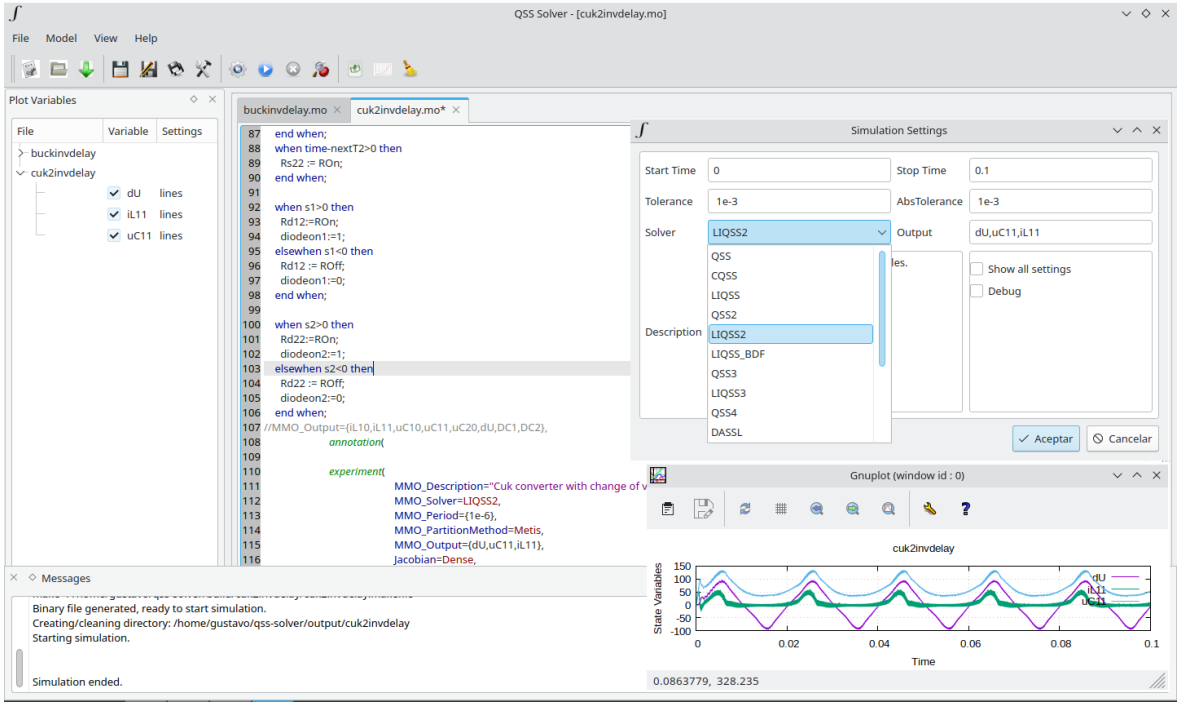


Figura 4.6: Interfaz gráfica del QSS Solver.

Esta herramienta dispone de toda la información estructural y puede realizar evaluaciones simbólicas de las matrices Jacobianas. Por este motivo, aún utilizando los métodos clásicos implementados en ella, se obtiene resultados de simulación significativamente más rápido que con otras herramientas. Además, el hecho de que se utilice el mismo fragmento de código para las EDOs en los diferentes algoritmos (QSS y métodos clásicos) permite realizar comparaciones de rendimiento justas entre ellos.

En el siguiente capítulo se utiliza esta herramienta para simular los modelos de DMSI PWM desarrollados en esta tesis.

5. SIMULACIÓN EFICIENTE DE DMSIS PWM CON MÉTODOS QSS

5.1 Introducción

Como se explicó anteriormente en esta tesis, las simulaciones eficientes de sistemas de electrónica de potencia son fundamentales para el diseño y la mejora de diversas aplicaciones en redes eléctricas ([61, 25]).

Una de las principales dificultades en la simulación de estos sistemas surge de la presencia de convertidores conmutados e inversores, ya que estos contienen elementos de conmutación que cambian su estado de conducción a alta frecuencia. Existen, básicamente, dos enfoques para intentar superar las dificultades asociadas a la simulación de este tipo de sistemas. Por un lado, se puede optar por utilizar y optimizar los algoritmos numéricos empleados en la simulación de los modelos resultantes; por otro, se pueden aplicar estrategias de modelado que se adecuén mejor al tipo de análisis que se desea realizar.

En muchas situaciones, es necesario realizar simulaciones en las que los fenómenos asociados a las conmutaciones tienen un papel relevante, como en el estudio del ruido, la interferencia electromagnética o el diseño detallado de convertidores e inversores. En estos casos no es posible utilizar modelos simplificados, ya que enmascaran dichos fenómenos; por lo tanto, deben emplearse modelos que presenten discontinuidades en sus EDOs.

La simulación de estos modelos mediante métodos clásicos, basados en la discretización temporal, resulta ineficiente, dado que el tiempo de ejecución (medido en tiempo de CPU) es considerablemente alto en comparación con el intervalo de tiempo sobre el cual se desea analizar el comportamiento del sistema (tiempo virtual de simulación).

Este capítulo analiza la estrategia más adecuada para la simulación de modelos híbridos de los DMSIs PWM desde el punto de vista de los algoritmos numéricos. En particular, dado que los métodos clásicos basados en la discretización temporal presentan importantes desventajas para este tipo de sistemas, se estudia la aplicación de los MIC como alternativa, con el objetivo de aprovechar las ventajas que ofrecen frente a la presencia de discontinuidades frecuentes.

Dado que los modelos híbridos de los DMSIs PWM representan los elementos de con-

mutación como impedancias que alternan entre valores muy altos y muy bajos, es esperable que, además de ser discontinuos, presenten rigidez estructural. Por este motivo, en este capítulo se inicia con un análisis de la rigidez de los modelos correspondientes a las distintas topologías básicas de esta familia de inversores, con el fin de seleccionar el MIC más adecuado para su simulación.

Posteriormente, se realiza una evaluación del desempeño en la simulación de los DMSIs PWM utilizando los MIC más adecuados, en particular el método LIQSS, mediante un análisis comparativo con los resultados obtenidos a partir de métodos clásicos avanzados, como DASSL y CVODE–BDF, comúnmente implementados en herramientas de simulación comercial.

5.2 Modelos de DMSI PWM

En esta sección se desarrollan los modelos correspondientes a tres topologías de DMSIs PWM, utilizando una estrategia de modelado realista que contempla explícitamente los elementos de conmutación, con el objetivo de preservar el nivel de detalle requerido en los resultados de simulación.

Primero se presentan los modelos de los elementos de conmutación; luego se derivan las ecuaciones del circuito y, finalmente, se traduce el modelo a μ–Modelica.

5.2.1 Modelado de la estrategia de cambio de estado de los elementos de conmutación

El modelado de elementos discontinuos en inversores PWM requiere representar con precisión el comportamiento no lineal y su evolución temporal, generalmente mediante EDOs con discontinuidades.

Como se introdujo en el Capítulo 3, en el caso de diodos y transistores que operan en estado de conducción o bloqueo, es necesario tener en cuenta las características discontinuas de tensión-corriente (v–i).

En este caso utilizamos el enfoque que representa los componentes de conmutación (Sección 3.2.1) de manera más *realista*, que consiste en representar el componente de conmutación mediante una resistencia que cambia de valor. Esta resistencia tiene un valor bajo (R_{on}) cuando el elemento conduce, y alto cuando está bloqueado (R_{off}). Los modelos así obtenidos no presentan cambios estructurales sino paramétricos, por lo que pueden ser simulados mediante herramientas estándar.

Para comprender la lógica de funcionamiento de los modelos de los DMSIs PWM, analizamos el comportamiento de conmutación de las ramas que los componen, independientemente de la topología particular de cada inversor. Con este objetivo, se recurre a la representación esquemática mostrada en la Figura 5.1 a), donde se observan las dos ramas que conforman estos inversores monofásicos, modeladas mediante transistores con diodo

antiparalelo. Cada transistor se representa como una resistencia $R_{T_{Pij}}$, donde el índice i indica la rama a la que pertenece el transistor.

Asimismo, en la misma figura se puede observar que cada modelo de transistor con diodo antiparalelo se descompone en dos resistencias en paralelo: una correspondiente al transistor ($R_{T_{ij}}$) y otra al diodo ($R_{D_{ij}}$), que conforman el conjunto del interruptor activo.

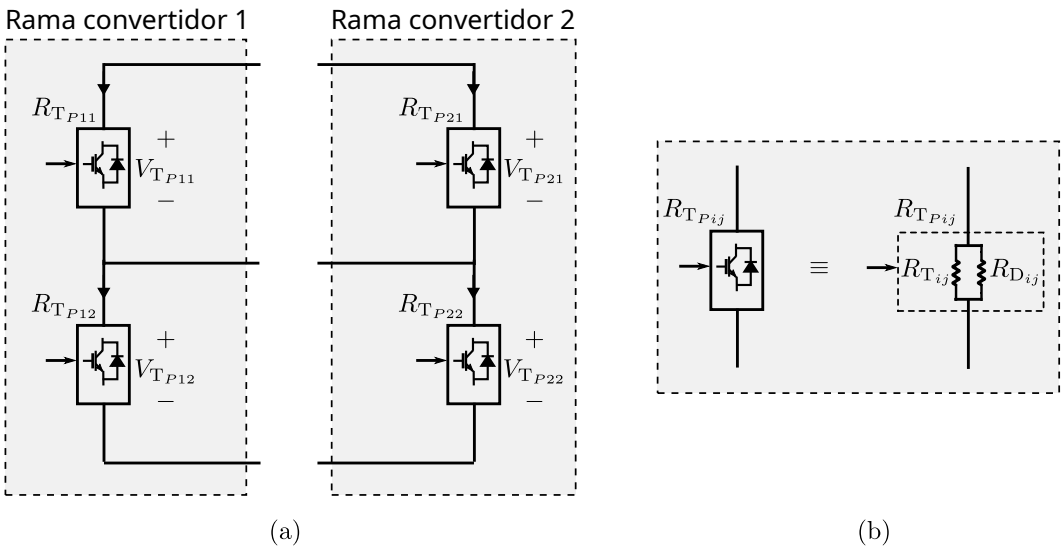


Figura 5.1: Modelos realistas de transistores a) en ramas de convertidores, b) transistor con diodo antiparalelo.

Para comprender la lógica de disparo de cada uno de los transistores representados por $R_{T_{ij}}$, en la Figura 5.2 se muestra un esquema temporal que representa las señales PWM utilizadas para controlar su activación.

La interpretación que se desprende de la Figura 5.2 es la siguiente: cuando la señal PWM correspondiente a la rama i está en nivel alto, conduce el transistor T_{i1} mientras que T_{i2} permanece en corte. Por el contrario, cuando la señal PWM está en nivel bajo, T_{i1} se encuentra en corte y conduce T_{i2}.

Además, el control de los transistores debe asegurar que ambos dispositivos de una misma rama no conduzcan simultáneamente, a fin de evitar cortocircuitos. Esto se consigue mediante la incorporación de un tiempo muerto (T_d) entre el apagado de un transistor y el encendido del otro. La inclusión de este detalle resulta fundamental para realizar estudios de validación en hardware, que requieren analizar los armónicos de corriente generados por dichos tiempos muertos, así como la disipación de potencia en los transistores [5, 86].

De esta manera, durante la conducción (modo de saturación), los transistores se modelan con una resistencia baja R_{on}, y durante el corte, con una resistencia alta R_{off}.

Esta lógica puede describirse en μ–Modelica dentro de la sección *algorithm*, imple-

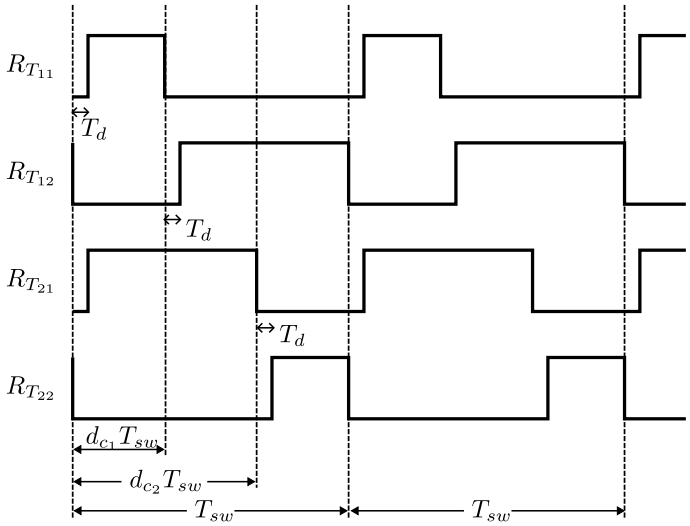


Figura 5.2: Esquema de disparo de los transistores

mentando tanto las señales PWM de cada convertidor como el control de los estados de conducción de cada transistor($R_{T_{ij}}$), mediante la asignación de los valores de impedancia correspondientes(R_{on} y R_{off}). En este modelo, ambos convertidores utilizan referencias de ciclo de trabajo senoidales ($DC1$ y $DC2$) desfasadas 180 grados y con una frecuencia de $50,$ Hz.

Además, se incorpora el tiempo muerto mencionado anteriormente (*Delay*) previo a la transición del estado de corte al de saturación de los transistores, con el fin de evitar posibles cortocircuitos. Esta medida resulta necesaria, ya que en la práctica el apagado de los transistores no ocurre de forma instantánea.

```
1  algorithm
2      when time > nextT then    //comienzo periodo de PWM
3        lastT:=nextT;
4        nextT:=nextT+T;
5        RT12 := Roff;
6        RT22 := Roff;
7        DC1:=(sin(2*50*3.14*time)+1)*0.3+0.2;
8        DC2:=1-DC1;
9        nextT0:=time+Delay; //Retardo para el encendido de los transistores
10     end when;
11     when time -nextT0>0 then //Encendido de los transistores retardado
12       RT11 := Ron;
13       RT21 := Ron;
14     end when;
15
16     when time-lastT-DC1*T>0 then //Cambio de estado del PWM1 (Buck 1)
17       RT11 := Roff;
18       nextT1:= time+Delay;  //Retardo para el encendido de los transistores
19     end when;
```

```
20      when time-nextT1>0 then //Encendido de los transistores retardado
21        RT12 := Ron;
22      end when;
23
24      when time - lastT-DC2*T>0 then //Cambio de estado del PWM2 (Buck 2)
25        RT21 := Roff;
26        nextT2:= time+Delay; //Retardo para el encendido de los transistores
27      end when;
28      when time-nextT2>0 then //Encendido de los transistores retardado
29        RT22 := Ron;
30      end when;
```

En el caso de un diodo, se debe detectar en qué estado de polarización se encuentra para seleccionar el valor adecuado de la impedancia que lo modela. Dado que el diodo comienza a conducir cuando $V_D > V_\gamma$ y deja de conducir cuando $i_D \leq 0$, la detección del estado de conducción se representa según:

$$diodeon = \begin{cases} 1 & \text{diodo conduce} \\ 0 & \text{diodo cortado} \end{cases} \tag{5.1}$$

A partir del estado de conducción $diodeon$, podemos definir la variable auxiliar $s = diodeon \cdot i_D + (1 - diodeon) \cdot V_D$, que representa tanto la tensión como la corriente, según el estado de conducción del diodo.

El comportamiento del diodo, según esta idea, puede describirse en μ–Modélica de la siguiente manera:

```
1 s = diodeon * iD + (1 - diodeon) * iD * Rd;
2 algorithm
3   when s > 0 then Rd := Ron; diodeon := 1;
4   elsewhen s < 0 then Rd := Roff; diodeon := 0;
5   end when;
```

En particular, la activación de cada uno de los diodos que integran las dos ramas de los inversores se representa en el código de cada modelo de la siguiente manera:

```
1  algorithm
2      when s1>0 then //Control del estado de conduccion de los diodos
3        Rd11:=Ron;
4        diodeon1:=1;
5      elsewhen s1<0 then
6        Rd11 := Roff;
7        diodeon1:=0;
8      end when;
9      when s2>0 then
10       Rd12:=Ron;
11       diodeon2:=1;
12     elsewhen s2<0 then
13       Rd12 := Roff;
14       diodeon2:=0;
15     end when;
```

Este enfoque optimiza la simulación del diodo al alternar eficientemente entre sus dos estados.

A partir de estos modelos elementales de los componentes discontinuos, se pueden construir fácilmente modelos *realistas* de DMSI PWM.

5.2.2 Modelo del DMSI PWM buck

La Figura 5.3 muestra el esquema circuital básico del DMSI PWM *buck*. En esta configuración la tensión de salida es una señal de alterna $v_{\text{out}}(t)$ cuya amplitud está acotada según $-V_{in} \leq v_{\text{out}}(t) \leq V_{in}$.

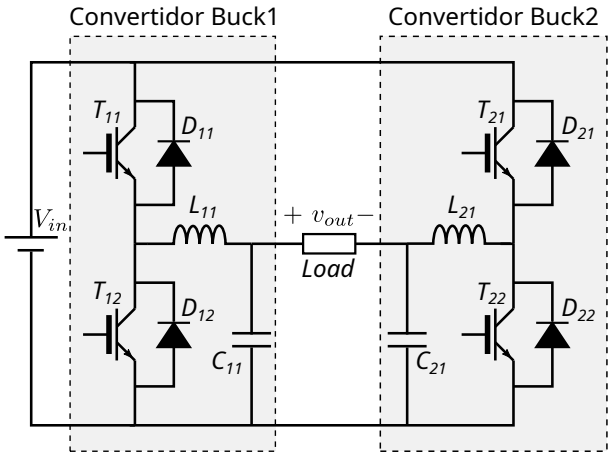


Figura 5.3: Esquema circuital del DMSI PWM buck.

A partir de este esquema, y teniendo en cuenta que el ij-ésimo transistor T_{ij} y diodo D_{ij} se modelan mediante las resistencias $R_{T_{ij}}$ y $R_{D_{ij}}$, respectivamente, se obtiene el siguiente modelo de orden 4 en EDOs para el DMSI PWM buck:

$$\begin{aligned}
\frac{di_{L_{11}}}{dt} &= \frac{-i_{D12}R_{D_{12}} - v_{C_{11}}}{L_{11}} \\
\frac{di_{L_{21}}}{dt} &= \frac{-i_{D22}R_{D_{22}} - v_{C_{21}}}{L_{21}} \\
\frac{dv_{C_{11}}}{dt} &= \frac{i_{L_{11}} - (v_{C_{11}} - v_{C_{21}})/R_{\text{load}}}{C_{11}} \\
\frac{dv_{C_{21}}}{dt} &= \frac{i_{L_{21}} - (v_{C_{21}} - v_{C_{11}})/R_{\text{load}}}{C_{21}}
\end{aligned} \tag{5.2}$$

En estas ecuaciones, el subíndice $i = 1, 2$ indica el convertidor (columna del inversor) correspondiente. Las variables $i_{L_{i1}}$ y $v_{C_{i1}}$ representan las corrientes en los inductores y las tensiones en los capacitores e $i_{D_{i2}}$ las corrientes en los diodos D_{i2}. Esta última se calcula según:

$$
\begin{aligned}
i_{D_{12}} &= \frac{(i_{L_{11}} R_{T_{11}} - V_{in}) R_{T_{12}}}{R_{T_{11}} R_{D_{12}} + R_{T_{11}} R_{T_{12}} + R_{D_{12}} R_{T_{12}}} \\
i_{D_{22}} &= \frac{(i_{L_{21}} R_{T_{21}} - V_{in}) R_{T_{22}}}{R_{T_{21}} R_{D_{22}} + R_{T_{21}} R_{T_{22}} + R_{D_{22}} R_{T_{22}}}
\end{aligned}
\tag{5.3}
$$

La tensión de salida de este inversor se define como $v_{\text{out}} = v_{C_{11}} - v_{C_{21}}$.

A continuación se presenta el código μ–Modelica de la implementación del modelo completo de la DMSI PWM buck.

El modelo está compuesto por dos secciones que pueden ser identificadas por las ecuaciones que definen la dinámica de las variables de estado y, por otro, el algoritmo de control de los transistores y diodos.

La sección *equation* del código implementa las ecuaciones (5.2)–(5.3):

```
equation
  //buck1
    iD12=(iL11*RT11-U)*RT12/(RT11*RT12+RT11*Rd12+RT12*Rd12);
    s1=diodeon1*iD12+(1-diodeon1)*iD12*Rd12;
    der(iL11) = (-iD12*Rd12- vC11)/L11;
    der(vC11) = (iL11 - (vC11-vC21)/Rload)/C11;
  //buck2
    iD22=(iL21*RT21-U)*RT22/(RT21*RT22+RT21*Rd22+RT22*Rd22);
    s2=diodeon2*iD22+(1-diodeon2)*iD22*Rd22;
    der(iL21) = (-iD22*Rd22- vC21)/L21;
    der(vC21) = (iL21 - (vC21-vC11)/Rload)/C21;
    vout=vC11-vC21; //Salida de tension diferencial
```

5.2.3 Modelos de los DMSIs PWM buck-boost y Ćuk

Para la construcción de los modelos de estas topologías del tipo reductores–elevadores, partimos de los esquemas circuitales básicos de estos inversores se muestran en la Figura 5.4.

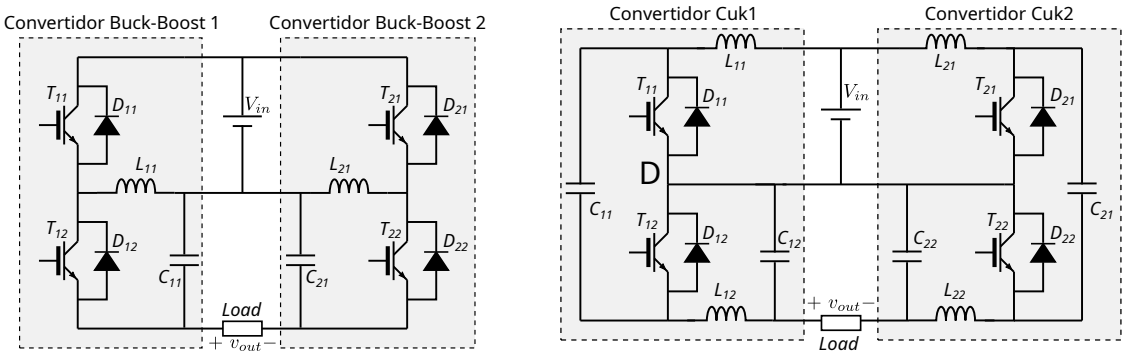


Figura 5.4: DMSIs PWM reductores–elevadores: esquemas circuitales de buck–boost (izq.) y Ćuk (der.).

Al igual que en el caso del DMSI buck, se obtuvieron las EDOs y las ecuaciones algebraicas necesarias para construir los modelos en μ–Modelica de los inversores buck–boost y Ćuk.

En particular, para el caso del DMSI PWM buck–boost, las EDOs se expresan en las Ecuaciones (5.4).

$$\begin{cases} \frac{di_{L_{i1}}}{dt} = \frac{-v_{C_{i1}} - i_{D_{i2}} R_{D_{i2}}}{L_{i1}} \\ \frac{di_{L_{i1}}}{dt} = \frac{-v_{C_{i1}} - i_{D_{i2}} R_{D_{i2}}}{L_{i1}} \\ \frac{dv_{C_{i1}}}{dt} = \frac{i_{D_{i2}} + i_{D_{i2}} R_{D_{i2}}/R_{T_{i2}} - (v_{C_{i,1}} - v_{C_{3-i,1}})/R_{\text{load}}}{C_{i1}} \\ \frac{dv_{C_{i1}}}{dt} = \frac{i_{D_{i2}} + i_{D_{i2}} R_{D_{i2}}/R_{T_{i2}} - (v_{C_{i,1}} - v_{C_{3-i,1}})/R_{\text{load}}}{C_{i1}} \end{cases} \tag{5.4}$$

Para este caso las corrientes en los diodos son expresadas por las Ecuaciones(5.5):

$$\begin{cases} i_{D_{i2}} = \frac{(i_{L_{i1}} R_{T_{i1}} - v_{C_{i1}} - V_{in}) R_{T_{i2}}}{R_{T_{i1}} R_{D_{i2}} + R_{T_{i1}} R_{T_{i2}} + R_{D_{i2}} R_{T_{i2}}} \\ i_{D_{i2}} = \frac{(i_{L_{i1}} R_{T_{i1}} - v_{C_{i1}} - V_{in}) R_{T_{i2}}}{R_{T_{i1}} R_{D_{i2}} + R_{T_{i1}} R_{T_{i2}} + R_{D_{i2}} R_{T_{i2}}} \end{cases} \tag{5.5}$$

En este caso, la tensión de salida del buck–boost se define como $v_{\text{out}} = v_{C_{11}} - v_{C_{21}}$.

El siguiente código μ–Modelica es una implementación del modelo completo del convertidor DMSI PWM buck-boost descripto por las Ecuaciones 5.4–5.5. Solo se presenta la sección del código de ecuaciones, porque el algoritmo de control de transistores y diodos ya fue explicado anteriormente.

```
1  equation
2  //buckboost1
3      iD12=(RT11*iL11-vC12-U)*RT12/(RT11*Rd12+RT11*RT12+Rd12*RT12);
4      s1=diodeon1*iD12+(1-diodeon1)*iD12*Rd12;
5      der(vC12) = (iD12+iD12*Rd12/RT12 - (vC12-vC21)/Rload)/C1;
6      der(iL11) =  (-vC12-iD12*Rd12)/L11;
7
8  //buckboost2
9      iD22=(RT21*iL21-vC21-V_{in})*RT22/(RT21*Rd22+RT21*RT22+Rd22*RT22);
10     s2=diodeon2*iD22+(1-diodeon2)*iD22*Rd22;
11     der(vC21) = (iD22+iD22*Rd22/RT22 - (vC21-vC12)/Rload)/C21;
12     der(iL21) =  (-vC21-iD22*Rd22)/L21;
13     vout=vC12-vC21; //Salida de tension diferencial
```

Para el caso del DMSI PWM Ćuk, se pueden deducir las EDOs que se presentan en las Ecuaciones (5.6).

$$\begin{cases} \frac{di_{L_{12}}}{dt} = \frac{-v_{C_{12}} - i_{D_{12}} R_{D_{12}}}{L_{12}} \\ \frac{di_{L_{22}}}{dt} = \frac{-v_{C_{22}} - i_{D_{22}} R_{D_{22}}}{L_{22}} \\ \frac{dv_{C_{11}}}{dt} = \frac{i_{D_{12}} + i_{D_{12}} R_{D_{12}} / R_{T_{12}} - i_{L_{12}}}{C_{11}} \\ \frac{dv_{C_{21}}}{dt} = \frac{i_{D_{22}} + i_{D_{22}} R_{D_{22}} / R_{T_{22}} - i_{L_{22}}}{C_{21}} \\ \frac{dv_{C_{12}}}{dt} = \frac{i_{L_{12}} - (v_{C_{12}} - v_{C_{22}}) / R_{\text{load}}}{C_{12}} \\ \frac{dv_{C_{22}}}{dt} = \frac{i_{L_{22}} - (v_{C_{22}} - v_{C_{12}}) / R_{\text{load}}}{C_{22}} \\ \frac{dphi_1}{dt} = \frac{V_{in} + v_{C_{12}} - v_{C_{11}}}{L_{12}} \\ \frac{dphi_2}{dt} = \frac{V_{in} + v_{C_{22}} - v_{C_{21}}}{L_{22}} \end{cases} \tag{5.6}$$

Las corrientes en los diodos son expresadas por las Ecuaciones(5.7):

$$\begin{cases} i_{D_{12}} = \frac{(R_{T_{11}}(i_{L_{12}} + i_{L_{11}}) - v_{C_{11}}) R_{T_{12}}}{R_{T_{11}} R_{D_{12}} + R_{T_{11}} R_{T_{12}} + R_{D_{12}} R_{T_{12}}} \\ i_{D_{22}} = \frac{(R_{T_{21}}(i_{L_{22}} + i_{L_{21}}) - v_{C_{21}}) R_{T_{22}}}{R_{T_{21}} R_{D_{22}} + R_{T_{21}} R_{T_{22}} + R_{D_{22}} R_{T_{22}}} \\ i_{L_{11}} = \frac{L_{12} phi_i + L_{12} i_{L_{12}}}{L_{11}} \\ i_{L_{21}} = \frac{L_{22} phi_i + L_{22} i_{L_{22}}}{L_{21}} \end{cases} \tag{5.7}$$

En este caso, la tensión de salida inversor Ćuk se define como $v_{\text{out}} = v_{C_{12}} - v_{C_{22}}$.

De igual modo, a continuación se presenta el código μ–Modelica correspondiente al modelo completo del convertidor DMSI PWM Ćuk. Al igual que en el caso del inversor buck–boost, solo se incluye la sección del código de ecuaciones, ya que el algoritmo de control de transistores y diodos es el mismo que fue descrito anteriormente.

```
equation
//cuk1
    iL11=(L12*phi1+L12*iL12)/L11;
    iD1=(RT11*(iL12+iL11)-vC11)*RT12/(RT11*RT12+RT11*Rd12+RT12*Rd12);
    s1=diodeon1*iD1+(1-diodeon1)*iD1*Rd12;
    der(vC11) = (iD1+iD1*Rd12/RT12 - iL12)/C11;
    der(phi1) = ( Vin+vC12-vC11)/L12;
    der(vC12) = (iL12 - (vC12-vC22)/R/C12;
    der(iL12) =  (-vC12-iD1*Rd12)/L12;

//cuk2
    iL21=(L20*phi2+L20*iL20)/L21;
    iD2=(RT21*(iL20+iL21)-vC21)*RT22/(RT21*RT22+RT21*Rd22+RT22*Rd22);
```

```
14    s2=diodeon2*iD2+(1-diodeon2)*iD2*Rd22;
15    der(uC21) = (iD2+iD2*Rd22/RT22 - iL20)/C21;
16    der(phi2) = ( Vin+vC20-vC21)/L20;
17    der(uC20) = (iL20 - (vC20-vC10)/R-0*vC20/R)/C20;
18    der(iL20) =  (-vC20-iD2*Rd22)/L20;
19    vout=vC12-vC22; //Salida de tension diferencial
```

Estos modelos pueden ser simulados tanto en QSS Solver como en OpenModelica a partir de los códigos presentados.

5.2.4 Análisis de rigidez de los modelos

Al modelar de manera realista los elementos de conmutación, el modelo resultante puede volverse rígido en determinadas condiciones de conducción, lo cual, combinado con las discontinuidades frecuentes, genera problemas a los algoritmos de simulación.

En general, cuanto mayor sea la diferencia entre los valores de R_{on} y R_{off}, mayor será la rigidez del sistema.

Para analizar la rigidez de un modelo, se debe estudiar la parte real de los autovalores de su matriz Jacobiana J [36]. A continuación, se presenta el análisis de rigidez de distintas topologías DMSIs PWM.

Análisis de rigidez del modelo DMSI buck

En el caso del DMSI PWM buck, si se denomina $R_{TP_{ij}}$ a la resistencia equivalente al paralelo de $R_{D_{ij}}$ y $R_{T_{ij}}$, se obtiene la siguiente matriz Jacobiana, la cual depende de los parámetros $R_{TP_{ij}}$, C_{i1}, L_{i1} y R_{load}.

Considerando el vector de estados $\mathbf{x}^\top = \begin{bmatrix} i_{L_{11}} & i_{L_{21}} & v_{C_{11}} & v_{C_{21}} \end{bmatrix}$, resulta:

$$J_{\text{buck}} = \begin{bmatrix} -\dfrac{\alpha_{TP_{11}}}{L_{11}} & 0 & -\dfrac{1}{L_{11}} & 0 \\ 0 & -\dfrac{\alpha_{TP_{21}}}{L_{21}} & 0 & -\dfrac{1}{L_{21}} \\ \dfrac{1}{C_{11}} & 0 & -\dfrac{1}{R_{\text{load}}C_{11}} & \dfrac{1}{R_{\text{load}}C_{11}} \\ 0 & \dfrac{1}{C_{21}} & \dfrac{1}{R_{\text{load}}C_{21}} & -\dfrac{1}{R_{\text{load}}C_{21}} \end{bmatrix} \tag{5.8}$$

donde

$$\alpha_{TP_{i1}} = \frac{R_{TPi1} \cdot R_{TPi2}}{R_{TPi1} + R_{TPi2}}$$

Teniendo en cuenta los estados alto o bajo de las señales PWM que controlan cada rama, se pueden presentar los siguientes casos:

$$\begin{array}{llll} R_{TP_{11}} = R_{\text{off}} & R_{TP_{12}} = R_{\text{on}} & R_{TP_{21}} = R_{\text{off}} & R_{TP_{22}} = R_{\text{on}} \\ R_{TP_{11}} = R_{\text{on}} & R_{TP_{12}} = R_{\text{off}} & R_{TP_{21}} = R_{\text{off}} & R_{TP_{22}} = R_{\text{on}} \\ R_{TP_{11}} = R_{\text{on}} & R_{TP_{12}} = R_{\text{off}} & R_{TP_{21}} = R_{\text{on}} & R_{TP_{22}} = R_{\text{off}} \\ R_{TP_{11}} = R_{\text{off}} & R_{TP_{12}} = R_{\text{on}} & R_{TP_{21}} = R_{\text{on}} & R_{TP_{22}} = R_{\text{off}} \end{array} \tag{5.9}$$

Entonces, considerando que $R_{\text{on}} \ll R_{\text{off}}$ (generalmente $R_{\text{on}} = R_{\text{off}}^{-1}$), resulta:

$$\alpha_{TP_{i1}} \approx \frac{R_{\text{on}} \cdot R_{\text{off}}}{R_{\text{on}} + R_{\text{off}}} \approx R_{\text{on}} \tag{5.10}$$

luego considerando $L_{11} = L_{21}$ y $C_{11} = C_{21}$, se puede definir:

$$a = \frac{\alpha_{TP_{i1}}}{L_{i1}} \approx \frac{R_{\text{on}}}{L_{i1}} \qquad b = \frac{1}{L_{i1}} \qquad c = \frac{1}{C_{i1}} \qquad d = \frac{1}{R_{\text{load}} C_{i1}} \tag{5.11}$$

y la matriz $J_{\textbf{buck}}$ puede escribirse como:

$$J_{buck} = \begin{bmatrix} -a & 0 & -b & 0 \\ 0 & -a & 0 & -b \\ c & 0 & -d & d \\ 0 & c & d & -d \end{bmatrix} \tag{5.12}$$

cuyos autovalores son:

$$\lambda_{1,2} = -\frac{a}{2} \pm \frac{1}{2}\sqrt{a^2 - 4bc} \tag{5.13}$$

$$\lambda_{3,4} = -\frac{a}{2} - d \pm \frac{1}{2}\sqrt{a^2 - 4ad - 4bc + 4d^2} \tag{5.14}$$

Luego, comparando los valores de a, b y c en la Ecuación 5.11, se observa que $a \ll b$. Además, dado que L_{ij} y C_{ij} son, en general, del mismo orden de magnitud, se tiene que $b \sim c$, por lo que también se cumple $a \ll c$.

En consecuencia, en la Ecuación 5.13 se verifica que $a^2 - 4bc \ll 0$, y por lo tanto la parte real de los autovalores se aproxima por $\text{Re}(\lambda_{1,2}) = -\frac{a}{2}$.

En la Ecuación 5.14, si $R_{\text{load}} > L$, lo cual se cumple en condiciones normales de operación de los DMSI PWM, entonces se verifica que $a^2 - 4ad - 4bc + 4d^2 < 0$, y por lo tanto $\text{Re}(\lambda_{3,4}) = -\frac{a}{2} - d$.

Además, teniendo en cuenta que $a \approx \frac{R_{\text{on}}}{L_{i1}}$, $d = \frac{1}{R_{\text{load}} C_{i1}}$, y que L_{i1} es del mismo orden de magnitud que C_{i1}, se obtiene que $d \sim a \cdot \frac{1}{R_{\text{load}} R_{\text{on}}}$. Dado que $R_{\text{on}} \ll R_{\text{load}}$, se concluye que $d \gg a$, y en consecuencia $\text{Re}(\lambda_{3,4}) \approx -d$.

Comparando las partes reales de $\lambda_{1,2}$ y $\lambda_{3,4}$, y teniendo en cuenta que $d \gg a$, se concluye que el sistema es rígido.

Esto implica que su integración numérica mediante métodos explícitos requeriría pasos de tiempo extremadamente pequeños para garantizar la estabilidad.

Análisis de rigidez del modelo DMSI buck–boost

Para llevar a cabo el análisis del modelo del DMSI PWM buck–boost, se analiza la siguiente matriz Jacobiana asociada al vector de estados $\mathbf{x}^\top = \begin{bmatrix} i_{L_{11}} & i_{L_{21}} & v_{C_{11}} & v_{C_{21}} \end{bmatrix}$:

$$J_{\text{B-B}} = \begin{bmatrix} -\frac{\alpha_{TP_{11}} R_{TP_{12}}}{L_{11}} & 0 & -\frac{\alpha_{TP_{11}}}{L_{11}} & 0 \\ 0 & -\frac{\alpha_{TP_{21}} R_{TP_{22}}}{L_{21}} & 0 & -\frac{\alpha_{TP_{21}}}{L_{21}} \\ \frac{1}{C_{11}}\left(\alpha_{TP_{11}}+1\right) & 0 & -\frac{1}{C_{11}}\left(\frac{1}{R_{\text{load}}}+\frac{\alpha_{TP_{11}}}{R_{TP_{11}}}\right) & \frac{1}{C_{11}R_{\text{load}}} \\ 0 & \frac{1}{C_{21}}\left(\alpha_{TP_{21}}+1\right) & \frac{1}{C_{21}R_{\text{load}}} & -\frac{1}{C_{21}}\left(\frac{1}{R_{\text{load}}}+\frac{\alpha_{TP_{21}}}{R_{TP_{21}}}\right) \end{bmatrix} \tag{5.15}$$

donde:

$$\alpha_{TP_{i1}} = \frac{R_{TPi1}}{R_{TPi1}+R_{TPi2}} \quad \text{con } i = 1, 2$$

si consideramos que los elementos de conmutación de las dos ramas de esta topología de DMSI PWM se manejan a través de las señales PWM de la misma manera que la DMSI PWM buck entonces las combinaciones posibles de $R_{TP_{11}}$ son las mismas que las mostradas en la Ecuación 5.9.

La matriz $J_{\text{B-B}}$ puede de manera simplificada como:

$$J_{\text{B-B}} = \begin{bmatrix} -a & 0 & -b & 0 \\ 0 & -a & 0 & -b \\ c & 0 & -d & d \\ 0 & c & d & -d \end{bmatrix} \tag{5.16}$$

teniendo en cuenta que esta matriz $J_{\text{B-B}}(a, b, c, d)$ tiene la misma estructura que la correspondiente a la buck, entonces los autovalores de la topología DMSI PWM buck-boost también están dados por las Ecuaciones 5.13 y 5.14:

$$\begin{aligned} \lambda_{1,2} &= -\frac{a}{2} \pm \frac{1}{2}\sqrt{a^2 - 4bc} \\ \lambda_{3,4} &= -\frac{a}{2} - d \pm \frac{1}{2}\sqrt{a^2 - 4ad - 4bc + 4d^2} \end{aligned}$$

Considerando la primera condición mostrada en la Ecuación 5.9 resulta $\alpha_{TP_{i1}} \approx 1$. Teniendo esto en cuenta, los parámetros de la matriz $J_{\text{B-B}}$ son:

$$a \approx \frac{R_{\text{on}}}{L_{i1}}, \qquad b = \frac{1}{L_{i1}}, \qquad c = \frac{2}{C_{i1}}, \qquad d \approx \frac{1}{R_{\text{load}} C_{i1}} \tag{5.17}$$

comparando estos valores con los de la buck se ve que solo el término c cambia siendo en este caso el doble que en la buck. Además, todo el análisis realizado sobre rigidez en la buck es también valido en este caso ya que la relación entre los parámetros a, b, c y d es la misma. La diferencia entre el valor de c entre ambos DMSI solo afecta a la parte compleja de los autovalores. De este modo se verifica que esta topología también presenta una estructura rígida (al menos en una de las condiciones de operación de los componentes de conmutación)

Análisis de rigidez del modelo DMSI Ćuk

Considerando el vector de estados $\mathbf{x} = \begin{bmatrix} i_{L_{11}} & i_{L_{12}} & i_{L_{21}} & i_{L_{22}} & v_{C_{11}} & v_{C_{12}} & v_{C_{21}} & v_{C_{22}} \end{bmatrix}$ la matriz Jacobiana para la topología de la DMSI PWM Ćuk resulta:

$$J_{\text{Ćuk}} = \begin{bmatrix} -\frac{\alpha_{TP_{11}}}{L_{11}} & \frac{\alpha_{TP_{11}}}{L_{11}} & 0 & 0 & -\frac{\alpha_{TP_{11}}}{R_{TP_{12}} L_{11}} & 0 & 0 & 0 \\ \frac{\alpha_{TP_{11}}}{L_{12}} & -\frac{\alpha_{TP_{11}}}{L_{12}} & 0 & 0 & -\frac{\alpha_{TP_{11}}}{R_{TP_{11}} L_{12}} & -\frac{1}{L_{12}} & 0 & 0 \\ 0 & 0 & -\frac{\alpha_{TP_{21}}}{L_{21}} & \frac{\alpha_{TP_{21}}}{L_{21}} & 0 & 0 & -\frac{\alpha_{TP_{21}}}{R_{TP_{22}} L_{21}} & 0 \\ 0 & 0 & \frac{\alpha_{TP_{21}}}{L_{22}} & -\frac{\alpha_{TP_{21}}}{L_{22}} & 0 & 0 & -\frac{\alpha_{TP_{21}}}{R_{TP_{21}} L_{22}} & -\frac{1}{L_{22}} \\ \frac{\alpha_{TP_{11}}}{R_{TP_{12}} C_{11}} & \frac{\alpha_{TP_{11}}}{R_{TP_{11}} C_{11}} & 0 & 0 & -\frac{\alpha_{TP_{11}}}{R_{TP_{11}} R_{TP_{12}} C_{11}} & 0 & 0 & 0 \\ 0 & \frac{1}{C_{12}} & 0 & 0 & 0 & -\frac{1}{R_{\text{load}} C_{12}} & 0 & \frac{1}{R_{\text{load}} C_{12}} \\ 0 & 0 & \frac{\alpha_{TP_{21}}}{R_{TP_{22}} C_{21}} & \frac{\alpha_{TP_{21}}}{R_{TP_{21}} C_{21}} & 0 & 0 & -\frac{\alpha_{TP_{21}}}{R_{TP_{21}} R_{TP_{22}} C_{21}} & 0 \\ 0 & 0 & 0 & \frac{1}{C_{22}} & 0 & \frac{1}{R_{\text{load}} C_{22}} & 0 & -\frac{1}{R_{\text{load}} C_{22}} \end{bmatrix} \tag{5.18}$$

donde:

$$\alpha_{TP_{i1}} = \frac{R_{TPi1} \cdot R_{TPi2}}{R_{TPi1} + R_{TPi2}} \text{ con } i = 1, 2$$

Considerando las siguientes aproximaciones: $C_{ij} \sim C$ y $L_{ij} \sim L$ para todo (i, j), L y C del mismo orden, que $R_{TP_{i1}} R_{TP_{i2}} = 1$ y que $\alpha_{TP_{i1}} = R_{\text{on}}$ con $i = 1, 2$ para cualquiera de las condiciones posibles de los elementos de conmutación entonces la matriz Jacobiana mostrada en la Ecuación 5.18 puede escribirse como:

$$J_{\text{Ćuk}} = \begin{bmatrix} -a & a & 0 & 0 & -b & 0 & 0 & 0 \\ a & -a & 0 & 0 & -c & -d & 0 & 0 \\ 0 & 0 & -a & a & 0 & 0 & -e & 0 \\ 0 & 0 & a & -a & 0 & 0 & -f & -d \\ b & c & 0 & 0 & -a & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & -g & 0 & g \\ 0 & 0 & e & f & 0 & 0 & -a & 0 \\ 0 & 0 & 0 & d & 0 & g & 0 & -g \end{bmatrix} \tag{5.19}$$

Dado que los elementos de conmutación son excitados de la misma manera que en la DMSI PWM buck, se siguen cumpliendo las condiciones establecidas en la Ecuación 5.9. Para cualquiera de estas condiciones se mantienen los siguientes parámetros:

$$a \approx \frac{R_{\text{on}}}{L} \sim \frac{R_{\text{on}}}{C} \quad d = \frac{1}{L} \sim \frac{1}{C} \quad g = \frac{1}{R_{\text{load}} C} \tag{5.20}$$

Sin embargo, los parámetros que cambian según el estado de los elementos de conmutación se muestran en la Tabla 5.1.

Condición	**Estado de transistores**	b	c	e	f
1	$R_{TP_{11}} = R_{TP_{21}} = R_{\text{off}}$, $R_{TP_{12}} = R_{TP_{22}} = R_{\text{on}}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$
2	$R_{TP_{11}} = R_{TP_{22}} = R_{\text{on}}$, $R_{TP_{12}} = R_{TP_{21}} = R_{\text{off}}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$
3	$R_{TP_{11}} = R_{TP_{21}} = R_{\text{on}}$, $R_{TP_{12}} = R_{TP_{22}} = R_{\text{off}}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{1}{L} \sim \frac{1}{C}$
4	$R_{TP_{11}} = R_{TP_{22}} = R_{\text{off}}$, $R_{TP_{12}} = R_{TP_{21}} = R_{\text{on}}$	$\frac{1}{L} \sim \frac{1}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{R_{\text{on}}^2}{L} \sim \frac{R_{\text{on}}^2}{C}$	$\frac{1}{L} \sim \frac{1}{C}$

Tabla 5.1: Valores aproximados de los parámetros b, c, e y f según el estado de los transistores

Observando la Tabla 5.1 se puede verificar que tanto en la condición 1 como en la 3 $b = e$ y $c = f$, definiendo $b_1 = b = e$ y $c_1 = c = f$, la matriz Jacobiana para estas dos

condiciones resulta en la matriz $J_{\acute{\text{C}}\text{uk}_{1,3}}$ de las Matrices 5.21.

$$J_{\acute{\text{C}}\text{uk}_{1,3}} = \begin{bmatrix} -a & a & 0 & 0 & -b_1 & 0 & 0 & 0 \\ a & -a & 0 & 0 & -c_1 & -d & 0 & 0 \\ 0 & 0 & -a & a & 0 & 0 & -b_1 & 0 \\ 0 & 0 & a & -a & 0 & 0 & -c_1 & -d \\ b_1 & c_1 & 0 & 0 & -a & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & -g & 0 & g \\ 0 & 0 & b_1 & c_1 & 0 & 0 & -a & 0 \\ 0 & 0 & 0 & d & 0 & g & 0 & -g \end{bmatrix}, \quad J_{\acute{\text{C}}\text{uk}_{2,4}} = \begin{bmatrix} -a & a & 0 & 0 & -b_2 & 0 & 0 & 0 \\ a & -a & 0 & 0 & -c_2 & -d & 0 & 0 \\ 0 & 0 & -a & a & 0 & 0 & -c_2 & 0 \\ 0 & 0 & a & -a & 0 & 0 & -b_2 & -d \\ b_2 & c_2 & 0 & 0 & -a & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & -g & 0 & g \\ 0 & 0 & c_2 & b_2 & 0 & 0 & -a & 0 \\ 0 & 0 & 0 & d & 0 & g & 0 & -g \end{bmatrix} \tag{5.21}$$

De manera equivalente, como en las condiciones 2 y 4 $b = f$ y $c = e$, entonces definiendo $b_2 = b = f$ y $c_2 = c = e$, la matriz Jacobiana para estas dos condiciones puede escribirse como la matriz $J_{\acute{\text{C}}\text{uk}_{2,4}}$ de las Matrices 5.21.

A pesar de las simplificaciones planteadas en los parámetros de la matriz $J_{\acute{\text{C}}\text{uk}}$, obtener una expresión analítica de sus autovalores resulta prácticamente imposible debido a la complejidad de esta matriz de 8×8, incluso cuando presenta ciertas características de simetría. Por este motivo, el análisis de rigidez en este caso se aborda mediante la teoría de matrices perturbadas [36, 30].

Para realizar el análisis dinámico de esta matriz se raliza la descomposición de la matriz $J_{\acute{\text{C}}\text{uk}}$ en una *matriz base* (D) y una de *perturbación* (F). De este modo:

$$J_{\acute{\text{C}}\text{uk}} = D + F, \tag{5.22}$$

donde D es:

$$D = \begin{bmatrix} -a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g \end{bmatrix}, \quad (5.23)$$

y F según las condiciones de los elementos de conmutación:

$$F_{1,3} = \begin{bmatrix} 0 & a & 0 & 0 & -b_1 & 0 & 0 & 0 \\ a & 0 & 0 & 0 & -c_1 & -d & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 & -b_1 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & -c_1 & -d \\ b_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & 0 & 0 & g \\ 0 & 0 & b_1 & c_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & g & 0 & 0 \end{bmatrix}, \quad F_{2,4} = \begin{bmatrix} 0 & a & 0 & 0 & -b_2 & 0 & 0 & 0 \\ a & 0 & 0 & 0 & -c_2 & -d & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 & -c_2 & 0 \\ 0 & 0 & a & 0 & 0 & 0 & -b_2 & -d \\ b_2 & c_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d & 0 & 0 & 0 & 0 & 0 & g \\ 0 & 0 & c_2 & b_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & g & 0 & 0 \end{bmatrix} \quad (5.24)$$

Esta descomposición permite analizar la influencia de los acoplamientos expresados en la matriz F sobre la posición de los autovalores λ_{D_i} de la matriz D. Según el teorema de Bauer–Fike de perturbación de autovalores, si se cumple que las normas de las matrices $\|F\| \ll \|D\|$, entonces la posición de los autovalores λ_i de $J_{\text{Ćuk}}$ es muy similar a la de los autovalores λ_{D_i} de D (los acoplamientos son débiles y no perturban significativamente el espectro de D).

En el caso de estudio $\|F\| \gg \|D\|$ ya que algunos elementos de la matriz de perturbación toman valores grandes. En este caso, a partir del *teorema de los discos de Geršgorin*, se puede analizar la influencia de los acomplamientos sobre los autovalores de la matriz no perturbada. El mismo permite acotar la zona donde quedaran finalmente los autovalores de

la matriz $J = D + F$ en el plano complejo.

Aplicado a la matriz $J_{\text{Ćuk}}$, este teorema afirma que sus autovalores λ_i estarán dentro de un disco C_i de radio R_i centrado en λ_i donde:

$$R_i = \sum_{j=1}^{8} |F_{ij}| \tag{5.25}$$

De este modo el radio R_i de cada disco se calcula como la suma de los valores absolutos de los elementos de la i-ésima fila. Cuanto más grandes sean los radios R_i, mayor será la incertidumbre en la localización de los autovalores.

En la Tabla 5.2 se muestran los radios de los discos de Geršgorin y su centro para las dos posibles matrices de perturbación resultantes de todos los estados posibles de los elementos de conmutación (Matrices 5.21).

En esta tabla puede verse que hay autovalores λ_{D_i} de la matriz base que tienen parte real muy diferente entre si (λ_{D_8} y λ_{D_6} son mucho menores que los demás dado que $a \ll g$ mientras $R_{\text{load}} \ll 1/R_{\text{on}}$). Aunque los autovalores de la matriz D no correspondan a los definitivos de $J_{\text{Ćuk}}$, es posible advertir que la coexistencia de autovalores con escalas muy distintas en D (dado que $a \ll g$) pone de manifiesto la posible presencia de rigidez estructural. Sin embargo, no se puede concluir sobre la rigidez del sistema ya que los radios de los discos centrados en λ_{D_8} y λ_{D_6} tienen un radio mayor que g por lo que estos discos incluyen a los demás autovalores.

Autovalor Matriz D	**Radio (R_i) Condiciones 1 y 3**	**Radio (R_i) Condiciones 2 y 4**
$\lambda_{D_1} = -a$	$R_1 = b_2$	$R_1 = b_1$
$\lambda_{D_2} = -a$	$R_2 = c_2 + d$	$R_2 = c_1 + d$
$\lambda_{D_3} = -a$	$R_3 = c_2$	$R_3 = b_1$
$\lambda_{D_4} = -a$	$R_4 = b_2 + d$	$R_4 = c_1 + d$
$\lambda_{D_5} = -a$	$R_5 = b_2 + c_2$	$R_5 = b_1 + c_1$
$\lambda_{D_6} = -g$	$R_6 = d + g$	$R_6 = d + g$
$\lambda_{D_7} = -a$	$R_7 = b_1 + c_1$	$R_7 = b_2 + c_2$
$\lambda_{D_8} = -g$	$R_8 = d + g$	$R_8 = d + g$

Tabla 5.2: Autovalores y radios de los discos de Geršgorin para distintas condiciones.

Por los motivos expuestos, la única forma de verificar la rigidez sugerida a partir de los autovalores de D en esta topología es mediante la evaluación numérica de los autovalores de la matriz $J_{\text{Ćuk}}$ ya parametrizada. En este sentido, se realizaron diversas pruebas con configuraciones paramétricas típicas, y la evaluación numérica confirmó efectivamente la

rigidez.

5.3 Resultados y comparativa de simulación

Esta sección presenta los resultados de simulación de los tres modelos de inversores DMSI PWM descritos anteriormente, utilizando los métodos numéricos basados en cuantificación de los estados LIQSS2 y de tiempo discreto CVODE–BDF y DASSL ya que son los más adecuados para simular modelos rígidos. Los parámetros de los modelos se muestran en la Tabla 5.3.

En los tres casos, se emplearon los siguientes ciclos de trabajo, que comandan cada convertidor operando a lazo abierto:

$$\begin{aligned} d_{c_1}(t) &= 0{,}5 + 0{,}3\sin(2\pi f_o t) \\ d_{c_2}(t) &= 1 - d_{c_1}(t) \end{aligned} \tag{5.26}$$

Parámetro	**DMSI PWM**		
	buck	buck-boost	Ćuk
U	$24\,V$	$24\,V$	$24\,V$
Frec.PWM	10kHz	$10\,kHz$	$10\,kHz$
R_{load}	$10\,\Omega$	$10\,\Omega$	$10\,\Omega$
$L_{11} = L_{21}$	$100\,\mu H$	$100\,\mu H$	$100\,\mu H$
$L_{12} = L_{22}$	$100\,\mu H$	$100\,\mu H$	$100\,\mu H$
$C_{11} = C_{12}$	$100\,\mu F$	$100\,\mu F$	$100\,\mu F$
$C_{21} = C_{22}$	-	-	$100\,\mu F$

Tabla 5.3: Parámetros de los DMSI PWM.

Con estas referencias de ciclo de trabajo, y tomando los valores $R_{\text{On}} = 1 \cdot 10^{-5}\,\Omega$ y $R_{\text{Off}} = 1 \cdot 10^{5}\,\Omega$ para las resistencias que modelan los elementos de conmutación (diodos y transistores), se obtienen las tensiones de CA de salida de los inversores mostradas en la Figura 5.5. Estos resultados de simulación se obtuvieron utilizando LIQSS2. Cabe mencionar que para obtener una onda de salida cuyo valor medio sea con un perfil más senoidal es necesario implementar un lazo de control de la tensión de salida.

En todos los casos se midió el tiempo de CPU para simular 1 segundo cada modelo utilizando LIQSS2 y los algoritmos clásicos DASSL y CVODE–BDF. Además, se computó el error relativo de los resultados obtenidos en cada simulación a partir de la siguiente ecuación:

$$e_{rr} = \sqrt{\frac{\sum (x(k) - x_{ref}(k))^2}{\sum x_{ref}^2(k)}} \tag{5.27}$$

Para el cálculo del error se tomaron como referencia los resultados de simulación ob-

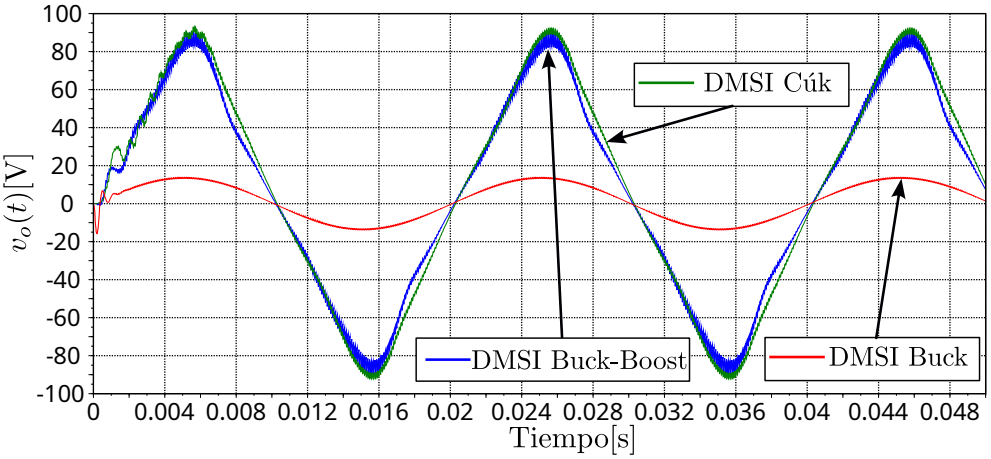


Figura 5.5: Tensión de salida de DMSIs tipo buck, buck–boost y Ćuk.

tenidos utilizando DASSL con una resolución de $1 \cdot 10^{-12}$. Las simulaciones para obtener estas referencias se realizaron en OpenModelica.

Como indicadores de desempeño se utilizaron no solo el error relativo, sino también el tiempo de CPU requerido por las simulaciones y el *factor de aceleración* (A_f) definido como:

$$A_f = t_{\text{virtual}}/t_{\text{CPU}} \tag{5.28}$$

donde $t_{\text{virtual}} = 1$ s es el tiempo de simulación. Debe tenerse en cuenta que este los valores obtenidos de este indicador de desempeño dependerán no solo de los algoritmos de simulación y modelos propuestos sino también de las herramientas de hardware utilizadas para la simulación. Observar que valores $A_f > 1$ indican simulación más rápida que el tiempo real, mientras que $A_f < 1$ denota ejecución más lenta.

Las simulaciones con LIQSS2 y con los métodos clásicos DASSL y CVODE–BDF se ejecutaron en la misma herramienta (el QSS Solver) para que la comparación fuese independiente de la herramienta donde estén implementados los métodos. Además, para medir el desempeño del QSS Solver, los mismos modelos fueron simulados tanto en esta herramienta como en OpenModelica 1.24.5 utilizando DASSL. Estas herramientas se ejecutaron en un sistema operativo Ubuntu, utilizando un procesador (CPU) Intel(R)Core (TM) i3-2350MCPU @ 2.30GHz.

En la Tabla 5.4 pueden verse los indicadores de desempeño obtenidos en las simulaciones del DMSI buck. Puede verse que en este caso LIQSS2 es 1.75 veces más rápido que CVODE–BDF y 3.84 veces más rápido que DASSL ante el mismo requerimiento de error de simulación (tolerancia de error = 10^{-3}). Además, se observa que, si bien todos los métodos cumplen con la tolerancia de error especificada en la simulación, el valor obtenido con LIQSS2 es 2 veces menor que con DASSL y 1.5 veces menor que con CVODE–BDF.

Método de Integración		Error Relativo	Tiempo de CPU [ms]	A_f
LIQSS2 (QSS-Solver)	$\Delta Q_i = 10^{-3}$	$0{,}7 \cdot 10^{-3}$	159,4	6,27
CVODE (QSS-Solver)	tol.err.=10^{-3}	$1{,}0 \cdot 10^{-3}$	278,9	3,59
DASSL (QSS-Solver)	tol.err.=10^{-3}	$1{,}85 \cdot 10^{-3}$	610,4	1,64
DASSL (OpenModelica)	tol.err.=10^{-3}	$1{,}51 \cdot 10^{-3}$	9449,0	0,11

Tabla 5.4: Resultados de simulación del DMSI buck con factor de aceleración A_f.

Si se observan los tiempos de CPU requeridos por el QSS Solver y OpenModelica para simular este modelo utilizando en ambos casos DASSL, resulta evidente que la primera herramienta es mucho más eficiente (15.5 veces más rápida). Esto se debe principalmente a que la detección de eventos implementada en el QSS Solver es considerablemente más eficiente.

De manera equivalente, la Tabla 5.5 muestra los índices de desempeño obtenidos en el caso del inversor buck–boost. En este caso, la simulación realizada con LIQSS2 es 1.71 veces más rápida que con CVODE–BDF, y 4.47 veces más rápida que con DASSL, ambos implementados en el mismo solver. Además, se observa que ante el mismo requerimiento de tolerancia de error establecido para la simulación, el error relativo obtenido con LIQSS2 es 4 veces menor que con DASSL y CVODE–BDF implementados en la misma herramienta, y 15 veces menor que el obtenido con DASSL en OpenModelica.

Método de Integración		Error Relativo	Tiempo de CPU [ms]	A_f
LIQSS2(QSS-Solver)	$\Delta Q_i = 10^{-3}$	$0{,}9 \cdot 10^{-3}$	143,8	6,95
CVODE(QSS-Solver)	tol.err.=10^{-3}	$3{,}6 \cdot 10^{-3}$	244,2	4,10
DASSL(QSS-Solver)	tol.err.=10^{-3}	$3{,}7 \cdot 10^{-3}$	640,9	1,56
DASSL (OpenModelica)	tol.err.=10^{-3}	$1{,}4 \cdot 10^{-2}$	13256,0	0,08

Tabla 5.5: Resultados de simulación del DMSI buck–boost con factor de aceleración A_f.

Respecto a la eficiencia de las herramientas de simulación, se observa que el QSS Solver es requiere 20 veces menos tiempo de CPU que OpenModelica al simular este modelo con DASSL y los mismos requerimientos de error de simulación. Además, notar que los resultados obtenidos en el QSS Solver cumplen con la resolución especificada mientras los obtenidos en OpenModelica no lo hacen. Esto puede atribuirse a un mayor error en la detección de las discontinuidades en OpenModelica.

Por último, los indicadores de desempeño obtenidos en el caso del inversor Ćuk pueden verse en la Tabla 5.6. En ella se observa que LIQSS2 es 1.3 veces más rápido que CVODE–BDF y 3 veces más rápido que DASSL, ambos ejecutados en el QSS Solver. Respecto al *error relativo*, se observa que todos los métodos implementados en el QSS Solver cumplen con la resolución especificada, mientras que el resultado obtenido en OpenModelica no lo hace. Esto puede deberse a la diferencia en la implementación de la detección de

discontinuidades. Además, OpenModelica resulta 17 veces más lento que el QSS Solver.

Método de Integración		**Error Relativo**	**Tiempo de CPU [ms]**	A_f
LIQSS2(QSS-Solver)	$\Delta Q_i = 10^{-3}$	$2{,}2 \cdot 10^{-3}$	$350{,}7$	$2{,}85$
CVODE(QSS-Solver)	tol.err.=10^{-3}	$5{,}0 \cdot 10^{-3}$	$447{,}7$	$2{,}23$
DASSL(QSS-Solver)	tol.err.=10^{-3}	$6{,}5 \cdot 10^{-3}$	$1057{,}2$	$0{,}95$
DASSL (OpenModelica)	tol.err.=10^{-3}	$2{,}2 \cdot 10^{-2}$	$18173{,}0$	$0{,}06$

Tabla 5.6: Resultados de simulación del DMSI Ćuk con factor de aceleración A_f.

En los tres DMSI evaluados (buck, buck–boost y Ćuk), LIQSS2 alcanzó los mejores compromisos *precisión–tiempo*, con $A_f \approx 6{,}27$ (buck), $6{,}95$ (buck–boost) y $2{,}85$ (Ćuk), manteniendo errores relativos del orden de 10^{-3}. CVODE mostró desempeños intermedios ($A_f \approx 3{,}59$, $4{,}10$ y $2{,}23$, respectivamente), coherentes con una integración continua eficiente pero con mayor sobrecarga por manejo de discontinuidades. DASSL dentro de QSS-Solver resultó claramente más lento ($A_f \lesssim 1{,}6$ en buck/buck–boost y $0{,}95$ en Ćuk), y DASSL en OpenModelica fue el escenario más costoso, con $A_f \ll 1$ en las tres topologías. En síntesis, para *modelos realistas* de DMSI PWM, LIQSS2 ofrece una aceleración sustantiva respecto de los métodos clásicos, preservando niveles de error prácticos para análisis y diseño.

A través de estos ejemplos se puede observar que los DMSIs PWM pueden ser simulados de manera más eficiente con LIQSS que con los métodos tradicionales. Esto se evidencia no solo en términos de disminución en el tiempo de CPU, sino también en el mejor cumplimiento de la tolerancia de error especificada para la simulación. Además, debe tenerse en cuenta que se está comparando un método de segundo orden (LIQSS2) con métodos de integración de quinto orden.

Si bien el uso de los métodos LIQSS logró mejoras significativas en la simulación de los DMSIs PWM, estas resultan insuficientes cuando se requiere simular sistemas que integran dichos inversores, como es el caso de grandes sistemas de generación de energía. En estos casos, además de tratarse de modelos con alta demanda computacional, los tiempos de simulación suelen ser extremadamente largos (del orden de horas o incluso días).

En el contexto de esta tesis, el concepto de *simulación de largo plazo* se refiere a integrar el modelo durante tiempos de simulación que son varios órdenes de magnitud mayores que el período de conmutación T_{sw} del DMSI PWM. Esta característica se verifica particularmente en la simulación de sistemas de generación, donde se requieren intervalos prolongados de simulación debido a las dinámicas lentas asociadas a procesos de carga de baterías, pilas de combustible o a fenómenos climatológicos, entre otros.

Considerando, por ejemplo, un único DMSI PWM con una frecuencia de conmutación de 20 kHz y una simulación de 10 horas, la relación entre el tiempo simulado y el período de conmutación es del orden de 7×10^9, lo que vuelve el problema computacionalmente prohibitivo. Incluso si sólo se simulase esa fuente aislada, el tiempo de ejecución podría alcanzar

aproximadamente 1.5 horas de CPU. Sin embargo, la situación se vuelve aún más crítica al integrar estos modelos dentro de sistemas de generación o conversión más complejos, como microrredes, sistemas híbridos o bancos de convertidores interconectados.

En dichos escenarios, la coexistencia de múltiples escalas de tiempo y de otras dinámicas lentas produce un crecimiento significativo del número de eventos y de las dependencias entre variables, de modo que los tiempos de ejecución —aun empleando LIQSS— comienzan a superar al tiempo virtual de simulación [59].

Estas limitaciones motivan el desarrollo de una nueva estrategia que combine las ventajas de los modelos promediados —en régimen estacionario—con el detalle de los modelos conmutados durante los transitorios. El siguiente capítulo presenta en detalle esta propuesta de *modelado mixto de DMSI PWM*, junto con su implementación práctica y los resultados comparativos que evidencian la mejora obtenida en los tiempos de simulación del modelo.

6. MODELADO MIXTO DE DMSI PWM

6.1 Introducción

Aunque el uso de los métodos LIQSS mejoró significativamente la eficiencia de la simulación de DMSIs PWM, estos avances resultan insuficientes en aplicaciones de largo plazo, en tiempo real o en sistemas de gran tamaño que integran estos inversores y requieren simular períodos extensos. Esta limitación motivó la búsqueda de un enfoque alternativo que no implicara desarrollar nuevos métodos numéricos que mejoren la eficiencia de la simulación de estos sistemas.

Este capítulo presenta una nueva *estrategia de modelado mixto* para DMSIss PWM, adecuada para simulaciones precisas y eficientes de este tipo de sistemas [10].

La estrategia propuesta consiste en implementar un único modelo que incluye un SWM —utilizado durante regímenes transitorios— y un AVM —aplicado en régimen estacionario—. La conmutación entre ambos se realiza automáticamente mediante un algoritmo supervisor que monitorea el estado del régimen. Además, durante la simulación con el modelo conmutado se estiman automáticamente los parámetros del modelo promediado.

Si bien esta estrategia ha demostrado ser eficaz en convertidores CC–CC, su aplicación directa a DMSIs presenta mayores desafíos, especialmente por la complejidad que implica detectar las condiciones para conmutar entre los submodelos promediados y conmutados. Esta dificultad radica principalmente en la naturaleza variable en el tiempo de las señales de los inversores. Esto motivó el estudio más profundo de estos sistemas e investigar una nueva estrategia de modelado mixto específicamente orientada a DMSIss PWM.

Todas las implementaciones de este capítulo se realizaron en Modelica [29], un lenguaje orientado a objetos basado en ecuaciones no causales, que permite definir modelos multidominio y clases reutilizables. Estas características facilitan la integración con componentes de otros dominios (eléctricos, térmicos, mecánicos). Esto hace que los modelos desarrollados sean fácilmente integrables en sistemas de mayor dimensión, lo cual resulta especialmente relevante en el contexto de sistemas de generación de energía renovable [13]. Herramientas como OpenModelica, Dymola y Wolfram System Modeler ofrecen entornos para la simulación de estos modelos.

6.2 Estrategia de modelado mixto para DMSI PWM

La metodología propuesta para topologías DMSI PWM se basa en la idea de la estrategia de modelado mixto aplicada a convertidores PWM CC–CC, es decir, en el uso alternado de submodelos conmutados (SWM) y promediados (AVM).

El SWM (preciso) se utiliza hasta que se detecta un Régimen Permanente Periódico (*Periodic Steady-State*) (PSS). Cuando se cumple esta condición, el modelo conmuta al AVM. Este modo se mantiene activo hasta que se detecta un cambio en el régimen de operación, momento en el cual se retorna al SWM. El esquema general de modelado mixto para DMSI PWM, basado en dos convertidores PWM CC–CC, se muestra en la Figura 6.1.

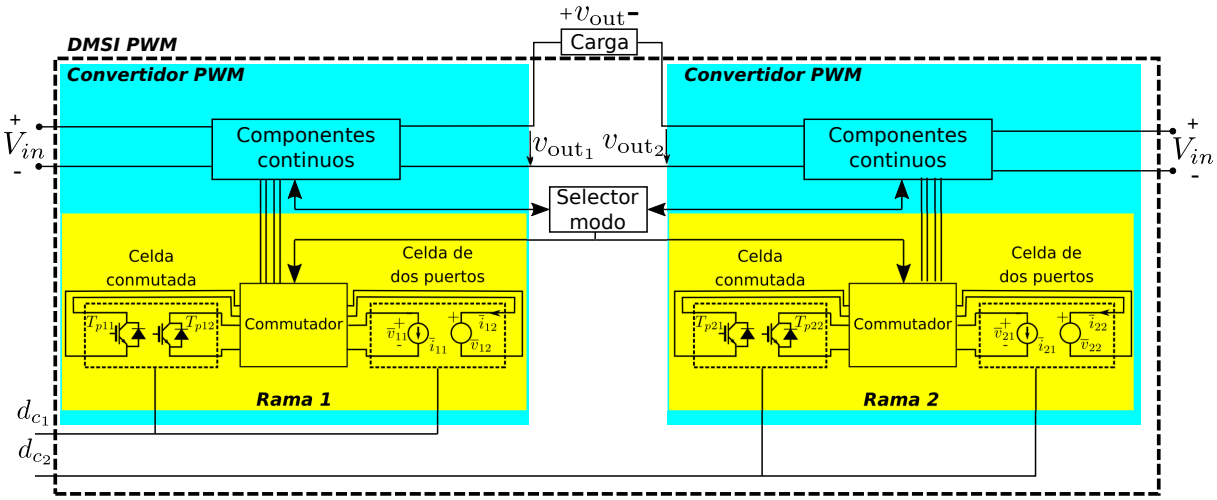


Figura 6.1: Esquema de la estrategia de modelado mixto para DMSIs PWM.

El modelo de los DMSIs PWM se construye conectando dos convertidores CC–CC bidireccionales, con una carga conectada diferencialmente entre sus salidas. En cada uno de estos convertidores pueden identificarse y separarse fácilmente los componentes continuos (resistencias, capacitores e inductancias) de los componentes de conmutación (en este caso, los que conforman cada columna o rama del inversor).

Siguiendo la idea del modelado mixto de convertidores PWM CC–CC, cada par de transistores que integra una rama se modela tanto como una celda de conmutación como una *celda de dos puertos*. La celda de conmutación está compuesta por los componentes que conmutan, mientras que la *celda de dos puertos* está formada por una fuente de corriente controlada y una fuente de tensión controlada.

Estas celdas se vinculan con los circuitos de los componentes continuos a través de los *conmutadores*. Dichos *conmutadores* se encargan de conectar la *celda de conmutación* o la *celda de dos puertos* con los componentes continuos, según lo que determine el bloque

selector de modo.

Dado que durante la simulación deben utilizarse los submodelos conmutados o promediados de forma simultánea en ambos convertidores (no puede emplearse el SWM en uno y el promediado en el otro), se implementa un único *selector de modo* que comanda ambos *conmutadores* en función del estado de las variables de ambos convertidores.

Este bloque ejecuta un algoritmo de supervisión que, a partir de la información proveniente de los componentes continuos, determina si el sistema está en régimen periódico estacionario o en transitorio. De acuerdo con esto, selecciona automáticamente el AVM o el SWM.

De esta manera, el modelo completo integra submodelos promediados y conmutados. Esta estrategia puede comprenderse a partir del análisis de las *celdas de conmutación*, de las *celdas de dos puertos* y del algoritmo de conmutación implementado en el *selector de modo*.

6.2.1 Celda conmutada

En los modelos mixtos de DMSI PWM se tiene una *celda conmutada* por cada i-ésima rama del inversor. Cada *celda conmutada* está compuesta por un par de transistores con diodos antiparalelo, configurados como se muestra en la Figura 6.1. El estado de los transistores es controlado por la señal de entrada d_{c_i}, de frecuencia $1/T_{\text{sw}}$, la cual activa los transistores de manera alternada durante el funcionamiento en SWM.

Esta celda se conecta a través del *conmutador* con los componentes continuos del circuito cuando el *selector de modo* indica que se debe utilizar el SWM. Cuando se utiliza el AVM, el *conmutador* desconecta la *celda conmutada* y mantiene inactivos ambos transistores, evitando así las conmutaciones.

6.2.2 Celda de dos puertos

A cada celda de conmutación le corresponde una *celda de dos puertos* asociada. Estas *celdas de dos puertos* implementan el modelo promediado circuital. Cada una incluye una fuente de tensión controlada y una fuente de corriente controlada como puede verse en la Figura 6.1. De esta manera, cuando el *selector de modo* decide que se debe utilizar el AVM, el *conmutador* conecta las celdas de dos puertos a los componentes continuos y desconecta la correspondiente celda de conmutación.

Las fuentes controladas de cada *celda de dos puertos* representan los valores medios, calculados en un período de conmutación, de las corrientes y las tensiones en los transistores con diodo antiparalelo ($\overline{i}_{i1}$, $\overline{i}_{i2}$, $\overline{v}_{i1}$ y $\overline{v}_{i2}$). En la Figura 6.2 se muestra el esquema de la *celda de dos puertos* asociada a la celda de conmutación correspondiente a la i-ésima rama.

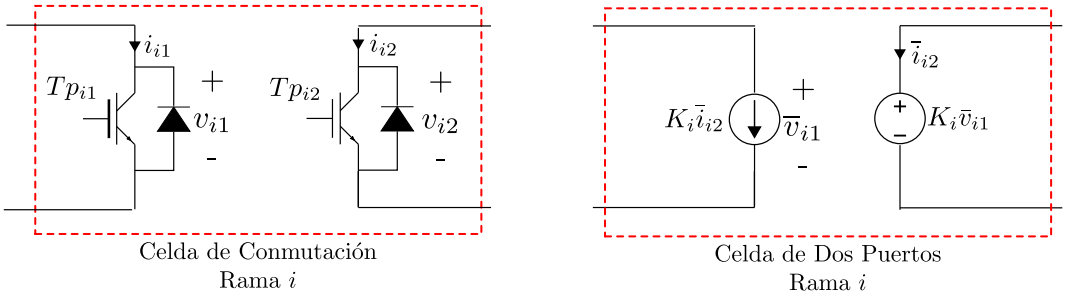


Figura 6.2: Celda de conmutación y su correspondiente *celda de dos puertos*.

Las ecuaciones que definen el comportamiento de estas fuentes controladas son:

$$\begin{aligned} \overline{v}_{i2}(t) &= K(d_{c_i}) \cdot \overline{v}_{i1}(t) \\ \bar{i}_{i1}(t) &= K(d_{c_i}) \cdot \bar{i}_{i2}(t) \end{aligned} \tag{6.1}$$

Como las fuentes controladas reemplazan a los $T_{p_{ij}}$, $\overline{v}_{ij}$ y $\bar{i}_{ij}$ representan la tensión y la corriente en las fuente de tensión y corriente controladas que reemplazan a los transistores $T_{p_{ij}}$ correspondientes.

El coeficiente de ganancia $K(d_{c_i})$ puede describirse mediante la siguiente ecuación:

$$K(d_{c_i}) = \frac{\overline{v}_{i2}}{\overline{v}_{i1}} = \frac{\bar{i}_{i1}}{\bar{i}_{i2}} = k_i \cdot \frac{1 - d_{c_i}}{d_{c_i}} \tag{6.2}$$

El modelo mixto de un DMSI PWM monofásico, requiere un par de ramas. Cada una de estas ramas se modela mediante una celda de conmutación y una *celda de dos puertos*. La celda de dos puertos de cada una de estas ramas posee su propio parámetro k_i que se estima cuando se utiliza el SWM. Además, las señales PWM que controlan cada una de estas ramas tienen la misma frecuencia de conmutación ($1/T_{\text{sw}}$) pero cada una de ellas posee su propio ciclo de trabajo (d_{c_1} y d_{c_2}).

6.2.3 Estrategia de conmutación

La selección de la celda correspondiente se realiza automáticamente mediante el algoritmo de conmutación incluido en el *selector de modo*.

Este algoritmo selecciona el modo adecuado según la evolución de ciertas variables (corrientes y tensiones). En el caso de los inversores PWM, el algoritmo debe detectar si la tensión y la corriente de salida del puente siguen una forma de onda en PSS o no.

El principio de funcionamiento de la estrategia de conmutación se puede entender a partir de dos situaciones claramente diferenciables. Por un lado, cuando se detecta un cambio en el régimen PSS y se está utilizando el AVM, se selecciona el SWM. Por otro lado, cuando

el SWM está activo y las variables permanecen en régimen PSS durante dos períodos, se selecciona el AVM.

Para detectar el régimen PSS, cada variable de salida (corriente y tensión) se compara con su correspondiente primer armónico. El algoritmo detecta un régimen PSS cuando ambas señales están lo suficientemente próximas a su primer armónico durante al menos dos períodos consecutivos. Para determinar si las señales están lo suficientemente cerca, se evalúa la siguiente ecuación:

$$s_x(t) = \frac{1 + x_F(t)^2}{1 + x(t)^2} \tag{6.3}$$

donde $x(t)$ representa la tensión $v(t)$ o la corriente $i(t)$ que se desea evaluar si se encuentra en PSS y $x_F(t)$ es el primer armónico de $x(t)$. Es importante notar que el denominador de esta ecuación no puede ser cero. La señal $x_F(t)$ se obtiene a partir de $x(t)$ mediante un filtro pasa banda de segundo orden con ganancia unitaria y error de fase nulo en la frecuencia central f_{out}:

$$F_x(s) = \frac{X_F(s)}{X(s)} = \left(\frac{1}{\pi f_{\text{out}}} \frac{s}{1 + \frac{s}{\pi f_{\text{out}}} + \frac{s^2}{(2\pi f_{\text{out}})^2}} \right) \tag{6.4}$$

La frecuencia fundamental de salida f_{out} del DMSIs PWM se estima mientras el SWM está seleccionado.

En principio, la Ecuación 6.3 permitiría detectar si $x(t)$ es una señal sinusoidal, ya que en ese caso se cumple $s_x(t) = 1$. Sin embargo, las señales $x(t)$ presentan una componente de ripple de alta frecuencia debido a la conmutación de los interruptores en el inversor. Esta componente también se manifiesta en $s_x(t)$, por lo que, aún cuando $x(t)$ sea igual a $x_F(t)$ con un ripple superpuesto, se verifica que $s_x(t) \neq 1$. Por este motivo, se utiliza una versión suavizada $y_x(t)$ de $s_x(t)$ (en adelante llamada señal de control) obtenida mediante un filtro pasa bajos de primer orden:

$$G_x(s) = \frac{Y_x(s)}{S_x(s)} = \frac{1}{1 + s\frac{k_f}{2\pi f_{\text{out}}}} \tag{6.5}$$

donde k_f determina la constante de tiempo del filtro. Un criterio para elegir su valor es seleccionar el mayor valor posible para mejorar la suavidad de $y_x(t)$, asegurando al mismo tiempo que la respuesta temporal del filtro t_r sea menor que $2T_{\text{out}}$. En consecuencia, k_f debe satisfacer:

$$3\frac{k_f}{2\pi\, f_{\text{out}}} \leq 2T_{\text{out}} \Rightarrow k_f \leq \frac{4}{3} \cdot \pi \tag{6.6}$$

Este resultado establece un límite superior para el valor de k_f, garantizando que el filtro suavice adecuadamente la señal sin comprometer la respuesta dinámica del sistema. En la práctica, seleccionar un valor próximo a este límite permite obtener una buena atenuación del ripple sin retrasar significativamente la evolución de $y_x(t)$ y en consecuencia, el tiempo en que se detectará que se esta en estado PSS.

Es importante destacar que si $x_F(t)$ coincide con $x(t)$, el valor medio de $y_x(t)$ es igual a 1. Dado que $x(t)$ no necesariamente es sinusoidal sino periódico, se cumple que $y_x(t) \approx 1$ cuando el sistema alcanza el régimen PSS.

La conmutación entre ambos modos puede explicarse a partir de la señal de salida del filtro. Si se denomina $y_v(t)$ a dicha salida al analizar si una tensión se encuentra o no en PSS, entonces el funcionamiento del algoritmo supervisor puede interpretarse a partir de las Figuras 6.3 y 6.4. En las mismas se muestra la evolución de $y_v(t)$ para distintos intervalos de tiempo. Estas figuras se obtuvieron por simulación tomando $x(t)$ como la tensión $v(t)$ de salida de un DMSI PWM.

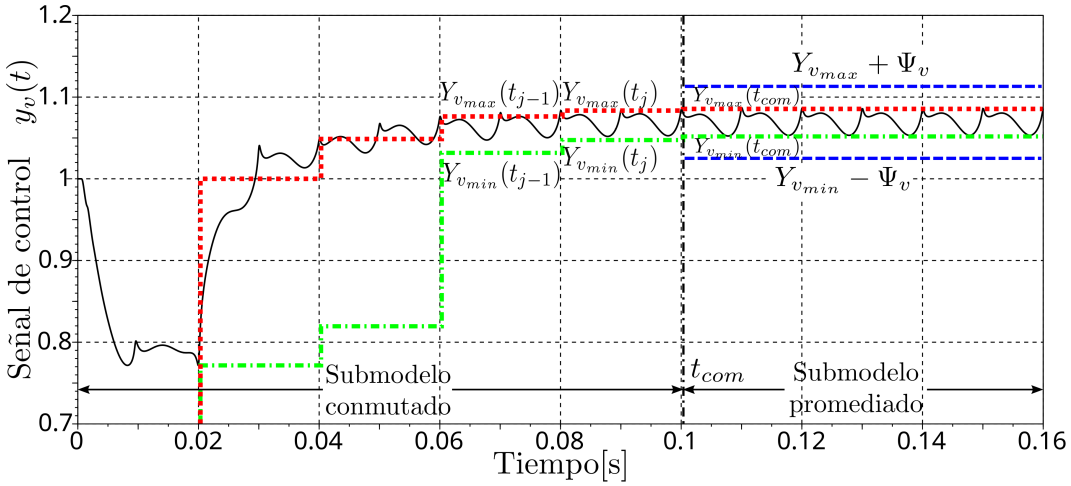


Figura 6.3: Detalle temporal de la señal de salida filtrada $y_v(t)$ que activa la conmutación, en un intervalo amplio.

Definiendo $Y_{v_{\text{min}}}$ y $Y_{v_{\text{max}}}$ a los valores mínimo y máximo de $y_v(t)$ en cada período T_{out} de la tensión de salida (Figuras 6.3 y 6.4), la detección de una situación transitoria o de régimen PSS puede realizarse del siguiente modo. Cuando $v(t)$ se aproxima a una onda PSS, estos límites permanecen aproximadamente constantes en períodos sucesivos. En cambio, durante un régimen transitorio, dichos límites varían entre ciclos consecutivos.

Los límites $Y_{v_{\text{min}}}$ y $Y_{v_{\text{max}}}$ son calculados por el algoritmo al final de cada período T_{out} mientras se usa el SWM y dado que dichos límites no cambian en un régimen PSS, no se recalculan mientras el AVM está activo. Mientras se está en AVM los límites $Y_{x_{\text{max}}}(t_{com})$ y $Y_{x_{\text{min}}}(t_{com})$ permanecen constantes con los valores calculados en el último período que se uso el SWM antes de pasar a AVM en el instante t_{com} como se muestra en las Figuras 6.3

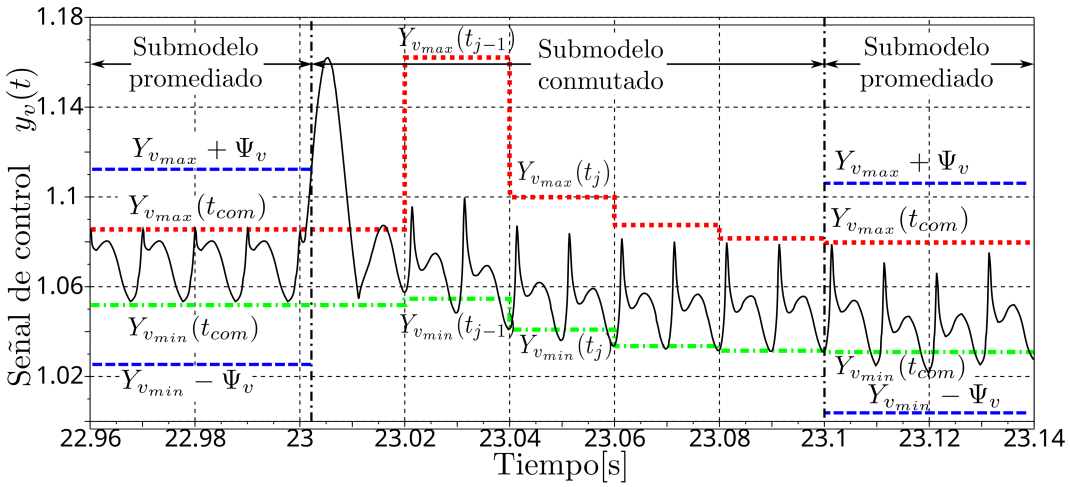


Figura 6.4: Detalle ampliado de la señal $y_v(t)$ en una región conmutada, donde se indica el umbral de decisión.

y 6.4. Estos límites almacenados se utilizan para detectar cualquier perturbación en el PSS.

Para comprender el funcionamiento de este algoritmo, puede analizarse un caso típico. La simulación comienza siempre utilizando el SWM, como se muestra en la Figura 6.3. Cuando la diferencia entre dos valores consecutivos de $Y_{v_{\text{min}}}$ y $Y_{v_{\text{max}}}$ es menor que un umbral previamente definido, ΔY_v, se detecta un régimen en PSS, y el algoritmo del *selector de modo* determina que puede utilizarse el AVM.

$$\begin{aligned} |Y_{v_{\text{max}}}(t_j) - Y_{v_{\text{max}}}(t_{j-1})| < \Delta Y_v \\ |Y_{v_{\text{min}}}(t_j) - Y_{v_{\text{min}}}(t_{j-1})| < \Delta Y_v \end{aligned} \tag{6.7}$$

Esta condición debe cumplirse para todas las variables cuya pertenencia al régimen en PSS se desea evaluar; es decir, en todos los límites $Y_{x_{\text{max}}}$ y $Y_{x_{\text{min}}}$ asociados a las señales de control $y_x(t)$.

La conmutación del SWM al AVM puede observarse en la Figura 6.3, en $t = 0{,}1$ s, y en la Figura 6.4, en $t = 23{,}1$ s. En ambas figuras se muestran los valores consecutivos de $Y_{v_{\text{max}}}$ y $Y_{x_{\text{min}}}$ mientras se utiliza el SWM.

Para evitar efectos transitorios durante la conmutación del SWM al AVM, el cambio efectivo entre ambos modos se realiza cuando la tensión de salida alcanza su valor medio. En ese instante, los valores de las variables de estado se fijan en sus respectivos valores promediados. Previo a la conmutación, la diferencia $\Delta x(t)$ entre los valores actuales de estado $x(t)$ y sus valores medios $\overline{x}(t)$ se almacena para ser utilizada en el próximo cambio del AVM al SWM.

Una vez seleccionado el AVM en t_{com}, se establecen los límites inferior y superior, $Y_{v_{\text{min}}} - \Psi_v$ y $Y_{v_{\text{max}}} + \Psi_v$, respectivamente, a partir de los valores $Y_{v_{\text{min}}}$ y $Y_{v_{\text{max}}}$ de las señales de control

$y_v(t)$, donde Ψ_v es un parámetro configurable por el usuario.

El SWM debe volver a seleccionarse cuando se detecta una señal no periódica. Esta situación ocurre cuando la señal $y_x(t)$ excede alguno de sus límites establecidos, es decir, cuando $y_x(t) > Y_{x_{\text{max}}} + \Psi_x$ o $y_x(t) < Y_{x_{\text{min}}} - \Psi_x$. En tal caso, el *selector de modo* activa la conmutación. La condición para cambiar del AVM al SWM se define según:

$$y_x(t) = \begin{cases} \text{Usar SWM}, & \text{si } y_x(t) < Y_{x_{\text{min}}} - \Psi_x \\ \text{Usar SWM}, & \text{si } y_x(t) > Y_{x_{\text{max}}} + \Psi_x \\ \text{sin cambios}, & \text{en otro caso} \end{cases} \tag{6.8}$$

La transición del AVM al SWM puede observarse en la Figura 6.4. En el instante $t = 23{,}001$ s, la señal $y_v(t)$ alcanza su límite superior, $Y_{v_{\text{max}}} + \Psi_v$, lo que indica la presencia de un transitorio. En ese momento, el algoritmo selecciona el SWM. Sin embargo, antes de la conmutación, las variables de estado $x(t)$ deben ajustarse adecuadamente para evitar transitorios, según:

$$x(t) \leftarrow \bar{x}(t) + \Delta x(t_{com}) \tag{6.9}$$

donde $\overline{x}(t)$ es el valor medio de $x(t)$, $\Delta x(t_{com}) = x(t_{com}) - \overline{x}(t_{com})$ y t_{com} es el instante de conmutación anterior de SWM a AVM.

La selección de los parámetros del modelo mixto (ΔY_x y Ψ_x) sigue un criterio análogo al de la tolerancia de error en simulaciones que emplean métodos de integración numérica de paso variable. Un valor pequeño de Ψ_x se traduce en un menor error transitorio ante cualquier perturbación en el DMSIs PWM, ya que controla la transición del modelo promediado (AVM) al modelo conmutado (SWM). De manera similar, el parámetro ΔY_x controla el cambio del modelo conmutado al modelo promediado. Reducir el valor de ΔY_x aumenta el tiempo de uso del modelo conmutado y disminuye el error final. La reducción de los valores de ΔY_x y/o Ψ_x mejora los detalles de la simulación, aunque incrementa el costo computacional.

Por lo tanto, la selección de estos parámetros debe hacerse de manera tal de lograr un equilibrio entre el tiempo de ejecución en la CPU (costo computacional) y el nivel de detalle alcanzado en la simulación.

Las operaciones del algoritmo de conmutación mencionadas anteriormente, tanto para la transición de SWM a AVM como de AVM a SWM, fueron implementadas en el lenguaje Modelica.

6.3 Implementación en Modelica

Para implementar la estrategia de modelado mixto para convertidores DMSI PWM, se construyó un bloque básico en Modelica denominado *full bridge* (ver Figura 6.5 a)). Este bloque modela un puente en H genérico, común a las distintas topologías de DMSI PWM.

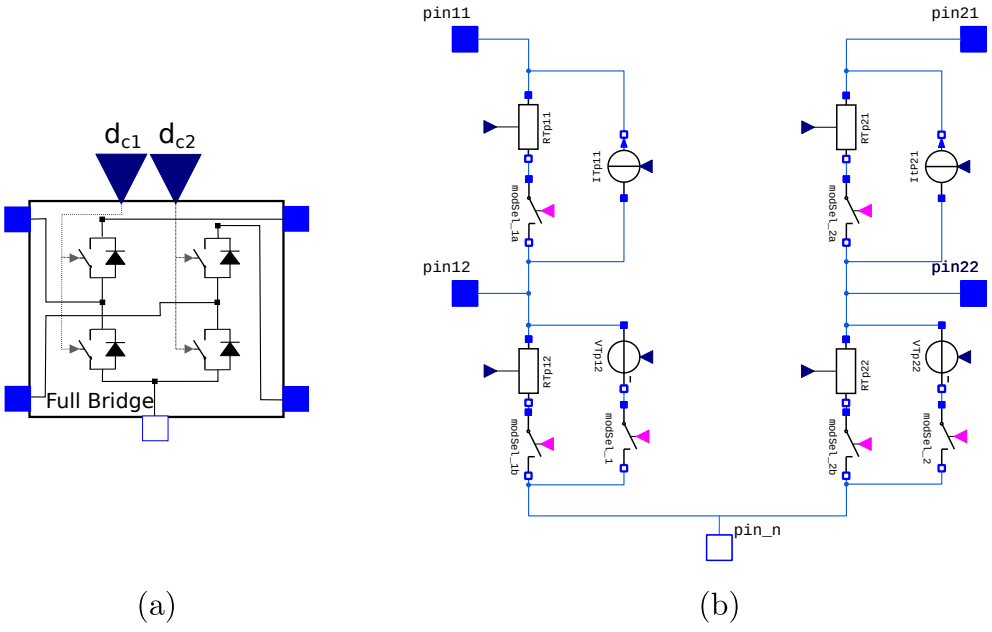


Figura 6.5: Modelo en Modelica del *Full Bridge*.

Como puede verse en la Figura 6.5 b), este bloque contiene exclusivamente el modelo de los componentes discontinuos de un inversor monofásico en puente completo, conformado por dos ramas de transistores con diodos en antiparalelo. El modelo incorpora tanto las *celdas conmutadas* como las *celdas de dos puertos*, junto con el algoritmo supervisor previamente descripto para la selección automática del modo de operación.

Cada *celda de conmutación* se implementa en Modelica mediante un par de resistencias controladas, $R_{Tp_{i1}}$ y $R_{Tp_{i2}}$, que conmutan su valor entre R_{on} y R_{off} según el estado de activación del transistor y el estado de conducción del diodo. De manera similar, cada *celda de dos puertos* se compone de una fuente controlada de tensión y una de corriente ($V_{Tp_{i2}}$ e $I_{Tp_{i1}}$). La selección entre las *celdas de conmutación* y las *celdas de dos puertos*, en función del modo de operación, se implementa mediante las llaves ideales `modeSel_i`, `modeSel_ia` y `modeSel_ib`. Este comportamiento se observa en el Algoritmo 1, donde puede verificarse que, mientras se utiliza el AVM, las resistencias controladas que modelan los componentes de conmutación no cambian de estado, y por lo tanto no se producen saltos en las EDOs que dependen de ellas.

Algorithm 1 Implementación del selector de celdas

▷ Este algoritmo activa las celdas de conmutación o las celdas de dos puertos según el modo seleccionado
si $modo == Promediado$
  para $i \leftarrow \{1, 2\}$ **hacer**
    modSel_i ← cerrado
    modSel_ia ← abierto
    modSel_ib ← abierto
    RTpi1 ← Roff
    RTpi2 ← Roff
  fin para
si no
  para $i \leftarrow \{1, 2\}$ **hacer**
    modSel_i ← abierto
    modSel_ia ← cerrado
    modSel_ib ← cerrado
    Manejo de Resistencias controladas RTp a cargo del PWM
  fin para
fin si

Algorithm 2 Cálculos adicionales en SWM.

▷ Este algoritmo se ejecuta al final de un período de conmutación mientras el SWM está seleccionado.
cuando $modo == Conmutado$ **y** finDelPeriodoDeConmutación **entonces**
  $t_j \leftarrow t$ ▷ Almacena el instante de tiempo al final del último período de conmutación T_{sw}
  $\bar{v}_{\text{out}} \leftarrow \frac{1}{T} \cdot \int_{t_j-T}^{t_j} v_{\text{out}}(\tau)\mathrm{d}\tau$ ▷ Calcula el valor medio de v_{out} sobre el último período de conmutación
  para $i \leftarrow \{1, 2\}$ **hacer**
  $\bar{v}_{i2} \leftarrow \frac{1}{T} \cdot \int_{t_j-T}^{t_j} v_{i2}(\tau)\mathrm{d}\tau$ ▷ Calcula los valores medios de v_{i1} y v_{i2} sobre el último período de conmutación
  $\bar{v}_{i1} \leftarrow \frac{1}{T} \cdot \int_{t_j-T}^{t_j} v_{i1}(\tau)\mathrm{d}\tau$
  $k_i \leftarrow \frac{\bar{v}_{i2}}{\bar{v}_{i1}} \cdot \frac{1-d_i}{d_i}$ ▷ Calcula el parámetro k_i de las fuentes controladas en el instante actual
  fin para
  para $x \leftarrow \{1, .., nEstados\}$ **hacer**
  $\bar{x}_i \leftarrow \frac{1}{T} \cdot \int_{t_k-T}^{t_k} x_i(\tau)\mathrm{d}\tau$ ▷ Calcula el valor medio de cada variable de estado
  fin para
fin cuando

Para completar la implementación del bloque *full bridge*, es necesario incorporar el *selector de modo*. Su implementación requiere, en primer lugar, la estimación de una serie de variables durante el funcionamiento en modo SWM. Estas variables son necesarias para aplicar la estrategia de conmutación, e incluyen el valor medio de la tensión de salida, las variables de estado y las tensiones en los transistores. Dichos cálculos se realizan en el Algoritmo 2.

El funcionamiento del *selector de modo* se encuentra formalizado en los Algoritmos 3–6, los cuales se presentan en formato de pseudocódigo, dado que el código completo en Modelica es considerablemente más extenso y no aporta información adicional relevante para la comprensión conceptual.

Algorithm 3 Condición para conmutar de SWM a AVM.

▷ Este algoritmo se ejecuta al final de un período de salida T_{out} en SWM.
cuando $modo == Conmutado$ **y** finDelPeriodoDetensiónDeSalida **entonces**
  para $x \leftarrow \{1, .., nEstados\}$ **hacer**
    $Y^*_{x_{max}} \leftarrow Y_{x_{\text{max}}}$
    $Y^*_{x_{min}} \leftarrow Y_{x_{\text{min}}}$
    $Y_{x_{\text{max}}} \leftarrow \text{máx}\{y_x(t); [t - T_{\text{out}}, t]\}$
    $Y_{x_{\text{min}}} \leftarrow \text{mín}\{y_x(t); [t - T_{\text{out}}, t]\}$
  fin para
 fin cuando
 cuando $\|Y^*_{x_{max}} - Y_{x_{\text{max}}}\| < \Delta Y_x$ **y** $\|Y^*_{x_{min}} - Y_{x_{\text{min}}}\| < \Delta Y_x$ **entonces**
  $habilitarPromediado \leftarrow Verdadero$ ▷ Bandera que habilita AVM
  $habilitarConmutado \leftarrow Falso$
 fin cuando
fin cuando

Algorithm 4 Conmutación de SWM a AVM.

▷ Cuando la bandera "habilitarPromediado.está en "verdaderoz la señal de tensión de salida cruza su valor medio, este algoritmo realiza la conmutación de SWM a AVM.
cuando $habilitarPromediado$ **y** $v_{\text{out}} == \bar{v}_{\text{out}}$ **entonces**
 $t_{com} \leftarrow t$ ▷ Almacena el tiempo de conmutación de SWM a AVM
 para $x \leftarrow \{1, .., nEstados\}$ **hacer**
  $\Delta x(t_{com}) \leftarrow x(t_{com}) - \bar{x}(t_{com})$
  $x(t) \leftarrow \bar{x}(t)$ ▷ Reinicializa las variables de estado a sus valores medios
 fin para
 $modo \leftarrow Promediado$ ▷ Se selecciona el AVM
fin cuando

Algorithm 5 Condición de transición de AVM a SWM.

▷ Detecta cuándo es necesario cambiar al SWM
cuando $modo == Promediado$ **y** $(\exists x : y_x(t) \geq Y_{x_{\text{max}}} + \Psi_x \quad \textbf{o} \quad y_x(t) \leq Y_{x_{\text{min}}} - \Psi_x)$ **entonces**
  $habilitarPromediado \leftarrow Falso$
  $habilitarConmutado \leftarrow Verdadero$ ▷ Se debe utilizar el SWM
fin cuando

Algorithm 6 Conmutación de AVM a SWM.

▷ Realiza la conmutación efectiva de AVM a SWM
cuando $modo == Promediado$ **y** $habilitarConmutado$ **entonces**
  $t_r \leftarrow t_{com} + T \cdot \text{floor}\left(\frac{t - t_{com}}{T} + 1\right)$
fin cuando
cuando $t == t_r$ **entonces**
  para $x \leftarrow \{1, .., nEstados\}$
    $x(t) \leftarrow \bar{x}(t) + \Delta x(t_{com})$ ▷ Reinicialización de las variables de estado
  fin para
  $modo \leftarrow Conmutado$ ▷ Se selecciona el SWM
fin cuando

El Algoritmo 3 evalúa la condición necesaria para conmutar del SWM al AVM. Para ello, verifica si las señales de control se encuentran en un PSS a partir de la comparación entre los valores máximos y mínimos consecutivos de las variables monitoreadas. Una vez cumplida esta condición, el Algoritmo 4 realiza la transición efectiva al AVM. En esta etapa, se almacenan las diferencias $\Delta x(t)$ entre los valores actuales y sus promedios, y se fijan las variables de estado en sus valores medios para evitar transitorios.

Por su parte, el Algoritmo 5 determina la condición para regresar al modelo conmutado. Esto se logra verificando si alguna señal promediada $y_x(t)$ excede los límites definidos por $Y_{x_{\text{min}}} - \Psi_x$ o $Y_{x_{\text{max}}} + \Psi_x$. Este hecho indica la presencia de un transitorio.

Finalmente, el Algoritmo 6 ejecuta la conmutación de AVM a SWM. Para evitar transitorios en esta transición, se utilizan los valores de $\Delta x(t)$ guardados previamente, ajustando los nuevos valores de las variables de estado respecto a su valor medio.

Esta arquitectura modular, basada en la implementación del bloque *full bridge* con selección automática de modo, permite su integración con componentes estándar de circuito, como inductores, capacitores y fuentes. De este modo, es posible construir distintas topologías de DMSIs PWM. En la Figura 6.6 se presenta un ejemplo donde como puede verse, al conectar directamente elementos continuos de la librería estándar de componentes eléctricos de Modelica con el bloque *full bridge* desarrollado se puede construir una DMSI PWM

(en este caso, de tipo boost).

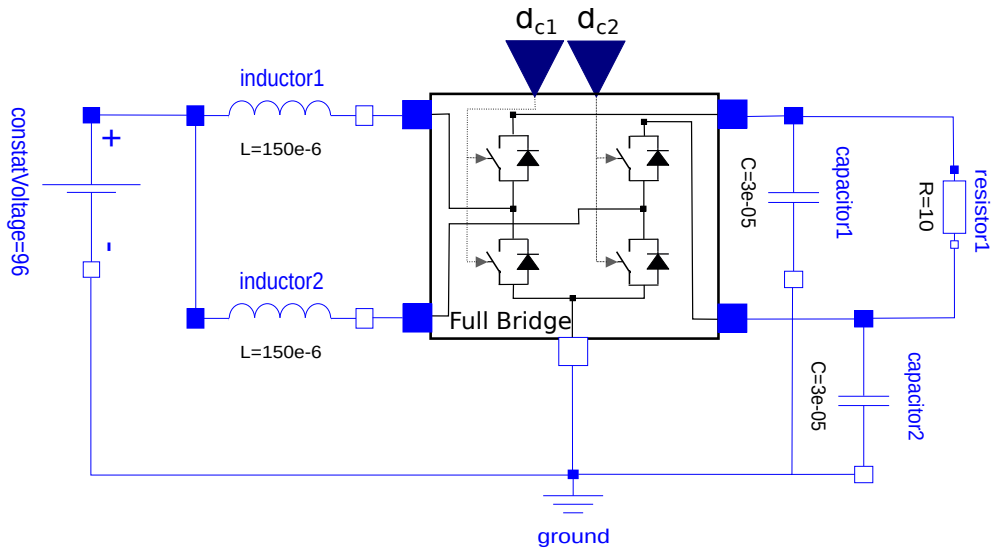


Figura 6.6: Modelo mixto del DMSI boost en Modelica.

Para completar el modelo mixto del inversor, es necesario establecer una correspondencia biunívoca entre las variables de estado utilizadas por el algoritmo supervisor y las variables físicas del circuito del inversor. Esta asociación resulta fundamental para, por ejemplo, asignar a las variables de estado sus respectivos valores medios cuando el AVM es seleccionado por el *selector de modo*, lo que garantiza la continuidad de las trayectorias durante las transiciones entre modos.

Tal como se explicó previamente, la implementación del modelo mixto requiere que el usuario configure ciertos parámetros adicionales, además de los propios del inversor, como la frecuencia de conmutación PWM, los valores de los componentes del circuito y la variación máxima permitida del ciclo de trabajo. Entre los parámetros adicionales se incluyen la constante de tiempo del filtro pasa-bajos k_f, así como los límites de histéresis Ψ_x y ΔY_x, asociados a la tensión y la corriente de salida.

Además del modelo mixto del DMSI boost, se implementó un modelo conmutado del mismo inversor en Modelica, utilizando el software *Wolfram System Modeler*, que sirve como referencia para la validación y la comparación sistemática entre ambas implementaciones.

6.4 Resultados de simulación

Para evaluar el desempeño de la estrategia de modelado mixto propuesta, se presentan dos ejemplos de un DMSI boost operando primero en lazo abierto y luego en lazo cerrado. Ambos casos tienen como objetivo realizar una simulación a largo plazo de un DMSI PWM

boost, con eventos externos esporádicos.

Las simulaciones se realizaron en Wolfram System Modeler versión 4.2.1, ejecutándose en un equipo con procesador Intel(R) Core(TM) i3-2350M CPU a 2.30 GHz. Los ejemplos fueron simulados utilizando el método DASSL, configurado con paso de integración variable y tolerancias de error relativo y absoluto de 10^{-6}, ya que este fue el método más adecuado para las simulaciones.

6.4.1 Simulaciones en lazo abierto

En este ejemplo, se realizaron simulaciones de 50 segundos tanto del modelo mixto del DMSI boost mostrado en la Figura 6.6, y de una versión puramente conmutada del mismo inversor. Los parámetros utilizados para la configuración de ambos modelos se detallan en la Tabla 6.1 [31].

Parámetro	**Valor**
Tensión de entrada de corriente continua	$V_{\text{in}} = 96\,V$
Inductores	$L_{11} = L_{12} = 750\,\mu H$
Capacitores	$C_{11} = C_{12} = 10\,\mu F$
Frecuencia de conmutación PWM	$f_{sw} = 20\,kHz$
Frecuencia de modulación	$f_{\text{out}} = 50\,Hz$
Constante del filtro pasa bajos	$k_f = 4$
Parámetros de histéresis	$\Psi_x = 0{,}025$, $\Delta Y_x = 0{,}005$

Tabla 6.1: Parámetros del modelo mixto de DMSI PWM boost a lazo abierto.

El parámetro k_f se seleccionó de acuerdo con a la Ecuación 6.6. El valor de Ψ_x se definió de modo tal que el algoritmo supervisor del selector de modo resultara lo suficientemente sensible para detectar todas las perturbaciones externas y capturar el inicio de cada transitorio. De manera análoga, ΔY_x se estableció con un valor lo suficientemente pequeño como para asegurar que el sistema se encontrara en régimen PSS antes de pasar al AVM.

Durante la simulación se introdujeron dos tipos de perturbaciones. La primera consistió en un cambio en la carga de salida, tal como se indica en las Ecuaciones (6.10):

$$R_{\text{LOAD}} = \begin{cases} 60\,\Omega, & \text{si } t < 23\,\text{s} \\ 15\,\Omega, & \text{si } t \geq 23\,\text{s} \end{cases} \tag{6.10}$$

Posteriormente, se modificó la amplitud del ciclo de trabajo, según lo definido en las Ecuaciones (6.11)–(6.12):

$$d_{c_1}(t) = \begin{cases} 0{,}5 + 0{,}25 \cdot \sin(2\pi f_{\text{out}} t) & \text{si } t < 40s \\ 0{,}5 - 0{,}3 \cdot \sin(2\pi f_{\text{out}} t) & \text{si } t \geq 40s \end{cases} \tag{6.11}$$

$$d_{c_2}(t) = \begin{cases} 0{,}5 - 0{,}25 \cdot \sin(2\pi f_{\text{out}} t) & \text{si } t < 40s \\ 0{,}5 + 0{,}3 \cdot \sin(2\pi f_{\text{out}} t) & \text{si } t \geq 40s \end{cases} \tag{6.12}$$

Las Figuras 6.7 y 6.8 muestran la evolución de la tensión de salida del modelo mixto, ilustrando la conmutación entre los modos SWM y AVM. Para validar el modelo mixto frente al comportamiento real del sistema, también se incluyen los resultados de simulación obtenidos con un modelo puramente conmutado. Como se observa en la Figura 6.7, el sistema inicia la simulación utilizando el SWM, y en el instante $t = 0{,}1s$, se produce la conmutación al AVM. Este cambio de modo puede interpretarse a partir de la Figura 6.3. En esta figura se ve la señal de control obtenida al simular este ejemplo. En la misma se verifica que la diferencia entre los valores consecutivos de $Y_{v_{\text{max}}}$ y $Y_{v_{\text{min}}}$ resulta menor que el umbral ΔY_v y por lo tanto se debe usar el AVM.

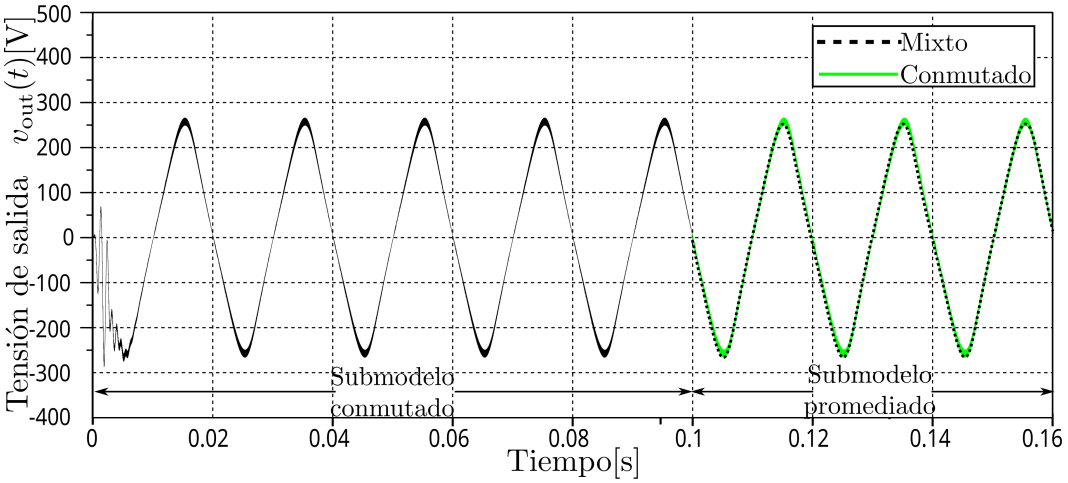


Figura 6.7: Señal de tensión de salida del DMSI boost en lazo abierto (transitorio inicial).

Luego, la Figura 6.8 muestra que, cuando la carga de salida cambia en $t = 23$ s, el modelo mixto conmuta al SWM a partir de $t = 23{,}001$ s. La detección y selección del SWM en ese instante puede explicarse a partir de la Figura 6.4. En la misma se ve que la señal $y_v(t)$ cruza el límite superior, $Y_{v_{\text{max}}} + \Psi_v$, en $t = 23{,}001$ s. Como consecuencia, el algoritmo supervisor hace que se seleccione nuevamente el SWM para capturar los transitorios y reflejar correctamente el cambio en el *ripple*. Puede observarse que la variación en la amplitud del *ripple*, evidenciada en la Figura 6.8, no podría haber sido detectada por un modelo puramente promediado. Luego, mediante un análisis similar a lo explicado para $t = 0{,}1s$, el algoritmo supervisor selecciona nuevamente el AVM en $t = 23{,}1s$ ya que detecta nueva-

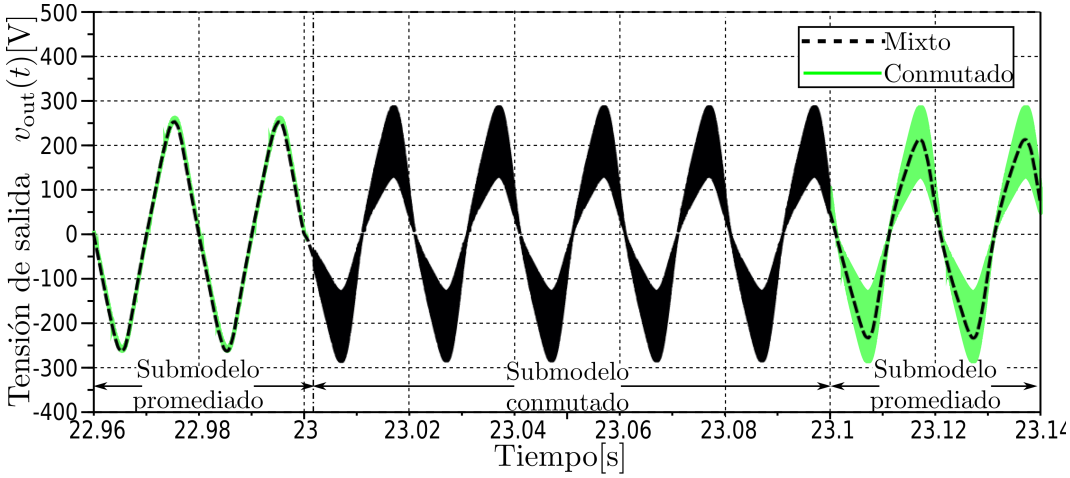


Figura 6.8: Señal de tensión de salida del DMSI boost en lazo abierto (cambio de carga en $t = 23s$).

mente un PSS.

En $t = 40s$, se produce el cambio en las amplitudes de d_{c_1} y d_{c_2} provocando una variación en la amplitud de la tensión de salida puede observarse en la Figura 6.9. Este cambio es detectado en $t = 40{,}0001s$ por el algoritmo supervisor del selector de modo y se selecciona el SWM. La detección del transitorio y la posterior selección del SWM se realizaron de forma análoga a lo explicado para el cambio de AVM a SWM en $23{,}001$s. Finalmente, con base en la evolución de $Y_{v_{\max}}$ y $Y_{v_{\min}}$, el algoritmo supervisor vuelve a seleccionar el AVM en $t = 40{,}136s$ siguiendo el mismo criterio explicado anteriormente para $t = 0{,}1s$.

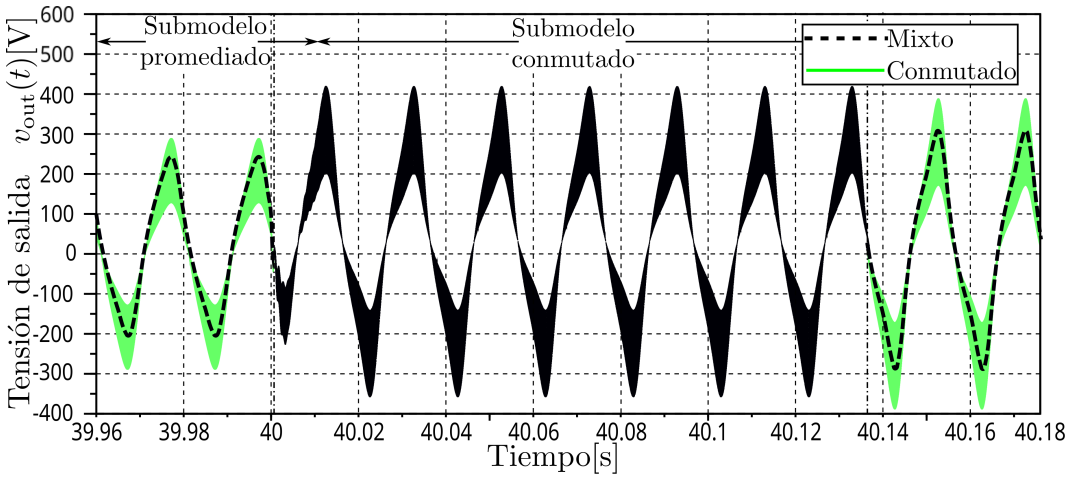


Figura 6.9: Señal de tensión de salida del DMSI boost en lazo abierto (cambio en d_c en $t = 40s$).

Como se muestra en la Tabla 6.2, el modelo mixto es aproximadamente 48 veces más rápido que el modelo puramente conmutado, sin pérdida de información en los resultados

de la simulación. La mejora en el tiempo de simulación se puede explicar a partir de la relación entre la cantidad de eventos en cada modelo ($34222104/55571 \approx 62$). La diferencia entre la relación de eventos (62) y la mejora en el tiempo de ejecución (48) se atribuye a los cálculos adicionales requeridos por el algoritmo supervisor.

Modelo	Número de eventos	Tiempo de CPU $[s]$
Mixto	55571	$68{,}9$
Conmutado	3422104	$3309{,}7$

Tabla 6.2: Comparación de costos computacionales simulación en lazo abierto.

El modelo puramente conmutado debe realizar tres detecciones de eventos por período PWM (conmutaciones de los interruptores) cuando opera en modo de conducción continua, lo que resulta en un total de aproximadamente $3\,000\,000$ de eventos $(3 \cdot 50\,\text{s} \cdot 20\,\text{kHz})$. La diferencia entre este valor estimado y el número de eventos reportado en la Tabla 6.2 puede atribuirse al funcionamiento en modo de conducción discontinua durante el transitorio inicial. En esta condición, se realizan cuatro detecciones de eventos por período PWM en lugar de tres.

En el caso del modelo mixto, el SWM se utiliza durante $0{,}32s$, lo que implica la generación de aproximadamente $19\,200$ eventos $(3 \cdot 0{,}32\,\text{s} \cdot 20\,\text{kHz})$. Los eventos adicionales registrados, más allá de los listados en la Tabla 6.2, son necesarios para la ejecución del algoritmo supervisor.

6.4.2 Simulación en lazo cerrado

Para analizar el desempeño del modelo mixto integrado dentro de un sistema más complejo se construyó un modelo de un DMSI PWM boost con un control de la tensión de salida que permite obtener una señal más senoidal y estable frente a perturbaciones. El control se implementó mediante dos lazos anidados, un lazo de corriente y un lazo de tensión en cada uno de los convertidores que conforman el DMSI PWM boost. La Figura 6.10 muestra el esquema del inversor con los lazos de control. Tanto la estrategia de control como los parámetros del inversor y de los controladores se tomaron del artículo [80].

En la Tabla 6.3 presenta los parámetros del DMSI PWM boost.

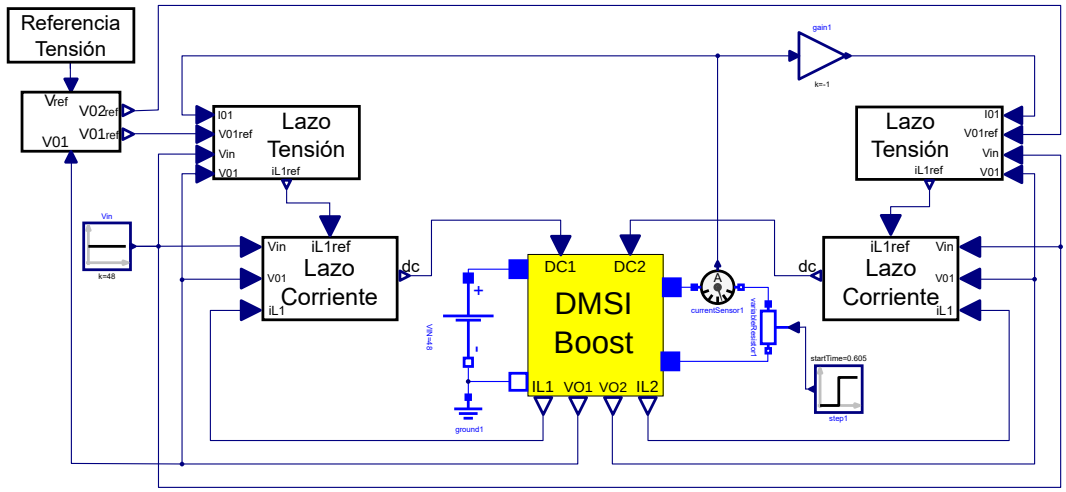


Figura 6.10: Esquema de un DMSI boost con control de regulación de doble lazo.

Parámetro	**Valor**
Tensión de entrada de corriente continua	$V_{in} = 48\,V$
Inductores	$L_{11} = L_{12} = 150\,\mu H$
Capacitores	$C_{11} = C_{12} = 30\,\mu F$
Frecuencia de conmutación PWM	$f_{sw} = 20\,kHz$
Frecuencia de salida	$f_{out} = 50\,Hz$
Constante del filtro pasa bajos	$k_f = 4$
Parámetros de histéresis	$\Psi_v = 0{,}025$, $\Psi_i = 0{,}01$, $\Delta Y_x = \Delta Y_i = 0{,}005$

Tabla 6.3: Parámetros del modelo DMSI PWM boost a lazo cerrado.

El diagrama en bloques del lazo interno de control de la corriente en el inductor se muestra en la Figura 6.11, y el diagrama en bloques del lazo externo de la tensión de salida se muestra en la Figura 6.12.

El control de corriente en el lazo interno calcula los ciclos de trabajo d_{c_1} y d_{c_2} a partir de las referencias de corriente $i_{L1_{\text{ref}}}$ e $i_{L2_{\text{ref}}}$, las cuales son generadas por el control de tensión en el lazo externo, junto con las corrientes en los inductores i_{L1} e i_{L2} del DMSI boost.

Por su parte, el control de tensión en el lazo externo calcula las referencias de corriente $i_{L1_{\text{ref}}}$ e $i_{L2_{\text{ref}}}$ en función de las referencias de tensión $v_{\text{out}_{1\text{ref}}}$ y $v_{\text{out}_{2\text{ref}}}$, y de las tensiones de salida v_{out_1} y v_{out_2} de cada convertidor boost.

En este ejemplo, las referencias de tensión $v_{\text{out}_{1\text{ref}}}$ y $v_{\text{out}_{2\text{ref}}}$ no deben establecerse de forma independiente, ya que esto conlleva diversas desventajas. Entre ellas, la tensión de salida v_{out} no estaría directamente regulada y la capacidad del sistema para rechazar perturbaciones externas se vería reducida. Por este motivo, se define una única referencia de tensión independiente para uno de los convertidores boost (v_{out_1}), mientras que la referencia

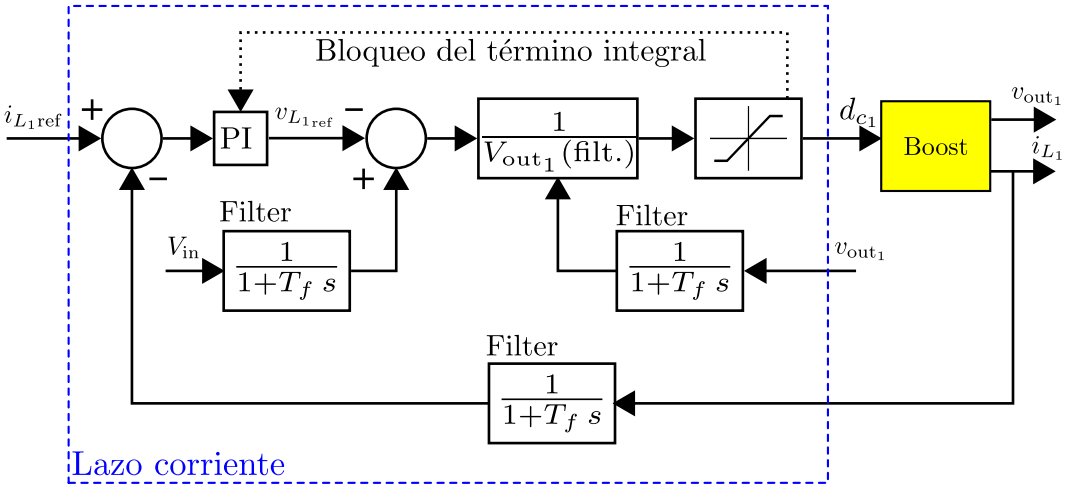


Figura 6.11: Esquema del control de corriente del inductor en lazo interno.

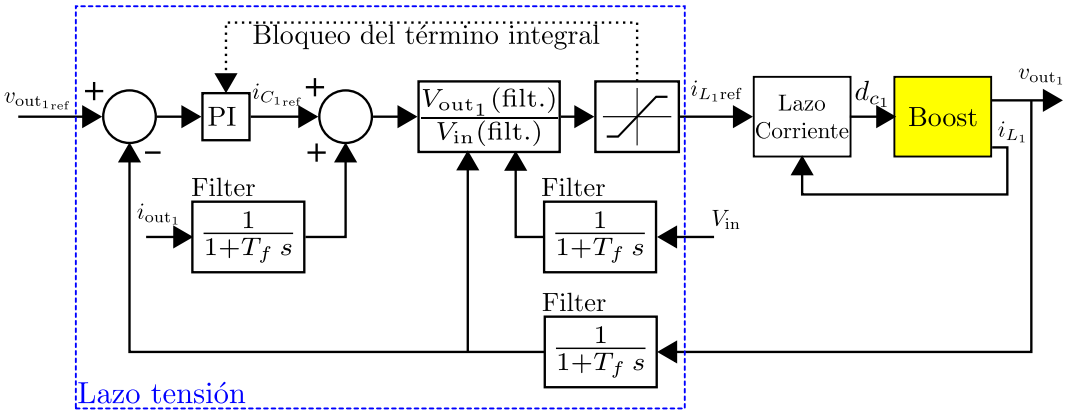


Figura 6.12: Esquema del control de tensión de salida en lazo externo.

restante (v_{out_2}) se ajusta para controlar directamente la tensión de salida del inversor v_{out}. De este modo, las referencias de tensión utilizadas son:

$$\begin{aligned} v_{\text{out}_{\text{ref}}} &= \sqrt{2}\,V_{\text{ref}}\sin(2\pi f t) \\ v_{\text{out}_{1\,\text{ref}}} &= V_{\text{DC}} + \frac{v_{\text{out}_{\text{ref}}}}{2} = V_{\text{DC}} + \frac{V_{\text{ref}}\sin(2\pi f t)}{\sqrt{2}} \\ v_{\text{out}_{2\text{ref}}} &= v_{\text{out}_1} - v_{\text{out}_{\text{ref}}} = v_{\text{out}_1} - \sqrt{2}\,V_{\text{ref}}\sin(2\pi f t) \end{aligned} \tag{6.13}$$

donde $v_{\text{out}_{\text{ref}}}$ es la referencia de tensión de salida del DMSI boost, $f_{\text{out}} = 50\,\text{Hz}$, V_{ref} es la tensión eficaz de referencia de salida, y $V_{\text{DC}} = 226\,\text{V}$ representa el *offset* (desplazamiento) de la tensión de salida de cada convertidor boost.

En la Tabla 6.4 se muestran los parámetros de los controladores utilizados.

Lazo de tensión	
Constante de filtro	$T_f = 2{,}5 \times 10^{-5}$ s
Ganancia proporcional	$k_{p,v} = 5{,}9 \times 10^{-2}$
Ganancia integral	$k_{i,v} = 4{,}99 \times 10^{-4}$
Lazo de corriente	
Constante de filtro	$T_f = 2{,}5 \times 10^{-5}$ s
Ganancia proporcional	$k_{p,i} = 3{,}529$
Ganancia integral	$k_{i,i} = 8{,}44 \times 10^{-5}$
Saturación inferior	$lowSat = 0{,}05$
Saturación superior	$uppSat = 0{,}95$

Tabla 6.4: Parámetros de los controladores del sistema: lazo de tensión y lazo de corriente.

Para analizar el comportamiento de la estrategia de modelado mixto, durante la simulación de este ejemplo en primer lugar se introdujo un cambio en la carga de salida:

$$R_{\text{LOAD}} = \begin{cases} 32{,}3\,\Omega & \text{si } t < 22{,}205\,\text{s} \\ 64{,}6\,\Omega & \text{si } t \geq 22{,}205\,\text{s} \end{cases} \tag{6.14}$$

Luego, se modificó en otro instante la referencia de amplitud de la tensión de salida:

$$V_{\text{ref}} = \begin{cases} 220\,\text{V} & \text{si } t < 40{,}205\,\text{s} \\ 180\,\text{V} & \text{si } t \geq 40{,}205\,\text{s} \end{cases} \tag{6.15}$$

La simulación comienza utilizando el SWM. Como en el ejemplo anterior, cuando el sistema alcanza un régimen de estado periódico estable (PSS) en $t = 0{,}0905$ s, se selecciona el modelo promediado.

Luego, en $t = 22{,}205$ s, la carga cambia y genera una perturbación en el régimen del estado. En $t = 22{,}2065$ s el algoritmo supervisor detecta automáticamente este régimen transitorio y selecciona el SWM. Este comportamiento se explica a partir de las Figuras 6.13 y 6.14. Dado que la tensión de salida está regulada, no se observan cambios significativos en la tensión de salida del DMSI boost v_{out}. En consecuencia, como se muestra en la Figura 6.13, la señal de control $y_v(t)$ permanece prácticamente constante y la detección no se realiza por tensión.

Sin embargo, la Figura 6.15 muestra que el cambio en la carga genera una variación considerable en la corriente de salida del puente completo, $i_{b_1}(t)$. Este cambio se refleja en la señal de control $y_i(t)$, como se observa en la Figura 6.14. En esta gráfica puede verse que en el instante $t = 22{,}2065$ s la señal de control $y_i(t)$ cruza el límite $Y_{i_{\text{max}}} + \Psi_i$ por lo que se detecta el régimen transitorio.

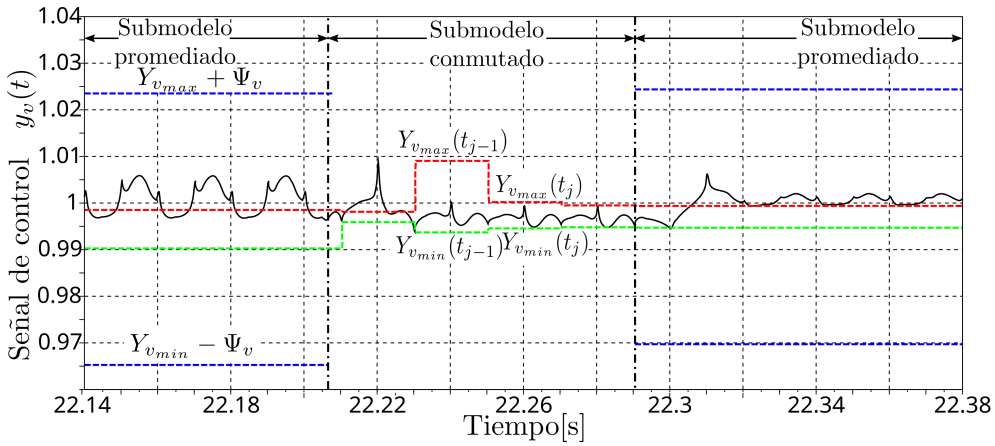


Figura 6.13: Evolución de la señal de control de conmutación $y_v(t)$ cuando cambia R_{Load}.

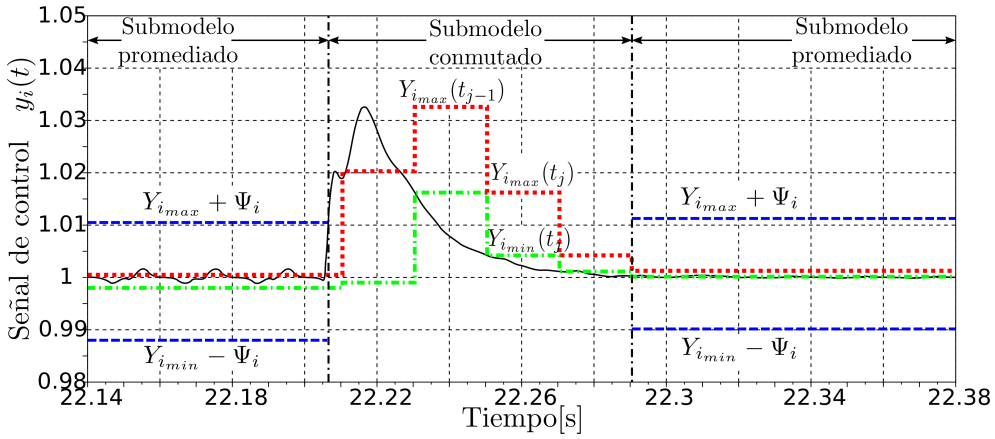


Figura 6.14: Evolución de la señal de control de conmutación $y_i(t)$ cuando cambia R_{Load}.

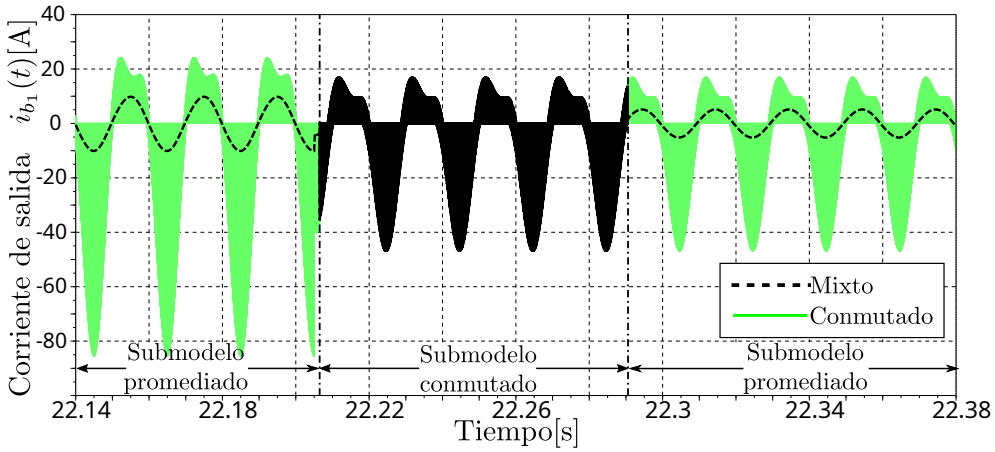


Figura 6.15: Comparación de la corriente de salida $i_{b_1}(t)$ entre los modelos de puente completo *mixto* y *puramente conmutado* (Cambio de carga).

Como se puede observar en las Figuras 6.13 y 6.14, el algoritmo supervisor detecta un nuevo PSS en el instante $t = 22{,}2905\,\text{s}$ ya que los valores consecutivos de $Y_{v_{\text{max}}}$, $Y_{v_{\text{min}}}$, $Y_{i_{\text{max}}}$ y $Y_{i_{\text{min}}}$ difieren en menos del umbral predefinido permitiendo así la selección del AVM.

Cuando la referencia de tensión de salida cambia en $t = 40{,}205\,\text{s}$, se producen variaciones significativas en la corriente i_{b_1} y la tensión v_{out} (Figura 6.16). Estos cambios también se reflejan en las señales $y_v(t)$ y $y_i(t)$, pero el algoritmo supervisor detecta primero el transitorio en $y_v(t)$, en $t = 40{,}20845\,\text{s}$. Finalmente, en $t = 40{,}3105\,\text{s}$, se identifica un nuevo régimen PSS y se selecciona nuevamente el AVM.

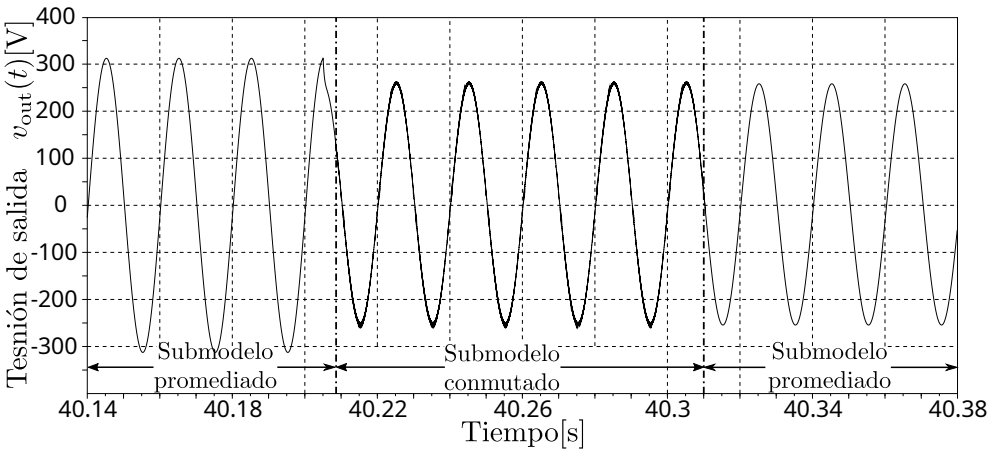


Figura 6.16: Tensión de salida $v_{\text{out}}(t)$ del modelo mixto de puente completo (Cambio en la referencia de tensión de salida).

Las Figuras 6.15 y 6.16 muestran aún frente a perturbaciones de distinto tipo que generan no solo cambios en la amplitud y forma de los valores medios de las señales observadas sino también en la amplitud de ripple, el modelo mixto preserva toda la información proporcionada por el modelo conmutado.

Para evaluar el rendimiento en tiempo de CPU de la estrategia de modelado mixto, se construyeron tanto un modelo puramente conmutado como un modelo mixto del sistema, y luego se compararon los tiempos de simulación obtenidos.

El tiempo de CPU requerido para simular un período de $t_{\text{sim}} = 50\,\text{s}$ utilizando ambos modelos se resume en la Tabla 6.5, la cual también indica el número de eventos ejecutados en cada simulación.

Modelo	Número de Eventos	Tiempo de CPU $[s]$
Mixto	27919	57,1
Conmutado	3885213	4983,7

Tabla 6.5: Comparación de costos computacionales simulación lazo cerrado.

Se puede observar que el modelo mixto es 87 veces más rápido que el modelo pura-

mente conmutado, mientras que la relación del número de eventos es 139. La mejora en el tiempo de CPU puede atribuirse al menor número de eventos detectados por el modelo mixto. La diferencia entre la razón de tiempos de CPU y la razón del número de eventos se explica por el costo computacional adicional asociado al algoritmo supervisor.

En cuanto al error, se compararon los resultados de simulación obtenidos mediante la estrategia de modelado mixto se compararon con los resultados de simulación obtenidos a partir de modelos puramente conmutados o puramente promediados , dependiendo del modo de operación utilizado por la estrategia mixta. Para el cálculo de las señales de referencia utilizadas para calcular el error se usó el solver DASSL con una tolerancia de error relativa de 10^{-6}. En el cálculo del error no se detectaron diferencias cuando se utiliza el AVM, en comparación con el modelo puramente promediado. Sin embargo, cuando se selecciona el SWM, aparece un error transitorio inicial. La Figura 6.17 ilustra el error en $i_{b_1}(t)$ luego de un cambio en la referencia de la carga de salida, al seleccionarse el SWM. Se observa que el error inicial es tres órdenes de magnitud inferior a la amplitud del ripple.

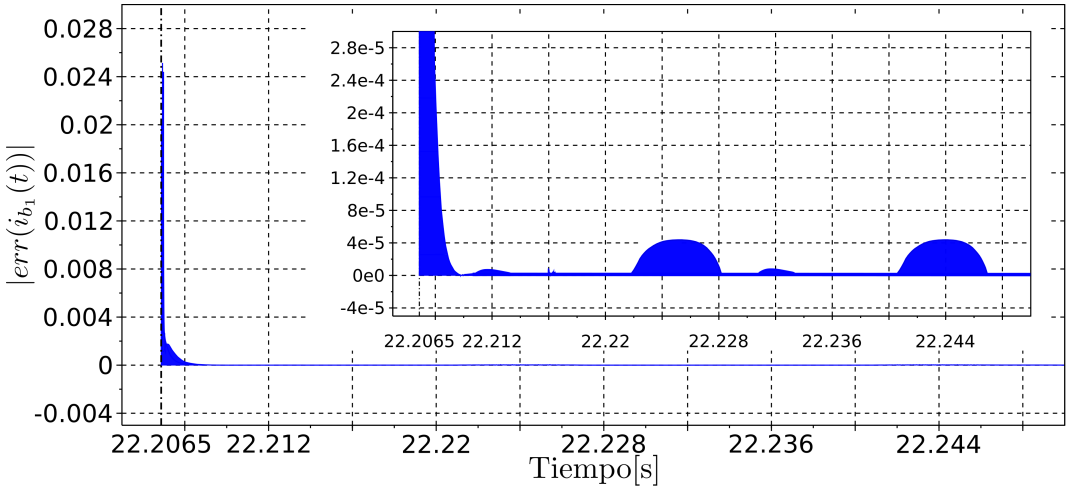


Figura 6.17: Error en la corriente $i_{b_1}(t)$.

Además, la Figura 6.18 muestra el error cuadrático medio de $i_{b_1}(t)$ para diferentes valores de Ψ_x. Se observa que, a medida que disminuye el valor de Ψ_x, el algoritmo supervisor detecta con mayor precisión el instante exacto en que ocurre una nueva situación transitoria, lo que reduce el error de simulación. Esta relación pone de manifiesto la influencia directa de la magnitud de Ψ_x sobre el error, tal como se representa en dicha figura.

Además, se evalúan las ventajas en el uso de CPU del modelo mixto en función de cuán frecuentes son los cambios en las entradas del sistema (tales como variaciones en la carga y en las referencias). El objetivo es identificar los casos en los que el modelo mixto presenta un desempeño superior al modelo puramente conmutado, y proporcionar recomendaciones sobre su aplicación en función de dichas condiciones.

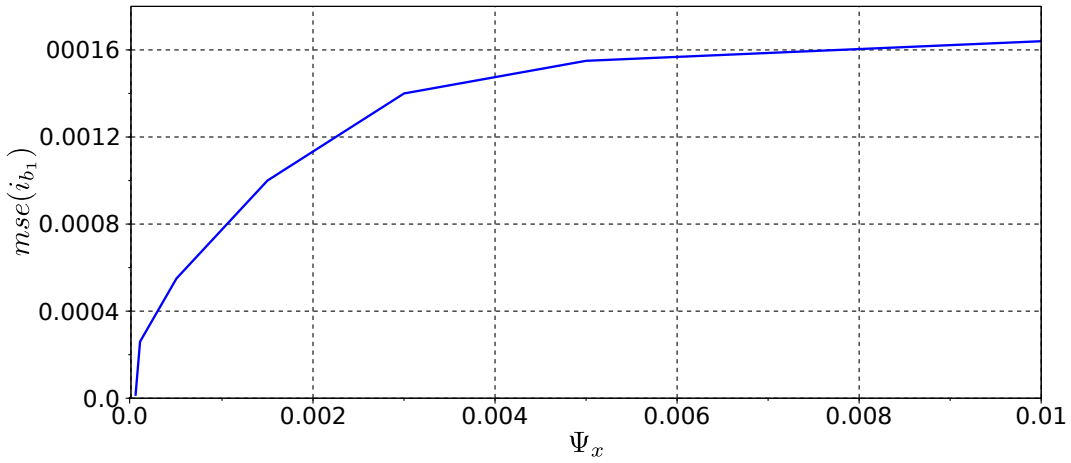


Figura 6.18: Error cuadrático medio en la corriente $i_{b_1}(t)$ vs Ψ_x.

La Figura 6.19 muestra la ganancia en tiempo de CPU (definida como la razón $CPU_{\text{SW}}/CPU_{\text{mix}}$) obtenida con el modelo mixto en comparación con el modelo puramente conmutado en función de la proporción del tiempo total de simulación durante el cual se utilizó el modo conmutado, es decir, $time_{\text{SW}_{\text{mode}}}/t_{\text{sim}}$. Se observa que cuando el SWM se emplea durante menos del 54 % del tiempo total de simulación, la estrategia de modelado mixto supera al modelo puramente conmutado en eficiencia, por lo que se recomienda su aplicación en tales casos.

Un criterio empírico para aplicar la estrategia de modelado mixto puede expresarse mediante la siguiente relación:

$$\frac{(N+1)\cdot T_{\text{out}}\cdot 10}{t_{\text{sim}}} < 50\,\% \tag{6.16}$$

donde N representa el número de eventos de perturbación externa, y T_{out} es el período de salida del sistema.

Sensibilidad de parámetros del modelo mixto

La estrategia de modelado mixto propuesta requiere definir un conjunto reducido de parámetros que determinan el comportamiento del algoritmo supervisor.

En particular, los parámetros de histéresis ΔY_x y Ψ_x controlan la detección de los cambios de régimen entre los modelos promediado y conmutado. El parámetro ΔY_x se utiliza para decidir cuándo conmutar desde el modelo conmutado (SWM) hacia el modelo promediado (AVM). Si se selecciona un valor muy pequeño, el sistema permanecerá la mayor parte del tiempo utilizando el modelo SWM, por lo que la simulación presentará menor error pero un mayor costo computacional (más lenta). Por el contrario, si se elige un valor grande de ΔY_x, la conmutación al modelo promediado ocurrirá con mayor frecuencia, lo que reduce

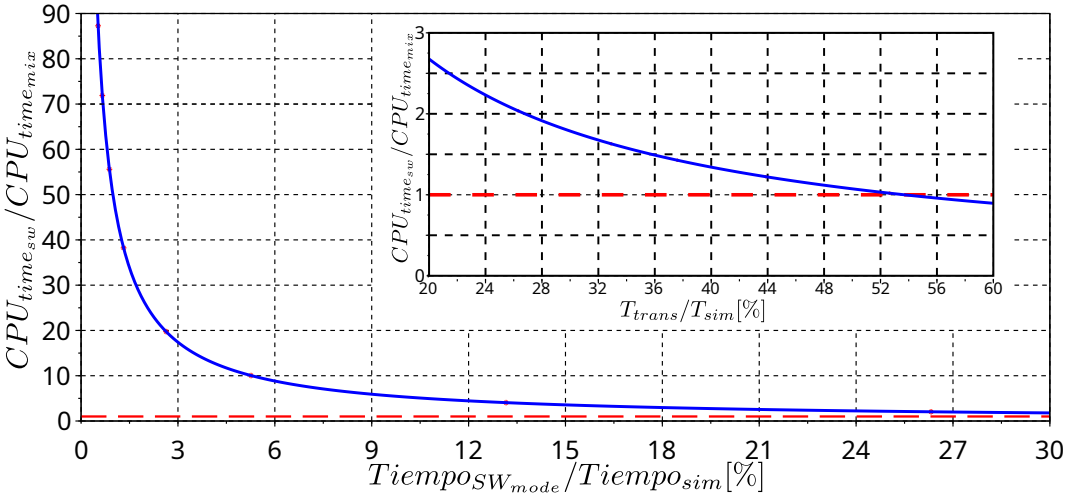


Figura 6.19: Ganancia en tiempo de CPU del modelo mixto respecto del modelo puramente conmutado, en función de la relación $time_{\mathrm{SW_{mode}}}/t_{\mathrm{sim}}$.

el tiempo de ejecución a costa de un aumento del error.

En cuanto al parámetro Ψ_x, que regula la transición inversa (de modelo promediado a conmutado), la situación es análoga. Un valor cercano a cero de Ψ_x hace que, ante la mínima perturbación, el algoritmo retorne al modelo conmutado, mejorando la precisión pero incrementando nuevamente el uso de CPU. Por el contrario, si Ψ_x se fija en un valor demasiado alto, el sistema tenderá a permanecer en el modelo promediado, logrando tiempos de simulación muy bajos a expensas de un error elevado.

La selección de estos parámetros es de carácter empírico, ya que no existe aún un criterio analítico que permita establecer su valor óptimo en función de un objetivo de error global predefinido. En principio, los parámetros pueden determinarse mediante un procedimiento de ajuste por convergencia, comparando resultados sucesivos hasta que la variación relativa entre magnitudes clave (corrientes, tensiones, potencias) sea menor que un umbral prefijado.

Si bien la sintonización actual de estos parámetros es manual, es factible desarrollar estrategias de ajuste automático basadas en control adaptativo de los umbrales en función de señales internas del sistema. Actualmente se está trabajando en este enfoque, y se dispone de resultados preliminares en los que se utiliza la información del *ripple* de la señal de control para sintonizar dinámicamente los parámetros de histéresis, manteniendo de forma adaptativa el compromiso entre precisión y eficiencia de simulación.

Ganancia y métrica de aceleración

Uno de los indicadores de desempeño utilizados para comparar las distintas estrategias de modelado es la *Ganancia de desempeño* (G). La misma se define como la relación

entre el tiempo de CPU requerido por modelo mixto respecto del requerido por el modelo puramente conmutado.

$$G = \frac{t_{\text{CPU, conmutado}}}{t_{\text{CPU, mixto}}} \tag{6.17}$$

Notar que valores de $G > 1$ indica una reducción efectiva del tiempo de ejecución con modelos mixtos respecto a los obtenidos con modelos puramente conmutados.

En la Figura 6.19 puede verse como cambia el valor de G al modificarse la fracción de tiempo tiempo virtual de simulación en que se utiliza el sub-modelo conmutado en el modelo mixto durante la simulación.

Los resultados experimentales obtenidos muestran valores de G comprendidos entre 48 y 87 según el ejemplo, lo que implica una aceleración de hasta *casi dos órdenes de magnitud* respecto del modelo puramente conmutado, sin pérdida de precisión en las variables eléctricas de interés.

En términos prácticos, esto significa que una simulación con un modelo puramente conmutado que podría requerir varias horas de CPU puede completarse con el modelos mixto en apenas algunos minutos, manteniendo detalles e información de las conmutaciones.

Por su carácter interpretativo y su relevancia práctica, esta métrica se incorporó como indicador central del aporte cuantitativo de la tesis.

Al igual que en el estudio de desempeño de los métodos LIQSS, uno de los indicadores utilizados para realizar este estudio en los modelos mixtos es el *factor de aceleración* (A_f, Ecuación (5.28)).

En modelos puramente conmutados se obtienen valores típicos de $A_f \simeq 1{,}5^{-2}$ mientras que con el uso de modelos mixtos este valor puede incrementarse hasta $A_f \simeq 0{,}75$

En términos prácticos, esto equivale a decir que si por ejemplo se debe simular 4 horas de tiempo virtual un modelo para apreciar los fenómenos estudiados, requerirá aproximadamente 266 horas de CPU utilizando modelos puramente conmutados. En cambio, al utilizar modelos mixtos este tiempo se reducirá a aproximadamente 5 horas, según el hardware empleado y parámetros de simulación.

Esta mejora cuantitativa resume el impacto principal de esta estrategia de modelado mixto para los DMSI PWM. Demostrando que es posible mantener la información de los modelos conmutados reduciendo el tiempo de cómputo mediante la estrategia de modelado mixto.

7. CONCLUSIONES Y TRABAJO FUTURO

7.1 Conclusiones

Las contribuciones desarrolladas en esta tesis muestran que tanto el uso de la estrategia de modelado mixto, así como el uso de métodos QSS para simular DMSI PWM permiten obtener mejoras significativas en lo que respecta a requerimientos de CPU y detalle en los resultados de simulación.

En primer lugar, se presentó la implementación de modelos realistas de tres topologías de DMSIs PWM. Estos modelos fueron programados utilizando el lenguaje μ–Modelica e implementados en el QSS Solver y en `OpenModelica`.

Se mostró que la simulación de estos modelos mediante métodos LIQSS permite obtener resultados con menor tiempo de cómputo e igual o mayor precisión que utilizando métodos basados en la discretización temporal. Esto permite reducir los tiempos de CPU sin pérdida de precisión en la simulación de este tipo de sistemas. Esta mejora en el rendimiento computacional se debe a que los métodos clásicos deben reinicializarse tras cada discontinuidad, lo que incrementa el tiempo de cómputo. Esto no ocurre en los métodos QSS ya que las discontinuidades son tratadas como un paso normal. Además, la detección de eventos temporales y de estado en los métodos QSS no implica un costo mayor al de un paso normal.

Si bien el uso del método LIQSS2 logró mejoras significativas en la simulación de los DMSIs PWM, estas resultan insuficientes en muchos casos como por ejemplo cuando se requieren elevados tiempos de simulación como es el caso de grandes sistemas de generación de energía.

Con el objetivo de lograr reducir aún más el tiempo de simulación se presentó una estrategia de modelado mixto para DMSIss PWM, que permite realizar simulaciones rápidas y precisas mediante la combinación de modelos promediados y modelos puramente conmutados. Se describió la nueva estrategia de modelado y se construyeron dos ejemplos de modelos de DMSI PWM boost utilizando el lenguaje `Modelica`.

Estos ejemplos fueron utilizados para probar y validar la estrategia, comparando los resultados de simulación con los obtenidos mediante modelos puramente conmutados. Los resultados muestran que la estrategia de modelado mixto puede reducir significativamente

el tiempo de CPU sin sacrificar detalles ni precisión en simulaciones de largo plazo. Esta mejora en el rendimiento computacional se logra reduciendo la cantidad de eventos detectados. durante la simulación. En este sentido se usan modelos promediados cuando no hay transitorios en la evolución de las señales del inversor y modelos conmutados precisos durante situaciones transitorias. Una ventaja adicional de esta estrategia es la estimación automática de los parámetros del modelo promediado durante la simulación, lo que asegura la precisión de los resultados incluso frente a cambios significativos en las condiciones de operación.

Si bien la metodología propuesta fue aplicada únicamente a un DMSI PWM boost, puede extenderse de manera sencilla a otras topologías de inversores PWM utilizando el bloque *full bridge* y conectando los componentes continuos restantes adecuadamente.

7.2 Trabajo futuro

Considerando que la utilización de los métodos QSS ha demostrado una notable reducción del costo computacional en la simulación de modelos conmutados de DMSIs PWM se propone profundizar en su aplicación a modelos mixtos. Esto se fundamenta en que si bien estos modelos utilizan modelos promediados gran parte del tiempo en los cuales no hay discontinuidades, durante las situaciones transitorias se usan modelos conmutados donde los métodos QSS han mostrado ventajas frente a los métodos tradicionales. En tal sentido, se plantea investigar el desempeño de los métodos de integración QSS combinándolos con la estrategia de modelado mixto en la simulación de DMSIs PWM. Se espera que el uso de algoritmos basados en la cuantificación del estado, en lugar de la discretización del tiempo, reduzca significativamente los requerimientos computacionales para la simulación de estos modelos.

En otra línea de trabajo, a partir del desempeño observado en la estrategia de modelado mixto, resulta necesario explorar su aplicación tanto en topologías de inversores trifásicos como en sistemas de gran escala que integran múltiples convertidores conmutados e inversores. Los resultados obtenidos en esta tesis permiten vislumbrar la posibilidad de aprovechar las ventajas del modelado mixto frente a los enfoques puramente conmutados para realizar simulaciones de largo plazo en sistemas de electrónica de potencia de gran escala, como los sistemas híbridos de generación.

8. PUBLICACIONES REALIZADAS DURANTE EL DESARROLLO DE LA TESIS

Los siguientes trabajos reflejan directamente los aportes principales de esta tesis. Se trata de publicaciones que surgen como resultado del desarrollo, implementación y validación de los modelos y estrategias propuestas en el presente trabajo.

Entre estas publicaciones, se destaca el trabajo titulado *Simulación de inversores PWM monofásicos en modo diferencial con métodos basados en cuantificación*, aceptado para su presentación en el simposio de modelado y simulación de las 54as Jornadas Argentinas de Informática e Investigación Operativa (JAIIO). Actualmente en prensa.

Uno de los aportes centrales de esta tesis es el desarrollo de una estrategia de modelado mixto para DMSIs PWM, junto con la implementación de un modelo mixto orientado a la simulación eficiente de estos inversores, utilizando el lenguaje *Modelica*. Este trabajo fue publicado en la revista *SIMULATION*, con el título *Mixed Modeling Approach for Efficient Simulation of Single-Phase PWM Inverters* [10].

Las siguientes publicaciones, si bien no forman parte directa de los aportes originales de esta tesis, constituyen antecedentes significativos del trabajo realizado, ya que aportaron desarrollos, herramientas y conocimientos que sirvieron de base para el planteo y resolución de los problemas abordados.

En primer lugar, se destaca el desarrollo de modelos y la implementación de una librería de fuentes conmutadas CC–CC en PowerDEVS, cuyos resultados fueron publicados en las actas de la RPIC [8]. Esta implementación constituyó una base conceptual y práctica para el estudio posterior de estrategias de simulación eficiente de fuentes conmutadas.

Durante la investigación orientada al diseño de nuevos MIC para simulación eficiente, se desarrolló un método básico aplicado al estudio de sistemas hamiltonianos (marginalmente estables). Este trabajo fue publicado inicialmente en los *Proceedings of AADECA* [9] y, posteriormente, ampliado y recopilado en el artículo *Quantized State System Simulation*, publicado en los *Proceedings of SummerSim 08* (2008 Summer Simulation Multiconference) [16].

En la misma línea, se participó en el desarrollo de nuevos métodos de integración para ecuaciones diferenciales ordinarias rígidas (*stiff*), basados en la cuantificación de esta-

dos, conocidos como LIQSS. Los aportes consistieron en analizar su aplicación a fuentes conmutadas CC–CC, y los resultados fueron publicados en la revista *Simulation Modelling Practice and Theory* [57].

BIBLIOGRAFÍA

[1] M. Albakri, A. Darwish, and P. Twigg. A comprehensive review of dc/ac single-phase differential-mode inverters for low-power applications. *Electronics*, 13(13):2474, 2024.

[2] L. Allain, A. Merdassi, L. Gerbaud, and S. Bacha. Automatic modelling of power electronic converter: Average model construction and Modelica model generation. In *Proceedings of the 7th International Modelica Conference*, volume 66 of *Linköping Electronic Articles in Computer and Information Science*, pages 576–586, Como, Italy, 2009. Linköping University Electronic Press.

[3] S. Bacha, I. Munteanu, and A. Bratcu. *Power Electronic Converters*. Springer, 2014.

[4] S. Banerjee and S. K. Mishra. An average circuit model of a single-phase grid-connected inverter. In *11th International Conference on Power Electronics and ECCE Asia*, pages 2188–2193. IEEE, 2023.

[5] Vincenzo Barba, Salvatore Musumeci, Fausto Stella, Fabio Mandrile, and Marco Palma. Investigation of dead time losses in inverter switching leg operation: Gan fet vs. mosfet comparison. *Energies*, 17(15):3855, 2024.

[6] Luca Barbierato, Michele Rossi, Alberto Leva, and et al. Facilitating smart grids integration through a hybrid multi-model co-simulation framework. *IEEE Access*, 12:104878–104897, 2024.

[7] F. Bergero, X. Floros, J. Fernández, E. Kofman, and F. E. Cellier. Simulating Modelica models with a stand-alone quantized state systems solver. In H. Elmqvist and M. Otter, editors, *Proceedings of the 9th International Modelica Conference*, volume 76 of *Linköping Electronic Conference Proceedings*, pages 237–246, Munich, Germany, September 3–5 2012. Linköping University Electronic Press.

[8] M. Bortolotto, F. Fontenla, E. Kofman, and M. Romero. Librería de simulación por eventos discretos de fuentes conmutadas. In *Proceedings of RPIC 2007*, Rio Gallegos, Argentina, 2007.

[9] M. Bortolotto, E. Kofman, and G. Migoni. Métodos de integración por cuantificación en sistemas hamiltonianos. In *Proceedings of AADECA 2008*, Buenos Aires, Argentina, 2008.

[10] M. Bortolotto and G. Migoni. Mixed modeling approach for efficient simulation of single-phase PWM inverters. *SIMULATION*, 101(1):73–85, 2025.

[11] R. Caceres and I. Barbi. A boost dc-ac converter: operation, analysis, control and experimentation. In *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*, volume 1, pages 546–551, 1995.

[12] CAMMESA. Informe anual 2022: Generación eléctrica en argentina. Technical report, Cámara Argentina de la Energía, 2022.

[13] F. Casella, A. Bartolini, S. Pasquini, and L. Bonuglia. Object-oriented modelling and simulation of large-scale electrical power systems using modelica: A first feasibility study. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 6298–6304. IEEE, 2016.

[14] R. Castro, M. Bergonzi, E. P. Marcosig, J. Fernández, and E. Kofman. Discrete-event simulation of continuous-time systems: evolution and state of the art of quantized state system methods. *Simulation*, 100(6):613–638, 2024.

[15] F. Cellier and E. Kofman. *Continuous System Simulation*. Springer, New York, 2006.

[16] F. Cellier, E. Kofman, G. Migoni, and M. Bortolotto. Quantized state system simulation. In *Proceedings of SummerSim 08 (2008 Summer Simulation Multiconference)*, Edinburgh, Scotland, 2008. Society for Modeling & Simulation International.

[17] R. Chang, Y. Luo, and R. Zhu. Simulated local climatic impacts of large-scale photovoltaics over the barren area of qinghai, china. *Renewable Energy*, 145:478–489, 2020.

[18] Marc Cheah-Mane, Agusti Egea-Alvarez, Eduardo Prieto-Araujo, Hasan Mehrjerdi, Oriol Gomis-Bellmunt, and Lie Xu. Modeling and analysis approaches for small-signal stability assessment of power-electronic-dominated systems. *WIREs Energy and Environment*, 12(1):e453, 2022.

[19] Fabio Cremona, Marten Lohstroh, David Broman, Patricia Derler, Stavros Tripakis, and Edward A. Lee. Hybrid co-simulation: It's about time. *Software and Systems Modeling*, 18(3):1655–1679, 2019.

[20] R. O. Cáceres and I. Barbi. A boost dc–ac converter: Analysis, design, and experimentation. *IEEE Transactions on Power Electronics*, 14(1):134–141, January 1999.

[21] B. De Kelperd, L. Dessaint, K. Al-Haddad, and H. Nakra. A comprehensive approach to fixed-step simulation of switched circuits. *IEEE Transactions on Power Electronics*, 17(2):216–224, 2002.

[22] N. Dhameliya. Power electronics innovations: Improving efficiency and sustainability in energy systems. *Asia Pacific Journal of Energy and Environment*, 9:71–80, 11 2022.

[23] L. Dieci and L. Lopez. A survey of numerical methods for IVPs of ODEs with discontinuous right-hand side. *Journal of Computational and Applied Mathematics*, 236(16):3967–3991, 2012.

[24] H. Elmqvist, D. Brueck, and M. Otter. *Dymola User's Manual*. Dynasim AB, Research Park Ideon, Lund, Sweden, 1995.

[25] R. W. Erickson and D. Maksimovic. *Fundamentals of Power Electronics*. Springer Cham, 3rd edition, 2020.

[26] J.M. Esposito, V. Kumar, and G.J. Pappas. Accurate event detection for simulating hybrid systems. In *HSCC*, volume 2034 of *Lecture Notes in Computer Science*, pages 204–217. Springer, 2001.

[27] Yuyao Feng, Xuejun Xiong, Christian Scheibe, Hanzhong Wang, Piergiovanni La Seta, and Holger Müller. Co-simulation of real-time and offline power system models: An application example. In *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, pages 1–5, 2023.

[28] J. Fernández and E. Kofman. A stand-alone quantized state system solver for continuous system simulation. *SIMULATION*, 90(7):782–799, 2014.

[29] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley-IEEE Press, 2nd ed. edition, 2015.

[30] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4 edition, 2013.

[31] A. Gopal, E. Devaraj, S. Saha, and S. Poddar. Modeling and voltage regulation of boost inverter. In *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pages 1–6. IEEE, 2018.

[32] E. Hairer, S. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer, Berlin, Germany, 2nd ed. edition, 1993.

[33] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*. Springer, Berlin, Germany, 2nd ed. edition, 1996.

[34] Simon Thrane Hansen, Cláudio Ângelo Gonçalves Gomes, Masoud Najafi, Torsten Sommer, Matthias Blesken, Irina Zacharias, Oliver Kotte, Pierre R Mai, Klaus Schuch, Karl Wernersson, et al. The fmi 3.0 standard interface for clocked and scheduled simulations. *Electronics*, 11(21):3635, 2022.

[35] H. Holttinen, J. Kiviluoma, D. Flynn, J. C. Smith, A. Orths, P. B. Eriksen, N. Cutululis, L. Soder, M. Korpas, A. Estanqueiro, J. MacDowell, A. Tuohy, T. K. Vrana, and M. O'Malley. System impact studies for near 100 % renewable energy systems dominated by inverter based variable generation. *IEEE Transactions on Power Systems*, 37(4):3249–3258, 2022.

[36] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2013.

[37] Md Tonmoy Hossain, Md Zunaid Hossen, Faisal R Badal, Md R Islam, Md Mehedi Hasan, Md F Ali, Md H Ahamed, SH Abhi, Md Manirul Islam, Subrata K Sarker, et al. Next generation power inverter for grid resilience: Technology review. *Heliyon*, 10(21), 2024.

[38] L. Iannelli. *Dynamics and Control of Switched Electronic Systems: Advanced Perspectives for Modeling, Simulation and Control of Power Converters*. Springer London, London, UK, 2012.

[39] Intergovernmental Panel on Climate Change. *Climate Change 2021: The Physical Science Basis*. Cambridge University Press, Cambridge, 2021.

[40] Saiful Islam, Nirosha Abeywickrama, and et al. A review of modelling techniques of power transformers for digital real-time simulation. *The Journal of Engineering*, 2022:1718–1731, 2022.

[41] S. Jiao, P.H. Huang, W. Xiao, S. Li, and J. Li. Balancing simulation speed and accuracy for grid-connected photovoltaic power systems. In *2023 IEEE International Future Energy Electronics Conference (IFEEC)*, pages 317–321, 2023.

[42] Andreas Junghanns, Cláudio Gomes, Christian Schulze, Klaus Schuch, Matthias Blaesken, Irina Zacharias, Andreas Pillekeit, Karl Wernersson, Torsten Sommer, Christian Bertsch, et al. The functional mock-up interface 3.0-new features enabling new applications. In *Modelica conferences*, pages 17–26, 2021.

[43] J. Kassakian. *Principles of Power Electronics*. Wesley Publishing Company, 1991.

[44] A. A. S. Khan and K. M. Rahman. Voltage mode control of single phase boost inverter. In *2008 International Conference on Electrical and Computer Engineering*, pages 665–670. IEEE Computer Society, 2008.

[45] E. Kofman. A Second Order Approximation for DEVS Simulation of Continuous Systems. *Simulation*, 78(2):76–89, 2002.

[46] E. Kofman. Discrete event simulation of hybrid systems. *SIAM Journal on Scientific Computing*, 25(5):1771–1797, 2004.

[47] E. Kofman. A third order discrete event simulation method for continuous system simulation. *Latin American Applied Research*, 36(2):101–108, 2006.

[48] E. Kofman and S. Junco. Quantized State Systems. A DEVS Approach for Continuous System Simulation. *Transactions of SCS*, 18(3):123–132, 2001.

[49] E. Kofman, J.S. Lee, and B. Zeigler. DEVS Representation of Differential Equation Systems. Review of Recent Advances. In *Proceedings of the 2001 European Simulation Symposium (ESS'01)*, pages 591–595, Marseille, France, 2001. Society for Computer Simulation International.

[50] A. Körner, F. Breitenecker, and N. Popper. About an alternative method of numerical iteration for state event finding and handling in system simulation of hybrid dynamical systems. In *Computer Modelling and Simulation (UKSim), 2013 UKSim 15th International Conference on*, pages 390–395. IEEE, 2013.

[51] J. Kwon, X. Wang, F. Blaabjerg, C. L. Bak, V.S. Sularea, and C. Busca. Harmonic interaction analysis in a grid-connected converter using harmonic state-space (HSS) modeling. *IEEE Trans. Power Electron.*, 32(9):6823–6835, 2017.

[52] Jose Daniel Lara, Rodrigo Henriquez-Auba, Deepak Ramasubramanian, Sairaj Dhople, Duncan Callaway, and Seth Sanders. Revisiting power systems time-domain simulation methods and models. *IEEE Transactions on Power Systems*, 2023.

[53] C. Liu, J. Hsieh, C. Chang, J. Bocek, and Y. Hsiao. A fast-decoupled method for time-domain simulation of power converters. *IEEE Transactions on Power Electronics*, 8(1):37–45, 1993.

[54] D. Logue and P. T. Krein. Simulation of electric machinery and power electronics interfacing using MATLAB/SIMULINK. In *Computers in Power Electronics, 2000. COMPEL 2000. The 7th Workshop on*, pages 34–39. IEEE, 2000.

[55] M. Mao, X. Zhang, and L. Chang. Research on fast modeling and super real-time simuation for grid-connected pv system. In *2017 Chinese Automation Congress (CAC)*, pages 7123–7128. IEEE, 2017.

[56] G. Migoni, F. Bergero, E. Kofman, and J. Fernández. Quantization-Based Simulation of Switched Mode Power Supplies. *Simulation: Transactions of the Society for Modeling and Simulation International*, 91(4):320–336, 2015.

[57] G. Migoni, M. Bortolotto, E. Kofman, and F. E. Cellier. Linearly implicit quantization-based integration methods for stiff ordinary differential equations. *Simulation Modelling Practice and Theory*, 35:118–136, 2013.

[58] G. Migoni, M. E. Romero, F. Bergero, and E. Kofman. A mixed modeling approach for efficient simulation of PWM switching mode power supplies. *IEEE Transactions on Power Electronics*, 34(10):9758–9767, 2019.

[59] G. Migoni, P. Rullo, F. Bergero, and E. Kofman. Efficient simulation of hybrid renewable energy systems. *international journal of hydrogen energy*, 41(32):13934–13949, 2016.

[60] S. M. Mirić and P. V. Pejović. A method for computer-aided analysis of differential mode input filters. *IEEE Transactions on Industrial Electronics*, 64(6):4741–4750, 2017.

[61] N. Mohan. *Power Electronics: A First Course*. Wiley, Hoboken, NJ, 2012.

[62] N. Mohan, W. P. Robbins, T.M. Undeland, R. Nilssen, and Mo O. Simulation of power electronic and motion control systems-an overview. *Proceedings of the IEEE*, 82(8):1287–1302, 1994.

[63] N. Mohan, T. M. Undeland, and W. P. Robbins. *Power Electronics: Converters, Applications, and Design*. John Wiley & Sons, 3rd ed. edition, 2003.

[64] M.K. Nguyen and T.T. Tran. A single-phase single-stage switched-boost inverter with four switches. *IEEE Transactions on Power Electronics*, PP(99):1–1, 09 2017.

[65] Annika Ofenloch, Jan Sören Schwarz, Deborah Tolk, Tobias Brandt, Reef Eilers, Rebeca Ramirez, Thomas Raub, and Sebastian Lehnhoff. Mosaik 3.0: Combining time-stepped and discrete event simulation. In *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)*, pages 1–5. IEEE, 2022.

[66] M Ouafi, Jean Mahseredjian, J Peralta, H Gras, S Dennetière, and B Bruned. Parallelization of emt simulations for integration of inverter-based resources. *Electric Power Systems Research*, 223:109641, 2023.

[67] L. Pagliero, E. Kofman, and S. Junco. Simulation of power electronic converters using quantized state system methods. In *Proceedings of RPIC'03 (Reunión de Trabajo en Procesamiento de la Información y Control)*, volume 2, pages 636–641, San Nicolás, Argentina, 2003. RPIC.

[68] H. Parildar and A. Leva. Mixed phasor and time domain modelling of ac networks with changeover management. In *Proceedings of the Linkoping University Electronic Press*, volume 56 of *Lecture Notes in Electrical Engineering*, pages 533–542, 2014.

[69] S. Park, W. Chen, A. M. Bazzi, and S. Park. A time-efficient approach for modelling and simulation of aggregated multiple photovoltaic microinverters. *Energies*, 10(4):465, 2017.

[70] Rohit Patil, K. Krithivasan, and A. Narayanaswamy. Robust hardware–software co-simulation framework for design and validation of hybrid systems. In *Proceedings of the 20th ACM–IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, pages 1–11, Shanghai, China, October 2022. IEEE.

[71] T. Pavlovic, T. Bjazic, and Z. Ban. Simplified averaged models of DC–DC power converters suitable for controller design and microgrid simulation. *IEEE transactions on power electronics*, 28(7):3266–3275, 2013.

[72] P. Pejovic and D. Maksimovic. A New Algorithm for Simulation of Power Electronic Systems Using Piecewise–Linear Device Models. *IEEE Trans. Power Electron.*, 10(3):340–348, 1995.

[73] L. R. Petzold. A description of DASSL: A differential/algebraic system solver. Technical Report UCRL-87244, Lawrence Livermore National Laboratory, 1982.

[74] L. R. Petzold. A description of DASSL: a differential/algebraic system solver. In *Scientific computing (Montreal, Quebec, 1982)*, pages 65–68. IMACS, New Brunswick, NJ, 1983.

[75] A. Pressman. *Switching Power Supply Design*. Mc.Graw Hill, 1998.

[76] S. Pérez, M. Vivert, R. Díez, and D. Patino. Modeling and control of a grid tied differential cuk inverter. In *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, pages 1–9, 2017.

[77] M. Rasheduzzaman, P. Fajri, J. Kimball, and B. Deken. Modeling, analysis, and control design of a single-stage boost inverter. *Energies*, 14(14):4098, 2021.

[78] J. Rodriguez, F. Blaabjerg, and M.P. Kazmierkowski. Energy transition technology: The role of power electronics. *Proceedings of the IEEE*, 111(4):329–334, 2023.

[79] C. R. Sánchez Reinoso, D.H. Milone, and R.H. Buitrago. Simulation of photovoltaic centrals with dynamic shading. *Applied Energy*, 103:657–666, 2013.

[80] P. Sanchis, A. Ursaea, E. Gubia, and L. Marroyo. Boost dc-ac inverter: a new control strategy. *IEEE Transactions on Power Electronics*, 20(2):343–353, 2005.

[81] C. Shah, J. D. Vasquez-Plaza, D. D. Campo-Ossa, J. F. Patarroyo-Montenegro, N. Guruwacharya, N. Bhujel, R. D. Trevizan, F. A. Rengifo, M. Shirazi, R. Tonkoski, R. Wies, T. M. Hansen, and P. Cicilio. Review of dynamic and transient modeling of power electronic converters for converter dominated power systems. *IEEE Access*, 9:82094–82117, 2021.

[82] F. A. Simões, M. G.and Farret. *Modeling Power Electronics and Interfacing Energy Conversion Systems*. Wiley-IEEE Press, Hoboken, NJ, 2016.

[83] J. Sun, D. Mitchell, M. Greuel, Krein P., and R. Bass. Averaged modeling of PWM converters operating in discontinuous conduction mode. *IEEE Transactions on Power Electronics*, 16(4):482–492, 2001.

[84] Y. Tan, Y. Shen, M. Xu, C. Shen, and J. Lei. Reduced model of droop-controlled converters for transient simulation analysis. *Chinese Journal of Electrical Engineering*, 8(3):112–122, 2022.

[85] A. M. Trzynadlowski, editor. *Power Electronic Converters and Systems: Frontiers and Applications*. The Institution of Engineering and Technology, London, 1st edition, 2015.

[86] Bo Wen, Rolando Burgos, Paolo Mattavelli, and Dushan Boroyevich. Experimental evaluation of voltage source inverter switching model with embedded c code controller. In *Proceedings of the 2013 Grand Challenges on Modeling and Simulation Conference*, page 8, 2013.

[87] W. Xiao. *Power Electronics Step-by-Step: Design, Modeling, Simulation, and Control*. McGraw-Hill Education, New York, 1st ed. edition, 2021.

[88] K. Yamashita, H. Renner, S. Martínez Villanueva, G. Lammert, P. Aristidou, J. Carvalho Martins, L. Zhu, L. D. Pabón Ospina, and T. Van Cutsem. Industrial recommendation of modeling of inverter-based generators for power system dynamic studies with focus on photovoltaic. *IEEE Power and Energy Technology Systems Journal*, 5(1):1–10, 2018.

[89] D. Yan, C. Yang, L. Hang, Y. He, P. Luo, L. Shen, and P. Zeng. Review of general modeling approaches of power converters. *Chinese Journal of Electrical Engineering*, 7(1):27–36, 2021.

[90] Dong Yan, Chenglin Yang, Lijun Hang, Yuanbin He, Ping Luo, Lei Shen, and Pingliang Zeng. Review of general modeling approaches of power converters. *Chinese Journal of Electrical Engineering*, 7(1):27–36, 2021.

[91] B. Zeigler, T.G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. Academic Press, New York, second edition, 2000.

[92] 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.

[93] B.P. Zeigler, A. Muzy, and E. Kofman. *Theory of Modeling and Simulation. 3rd Edition.* Academic Press, New York, 2018.

[94] T. Ören, B. P. Zeigler, and A. Tolk, editors. *Body of Knowledge for Modeling and Simulation: A Handbook by the Society for Modeling and Simulation International*. Springer, Cham, 2023. Edited by T. Ören, B. P. Zeigler, and A. Tolk.

A. CÓDIGO FUENTE DEL MODELO MIXTO

A.1 Modelo `FullBridge` en Modelica

El siguiente código corresponde al modelo mixto `FullBridge`, implementado en Modelica. Este modelo representa una celda conmutada de puente completo, capaz de alternar entre un modelo switcheado (SWM) y un modelo promediado (AVM) según ciertas condiciones del régimen de operación.

```
1  model FullBridge
2    Modelica.Blocks.Interfaces.RealInput uC1 "Tension en el capacitor 1"
       ;
3    Modelica.Blocks.Interfaces.RealInput uC2 "Tension en el capacitor 2"
       ;
4    Modelica.Blocks.Interfaces.RealInput iL1 "Corriente en el inductor 1
       ";
5    Modelica.Blocks.Interfaces.RealInput iL2 "Corriente en el inductor 2
       ";
6    Modelica.Electrical.Analog.Interfaces.Pin pin11;
7    Modelica.Electrical.Analog.Interfaces.Pin pin21;
8    Modelica.Electrical.Analog.Interfaces.Pin pin12;
9    Modelica.Electrical.Analog.Interfaces.Pin pin22;
10   Modelica.Electrical.Analog.Interfaces.NegativePin pin_n;
11   Modelica.Electrical.Analog.Basic.VariableResistor Rtp11;
12   Modelica.Electrical.Analog.Basic.VariableResistor Rtp12;
13   Modelica.Electrical.Analog.Basic.VariableResistor Rtp21;
14   Modelica.Electrical.Analog.Basic.VariableResistor Rtp22;
15   Modelica.Electrical.Analog.Sources.SignalVoltage Vtp12;
16   Modelica.Electrical.Analog.Sources.SignalCurrent Itp11;
17   Modelica.Electrical.Analog.Sources.SignalCurrent Itp21;
18   Modelica.Electrical.Analog.Sources.SignalVoltage Vtp22;
19   parameter Real Ron(unit = "Ohm") = 1e-5 "Resistencia equivalente de
       elemento de conmutacion cortado";
20   parameter Real Roff(unit = "Ohm") = 1e5 "Resistencia equivalente de
       elemento de conmutacion conduciendo";
21   //Parametros de la PWM
```

```
22   parameter Real Tsw(unit = "s") = 1e-4 "Periode␣de␣la␣se\~nal␣PWM";
23   parameter Real minDC = 0.02 "minimo␣valor␣del␣duty-cycle␣de␣la␣PWM";
24   parameter Real maxDC = 0.98 "maximo␣valor␣del␣duty-cycle␣de␣la␣PWM";
25   //Indicates the new mode (true=switched false=averaged)
26   discrete Real modSel(start = 1);
27   //Indicates the used mode (1=switched 0=averaged)
28   Modelica.Electrical.Analog.Ideal.IdealClosingSwitch modSel1;
29   Modelica.Electrical.Analog.Ideal.IdealClosingSwitch modSel2;
30   Modelica.Blocks.Interfaces.RealInput dc1 ;
31   Modelica.Blocks.Interfaces.RealInput dc2 ;
32   discrete Boolean newPwmPeriod(start = false);
33   atomicosNew.mode_selectorNew mode_selectorN(deltaSignalControlToAv =
         4);
34
35   Boolean modSelChange(start = true);
36   discrete Real tNextStart(start = 1e-9);
37 protected
38   discrete Real Rtp11d(start = Roff);
39   discrete Real Rtp12d(start = Roff);
40   discrete Real Rtp21d(start = Roff);
41   discrete Real Rtp22d(start = Roff);
42   discrete Real tNext1(start = 2e-9);
43   discrete Real tNext2(start = 2e-9);
44   discrete Real dcDisc1(start = 0.5);
45   discrete Real dcDisc2(start = 0.5);
46   Real dcSat1;
47   Real dcSat2;
48 equation
49   connect(Rtp12.p, Rtp22.p);
50   connect(Rtp11.p, Rtp12.n);
51   connect(Rtp21.p, Rtp22.n);
52   connect(pin22, Rtp22.n);
53   connect(pin21, Rtp21.n);
54   connect(pin11, Rtp11.n);
55   connect(pin12, Rtp12.n);
56   connect(Itp11.n, Rtp11.n);
57   connect(Itp11.p, Rtp11.p);
58   connect(Itp21.n, Rtp21.n);
59   connect(modSel1.p, Vtp12.n);
60   connect(modSel2.p, Vtp22.n);
61   connect(modSel2.n, Rtp22.p);
62   connect(pin_n, Rtp12.p);
63   connect(Itp21.p, Rtp21.p);
```

```
64    connect(Vtp22.p, Rtp22.n);
65    connect(Vtp12.p, Rtp12.n);
66    modSelChange = mode_selectorN.modSelChange;
67    (dcSat1, dcSat2) = saturaDC(dc1, dc2, Tsw, minDC, maxDC);
68    Itp11.i = Vtp12.i * (1 - modSel) * (1 - dcSat1) / dcSat1;
69    Itp21.i = Vtp22.i * (1 - modSel) * (1 - dcSat2) / dcSat2;
70    Vtp12.v = -Itp11.v * (1 - modSel) * (1 - dcSat1) / dcSat1;
71    Vtp22.v = -Itp21.v * (1 - modSel) * (1 - dcSat2) / dcSat2;
72    modSel1.control = not modSelChange;
73    modSel2.control = not modSelChange;
74    Rtp11.R = Rtp11d;
75    Rtp12.R = Rtp12d;
76    Rtp21.R = Rtp21d;
77    Rtp22.R = Rtp22d;
78    mode_selectorN.u1 = pin21.i;
79    mode_selectorN.u2 = uC1 - uC2;
80    mode_selectorN.uC1 = uC1;
81    mode_selectorN.uC2 = uC2;
82    mode_selectorN.iL1 = iL1;
83    mode_selectorN.iL2 = iL2;
84    mode_selectorN.Tsw = Tsw;
85    mode_selectorN.newPwmPeriod = newPwmPeriod;
86    mode_selectorN.Rt11 = Rtp11d;
87    mode_selectorN.Rt12 = Rtp12d;
88    mode_selectorN.Rt21 = Rtp21d;
89    mode_selectorN.Rt22 = Rtp22d;
90    mode_selectorN.deltaTNextStart = tNextStart - time;
91    mode_selectorN.deltaTNext1 = tNext1 - time;
92    mode_selectorN.deltaTNext2 = tNext2 - time;
93  algorithm
94
95    when modSelChange then
96      //Se comienza a usar el modelo conmutado
97      modSel := 1;
98      Rtp11d := mode_selectorN.lastRt11d;
99      Rtp12d := mode_selectorN.lastRt12d;
100     Rtp21d := mode_selectorN.lastRt21d;
101     Rtp22d := mode_selectorN.lastRt22d;
102     if mode_selectorN.lastDeltaTNextStart > 0 then
103       tNextStart := time + mode_selectorN.lastDeltaTNextStart;
104     else
105       tNextStart := 1e10;
106     end if;
```

```
107     if mode_selectorN.lastDeltaTNext1 > 0 then
108       tNext1 := time + mode_selectorN.lastDeltaTNext1;
109     else
110       tNext1 := 1e10;
111     end if;
112     if mode_selectorN.lastDeltaTNext2 > 0 then
113       tNext2 := time + mode_selectorN.lastDeltaTNext2;
114     else
115       tNext2 := 1e10;
116     end if;
117     //    tNextStart:=time;
118   end when;
119   when not modSelChange then
120     //Se comienza a usar el modelo promediado
121     Rtp11d := Roff;
122     Rtp12d := Roff;
123     Rtp21d := Roff;
124     Rtp22d := Roff;
125     tNextStart := 1e10;
126     tNext1 := 1e10;
127     tNext2 := 1e10;
128     modSel := 0;
129   end when;
130   //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
131   // PWM
132   //%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
133   when time * modSel > tNextStart then
134     //poner TnextStart en valor alto en promediado
135     dcDisc1 := dcSat1;
136     dcDisc2 := dcSat2;
137     tNext1 := time + dcDisc1 * Tsw;
138     tNext2 := time + dcDisc2 * Tsw;
139     tNextStart := time + Tsw;
140     Rtp11d := Roff;
141     Rtp12d := Ron;
142     Rtp21d := Roff;
143     Rtp22d := Ron;
144     newPwmPeriod := true;
145   end when;
146   when time * modSel > tNext1 then
147     Rtp11d := Ron;
148     Rtp12d := Roff;
149     newPwmPeriod := false;
```

```
150   end when;
151   when time * modSel > tNext2 then
152     Rtp21d := Ron;
153     Rtp22d := Roff;
154     newPwmPeriod := false;
155   end when;
156 end FullBridge;
```

A.1.1 Atómico Mode Selector en Modelica

```
1  model mode_selector
2    Modelica.Blocks.Interfaces.RealInput u1 "Variable 1 de control";
3    Modelica.Blocks.Interfaces.RealInput u2 "Variable 2 de control";
4    Modelica.Blocks.Interfaces.RealInput uC1 "Tension en el capacitor 1"
       ;
5    Modelica.Blocks.Interfaces.RealInput uC2 "Tension en el capacitor 2"
       ;
6    Modelica.Blocks.Interfaces.RealInput iL1 "Corriente en el inductor 1
       ";
7    Modelica.Blocks.Interfaces.RealInput iL2 "Corriente en el inductor 2
       ";
8    Modelica.Blocks.Interfaces.RealInput Tsw "Periodo de conmutacion de
       la PWM";
9    Modelica.Blocks.Interfaces.BooleanOutput modSelChange "true=Modelo
       conmutado - False Modelo Promediado";
10   Modelica.Blocks.Interfaces.BooleanInput newPwmPeriod;
11   signalControlGenerator signalControlGen1;
12   global_max_min_fun max_minCalcM;
13   filtroArmonicosG filtroG;
14   Modelica.Blocks.Interfaces.RealOutput iL1Reset "Corriente iL1 que se
       debe fijar al cambiar de modelos";
15   Modelica.Blocks.Interfaces.RealOutput iL2Reset "Corriente iL2 que se
       debe fijar al cambiar de modelos";
16   Modelica.Blocks.Interfaces.RealOutput uC1Reset "Tension uC1 que se
       debe fijar al cambiar de modelos";
17   Modelica.Blocks.Interfaces.RealOutput uC2Reset "Tension uC2 que se
       debe fijar al cambiar de modelos";
18   Real max_U1;
19   Real max_U2;
20   Real min_U1;
21   Real min_U2;
22   discrete Real modSel;
```

```
23   parameter Real deltaSignalControlToAv(start = 5);
24   //maxima variacion porcentual de los max y min respecto al riple de
       la se\~nal de control.
25   discrete Real timeToSwitched(start = 1e10);
26   discrete Integer nPeriods(start = 0);
27   discrete Real newtimePeriod(start = 0.02);
28   discrete Real lastDeltaiL1(start = 0);
29   discrete Real lastDeltaiL2(start = 0);
30   discrete Real lastDeltauC1(start = 0);
31   discrete Real lastDeltauC2(start = 0);
32   Modelica.Blocks.Interfaces.RealInput Rt11;
33   Modelica.Blocks.Interfaces.RealInput Rt12;
34   Modelica.Blocks.Interfaces.RealInput Rt21;
35   Modelica.Blocks.Interfaces.RealInput Rt22;
36   Modelica.Blocks.Interfaces.RealInput deltaTNextStart;
37   Modelica.Blocks.Interfaces.RealInput deltaTNext1;
38   Modelica.Blocks.Interfaces.RealInput deltaTNext2;
39   discrete Real lastDeltaTNextStart, lastDeltaTNext1, lastDeltaTNext2;
40   discrete Real lastRt11d, lastRt12d, lastRt21d, lastRt22d;
41   discrete Real iL1Resetd(start = 0);
42   discrete Real iL2Resetd(start = 0);
43   discrete Real uC1Resetd(start = 0);
44   discrete Real uC2Resetd(start = 0);
45   Real iL1ave(start = 0);
46   Real iL2ave(start = 0);
47   Real uC1ave(start = 0);
48   Real uC2ave(start = 0);
49   Real Ripple1;
50   //borrar
51   Real Ripple2;
52   //borrar
53   Real Ripple3;
54   //borrar
55   Real Ripple4;
56   //borrar
57 protected
58   //  discrete Real newHalfPeriod(start = 0.02);
59   discrete Real last_max_U1(start = 2);
60   discrete Real last_max_U2(start = 2);
61   discrete Real last_min_U1(start = 2);
62   discrete Real last_min_U2(start = 2);
63   discrete Real modSel1(start = 1);
64   discrete Real modSel2(start = 1);
```

```
65   discrete Real sum_dif_max_U1(start = 1);
66   discrete Real sum_dif_max_U2(start = 1);
67   discrete Real sum_dif_min_U1(start = 1);
68   discrete Real sum_dif_min_U2(start = 1);
69   discrete Real ripple_1(start = 0.001);
70   discrete Real ripple_2(start = 0.001);
71   discrete Real last_last_max_U1(start = 2);
72   discrete Real last_last_max_U2(start = 2);
73   discrete Real last_last_min_U1(start = 2);
74   discrete Real last_last_min_U2(start = 2);
75   discrete Real lastTtoAv(start = 0);
76   discrete Real squareReset(start = 0);
77 equation
78   //Genero las se\~nales de control
79   signalControlGen1.u1 = u1;
80   signalControlGen1.u2 = u2;
81   //calculo los maximos y minimos de las se\~nales de control
82   max_minCalcM.u1 = signalControlGen1.y1;
83   max_minCalcM.u2 = signalControlGen1.y2;
84   max_U1 = max_minCalcM.y1_max;
85   max_U2 = max_minCalcM.y2_max;
86   min_U1 = max_minCalcM.y1_min;
87   min_U2 = max_minCalcM.y2_min;
88   //Calculo de los valores medios de las variables en los
        almacenadores
89   filtroG.newPWMperiod = newPwmPeriod;
90   filtroG.u[1] = iL1;
91   filtroG.u[2] = iL2;
92   filtroG.u[3] = uC1;
93   filtroG.u[4] = uC2;
94   iL1ave = filtroG.yf1;
95   iL2ave = filtroG.yf2;
96   uC1ave = filtroG.yf3;
97   uC2ave = filtroG.yf4;
98    //Almaceno los valores de los valores de reset de los almacenadores
        (iL y uC)
99   iL1Reset = iL1Resetd;
100  iL2Reset = iL2Resetd;
101  uC1Reset = uC1Resetd;
102  uC2Reset = uC2Resetd;
103  max_minCalcM.squareReset = squareReset;
104  filtroG.modeSel = modSel;
105  Ripple1 = iL1 - iL1ave;
```

```
106   Ripple2 = iL2 - iL2ave;
107   Ripple3 = uC1 - uC1ave;
108   Ripple4 = uC2 - uC2ave;
109 algorithm
110   when {time > newtimePeriod} then
111     nPeriods := nPeriods + 1;
112     newtimePeriod := time + 0.02;
113     //    newHalfPeriod := time + 0.02 / 2;
114     last_last_max_U1 := last_max_U1;
115     last_last_max_U2 := last_max_U2;
116     last_last_min_U1 := last_min_U1;
117     last_last_min_U2 := last_min_U2;
118     last_max_U1 := pre(max_U1);
119     last_max_U2 := pre(max_U2);
120     last_min_U1 := pre(min_U1);
121     last_min_U2 := pre(min_U2);
122     max_minCalcM.resetSignal_in := true;
123     squareReset := 1 - squareReset;
124     ripple_1 := pre(max_U1) - pre(min_U1);
125     ripple_2 := pre(max_U2) - pre(min_U2);
126     sum_dif_max_U1 := abs(last_last_max_U1 - last_max_U1) + abs(
          last_max_U1 - pre(max_U1));
127     sum_dif_max_U2 := abs(last_last_max_U2 - last_max_U2) + abs(
          last_max_U2 - pre(max_U2));
128     sum_dif_min_U1 := abs(last_last_min_U1 - last_min_U1) + abs(
          last_min_U1 - pre(min_U1));
129     sum_dif_min_U2 := abs(last_last_min_U2 - last_min_U2) + abs(
          last_min_U2 - pre(min_U2));
130     modSel1 := 1;
131     modSel2 := 1;
132     modSel := 1;
133     modSelChange := true;
134     if nPeriods > 3 then
135       //Chequeo si la se\~nal de control1 esta en regimen
136       if sum_dif_max_U1 + sum_dif_min_U1 < 2 * ripple_1 *
            deltaSignalControlToAv / 100 then
137         modSel1 := 0;
138       end if;
139       //Chequeo si la se\~nal de control2 esta en regimen
140       if sum_dif_max_U2 + sum_dif_min_U2 < 2 * ripple_2 *
            deltaSignalControlToAv / 100 then
141         modSel2 := 0;
142       end if;
```

```
143       //Indica que se debe pasar al modelo promediado
144       if modSel1 + modSel2 < 0.5 then
145         iL1Resetd := pre(iL1ave);
146         iL2Resetd := pre(iL2ave);
147         uC1Resetd := pre(uC1ave);
148         uC2Resetd := pre(uC2ave);
149         lastRt11d := pre(Rt11);
150         lastRt12d := pre(Rt12);
151         lastRt21d := pre(Rt21);
152         lastRt22d := pre(Rt22);
153         lastDeltaTNextStart := pre(deltaTNextStart);
154         lastDeltaTNext1 := pre(deltaTNext1);
155         lastDeltaTNext2 := pre(deltaTNext2);
156         timeToSwitched := -1;
157         newtimePeriod := 1e10;
158         //Corregir a numero grande
159         //         newHalfPeriod := 1e10;
160         //Corregir a numero grande
161         max_minCalcM.enable := 0;
162         lastTtoAv := time;
163         lastDeltaiL1 := iL1 - iL1Resetd;
164         lastDeltaiL2 := iL2 - iL2Resetd;
165         lastDeltauC1 := uC1 - uC1Resetd;
166         lastDeltauC2 := uC2 - uC2Resetd;
167         modSel := 0;
168         modSelChange := false;
169       end if;
170     end if;
171   end when;
172   //CONMUTACION A MODELO CONMUTADO
173   when max_minCalcM.u1 > max_U1 + 3 * ripple_1 *
        deltaSignalControlToAv / 100 then
174     timeToSwitched := lastTtoAv + (floor((time - lastTtoAv) / Tsw) +
          1) * Tsw;
175   end when;
176   when max_minCalcM.u2 > max_U2 + 3 * ripple_2 *
        deltaSignalControlToAv / 100 then
177     timeToSwitched := lastTtoAv + (floor((time - lastTtoAv) / Tsw) +
          1) * Tsw;
178   end when;
179   when max_minCalcM.u1 < min_U1 - 3 * ripple_1 *
        deltaSignalControlToAv / 100 then
```

```
180     timeToSwitched := lastTtoAv + (floor((time - lastTtoAv) / Tsw) +
            1) * Tsw;
181   end when;
182   when max_minCalcM.u2 < min_U2 - 3 * ripple_2 *
         deltaSignalControlToAv / 100 then
183     timeToSwitched := lastTtoAv + (floor((time - lastTtoAv) / Tsw) +
            1) * Tsw;
184   end when;
185   when time > timeToSwitched then
186     iL1Resetd := 1.4 * lastDeltaiL1 + iL1;
187     iL2Resetd := 0.55 * lastDeltaiL2 + iL2;
188     uC1Resetd := 1.55 * lastDeltauC1 + uC1;
189     uC2Resetd := 2 * lastDeltauC2 + uC2;
190     modSel := 1;
191     modSelChange := true;
192     nPeriods := 0;
193     max_minCalcM.enable := 1;
194   end when;
195   when modSel > 0.5 then
196     newtimePeriod := floor(time / 0.02) * 0.02 + 0.02;
197   end when;
198   when initial() then
199     modSel := 1;
200     modSelChange := true;
201     max_minCalcM.enable := 1;
202   end when;
203 end mode_selector;
```

A.1.2 Atómico *signalControlGenerator* en Modelica

```
1  model signalControlGenerator
2    Modelica.Blocks.Interfaces.RealInput u1 "se\~nal␣de␣entrada";
3    Modelica.Blocks.Interfaces.RealOutput y1 "Salida␣filtrada";
4    Modelica.Blocks.Interfaces.RealInput u2 "se\~nal␣de␣entrada";
5    Modelica.Blocks.Interfaces.RealOutput y2 "Salida␣filtrada";
6    atomicosNew.filtroPBOrd2fase0 filtroPBn11(fo = 50, y_start = 0);
7    atomicosNew.filtroPBOrd2fase0 filtroPBn12(fo = 50, y_start = 0);
8    atomicosNew.filtroPBOrd2fase0 filtroPBn21(fo = 50, y_start = 0);
9    atomicosNew.filtroPBOrd2fase0 filtroPBn22(fo = 50, y_start = 0);
10   atomicosNew.similEst similEst1;
11   atomicosNew.similEst similEst2;
```

```
12   atomicosNew.filtroOrden1 filtroPBorden1_1(y_start = 1, tau = 4 / (2
       * 3.141516 * 50));
13   atomicosNew.filtroOrden1 filtroPBorden1_2(y_start = 1, tau = 4 / (2
       * 3.141516 * 50));
14 protected
15   Real X1;
16   Real Xf1;
17   Real X2;
18   Real Xf2;
19 equation
20   //se\~nal1
21   filtroPBn11.u = u1;
22   X1 = filtroPBn11.y;
23   filtroPBn12.u = X1;
24   Xf1 = filtroPBn12.y;
25   similEst1.X = X1;
26   similEst1.Xf = Xf1;
27   filtroPBorden1_1.u = similEst1.y;
28   y1 = filtroPBorden1_1.y;
29   //se\~nal 2
30   filtroPBn21.u = u2;
31   X2 = filtroPBn21.y;
32   filtroPBn22.u = X2;
33   Xf2 = filtroPBn22.y;
34   similEst2.X = X2;
35   similEst2.Xf = Xf2;
36   filtroPBorden1_2.u = similEst2.y;
37   y2 = filtroPBorden1_2.y;
38 end signalControlGenerator;
```

A.1.3 Atómico filtroPBOrd2fase0 en Modelica

```
1  model filtroPBOrd2fase0
2    //implementa el filtro de primer orden y ganancia unitaria:
3    //              1
4    //     Y(s)= --------
5    //           1+tau.s
6    // Conectores de entrada y salida
7    Modelica.Blocks.Interfaces.RealInput u "se\~nal␣de␣entrada";
8    Modelica.Blocks.Interfaces.RealOutput y "Salida␣filtrada";
9    parameter Real fo = 50 "Frecuencia␣central␣␣[Hz]";
10   parameter Real y_start = 0 "Valor␣inicial␣de␣salida";
```

```
  constant Real pi = 3.1415926535 "Valor de pi";
protected
  parameter Real b = 1 / (pi * fo);
  parameter Real a0 = 1 / (pi * fo);
  parameter Real a1 = 1 / (2 * pi * fo) ^ 2;
  // Real y_int(start = y_start) "Variable interna de estado";
  Real x1;
  Real x2;
equation
  der(x1) = (x2 + b * u) / a1;
  der(x2) = -x1 - a0 * ((x2 + b * u) / a1);
  y = x1;
end filtroPBOrd2fase0;
```

A.1.4 Atómico *similEst* en Modelica

```
model similEst
  //Evalua cuan similares son dos se\~nales X y Xf calculando
  //           1 + Xf
  //      y= ---------
  //           1 + X
  Modelica.Blocks.Interfaces.RealInput X "se\~nal de entrada";
  Modelica.Blocks.Interfaces.RealInput Xf "se\~nal de entrada";
  Modelica.Blocks.Interfaces.RealOutput y "Salida filtrada";
protected
  Real num;
  Real den;
  Real y_int "Variable interna de estado";
equation
  num = 100 + X * X;
  den = 100 + Xf * Xf;
  y_int = den / num;
  y = y_int;
end similEst;
```

A.1.5 Atómico *filtroOrden1* en Modelica

```
model filtroOrden1
  //implementa el filtro de primer orden y ganancia unitaria:
  //              1
  //     Y(s)= --------
  //           1+tau.s
```

```
  // Conectores de entrada y salida
  Modelica.Blocks.Interfaces.RealInput u "se\~nal de entrada";
  Modelica.Blocks.Interfaces.RealOutput y "Salida filtrada";
  parameter Real tau = 0.1 "Constante de tiempo [s]";
  parameter Real y_start = 0 "Valor inicial de salida";
protected
  Real y_int(start = y_start) "Variable interna de estado";
equation
  der(y_int) = (u - y_int) / tau;
  y = y_int;
end filtroOrden1;
```

A.1.6 Atómico *global max min fun* en Modelica

```

model global_max_min_fun
  Modelica.Blocks.Interfaces.RealInput u1 "se\~nal de entrada 1";
  Modelica.Blocks.Interfaces.RealInput u2 "se\~nal de entrada 2";
  Modelica.Blocks.Interfaces.BooleanInput resetSignal_in(start = false
      ) "resetea ymax=ymin=u";
  Modelica.Blocks.Interfaces.RealInput enable(start = 1) "Abilita la
      deteccion de maximos y minimos";
  Modelica.Blocks.Interfaces.RealOutput y1_max "Maximo global de u";
  Modelica.Blocks.Interfaces.RealOutput y2_max "Maximo global de u";
  Modelica.Blocks.Interfaces.RealOutput y1_min "Maximo global de u";
  Modelica.Blocks.Interfaces.RealOutput y2_min "Maximo global de u";
  atomicosNew.max_min_search_fun max_min_calc_1;
  atomicosNew.max_min_search_fun max_min_calc_2;
  Boolean resetSignal(start = false);
  discrete Real squareReset(start = 0);
  //Onda cuadrada que resetea cuando cambia
protected
  discrete Real maxd1;
  discrete Real maxd2;
  discrete Real mind1;
  discrete Real mind2;
equation
  max_min_calc_1.u = u1;
  max_min_calc_2.u = u2;
  max_min_calc_1.enable = enable;
  max_min_calc_2.enable = enable;
  y1_max = maxd1;
```

```
27   y2_max = maxd2;
28   y1_min = mind1;
29   y2_min = mind2;
30   resetSignal = resetSignal_in;
31   max_min_calc_1.resetSignal = squareReset;
32   max_min_calc_2.resetSignal = squareReset;
33 algorithm
34   when change(squareReset) then
35     maxd1 := u1;
36     if maxd1 < max_min_calc_1.y_max then
37       maxd1 := max_min_calc_1.y_max;
38     end if;
39     maxd2 := u2;
40     if maxd2 < max_min_calc_2.y_max then
41       maxd2 := max_min_calc_2.y_max;
42     end if;
43     mind1 := u1;
44     if mind1 > max_min_calc_1.y_min then
45       mind1 := max_min_calc_1.y_min;
46     end if;
47     mind2 := u2;
48     if mind2 > max_min_calc_2.y_min then
49       mind2 := max_min_calc_2.y_min;
50     end if;
51   end when;
52 end global_max_min_fun;
```

A.1.7 Atómico *max min search fun* en Modelica

```
1  model max_min_search_fun
2    Modelica.Blocks.Interfaces.RealInput u "se\~nal␣de␣entrada";
3    Modelica.Blocks.Interfaces.RealOutput y_max "Maximo␣global␣de␣u";
4    Modelica.Blocks.Interfaces.RealOutput y_min "Minimo␣global␣de␣u";
5    Modelica.Blocks.Interfaces.RealInput enable(start = 1) "Abilita␣la␣
       deteccion␣de␣maximos␣y␣minimos";
6    discrete Real resetSignal;
7  protected
8    Real dU;
9    Boolean pendienteNegativa;
10   Boolean lastPte;
11   discrete Real maxU(start = 0);
12   discrete Real minU(start = 2);
```

```
  Real X;
equation
  pendienteNegativa = dU < 0;
  y_max = pre(maxU);
  y_min = pre(minU);
  der(X) = 10000 * (u - X) * enable;
  dU = 1000 * (u - X) * enable;
algorithm
  when change(resetSignal) then
    maxU := pre(u);
    minU := pre(u);
  end when;
  when pendienteNegativa then
    lastPte := false;
    if maxU < u then
      maxU := u;
    end if;
  end when;
  when not pendienteNegativa then
    lastPte := true;
    if minU > u then
      minU := pre(u);
    end if;
  end when;
end max_min_search_fun;
```

A.1.8 Atómico saturaDC en Modelica

```
function saturaDC
  input Real dc1;
  input Real dc2;
  input Real Tsw;
  input Real limInf;
  input Real limSup;
  output Real dcSat1;
  output Real dcSat2;
algorithm
  dcSat1 := dc1;
  if dcSat1 < limInf then
    dcSat1 := limInf;
  end if;
  if dcSat1 > limSup then
```

```
15      dcSat1 := limSup;
16    end if;
17    dcSat2 := dc2;
18    if dcSat2 < limInf then
19      dcSat2 := limInf;
20    end if;
21    if dcSat2 > limSup then
22      dcSat2 := limSup;
23    end if;
24  end saturaDC;
```

A.1.9 Atómico `filtroArmonicosG` en Modelica

```
1   model filtroArmonicosG
2     Real u[4];
3     //Entrada
4     Real y[4];
5     //Salida
6     Modelica.Blocks.Interfaces.RealOutput yf1;
7     Modelica.Blocks.Interfaces.RealOutput yf2;
8     Modelica.Blocks.Interfaces.RealOutput yf3;
9     Modelica.Blocks.Interfaces.RealOutput yf4;
10    Real modeSel;
11    //Entrada
12    Modelica.Blocks.Interfaces.BooleanInput newPWMperiod(start = false);
13    parameter Real Ts = 1 / 10000 "Periodo de conmutacion de la PWM";
14    Real meanU[4];
15    discrete Real y2[4];
16    discrete Real y1[4];
17    discrete Real y0[4];
18    discrete Real t2(start = 1);
19    discrete Real t1(start = 1);
20    discrete Real t0(start = 1);
21    discrete Real timeReset(start = 1);
22    Real tau(start = 1);
23    discrete Boolean calculationsDone(start = false);
24  equation
25    tau = (time - t1) * modeSel;
26    for i in 1:4 loop
27      der(meanU[i]) = u[i] * modeSel;
28      y[i] = pre(y0[i]) * (tau * (tau + Ts)) / (2 * Ts ^ 2) - y1[i] * ((
          tau - Ts) * (tau + Ts)) / Ts ^ 2 + y2[i] * ((tau - Ts) * tau)
```

```
        / (2 * Ts ^ 2);
  end for;
  yf1 = y[1];
  yf2 = y[2];
  yf3 = y[3];
  yf4 = y[4];
  when change(timeReset) then
    reinit(meanU[1], 0);
    reinit(meanU[2], 0);
    reinit(meanU[3], 0);
    reinit(meanU[4], 0);
  end when;
algorithm
  when newPWMperiod then
    t2 := t1;
    t1 := t0;
    t0 := time - Ts / 2;
    for i in 1:4 loop
      y2[i] := y1[i];
      y1[i] := y0[i];
      y0[i] := meanU[i] / Ts;
    end for;
    calculationsDone := true;
    timeReset := 1 - timeReset;
  end when;
end filtroArmonicosG;
```

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS