

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



PROYECTO FIN DE CARRERA

CONTROL DIGITAL DE FUENTES DE ALIMENTACIÓN

Ingeniería de Telecomunicación

Alejandro García Talón
Abril 2009

Resumen

Para cumplir con las normativas de emisión de armónicos, tales como la EN61000-3-2, en las fuentes de alimentación es necesario realizar corrección del factor de potencia. La implementación de la corrección del factor de potencia ha sido tradicionalmente analógica e implica el diseño de un lazo de tensión y otro de corriente. En este proyecto se describe un método para realizar un sistema de control de un convertidor elevador basado en un dispositivo programable FPGA. El funcionamiento del sistema se basa en calcular los valores del ciclo de trabajo con antelación e introducirlos en una memoria de la FPGA, a partir de los cuales se genera la señal modulada por ancho de pulso que se conecta con el elemento conmutador del convertidor. De modo que una de las principales ventajas de este sistema es que no se requiere ni medida ni lazo de corriente, además de que el coste computacional es muy bajo. Para realizar las medidas de las tensiones de salida y entrada se han diseñado e implementado conversores sigma-delta, que requieren pocos componentes analógicos externos, por lo que se pueden integrar con el sistema de control. En este documento se analiza la técnica empleada y las ventajas y aspectos críticos que este método presenta, a través de simulaciones y pruebas experimentales.

Abstract

In order to comply with the requirements of harmonics emissions standards, such as EN61000-3-2, in power supplies, power factor correction is needed. The controllers implemented for power factor correction have been typically analog, and imply the design of both current and voltage loops. This project describes a method for power factor correction on a boost converter based on a programmable FPGA device. System operation is based on calculating in advance the duty cycle values and storing them in a memory of the FPGA, from which the pulse width modulation signal is generated, that is connected with the boost switching device. Hence, one of the main advantages of this technique is that neither current loop nor current measurement are required, and the computational cost is very low. To measure the input and output voltages, sigma delta analog-to-digital converters have been designed and implemented, which require few analog external components, and can be easily integrated with the control system. This documents discusses this technique and its advantages and critical issues through simulations and experimental tests.

Agradecimientos

Quisiera agradecer este proyecto a mis padres y a mi hermana, por todo el apoyo y motivación que me han dado a lo largo de los años de carrera, y por todo el amor y dedicación que me han brindado siempre.

Muchas gracias a Noe por darme todo su cariño y apoyo, por ayudarme en los malos momentos, reír conmigo en los buenos, y por haber compartido conmigo los mejores años de carrera y de mi vida.

Muchas gracias a mi tutor, Ángel, por toda la ayuda prestada, por estar siempre disponible para resolver mis dudas, y por haber sido además de un buen tutor, un buen amigo.

Quiero agradecer también a mis compañeros de laboratorio, Alberto, David, Fernando, Javier y Ricardo por su ayuda y por los buenos momentos que he pasado con ellos. También quiero agradecer a Guillermo y a Óscar la ayuda que me han dado con la parte más hardware de este proyecto.

Y a los buenos compañeros y amigos que he hecho durante la carrera, en especial a Héctor, Isa, Jesús Muñecas, Jesús Zamarreño, Sergio y Víctor, con los que he pasado grandes momentos y gracias a ellos, estudiar durante tantos años ha sido más sencillo.

Índice general

Índice de figuras	x
1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos y alcance del proyecto	2
1.3. Organización de la memoria	2
2. Estado de la Técnica	3
2.1. Técnicas de conversión alterna continua	3
2.1.1. Topologías de convertidores conmutados	4
2.2. Control y corrección del factor de potencia	5
2.2.1. Corrección de factor de potencia y control: doble lazo	7
2.3. Sistemas embebidos	7
2.3.1. Alternativas en la implementación de sistemas embebidos	8
3. Conversor analógico digital sigma-delta	11
3.1. Conversión analógica digital	11
3.2. Tasa de muestreo y aliasing	12
3.3. Tipos de convertidores analógico digitales	13
3.4. Modulación sigma delta	14
3.4.1. Ruido en convertidores sigma delta	16
3.5. Diseño del conversor	16
3.5.1. Versión básica	17
3.5.2. Versión mejorada	17
3.6. Simulaciones	18
3.7. Diseño y construcción del PCB del conversor	19
3.8. Resultados experimentales	21
3.8.1. Análisis paramétrico	22

4. Etapa de potencia	25
4.1. Diseño del convertidor utilizado	25
4.1.1. Cálculo de pérdidas en semiconductores	28
4.1.2. Diseño y construcción del PCB del convertidor elevador	29
5. Control de corriente: ciclo de trabajo precalculado	31
5.1. Cálculo del ciclo de trabajo	31
5.1.1. Modo de conducción continuo y discontinuo	34
5.2. Simulaciones y análisis paramétrico	35
5.2.1. Efecto de la cuantificación y limitación del ciclo de trabajo	35
5.2.2. Efecto de la potencia de salida	36
5.2.3. Efecto de la bobina	37
5.2.4. Efecto de la variación del condensador en la corriente de entrada	37
5.3. Implementación del diseño en VHDL	38
5.3.1. Protecciones en el ciclo de trabajo	41
6. Sincronización	43
6.1. Diseño del mecanismo de sincronización	43
6.2. Implementación y simulaciones	45
6.3. Resultados experimentales	47
7. Sistema de control en lazo cerrado de tensión	49
7.1. Introducción a los sistemas de control: regulación automática	49
7.1.1. Fundamentos matemáticos de los sistemas de control	50
7.1.2. Ecuación en diferencias	51
7.1.3. Sistemas de control en lazo cerrado	51
7.1.4. Elementos de control	52
7.2. Implementación digital de funciones de transferencia	53
7.3. Diseño e implementación del lazo de tensión	54
8. Resultados experimentales. Sistema completo	59
8.1. Descripción de las pruebas	59
8.2. Sistema en lazo abierto	61
8.3. Efecto de la sincronización	62
8.4. Sistema en lazo cerrado	63

9. Conclusiones y trabajo futuro	67
9.1. Conclusiones	67
9.2. Trabajo futuro	68
9.3. Publicaciones derivadas de este proyecto	68
Esquemáticos	I
Código VHDL. Sistema de control	III
Código VHDL. Modelos digitales	XIX
Presupuesto	XXIII
Pliego de condiciones	XXV

Índice de figuras

2.1. Topologías típicas de convertidores: a) elevador. b) reductor. c) elevador-reductor	4
2.2. Emulador de resistencia: corrector del factor de potencia ideal	6
2.3. Corriente, tensión y potencia en CFP	6
2.4. Lazo de tensión y corriente para PFC	7
2.5. arquitectura basica de una Spartan 3	8
3.1. Esquema del proceso de conversión analógico digital	12
3.2. Esquema de un ADC de aproximaciones sucesivas	13
3.3. Esquema de un ADC tipo flash	14
3.4. Esquema de un ADC de tipo integrador de rampa doble	14
3.5. Señal sinusoidal modulada sigma-delta	15
3.6. Esquema básico de un modulador sigma-delta	16
3.7. Densidad espectral de potencia de ruido con oversampling y noise shaping	16
3.8. Diagrama de bloques de la primera versión del conversor sigma delta . .	17
3.9. Diagrama de bloques de la versión final del conversor sigma delta	18
3.10. Simulaciones del efecto del retardo y la histéresis del comparador	19
3.11. Fotografía de la placa construida para los conversores sigma-delta	20
3.12. Fotografía del montaje del conversor en una placa de prototipos	21
3.13. diseño del PCB del conversor sigma delta	21
3.14. Linealidad en función de N_{cycles}	22
3.15. Linealidad en función de τ	23
4.1. Esquema electrico del convertidor elevador	25
4.2. Corrientes de subida y bajada en la bobina de un boost	27
4.3. Captura del programa PExprt	29
4.4. Diseño del PCB del convertidor elevador	29
4.5. Fotografía del prototipo de boost construido	30
5.1. Corrientes en un convertidor elevador	32

5.2. Componentes del ciclo de trabajo	34
5.3. Corrientes en modo de conducción continuo y discontinuo	34
5.4. Efecto de la cuantificación y limitación del ciclo de trabajo en corriete de entrada	36
5.5. Efecto de la variación de la potencia entregada en la corriente de entrada	37
5.6. Efecto de la variación del valor de la bobina en la corriente de entrada . .	38
5.7. Efecto del condensador en la corriente de entrada	39
5.8. Diagrama de bloques: señal PWM	39
5.9. Generación de la señal PWM	40
5.10. La ultima dirección se mantiene hasta que se activa la señal de restart . .	41
6.1. Caso ideal y real de la tensión de entrada rectificada	44
6.2. Detección de falsos flancos de subida o bajada	44
6.3. Esquema del mecanismo de sincronización	45
6.4. Simulación del sistema de sincronización	46
6.5. Esquema de las operaciones para el cálculo de R	47
6.6. Representación de la desviación de la señal de sincronismo	47
6.7. Captura del desviamiento de la señal de restart	48
6.8. Captura de la señal de sincronización	48
7.1. Digrama de bloques para un ejemplo de función de transferencia	54
7.2. Esquema de la función de transferencia del elemento de control	56
7.3. Diagrama de bloques del elemento de control y el bloque PWM	56
8.1. Esquema del montaje para las pruebas experimentales	60
8.2. Fotografía de las cargas construidas y empleadas para las pruebas experimentales	60
8.3. Capturas del sistema completo para tensiones de entrada de 33 V y 55 V. Tensión de entrada (amarillo), y de salida (verde) (50 V/div.), corriente de entrada (morado) (1.06 A/div.) y señal de sincronismo (rojo)	61
8.4. Capturas del sistema completo para tensiones de 130 V, 200 V, 210 V y 220 V. Tensión de entrada (amarillo), y de salida (verde) (50 V/div.), y corriente de entrada (morado) (1.06 A/div.)	62
8.5. Efecto del desfase de sincronismo en la corriente de entrada	63
8.6. Capturas del sistema completo. Comparativa con el lazo activado y sin activar	64
8.7. Catura del transitorio cuando se activa el lazo de tensión	65

1

Introducción

1.1. Motivación del proyecto

Los convertidores, en sus diferentes topologías, son dispositivos eléctricos que realizan una conversión de la energía que se les aplica a la entrada hacia la carga de salida. Los convertidores conmutados necesitan señales de control que actúen sobre los interruptores de la fuente, permitiendo así la conversión adecuada de energía entre la entrada y la salida.

En las últimas décadas los dispositivos eléctricos han experimentado una fuerte transición de lo analógico a lo digital debido a las ventajas que presenta:

- Reproducibilidad de resultados
- Facilidad de diseño
- Flexibilidad y funcionalidad
- Programabilidad
- Velocidad
- Avance tecnológico constante
- Coste

Aunque no siempre se ha producido esta transición de manera rápida y sencilla debido a una mayor complejidad en el diseño o un aumento del coste. Unos de los dispositivos que tradicionalmente emplea circuitería analógica son los sistemas de control de las fuentes conmutadas. En esta línea de evolución del mundo analógico al digital surge este proyecto, en el cual se busca realizar un sistema de control de un convertidor elevador con corrección de factor de potencia de manera digital basado en FPGA, intentando minimizar el coste, complejidad, tamaño y componentes necesarios.

1.2. Objetivos y alcance del proyecto

El objetivo principal de este proyecto es diseñar e implementar un sistema de control digital basado en FPGA para un convertidor elevador con corrección de factor de potencia **sin medida de corriente**.

Debido a que se requiere monitorizar diversas señales (tensión de entrada y salida) se hace necesario realizar una conversión de las señales analógicas a señales digitales. Por lo tanto, otro de los objetivos del proyecto es diseñar e implementar conversores analógico digitales que puedan integrarse con el sistema de control. Para minimizar el coste en componentes analógicos, y aprovechando que el sistema de control se realiza sobre un dispositivo programable (FPGA), se busca un diseño específico que requiera un número mínimo de componentes analógicos, aunque suponga una mayor utilización del dispositivo programable o un incremento en el coste computacional.

El sistema de control consiste en inyectar a un transistor del convertidor elevador una señal con modulación de ancho de pulso. Conociendo de antemano la señal modulada necesaria en cada semiciclo de tensión para conseguir una corrección del factor de potencia, se puede evitar la medida de corriente o el empleo de algoritmos de reconstrucción e inyectar dicha señal de manera directa cuyos valores han sido previamente calculados y almacenados en una memoria. Para que funcione correctamente, los valores de esta señal modulada y la tensión de entrada deben estar sincronizados, por lo que otro de los objetivos es diseñar e implementar, también en la FPGA, un sistema de sincronización, que como veremos posteriormente, es un elemento crítico del diseño. Para mantener la tensión de salida a un nivel estable se requiere también diseñar un lazo de control que mida la tensión de salida y actúe sobre los valores pre calculados del ciclo de trabajo. Dicho lazo se implementa también en la FPGA.

De esta manera se busca realizar un diseño compacto e integrado basado en FPGA con un número mínimo de componentes, lo que permitirá reducir su tamaño y coste. Para poder realizar una verificación experimental del diseño se han diseñado y construido las placas de circuito impreso del convertidor elevador y de los componentes analógicos de los ADC.

1.3. Organización de la memoria

La memoria se organiza en general de manera que cada capítulo corresponde a un bloque funcional del diseño. Tras un capítulo de introducción a los conceptos teóricos de los convertidores conmutados y la corrección del factor de potencia se presentan la solución adoptada para la conversión analógica digital. El siguiente capítulo se centra en el diseño y construcción de la etapa de potencia. A continuación se hace un estudio de la técnica de control basada en el ciclo de trabajo precalculado y la forma de generar la señal de control. El capítulo siguiente aborda el mecanismo de sincronización, la forma en que se ha diseñado y la comprobación de su funcionamiento. En el penúltimo capítulo, tras una introducción teórica a los sistemas de control realimentados y sus fundamentos matemáticos, se estudia la forma de implementarlos en un dispositivo digital programable y la solución adoptada para el lazo de tensión. Finalmente se presentan las conclusiones extraídas del proyecto y las posibles líneas de trabajo futuro.

2

Estado de la Técnica

2.1. Técnicas de conversión alterna continua

Todos los sistemas eléctricos necesitan de una fuente de energía para funcionar. Para proporcionársela se utilizan fuentes de alimentación que se basan en transformar la energía que se les aplica a su entrada para obtener valores apropiados en su salida. Podemos distinguir entre dos tipos de fuentes de alimentación:

- Reguladores lineales: Se basan en transformar el voltaje de entrada en uno menor a la salida mediante la disipación de la tensión sobrante en forma de calor.
- Fuentes conmutadas (o convertidores): Su funcionamiento se basa en la conmutación de transistores. En función de una señal de control que impone un ciclo de trabajo determinado el transistor entra en conmutación permitiendo o no el paso de energía a los distintos componentes de la fuente.

Las fuentes conmutadas presentan diversas ventajas respecto a los reguladores lineales:

- Eficiencia, calor y energía disipada: como se ha comentado, los reguladores lineales se basan en la disipación de energía excedente en forma de calor, mientras que en las fuentes conmutadas la regulación se basa en una señal de control en la que idealmente no se produce ninguna disipación, y teóricamente se puede alcanzar una eficiencia del 100 %. Las pérdidas producidas son por las características no ideales de los componentes.
- Voltaje de salida: los reguladores lineales sólo pueden proporcionar una tensión mas baja que la que se presenta en la entrada, mientras que en una fuente conmutada el valor a la salida puede ser mayor o menor que el voltaje aplicado a la entrada
- Relación peso-potencia: los reguladores lineales utilizan transformadores de bajas frecuencias, los cuales son considerablemente más grandes y pesados que los que usados típicamente en una fuente conmutada de frecuencias superiores. La tendencia

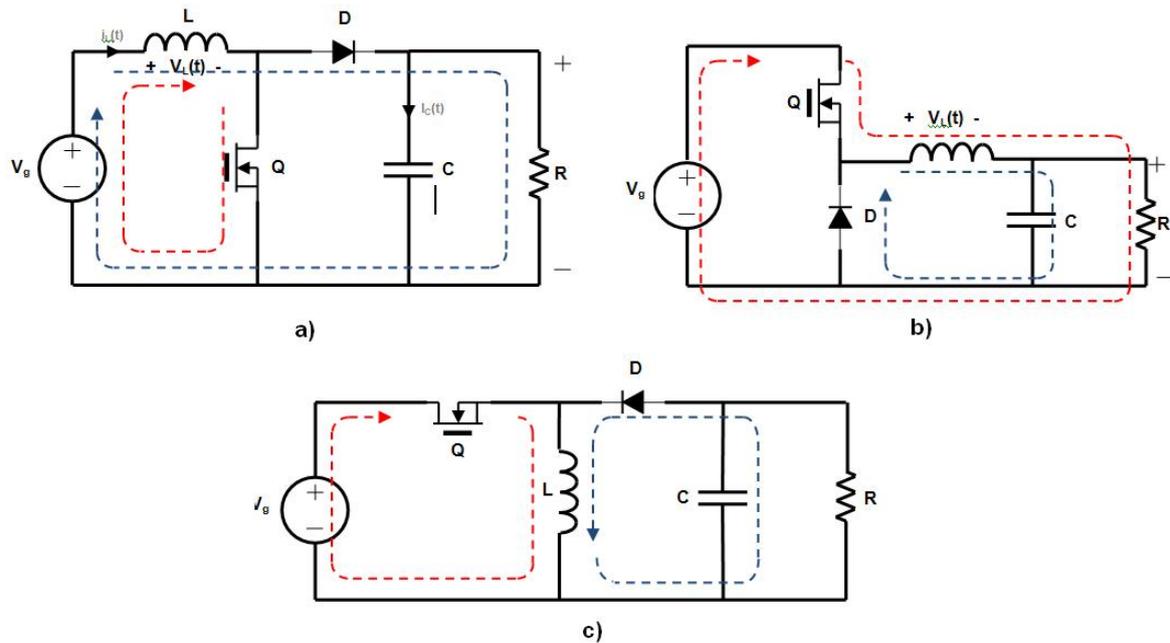


Figura 2.1: Topologías típicas de convertidores: a) elevador. b) reductor. c) elevador-reductor

es usar frecuencias lo más altas posibles mientras los transistores de la fuente lo permitan, para disminuir el tamaño de los condensadores, bobinas y transformadores.

- Factor de potencia (explicado en el apartado 2.2): en un regulador lineal el factor de potencia resultante es bajo pero con un diseño adecuado las fuentes conmutadas pueden ofrecer la regulación del factor de potencia.

Pero la utilización de las fuentes conmutadas también tienen ciertos inconvenientes respecto a los reguladores lineales. Presentan una mayor complejidad de diseño ya que es necesario incluir una etapa de control que genere la señal modulada por ancho de pulso que se inyecta al transistor. Típicamente esta señal se genera mediante un dispositivo de tipo controlador (generalmente DSP), pero en este proyecto, como se verá mas adelante, para realizar el control se ha utilizado un dispositivo FPGA. Además, debido a los cambios abruptos de corrientes que se dan en un convertidor conmutado, se pueden generar radiaciones electromagnéticas indeseadas, por lo que a veces es necesario emplear filtros EMI y blindajes externos para reducir las interferencias. Como las señales de conmutación son de altas frecuencias es necesario un diseño adecuado de las pistas del circuito de la fuente, ya que las capacidades parásitas no son despreciables.

2.1.1. Topologías de convertidores conmutados

Existen muchas topologías diferentes para los convertidores elevadores, pero las más usadas son el convertidor elevador, el reductor y el reductor-elevador (en inglés *boost*, *buck*, y *buck-boost* respectivamente). Se diferencian entre sí en la disposición de los elementos del circuito, teniendo cada uno distintos caminos de carga y descarga de la bobina. Todas ellas incluyen una bobina, un interruptor (generalmente un transistor MOSFET), una bobina y un condensador de salida. En la figura 2.1 podemos ver un esquema de las tres topologías y los caminos de carga (rojo) y descarga (azul) de la bobina:

2.2. Control y corrección del factor de potencia

Uno de los objetivos principales de este proyecto es realizar un convertidor conmutado de alterna - continua con corrección de factor de potencia (*power factor correction*), lo que implica introducir la menor cantidad posible de armónicos. El factor de potencia lo podemos definir como el cociente entre la potencia activa y la potencia aparente presentes en un convertidor, o el coseno del ángulo entre las tensión y la corriente cuando la forma de onda es sinusoidal pura:

$$PF = \frac{\text{potencia activa}}{\text{potencia aparente}} = \frac{\frac{1}{T} \int_0^T u \cdot i \cdot dt}{\sqrt{\frac{1}{T} \int_0^T i^2 \cdot dt} \sqrt{\frac{1}{T} \int_0^T u^2 \cdot dt}}$$

Si la tensión de entrada es sinusoidal, como es el caso que nos ocupa, el factor de potencia puede definirse como:

$$PF = \frac{V_{ef} \cdot I_{ef1} \cdot \cos(\theta)}{V_{ef} \cdot I_{ef}} = \frac{I_{ef1}}{I_{ef}} \cos(\theta) = K_d \cdot K_\theta$$

donde I_{ef1} se refiere a la corriente eficaz del primer armónico, K_d es el factor de distorsión y K_θ es el factor de desplazamiento.

Realizar la corrección del factor de potencia presenta la ventaja de que la mayor parte de la potencia es activa y la distorsión de la tensión se reduce, además de que presenta menores corrientes eficaces y menores pérdidas, aunque también existen desventajas, tales como un mayor coste y complejidad del convertidor y una menor fiabilidad del sistema.

Para obtener corrección del factor de potencia el convertidor debe comportarse idealmente como una resistencia, de manera que ante una tensión de entrada sinusoidal demandara una corriente de entrada también sinusoidal en fase con la tensión (figura 2.2). En este caso la potencia de entrada es una senoide de valor medio $V_{in\ pico}^2 / (2 \cdot R_{in})$ y del doble de frecuencia que la tensión:

$$V_{in} = V_{in\ pico} \cdot \text{sen}(\omega t)$$

$$I_{in}(t) = \frac{V_{in\ pico}}{R_{in}} \cdot \text{sen}(\omega t)$$

$$P_{in}(t) = V_{in}(t) \cdot I_{in}(t) = \frac{V_{in\ pico}^2}{2 \cdot R_{in}} \cdot (1 - \cos(2\omega t))$$

Podemos ver estas señales en la figura 2.3. Para poder proporcionar a la salida una potencia constante se utiliza un condensador de gran capacidad al final del convertidor, gracias al cual la tensión de salida es prácticamente constante y por tanto la potencia y la corriente de salida. En este caso la potencia de salida es constante e igual al valor medio de la de entrada:

$$P_{out}(t) = \frac{V_{in\ pico}^2}{2 \cdot R_{in}}$$

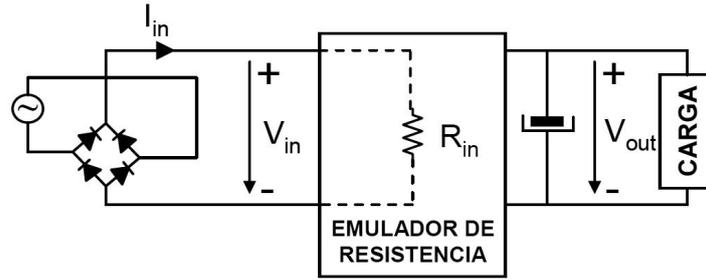


Figura 2.2: Emulador de resistencia: corrector del factor de potencia ideal

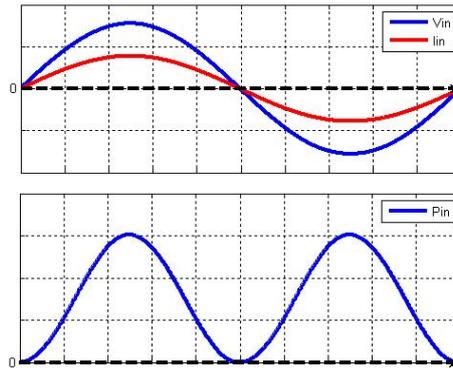


Figura 2.3: Corriente, tensión y potencia en CFP

El condensador de salida mantiene la potencia constante suministrando o absorbiéndola en cada momento según la siguiente expresión:

$$P_c(t) = P_{out}(t) - P_{in}(t) = \frac{V_{in\ pico}^2}{2 \cdot R_{in}} \cdot \cos(2\omega t)$$

sabiendo que la corriente en el condensador es prácticamente constante e igual a:

$$I_c(t) = \frac{P_c(t)}{V_{out}} = \frac{V_{in\ pico}^2}{2 \cdot R_{in} \cdot V_{out}} \cdot \cos(2\omega t)$$

podemos obtener la tensión en el condensador:

$$V_{out}(t) = V_{out} + \frac{V_{in\ pico}^2}{4 \cdot R_{in} \cdot V_{out} \cdot \omega \cdot C} \text{sen}(2\omega t) = V_{out} + \frac{P_{out}}{2 \cdot V_{out} \cdot \omega \cdot C}$$

Por lo tanto la tensión de salida tiene un rizado del doble de frecuencia que la tensión de entrada y con una amplitud pico a pico:

$$\Delta V_{out} = \frac{P_{out}}{\omega \cdot C \cdot V_{out}}$$

Se debe tener en cuenta que para corregir el factor de potencia este rizado debe existir. Cuanto mayor sea el condensador de salida menor será el rizado de la tensión, pero

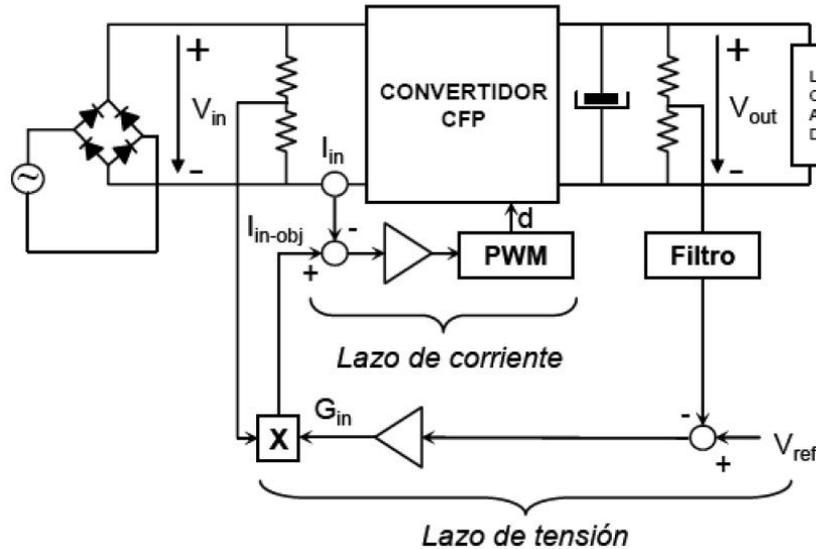


Figura 2.4: Lazo de tensión y corriente para PFC

cualquier técnica de corrección de factor de potencia debe respetar y tener presente este rizado.

2.2.1. Corrección de factor de potencia y control: doble lazo

Para realizar el control del convertidor con corrección del factor de potencia se emplea típicamente un doble lazo de tensión y de corriente. En la figura 2.4 se puede ver un esquema de funcionamiento del doble lazo. El lazo de tensión se encarga de medir la tensión a la salida y compararla con un cierto valor de consigna. El valor del error obtenido sirve para actuar sobre el lazo de corriente, el cual corrige el ciclo de trabajo medio que se impone en el elemento de conmutación. El lazo de corriente tiene como objetivo controlar la corriente que circula por el convertidor para que sea proporcional a la tensión de entrada y conseguir así la corrección del factor de potencia. El lazo de tensión tiene una dinámica lenta en comparación al lazo de corriente.

Debido a que la medida de corriente es un proceso costoso y complejo si se realiza de manera precisa se han buscado diversas alternativas. Por ejemplo, en [2] se realiza un sistema que permite la reconstrucción de los valores de la corriente mediante un algoritmo en cada ciclo de conmutación. En este proyecto se propone una técnica que no necesita de la medida de la corriente ni de su reconstrucción, sino que los valores del ciclo de trabajo adecuados se calculan de antemano, y sólo es necesario el lazo de tensión.

2.3. Sistemas embebidos

La forma de realizar el control descrito en el apartado anterior se ha hecho tradicionalmente de forma analógica. En este proyecto el control realizado es un sistema embebido basado en un dispositivo programable FPGA.

Un sistema embebido es un sistema digital que permite realizar diferentes funciones

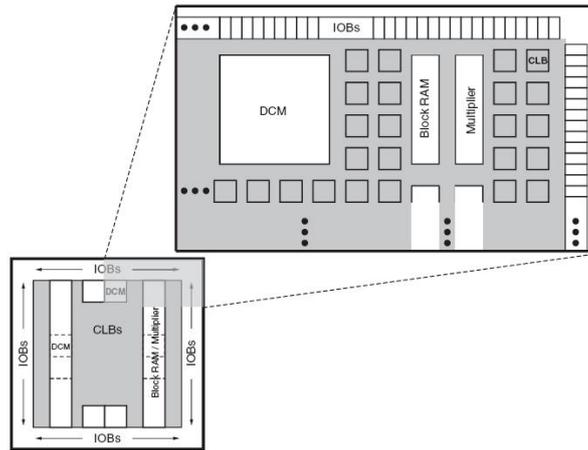


Figura 2.5: arquitectura básica de una Spartan 3

diseñadas específicamente para un propósito determinado, de manera transparente para el usuario, como lo es el sistema de control que se plantea en este proyecto. Por el contrario, los sistemas no embebidos, como por ejemplo un computador personal pueden realizar diferentes funciones dependiendo de la programación del mismo. Ejemplos de sistemas embebidos son la electrónica de control de un automóvil, o un reproductor MP3. Las instrucciones que se escriben para programar un sistema embebido se conocen como firmware, y se suelen almacenar en memorias flash externas.

2.3.1. Alternativas en la implementación de sistemas embebidos

Los sistemas embebidos suelen ser implementados en una estructura SoC (*System on Chip*), la cual consiste en integrar todos los elementos necesarios, tales como procesadores, memorias, multiplicadores, interfaces, conversores o incluso electrónica analógica en un único circuito integrado. Las diferentes alternativas para realizar un sistema embebido son las siguientes.

- Microcontrolador: son dispositivos programables que operan de manera secuencial y que integran diversos periféricos como pueden ser conversores analógico digitales, interfaces de comunicaciones, temporizadores, etc. Funcionan ejecutando las instrucciones programadas de manera secuencial.
- FPGA (*field programmable gate array*): son dispositivos programables que surgen históricamente como una evolución de los conceptos aplicados en los CPLD (*complex programmable logic device*) que contienen bloques de lógica cuya interconexión y funcionalidad se puede programar. Pueden contener bloques específicos tales como multiplicadores, memorias e incluso microprocesadores embebidos.

La FPGA utilizada en este proyecto es una *Spartan 3* del fabricante *Xilinx*. Podemos ver en la figura 2.5 un diagrama de su arquitectura básica.

- ASIC (*application-specific integrated circuit*): son circuitos integrados diseñados y fabricados a medida para un propósito particular. Un ejemplo son los chips empleados en telefonía móvil. Al ser dispositivos totalmente a la medida de cada aplicación permiten optimizar el área y la funcionalidad, pero presentan la desventaja de que

requieren un mayor tiempo de desarrollo y fabricación, costos fijos mayores y herramientas de diseño CAD mas complejas, por lo que se emplean en la producción de grandes lotes para amortizar los costes. Un proceso común es migrar el diseño de una FPGA a un ASIC.

Los sistemas de control de convertidores conmutados de naturaleza digital están basados generalmente en un microcontrolador. En este proyecto se ha adoptado como solución un dispositivo FPGA debido a su gran proliferación, versatilidad, velocidad y la posibilidad de un reducido coste si se traslada el diseño a un ASIC en grandes producciones. Debido a que muchos procesos del sistema de control, como son la conversión analógica digital, o la generación de la señal PWM son concurrentes (se deben procesar simultáneamente en el tiempo), la utilización de una FPGA es más adecuada que un microcontrolador, en el que los procesos son secuenciales.

3

Conversor analógico digital sigma-delta

En este capítulo se hace un estudio sobre los conversores analógico digitales sigma delta. Se presenta una introducción teórica para posteriormente analizar cómo se han diseñado, las diferentes versiones creadas, las simulaciones hechas para verificar su correcto funcionamiento y el diseño y construcción del PCB de la parte analógica. Finalmente se presentan las pruebas experimentales y un estudio paramétrico del conversor creado.

3.1. Conversión analógica digital

El proceso de conversión digital de una señal analógica está compuesto por los siguientes pasos:

- **Muestreo:** es el proceso explicado en más detalle en el siguiente apartado. A partir de la señal continua se obtienen muestras de la misma con una frecuencia de muestreo que debe ser al menos superior a la de *Nyquist* para evitar el solapamiento espectral y poder representar correctamente la señal analógica a partir de sus muestras.
- **Retención:** es un paso previo a la cuantificación, en el cual las muestras se mantienen el tiempo suficiente mediante un circuito de retención (*hold*) para poder evaluar correctamente el nivel de cuantificación.
- **Cuantificación:** a partir de la señal muestreada en tiempo discreto se asigna a margen de tensiones de la entrada un único valor de la salida. Esto introduce en la conversión un error de cuantificación, que será menor cuanto mayor sea la resolución del conversor.
- **Codificación:** consiste en asignar a cada nivel de cuantificación el código binario que representa la tensión de entrada.

En la figura 3.1 podemos ver un esquema de este proceso.

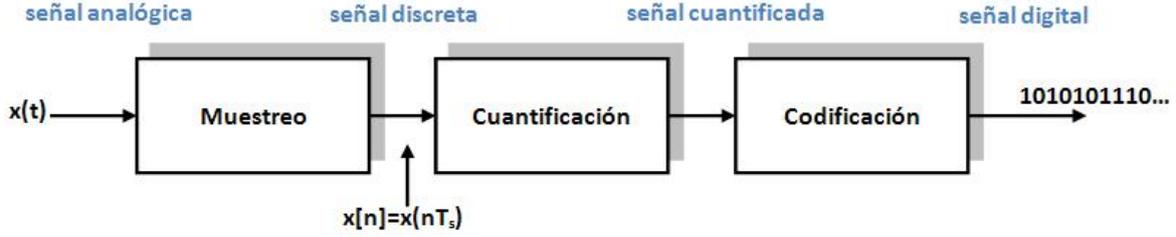


Figura 3.1: Esquema del proceso de conversión analógico digital

3.2. Tasa de muestreo y aliasing

Para poder obtener medidas sobre las señales analógicas y poder procesarlas en un sistema digital es necesario realizar una conversión. Para realizar la conversión primeramente es necesario un muestreo de la señal continua $x(t)$ a frecuencia $f_s = 1/T_s$, de forma que la señal muestreada es $x[n] = x_c(nT)$. La señal que se utiliza para realizar el muestreo es un tren de impulsos:

$$s(t) = \sum_{K=-\infty}^{\infty} \delta(t - KT)$$

Dado que esta señal es periódica y continua, utilizamos el desarrollo en serie de *Fourier* para hacer un estudio en el dominio transformado:

$$a_K = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} \delta(t) e^{-jK\Omega t} dt = \frac{1}{T_s} = f_s$$

de forma que la transformada en tiempo discreto de la señal $x[n]$ es:

$$X(j\Omega) = \sum_{K=-\infty}^{\infty} 2\pi a_K (\Omega - K\Omega_s) = \frac{2\pi}{T} \sum_{K=-\infty}^{\infty} \delta(\Omega - K\frac{2\pi}{T})$$

Podemos observar que la transformada del tren de impulsos periódico es otra señal periódica en el dominio frecuencial. Como la multiplicación en el dominio temporal es equivalente a la multiplicación en el dominio frecuencial, la transformada de la señal muestreada es por tanto:

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * \frac{2\pi}{T} \sum_{K=-\infty}^{\infty} X_c(j(\Omega - K\Omega_s))$$

donde $\Omega_s = \frac{2\pi}{T}$. De lo anterior podemos concluir que para poder recuperar la señal a partir de sus muestras se debe cumplir que $\Omega_s = \frac{2\pi}{T} > 2\Omega_N$, donde Ω_N es la frecuencia máxima de la señal discreta limitada en banda. Este teorema es conocido como **teorema de Nyquist** y nos indica la frecuencia de muestreo mínima para evitar el **aliasing** o solapamiento espectral.

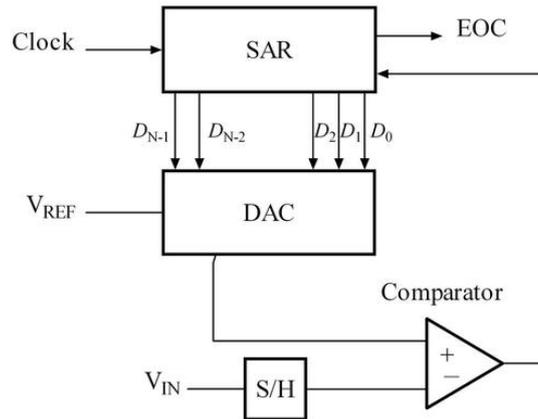


Figura 3.2: Esquema de un ADC de aproximaciones sucesivas

En el caso particular de un conversor sigma-delta la señal se sobremuestra a una frecuencia mucho mayor a la impuesta por *Nyquist* para evitar en mayor medida el solapamiento, disminuyendo la resolución de cada medida, pero aumentando la velocidad de muestreo. Esto permite utilizar filtros *antialiasing* mucho más sencillos (incluso de primer orden).

3.3. Tipos de conversores analógico digitales

Existen muchos tipos de conversores cada uno con sus ventajas e inconvenientes. Se presentan aquí los modelos más comunes y sus características principales:

- **De aproximaciones sucesivas** (Fig. 3.2): consta de un DAC, un comparador y un registro de aproximaciones sucesivas. Se inicializa a 1 el bit más significativo y se compara el valor de entrada con el valor del registro. Si es mayor el bit más significativo de salida se mantiene a 1, y así sucesivamente para el resto de los bits de salida. Es muy utilizado, ya que ofrece un buen compromiso entre velocidad y resolución, alcanzando velocidades de conversión de MHz con resoluciones de 12 a 16 bits.
- **Simultáneo o flash** (Fig. 3.3): está formado por tantos comparadores como niveles de cuantificación se requieran representar y un decodificador. Son muy rápidos pero presentan una gran complejidad por el alto número de comparadores requeridos. El modelo pipeline está basado en el mismo principio pero cuantificando los primeros $N/2$ bits más significativos, menos afectados por el ruido, estos bits se introducen a un DAC, restando su salida con la de entrada obteniendo así un voltaje residual, que se digitaliza de nuevo produciendo los restantes bits menos significativos. Este es más lento que el anterior pero reduce exponencialmente el número de comparadores necesarios.
- **Tipo integrador** (Fig. 3.4): se basan en acumular la tensión de entrada durante un tiempo determinado y compararla con un valor de referencia. Presentan gran inmunidad al ruido superpuesto a la señal a convertir (siempre que la frecuencia del ruido sea mayor que la de integración) y una gran linealidad, pero el tiempo de conversión es muy lento. Pueden ser de rampa simple o doble.

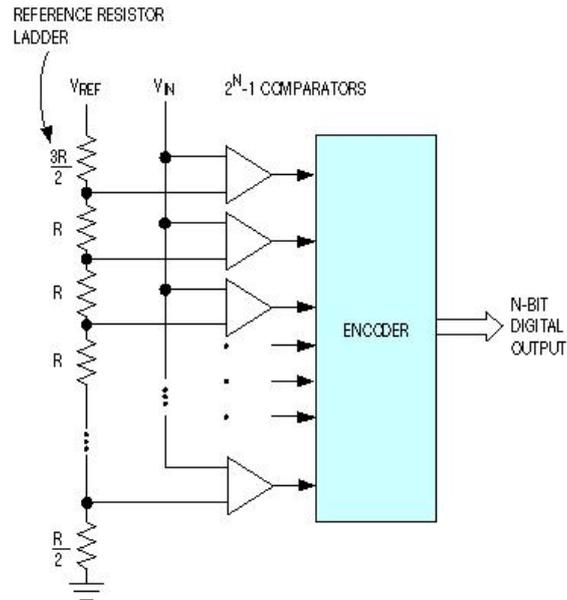


Figura 3.3: Esquema de un ADC tipo flash

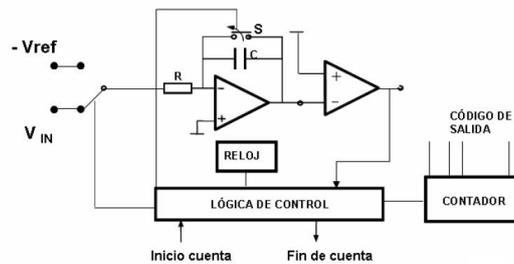


Figura 3.4: Esquema de un ADC de tipo integrador de rampa doble

- **Convertidores sigma-delta:** se explican en el apartado siguiente y son los utilizados en este para realizar las conversiones.
- **Híbridos:** combinan alguna de las técnicas anteriores, por ejemplo utilizando un tipo flash para los bits menos significativos, y uno de aproximaciones sucesivas para los más significativos.

3.4. Modulación sigma delta

Como se ha comentado anteriormente uno de los objetivos del proyecto es minimizar el tamaño y el número de componentes del control del convertidor elevador, por lo cual se intenta que todos los elementos que componen el sistema queden integrados en el dispositivo programable y que los componentes externos necesarios sean los mínimos. Además de esos requisitos el convertidor elegido debe tener una resolución suficiente para representar la señal correctamente y ser lo suficientemente rápido para seguir la señal de tensión. La tensión de entrada tiene un frecuencia de 50 Hz rectificada (frecuencia de red europea, 60 Hz si se usa la frecuencia de red americana) y la frecuencia de salida presenta



Figura 3.5: Señal sinusoidal modulada sigma-delta

un rizado del doble de frecuencia, por lo que dichas frecuencias son muy bajas para la mayoría de los conversores analógico digitales. Por tanto, la velocidad de conversión no es un parámetro decisivo en nuestro ADC.

El conversor sigma-delta cumple con los requisitos anteriormente expuestos y por eso se ha elegido como solución para este proyecto. La solución empleada es un conversor de primer orden.

Los fundamentos de estos conversores se basan en la modulación sigma delta y ya se conocen desde los años 60, pero es gracias a las técnicas de integración de circuitos basados en silicio que no han prosperado hasta fechas más recientes. En la figura 3.5 podemos ver un ejemplo de señal sinusoidal modulada sigma delta. A esta modulación también se la conoce como modulación por densidad de pulso (en inglés *pulse density modulation*).

Los conversores sigma delta constan típicamente de un modulador seguido de un decimador digital. Se basan en cambiar tasa de muestreo por resolución. Las señales a convertir son muestreadas a un tasa superior a la impuesta por Nyquist para evitar el *aliasing* pero sólo con un bit de resolución. El modulador consta en general de un integrador, un convertidor DAC (*digital to analog converter*) de un único bit, un comparador y un sumador. El voltaje de entrada se suma algebraicamente al voltaje de salida ofrecido por el DAC y después se pasa por un integrador. Si la salida del integrador es mayor que 0 el comparador da un '0' a su salida y al contrario, si es mayor de 0 V la salida del comparador es un '1'. El lazo de realimentación formado por el DAC se encarga de ajustar la salida del comparador, intentando mantener la salida del integrador siempre a cero.

La salida del comparador alimenta un filtro digital con un determinado ancho de banda que promedia la señal de un bit y ofrece valores codificados con varios bits que representan la señal de entrada. En la figura 3.6 podemos ver el diagrama de bloques de un modulador sigma delta de primer orden. Para obtener medidas con una mayor resolución se suelen utilizar técnicas de sobremuestreo promediando la salida del modulador, lo cual se consigue mediante un filtro paso bajo seguido de un decimador, lo que permite obtener la señal de un único bit en una señal de varios bits a la frecuencia de *Nyquist*.

Las ventajas de utilizar este tipo de conversores es que la mayor parte de la circuitería necesaria es digital, presentan una gran linealidad y ofrecen una alta resolución. El inconveniente es la velocidad requerida para realizar la conversión, por eso típicamente se han empleado en aplicaciones de alta resolución y baja frecuencia y recientemente en aplicaciones de audio a velocidades medias.

Para la aplicación concreta de este proyecto están muy indicados ya que las velocidades que se pueden manejar con la FPGA son más que suficientes para realizar la conversión

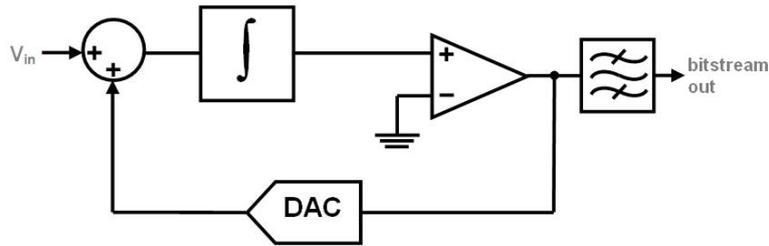


Figura 3.6: Esquema básico de un modulador sigma-delta

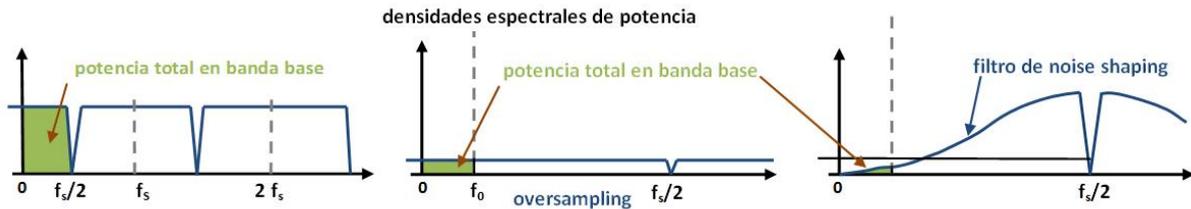


Figura 3.7: Densidad espectral de potencia de ruido con oversampling y noise shaping

con suficiente resolución y permiten un empleo mínimo de componentes externos.

3.4.1. Ruido en convertidores sigma delta

Otra de las ventajas de utilizar un convertidor sigma delta es que son muy robustos frente al ruido. Los convertidores sigma delta realizan una conformación de la densidad espectral del ruido trasladando gran parte del ruido de bajas frecuencias a altas frecuencias, que puede ser eliminado posteriormente con un filtro paso bajo sencillo. Esto se da por dos razones, primeramente por la utilización de técnicas de sobremuestreo, que consisten en tomar muestras de la señal a una frecuencia superior a la de *Nyquist*, y segundo por que el hecho de tener un lazo de realimentación y un integrador supone por sí solo una conformación del ruido como se ha explicado (*noise shaping*).

3.5. Diseño del convertidor

El convertidor sigma delta implementado en este proyecto es una versión modificada del explicado en el apartado anterior, intentando un diseño que incluyera el menor número posible de componentes analógicos. Si bien se podrían haber utilizado algún convertidor de los muchos disponibles comercialmente lo que se busca es poder integrar todos los componentes digitales en un único dispositivo para que sea posible su construcción en un ASIC, lo cual simplifica su construcción, reduce el tamaño y abarata los costes, ya que cuantos menos componentes externos a la FPGA haya, más fácilmente se podrán integrar en el ASIC. En esta versión modificada del convertidor los componentes analógicos necesarios son simplemente un filtro R-C y un comparador. La parte digital que se implementa en la FPGA incluye principalmente un contador, un acumulador y un sumador. Se realizaron dos versiones del convertidor hasta obtener una solución que cumpliera con los requisitos.

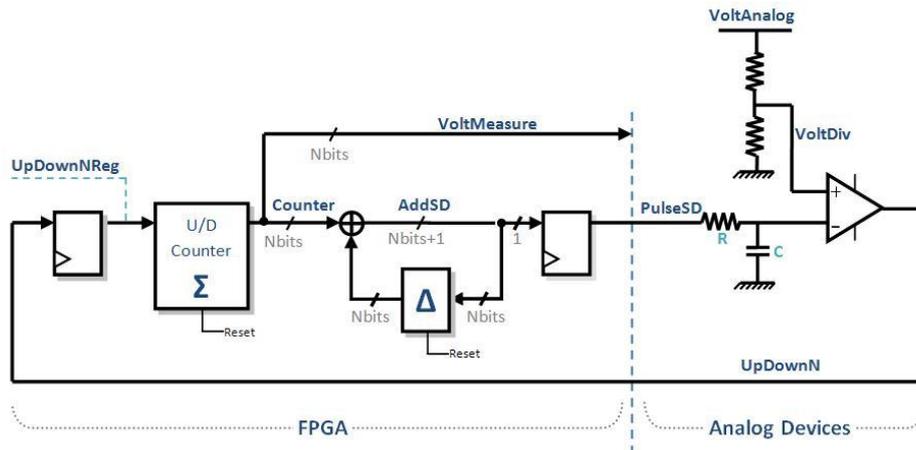


Figura 3.8: Diagrama de bloques de la primera versión del convertor sigma delta

3.5.1. Versión básica

La primera es una versión básica del convertor sigma delta implementado. Podemos ver un diagrama de bloques de este convertor en la figura 3.8. La señal *PulseSD* contiene el bit más significativo de la señal del contador acumulada. Esta señal se filtra analógicamente dando un valor de tensión que se compara con la tensión de entrada que se quiere digitalizar. Si el valor de *PulseSD* filtrado es mayor que la tensión de entrada, la salida del comparador es un '0' y a la inversa. Esta señal, que llamamos *UpDownN*, se registra y se conecta a la entrada del contador. Este contador incrementa en uno su cuenta si la señal de entrada es un '1' y la decrementa si es un '0'. La señal de la cuenta de $N+1$ bits se suma con los N bits de la cuenta anterior y el bit más significativo es el de la señal *PulseSD*. La señal de cuenta es precisamente la tensión analógica de entrada representada digitalmente en N bits.

3.5.2. Versión mejorada

En la versión anterior se observó al realizar las pruebas experimentales que los valores digitales de la tensión de entrada presentaban una fuerte oscilación alrededor del valor solución, y además esta oscilación no se veía disminuida al cambiar los valores del filtro R-C. Por tanto se optó por realizar algo parecido a un filtrado digital que redujese las oscilaciones del ADC y proporcionara medidas más estables.

La idea es permitir que el contador disminuya o incremente su cuenta sólo cada determinado número de ciclos (*Ncycles*). Durante *Ncycles* se cuenta las veces que la señal *UpdownN* se encuentra a '1' y '0'. Si se encuentra más veces a '1' que a '0' al cabo de *Ncycles* el contador se incrementa en una unidad y a la inversa. Esto se implementa añadiendo un contador que recorre desde 0 hasta *Ncycles* y que se reinicia al terminar la cuenta, dos contadores para calcular el número de veces que la señal se encuentra a '1' y '0', y dos comparadores. Las salidas de los comparadores alimentan el contador principal. Puede verse un diagrama de bloques de esta versión mejorada en la figura 3.9.

Esta solución presenta oscilaciones más moderadas y es la que se ha adoptado como versión final para las medidas de las tensiones de entrada y de salida.

El diseño realizado en lenguaje VHDL tiene como entradas las señales de reloj y *reset*,

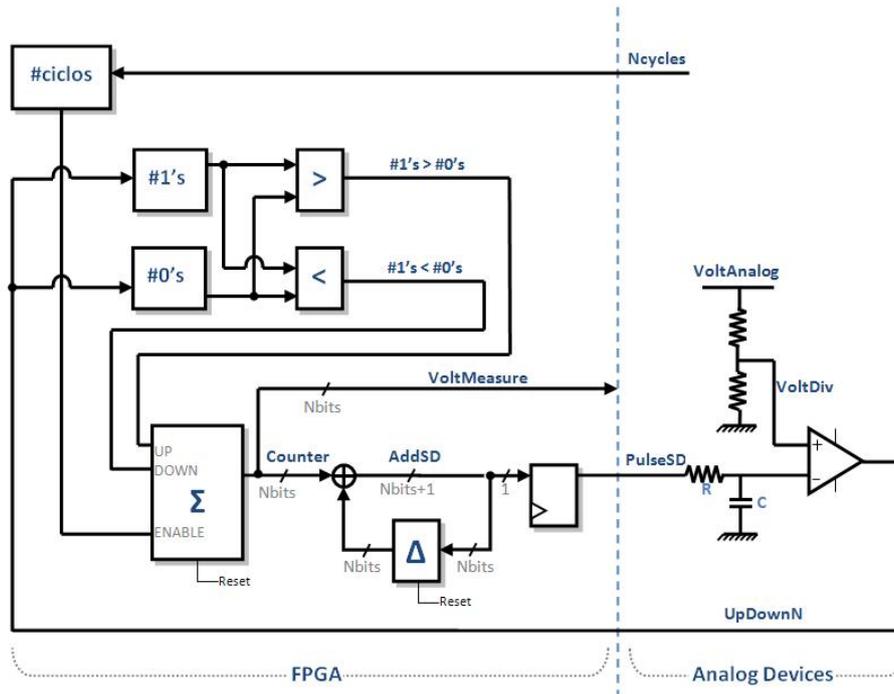


Figura 3.9: Diagrama de bloques de la versión final del convertidor sigma delta

la señal $UpDownN$ proveniente del comparador y la variable $Ncycles$, y como salidas la señal $VoltMeas$ de N bits y la señal $PulseSD$ que alimenta el filtro R-C. El código del convertidor sigma delta y el modelo digital de los componentes analógicos pueden verse en los anexos.

3.6. Simulaciones

Para comprobar el funcionamiento del convertidor se han realizado una serie de simulaciones en el programa *ModelSim*. De manera similar a las simulaciones realizadas para el elevador, tal y como se explica en el apartado 5.2, se han descrito los componentes analógicos del convertidor (filtro R-C y comparador) simulando los pasos de integración con tiempos de 1 ns.

Para obtener simulaciones representativas de lo que sucede en la realidad se han tenido en cuenta dos factores. El primero es el retardo de conmutación del comparador, que como puede comprobarse en los *datasheets* del fabricante, [15] es de 600 ns aproximadamente en tensiones de 5 V. El segundo es la histéresis del mismo, que es la diferencia de voltaje mínima necesaria entre las entradas del comparador para obtener un cambio en la señal de salida. Se ha comprobado en las simulaciones que el efecto de la histéresis es crítico en la velocidad de variación de la señal $UpDownN$ lo cual afecta en la tensión obtenida a la salida del filtro. El efecto del retardo del comparador no afecta en la variación de la señal, pero cuanto más retardo presente el comparador más lento será el convertidor.

En la figura 3.10 podemos ver capturas de simulaciones con los parámetros siguientes:

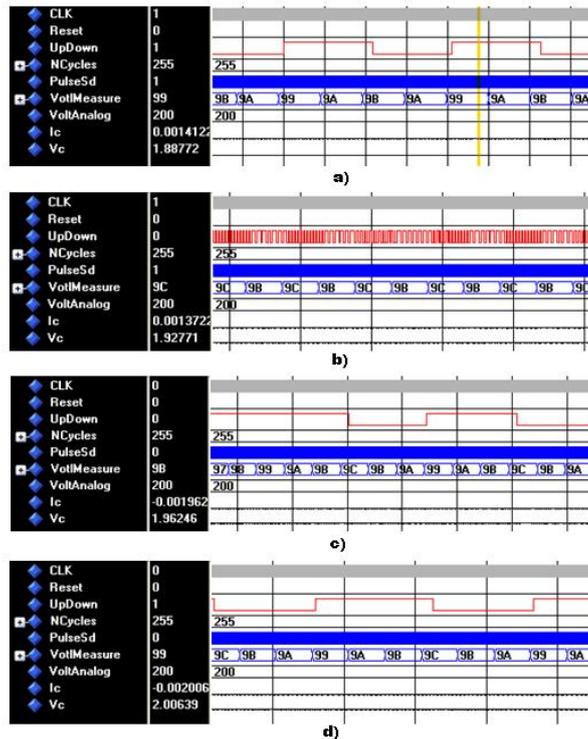


Figura 3.10: Simulaciones del efecto del retardo y la histéresis del comparador

figura	$V_{in}(V)$	$t_{retardo} (ns)$	$R(k\Omega)$	$C(\mu F)$	Histéresis (V)
3.10.a)	2.0	500	1	220	0.09
3.10.b)	2.0	500	1	220	0.06
3.10.c)	2.0	100	1	220	0.09
3.10.d)	2.0	300	1	220	0.09

En las simulaciones *a)* y *b)* puede comprobarse el efecto significativo que tiene sobre la señal de *UpDownN* variar unas pocas décimas de voltio la histéresis. Entre las figuras *c)* y *d)* podemos ver el efecto de la variación del retardo del comparador, si bien es menos significativo que el efecto de la histéresis en la señal *UpDownN*. Estas capturas se han realizado con una tensión de entrada fija para comprobar el funcionamiento del convertor y caracterizar estos parámetros. En las pruebas experimentales se toma como tensión de entrada una señal sinusoidal de frecuencia máxima de 100 Hz que simula las que realmente se tienen que medir en el sistema.

Una vez caracterizados estos parámetros y comprobado el correcto funcionamiento del convertor se han realizado las pruebas experimentales en las que se ha hecho un análisis paramétrico más exhaustivo.

3.7. Diseño y construcción del PCB del convertor

Para realizar pruebas y ensayos del funcionamiento del convertor de manera cómoda se realizó un primer montaje en una placa de prototipos (figura 3.12) incluyendo el filtro R-C, el comparador y un divisor de resistivo de tensión con un potenciómetro para poder introducir fácilmente distintos valores de tensión. En este tipo de placas se pueden

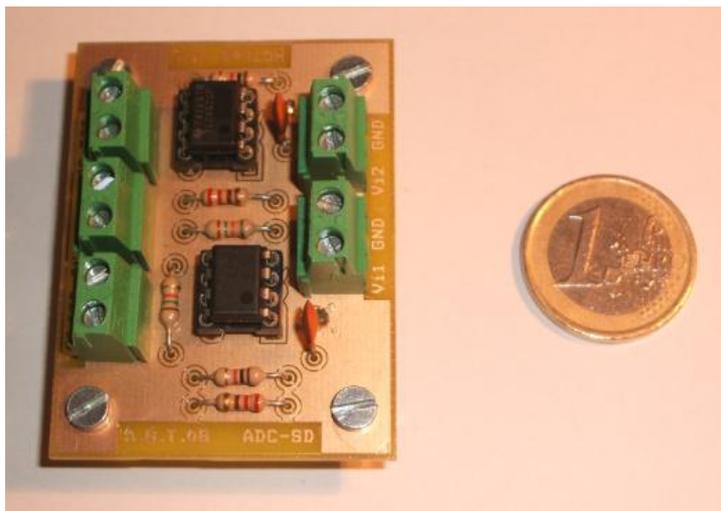


Figura 3.11: Fotografía de la placa construida para los convertidores sigma-delta

insertar y retirar los componentes de manera sencilla lo que nos permite comprobar el funcionamiento con distintos valores del filtro R-C y distintos comparadores.

Para un funcionamiento correcto es necesario añadir dos componentes más al diseño. Se trata de dos resistencias, una a la entrada de la FPGA de $1.5\text{ k}\Omega$ para limitar la corriente de entrada, y una segunda resistencia de *pull-up* de $220\ \Omega$ necesaria para el comparador indicada por la hoja de datos del fabricante.

El comparador utilizado es el TLC352CP de *Texas Instruments* [15] el cual presenta un tiempo de conmutación típico de 600 ns a una tensión de alimentación de 5 V . Inicialmente se realizaron pruebas utilizando amplificadores operacionales en saturación, como el LM6172B [14] para realizar la comparación pero se descartó su uso para evitar la necesidad de generar las tensiones simétricas de alimentación, además de que las velocidades de conmutación ofrecidas son similares. Se podrían haber elegido otros modelos de comparadores con mayores velocidades de conmutación, pero el elegido era el más rápido disponibles en formato DIP (*dual in-line package*) para montaje *through hole*, lo cual facilita mucho su utilización en placas de prototipos o la realización de medidas sobre el componente. Podemos obtener fácilmente la tensión de 5 V necesaria para el comparador de los conectores de expansión de la tarjeta de desarrollo de la *Spartan 3*, que es la placa con la que se han realizado todas las pruebas experimentales para este proyecto, debido a su bajo coste y amplia disponibilidad de conexiones externas.

Una vez comprobado el funcionamiento del convertidor con distintos valores del filtro y realizado un estudio paramétrico se realizó un diseño de una placa de circuito impreso con los componentes calculados para optimizar el funcionamiento del convertidor en cuanto a ruidos y tener un prototipo más robusto, ya que en el elevador están presentes corrientes elevadas que se encuentran constantemente en conmutación y que añaden ruidos e interferencias. La placa diseñada contiene los componentes necesarios para realizar dos convertidores, uno para la tensión de entrada y otro para la de salida. Se trata de una placa sencilla con pocos componentes (puede verse el diseño del PCB en la figura 3.13 y una fotografía de la placa construida en la fotografía de la figura 3.11). Los conectores de las tensiones de entrada y salida se han diseñado con sendas conexiones a masa para facilitar trenzar los cables de las señales de tensión con uno de masa y disminuir los ruidos e interferencias.

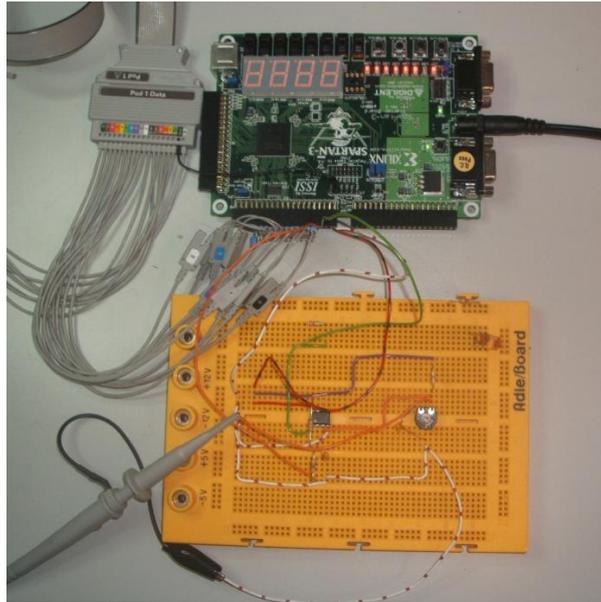


Figura 3.12: Fotografía del montaje del convertor en una placa de prototipos

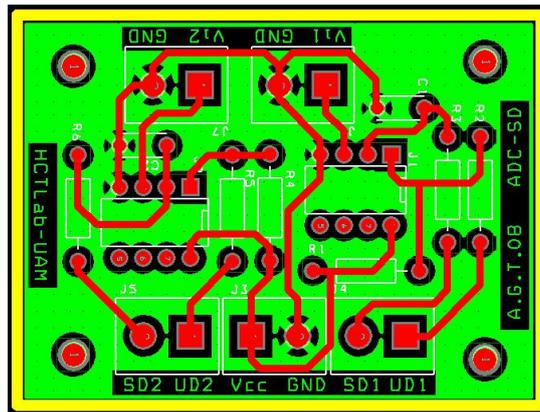
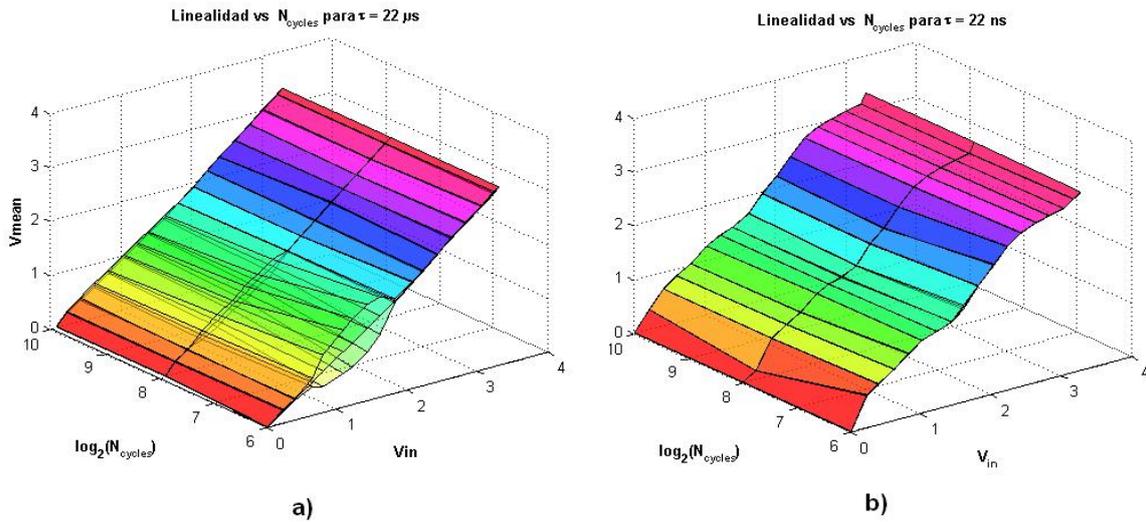


Figura 3.13: diseño del PCB del convertor sigma delta

A diferencia de la placa del *boost* que se construyó mediante la técnica de revelado y atacado químico, la placa de los convertidores se ha fabricado mediante una máquina de control numérico. La máquina empleada ha sido la *Protomat S62* del fabricante *LPKF*. Esta máquina se encuentra en los laboratorios de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid. Se trata de una máquina de altas prestaciones que fabrica las placas eliminando cobre mediante una fresadora. La resolución de la máquina está muy por debajo de 1 mm y su precio aproximado es de 20.000 euros.

3.8. Resultados experimentales

En esta sección se presentan los resultados que se han obtenido de la realización de las pruebas experimentales del convertor sigma delta. Se ha utilizado la placa de desarrollo comentada anteriormente y la parte analógica construida sobre una placa de prototipos. Para la medición de las señales se utiliza un osciloscopio con sondas digitales y analógicas para comprobar el estado de las señales de salida y entrada respectivamente, tal y como

Figura 3.14: Linealidad en función de N_{cycles}

puede verse en la figura 3.13. Las tensiones de entrada se seleccionan mediante un divisor de tensión con un potenciómetro.

Todas las pruebas realizadas se han realizado tanto para una versión de 8 bits como de 10 bits. Los resultados son similares en ambos casos y aquí presentamos los resultados para el de 10 bits, que es el que se utiliza en el sistema final.

3.8.1. Análisis paramétrico

Para calcular los valores del filtro y de otras variables que nos proporcionen un correcto funcionamiento del convertor se ha realizado un estudio paramétrico experimental. Los dos parámetros que intervienen en el estudio más importantes son:

Manipulando estas dos variables para distintos valores de tensión de entrada se comprueban dos aspectos:

- Linealidad del convertor.
- Oscilación de la salida: como se ha comentado anteriormente los valores de la salida presentan una oscilación alrededor del valor solución y nos interesa que esa oscilación sea mínima.

En la figura 3.14 pueden verse los resultados del valor máximo, mínimo y medio de la oscilación de salida para valores de N_{cycles} 64, 256 y 1024 cuando variamos la tensión de entrada de 0 a 3,3V (tensión de saturación). El eje de N_{cycles} se representa de forma logarítmica. La gráfica *a* se ha realizado con un valor de $\tau = 22 \mu s$ y la gráfica *b* con $\tau = 22 ns$. En ambas gráficas podemos comprobar que la oscilación de salida es muy pequeña salvo para el caso *a* donde puede verse que la oscilación es mayor para valores de N_{cycles} menores donde las superficies del gráfico se separan, con la particularidad de que la oscilación se reduce drásticamente en tensiones cercanas a 2 V. Comparando ambas gráficas puede comprobarse que para valores menores de la constante de tiempos las oscilaciones son mínimas pero el convertor presenta una menor linealidad.

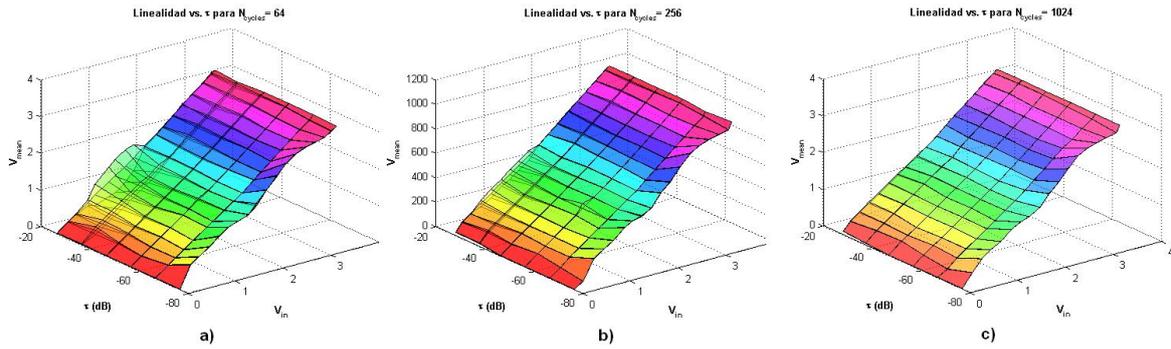


Figura 3.15: Linealidad en función de τ

Para comprobar los efectos que supone un cambio en el valor de τ se han realizado gráficas similares a las anteriores, pero en este caso el valor de N_{cycles} se mantiene fijo en cada gráfica. Podemos ver los resultados en la figura 3.15. Las gráficas 3.15.a, 3.15.b y 3.15.c corresponden a valores de N_{cycles} 64, 256 y 1024 respectivamente. Los distintos valores de τ para los que se han tomado medidas son los de la tabla siguiente:

τ (s)	R ($K\Omega$)	C (nF)
$2,2 \cdot 10^{-8}$	1	0,022
$2,2 \cdot 10^{-7}$	1	0,22
$2,2 \cdot 10^{-6}$	1	2,2
$2,2 \cdot 10^{-5}$	1	22
$2,2 \cdot 10^{-4}$	1	220
$2,2 \cdot 10^{-3}$	10	220

Observando la gráfica 3.15.a podemos comprobar que cuanto mayor es el valor de τ las oscilaciones son mayores, en mayor medida cuando el valor de N_{cycles} es menor. Como en el caso anterior las oscilaciones aumentan en tensiones próximas a 2 V, valor a partir del cual las oscilaciones se reducen de manera significativa. Se observa en las gráficas que la linealidad del convertor es mayor si el aumenta N_{cycles} y disminuye τ . Por otro lado, si se reduce drásticamente el valor de τ se producen pérdidas de linealidad y precisión en el convertor.

Con las gráficas anteriores y los datos obtenidos se ha concluido que los valores óptimos para el diseño del filtro que nos ofrecen una linealidad y estabilidad suficientes son los de la tabla siguiente, y son los que se ha utilizado para la construcción de los convertores y la realización de todas las pruebas experimentales:

C(pF)	R($k\Omega$)	τ (ns)	N_{cycles}	N_{bits}
220	1	22	1024	10

4

Etapa de potencia

En este capítulo se va a analizar el diseño de la etapa de potencia, el cálculo de los componentes necesarios y una estimación de las pérdidas en los semiconductores. Además se verá cómo se ha diseñado y construido la placa del convertidor.

Como se ha comentado anteriormente los convertidores elevadores (conocidos como *boost*) son convertidores muy estudiados y conocidos y tienen la característica de que el voltaje de salida es mayor que el voltaje introducido. Recordamos que su topología es la mostrada en la figura 4.1.

4.1. Diseño del convertidor utilizado

Las especificaciones de la aplicación son las siguientes:

1. La tensión de entrada eficaz nominal es de 230 V a una frecuencia de 50 Hz.
2. La tensión nominal de salida alcanzada será 400 V.

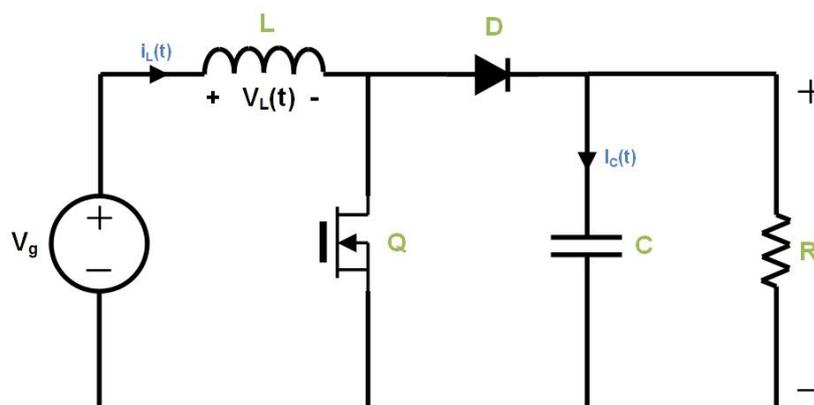


Figura 4.1: Esquema eléctrico del convertidor elevador

3. La potencia nominal es de 300 W.
4. La frecuencia de conmutación del *MOSFET* es de 100 kHz y la de la lógica de control de 100 MHz.
5. Rizado de la tensión de salida del 3 %.
6. Rizado máximo de corriente del 30 %.

Para la realización de las diferentes pruebas a lo largo de todo el proyecto se han utilizado tensiones de entrada más bajas y se han ido aumentando progresivamente. En cualquier caso las especificaciones máximas son las anteriores y son las que nos permiten calcular los valores de los componentes necesarios y diseñar la etapa de potencia.

Para el diseño del convertidor hay que tener en cuenta también las especificaciones del resto del sistema, ya que las tensiones de entrada de los conversores sigma-delta deben tener un rango adecuado, por lo que los divisores de tensión se deben adaptar durante las pruebas realizadas a las tensiones de entrada que se van aplicando. Los valores aquí presentados son para tensión de entrada nominal.

Los componentes más importantes en un convertidor son la bobina, cuyo valor vendrá determinado por el rizado de conmutación de la corriente de entrada, y el condensador que es el que mantiene la tensión de salida con determinado rizado.

El valor de la bobina se calcula a partir del rizado de corriente máximo en ella. Para dimensionarla tenemos en cuenta el peor caso (el de mayor rizado), así la tensión máxima (de pico) es $V_{Lmax} = 230 \cdot \sqrt{2}V$ y la corriente máxima en la bobina es $i_L = 1,304 \cdot \sqrt{2}A$. Debido a que se impone un rizado de tensión del 30 % el incremento de corriente en la bobina es $\Delta i_L = i_L \cdot 0,3$ y su valor en el pico es $\Delta i_{L-max} = i_L \cdot 0,3 \cdot \sqrt{2}$. El valor del ciclo de trabajo en el punto máximo se calcula a partir de la ecuación:

$$\frac{V_{in}}{1 - D} = V_{out}$$

por lo que se obtiene que el ciclo de trabajo en ese punto es $D = 0,1868$. La ecuación anterior se demuestra en el capítulo siguiente.

A partir de estos valores se puede calcular el valor necesario de la inductancia para cumplir con las especificaciones. Si se observa la figura 4.2 se puede comprobar que la pendiente de la corriente de subida es $V_{L-up} = V_{in}/L$ y la de bajada $V_{L-down} = (V_{in} - V_{out})/L$, por lo cual se deduce que el valor de la bobina es:

$$L = \frac{V_L \cdot d}{f \cdot \Delta i_L} = \frac{230 \cdot \sqrt{2} \cdot 0,1868}{100 \cdot 10^3 \cdot 1,304 \cdot 0,3 \cdot \sqrt{2}} = 1,098 \cdot 10^{-3}H$$

Para mantener el valor de rizado de corriente por debajo de este valor se necesita una bobina con inductancia igual o mayor que esa, pero como se verá en el capítulo 5 el factor de potencia mejora considerablemente cuanto mayor es el valor de la bobina, por lo que tras simular con distintos valores se ha decidido utilizar una bobina de 5 mH.

Para el dimensionado del condensador de salida hemos de tener en cuenta el rizado de la tensión de salida, el cual lo fijamos en un 3 %. El valor de la tensión pico a pico en el condensador para ese rizado es $\Delta V_{cnp} = 400 \cdot 0,03 = 12 V$. Si la pulsación de la señal de salida es $\omega(rad/s) = 100Hz \cdot 2\pi$, el valor necesario del condensador es [8]:

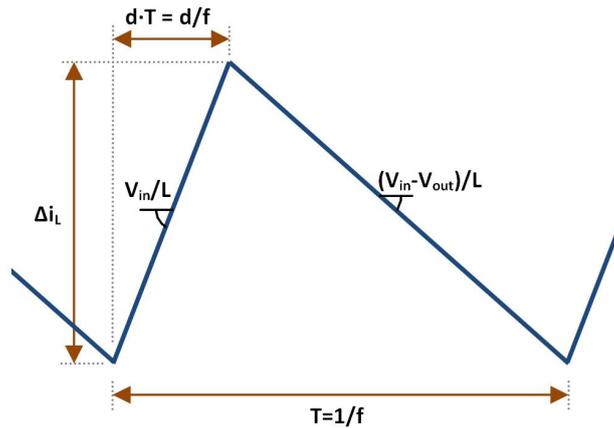


Figura 4.2: Corrientes de subida y bajada en la bobina de un boost

$$C = \frac{P_{out}}{\omega \cdot V_{out} \cdot c_{pp}} = 99,47 \cdot 10^{-6} F$$

por lo que se ha utilizado un condensador de $100 \mu F$.

El *MOSFET* debe cumplir las especificaciones de tensión máxima y además tener una respuesta rápida para funcionar a la frecuencia de conmutación por lo que se ha elegido un IRFP450 de *Fairchild Semiconductors* que soporta una tensión máxima de 500 V y tiene una resistencia de conducción de $0,4 \Omega$.

Para conmutar el *MOSFET* es necesario un *driver* que proporcione la tensión y corriente adecuadas, ya que los 3,3 V ofrecidos por las salidas lógicas de la FPGA no son suficientes. El *driver* utilizado es el IR2181 [12] que ofrece una señal de 15 V a su salida y es lo suficientemente rápido para esta aplicación.

El diodo de potencia utilizado es el DYV29-500, el cual cumple con las especificaciones de tensión y corriente del sistema y tiene un tiempo de conmutación menor de 60 ns.

El puente de diodos para rectificar la tensión de entrada elegido es FBL2504L [13] que soporta 400V de tensión de pico y una corriente de 25 A.

Para eliminar el ruido de alta frecuencia en la salida y en la entrada se añaden un condensador de 470 nF que soporten las tensiones de entrada y salida respectivamente.

Para la realización de las simulaciones de todas las pruebas del sistema de control se hace necesario un modelo del elevador. Dado que la mayoría de las simulaciones se realizan en *Modelsim* se ha descrito en lenguaje VHDL un modelo del elevador. Para definir un modelo analógico en un lenguaje de descripción de naturaleza digital se define una variable dt que expresa el paso de integración y una frecuencia de reloj para realizar las operaciones que emulan el *boost* 10 veces más rápido que la frecuencia a la que funciona la lógica de la FPGA. La idea es la misma que para simular la parte analógica de los convertidores sigma delta. Puede verse el código del modelo analógico del elevador en los anexos.

4.1.1. Cálculo de pérdidas en semiconductores

En este capítulo se va a hacer una estimación de las pérdidas térmicas que se producen en los semiconductores del convertidor elevador, para poder dimensionar los disipadores térmicos que deben llevar el diodo y el MOSFET.

Para calcular las pérdidas térmicas en el MOSFET las dividimos en distintas componentes [8]:

- Pérdidas de conducción ($P_{COND} = 1,6 \cdot R_{SDson} \cdot I_{ef}^2$)
- Pérdidas del condensador drenador-fuente ($P_{COND} = \frac{1}{2} \cdot C_{OSS} \cdot U^2 \cdot f$)
- Pérdidas de convivencia tensión corriente ($P_{CONV} = \frac{1}{3} \cdot U \cdot I \cdot f \cdot (T_{RISE} + T_{FALL})$)

Para calcular las pérdidas anteriores necesitamos los siguientes parámetros:

- Máxima tensión en abierto (U).
- Corriente media en el punto de máxima potencia (I).
- Resistencia en conducción ($R_{SDson} = 0,38\Omega$).
- Tiempo de encendido ($T_{RISE} = 8ns$).
- Tiempo de apagado ($T_{FALL} = 8ns$).
- Capacidad parásita de la salida ($C_{OSS} = 400pF$).

Sumando las componentes anteriores se obtienen la estimación de pérdidas térmicas del MOSFET:

$$P_{COND} + P_{CONV} + P_{COND} = 1,124W + 0,29W + 3,2W = 4,614W$$

En las pérdidas térmicas en el diodo intervienen dos parámetros, la corriente media en conducción I_D y su caída de tensión en conducción V_F , de forma que las pérdidas son:

$$P_{DIODO} = V_F \cdot I_D = 0,77W$$

Una vez que se han estimado las pérdidas en los semiconductores se debe calcular el valor máximo de impedancia térmica necesaria para poder dimensionar los disipadores térmicos:

$$\theta_{MOSFET} = \frac{\Delta T}{P} = \frac{100C - 30C}{4,614W} = 15,2C/W$$

$$\theta_{DIODO} = \frac{\Delta T}{P} = \frac{100C - 30C}{0,77W} = 90,9C/W$$

Para disponer de un margen de seguridad de funcionamiento los disipadores se eligen con una menor resistencia térmica, pero se ha de mantener un compromiso con el tamaño de los mismos pues las pistas que unen el diodo y el MOSFET deben ser de la menor longitud posible ya que por ellas circulan señales de altas frecuencias. Puede verse los disipadores escogidos para los dos componentes en la figura 4.5.

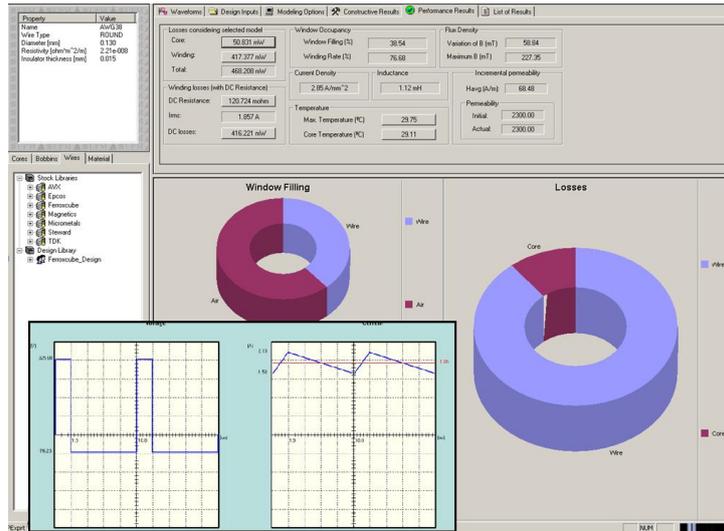


Figura 4.3: Captura del programa PEXprt

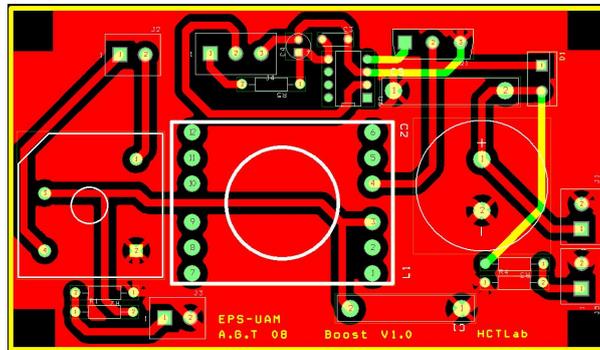


Figura 4.4: Diseño del PCB del convertidor elevador

4.1.2. Diseño y construcción del PCB del convertidor elevador

Para el diseño y construcción de la placa del convertidor se utilizaron los programas *OrCad Capture* para la realización del esquemático y el *OrCad Layout* para realizar el rutado del PCB. Se realizaron varias versiones del rutado y disposición de los componentes intentando minimizar al máximo las pistas entre el condensador de salida, la señal del driver y el *MOSFET*, pues es en estas pistas donde se encuentran las señales a altas frecuencias y es necesario que sean de la menor longitud posible para optimizar el funcionamiento del elevador. Existen dos zonas de masa en el PCB, el plano de masa que cubre un lado de la placa es el referente a la tensión de entrada, y la otra masa es la referente a la señal del driver. Estas dos masas están unidas por un único punto en el PCB para aislar los ruidos de la etapa de potencia en la parte de control.

El esquema definitivo del PCB puede verse en la figura 4.4. Para la construcción del mismo se empleó la técnica de revelado y atacado químico posterior. Se ha realizado en los laboratorios de la EPS. La bobina es un elemento que hubo que construir a medida. Esto implica la elección del núcleo apropiado y el número de espiras de cable de sección determinadas así como el espesor adecuado del entrehierro. La bobina se diseñó y construyó en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid. El programa utilizado para el diseño es el *PEXprt*, el cual nos



Figura 4.5: Fotografía del prototipo de boost construido

indica el tipo de cable necesario y el número de espiras para las especificaciones dadas. En este caso son 172 vueltas y un cable de sección AWG19. Dicha bobina se construyó a mano por parte del alumno incluyendo la colocación del entrehierro adecuado. En la figura 4.3 puede verse una captura del programa empleado.

En la figura 4.5 se puede ver una fotografía del *boost* construido y con el que se han realizado todas las pruebas experimentales.

5

Control de corriente: ciclo de trabajo precalculado

En este capítulo se va a realizar un estudio sobre los valores del ciclo de trabajo necesarios para conseguir corrección del factor de potencia, cómo calcularlos y la forma de generar la señal PWM a partir de esos valores. Después veremos la forma de implementarlos en VHDL y finalmente un análisis y estudio paramétrico de los efectos que tienen las distintas variables sobre el factor de potencia y la corriente de entrada.

5.1. Cálculo del ciclo de trabajo

El objetivo principal del proyecto es proponer un método de control cuya mayor ventaja es que permite **evitar realizar medidas de corriente** ya que es un proceso más complicado y costoso que realizar medidas de tensión.

En [8-9] se realiza un planteamiento similar, se evita la medida de corriente, pero se emplean algoritmos para reconstruirla continuamente, con el consiguiente gasto computacional, además de que dicho método es muy sensible a ciertos parámetros críticos como el retardo de encendido y apagado en el *driver* y en el MOSFET. Aquí se presenta un sistema en el que no es necesaria la medida de la corriente ni su reconstrucción, si no que se calculan las curvas del ciclo de trabajo de antemano a partir de los valores de tensión de entrada y salida, carga y otros parámetros conocidos, de tal forma que sincronizándolas con la tensión de entrada, se consiga corrección del factor de potencia.

A continuación veremos cómo se calculan dichas curvas del ciclo de trabajo.

Como se ha comentado anteriormente la topología utilizada es el elevador o *boost* (figura 4.5). La energía que se almacena en la inductancia es $E = \frac{1}{2} \cdot L \cdot I^2$ y viene determinada por el cambio en la corriente ΔI_L . Dicha corriente depende del voltaje aplicado a través de ella durante un tiempo determinado Δt de la forma:

$$\Delta I_L = \frac{V_L \cdot \Delta t}{L}$$

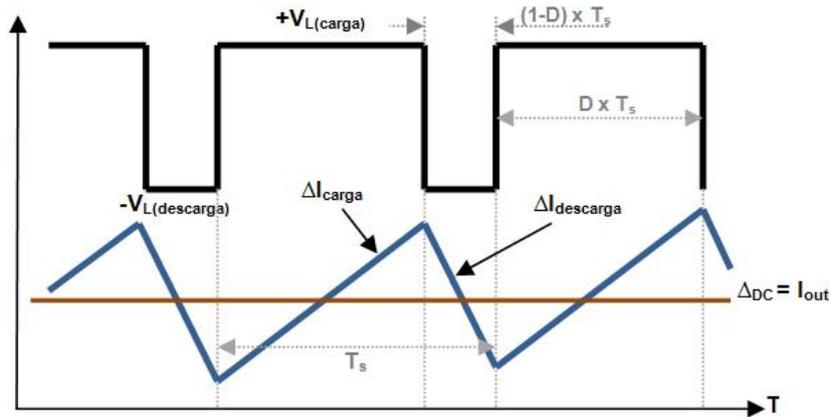


Figura 5.1: Corrientes en un convertidor elevador

La corriente ΔI_L es lineal dado que se aplica un voltaje casi constante a la inductancia en la fase de carga.

En un convertidor conmutado podemos distinguir dos etapas en función del estado de conducción del MOSFET, la de carga y la de descarga. En un elevador, durante la fase de carga el MOSFET está en conducción, se almacena energía en la bobina, el diodo se encuentra polarizado en inversa y la corriente aplicada en la inductancia aumenta porque la tensión aplicada en ella (V_L) es positiva. Durante la fase de descarga el MOSFET no conduce y el diodo se polariza en directa, por lo que la bobina transfiere su energía al condensador y a la carga.

La señal que controla el encendido y apagado del transistor es una señal modulada por ancho de pulso o *pulse with modulation* (PWM). De esta señal podemos definir dos parámetros, el tiempo de encendido y el periodo de conmutación. Sea T_{on} el tiempo que está en estado de conducción el MOSFET y T_s el periodo de conmutación, definimos el ciclo de trabajo de la señal PWM como $D = \frac{T_{on}}{T_s}$. Una vez conocido T_s , cuyo valor viene impuesto por el diseño del *boost*, se puede generar la señal PWM a partir de los valores del ciclo de trabajo, y variando dicha señal adecuadamente podemos controlar las corrientes de carga y descarga de la inductancia.

La corriente de la inductancia es el resultado de la superposición de las corrientes de carga y descarga y de una componente continua (figura 5.1), por lo que calculando adecuadamente los valores de D en cada periodo de conmutación podemos imponer que la corriente de entrada sea proporcional a la tensión aplicada y conseguir así corrección del factor de potencia.

Para calcular el ciclo de trabajo lo descomponemos en dos componentes D_1 y D_2 . La primera componente es función de las tensiones aplicadas a la inductancia en las fases de carga y descarga, y la segunda componente se introduce para compensar los efectos no tenidos en cuenta en D_1 de almacenar y extraer energía de la bobina.

Calculamos primero D_1 . Sabemos que la corriente en estado estacionario es la misma en el inicio y el final de cada ciclo de carga y descarga, por lo que:

$$I_{\text{carga}} = I_{\text{descarga}}$$

$$\frac{V_{L(carga)} \cdot T_s \cdot D_1}{L} = \frac{V_{L(descarga)} \cdot T_s \cdot (1 - D_1)}{L}$$

$$V_{L(carga)} \cdot D_1 = V_{L(descarga)} \cdot (1 - D_1)$$

Como en un convertidor elevador $V_{L(carga)} = V_{in}$ y $V_{L(descarga)} = V_{out} - V_{in}$ debido a su topología se tiene que:

$$V_{in} \cdot D_1 = V_{out} - V_{in} \cdot (1 - D_1)$$

y por tanto:

$$D_1 = \frac{V_{out} - V_{in}}{V_{out}}$$

Considerar este valor de D_1 sería suficiente si el convertidor fuera DC/DC (de continua a continua) pero por ser el convertidor diseñado del tipo AC/DC (de alterna a continua) con corrección de factor de potencia y dado que la tensión de salida tiene un rizado, es necesario tenerlo en cuenta para el cálculo de D_1 y lo podemos definir de la siguiente manera:

$$V_{riz} = \frac{P_{out}}{C \cdot \omega_r \cdot V_{out}} \sin(\omega_r \cdot t)$$

donde P_{out} es la potencia de salida, C es la capacidad del condensador de salida y ω_r es la pulsación del rizado de tensión, que en este caso es el doble de la frecuencia de red, es decir, 100 Hz. Teniendo en cuenta este rizado el valor de D_1 lo calculamos finalmente como:

$$D_1 = \frac{(V_{out} - V_{riz}) - V_{in}}{V_{out} - V_{riz}}$$

Para conseguir el mejor factor de potencia la componente D_1 no es suficiente debido a que no tiene en cuenta los efectos de la energía que se almacena o desprende de la bobina, fruto de la variación de corriente media a 100 Hz por tratarse de un convertidor con corrección de factor de potencia. Sea la diferencia de corriente en la bobina en un ciclo de conmutación k :

$$V_L(t) = L \left(\frac{di_L}{dt} \right) \approx \left(\frac{L}{T_s} \right) [i_L(k+1) - i_L(k)]$$

se define la componente $D_2(k)$ en cada ciclo de conmutación como:

$$D_2(k) = \frac{\left(\frac{L}{T_s} \right) [i_{out}(k+1) - i_{out}(k)]}{V_{out}}$$

Como puede verse en las gráficas de ejemplo (5.2) esta curva tiene forma de coseno y es varios órdenes de magnitud más pequeña que la componente D_1 , pero ha de tenerse

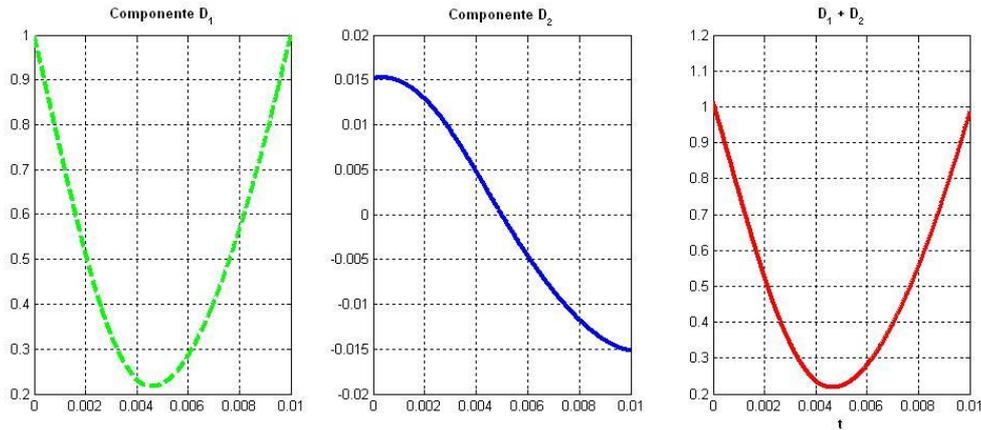


Figura 5.2: Componentes del ciclo de trabajo

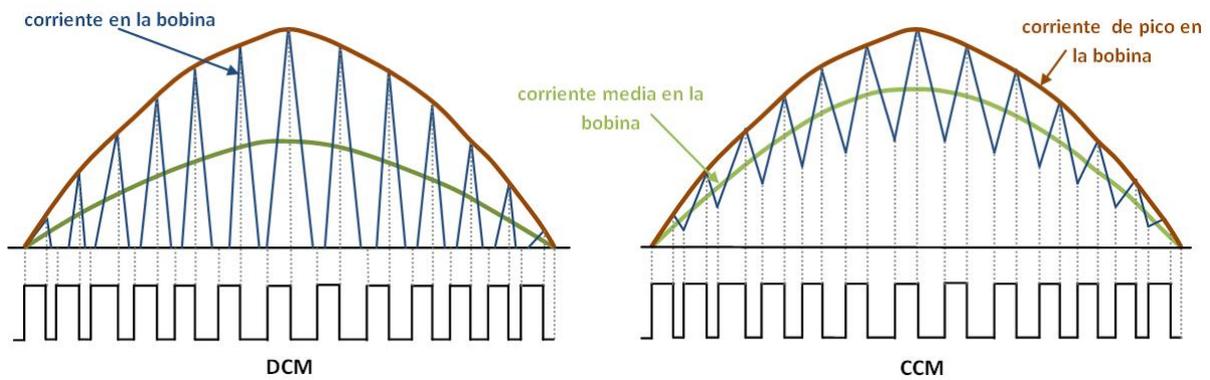


Figura 5.3: Corrientes en modo de conducción continuo y discontinuo

en cuenta para optimizar el factor de potencia obtenido. D_1 es la parte principal del ciclo de trabajo que marca la relación de tensiones, y D_2 hace que la corriente vaya variando a la frecuencia de red, es decir, lentamente. Los valores de D_1 son fijos independientemente de la carga, pero el valor de D_2 crece al aumentar el valor de la carga.

Finalmente, el ciclo de trabajo se obtiene de la superposición de estas dos componentes, $D = D_1 + D_2$. En la figura 5.2 podemos ver un ejemplo de una curva de ciclo de trabajo para un semiciclo de tensión para valores de tensión de entrada 55 V , 100 V de tensión de salida, una inductancia de 5 mH, y condensador de salida de 100 μ F.

5.1.1. Modo de conducción continuo y discontinuo

En un convertidor podemos distinguir entre los modos de conmutación continuo y discontinuo (CCM y DCM respectivamente). En el modo de conmutación discontinuo la corriente se hace cero, mientras que en el modo de conducción continuo la corriente de la bobina permanece próxima a la corriente sinusoidal media. Podemos ver una representación de los dos modos de conmutación en la figura 5.3. Existe un tercer estado en el que no conducen ni el MOSFET ni el diodo, y la bobina ya está descargada.

Si definimos $M = V_{out}/V_{in}$ como la relación de conversión, la componente D_1 del ciclo de trabajo para el modo de conducción continuo en un convertidor elevador es:

$$M(D) = \frac{1}{1-D}(CCM)$$

Cuando el convertidor está en modo de conducción discontinuo puede demostrarse que la relación de conversión es:

$$M(D, K) = \frac{1 + \sqrt{1 + 4D^2/K}}{2}(DCM)$$

donde K es un parámetro adimensional que indica la tendencia de un conversor para operar en modo continuo o discontinuo y es igual a $K = \frac{2L}{RT_s}$. Valores mayores de K indican una propensión a operar en modo continuo y a la inversa. Esto implica que los valores del ciclo de trabajo necesarios son diferentes según nos encontremos en modo de conducción continuo o discontinuo. El valor de K que indica el punto en el que el convertidor pasa de un modo de conmutación a otro se le conoce como $K_{critica}(D)$ y es función del ciclo de trabajo. En un *boost* es igual a $D(1-D)^2$. Por lo tanto conociendo esta $K_{critica}(D)$ podemos saber en que modo de conmutación se encuentra el convertidor:

- Si $K < K_{critica}(D) \Rightarrow DCM$
- Si $K > K_{critica}(D) \Rightarrow CCM$

En el modelo empleado para el proyecto se ha comprobado que el elevador siempre se encuentra en modo de conducción continuo, debido a que con la bobina empleada y los valores de potencia y tensión para los que se han realizado las pruebas se obtiene un valor alto de K , por lo que las gráficas de las curvas del ciclo de trabajo son las presentadas en el apartado anterior. En cualquier caso los programas creados en este proyecto para el cálculo de D tienen en cuenta los dos modos de conmutación.

5.2. Simulaciones y análisis paramétrico

Para verificar el funcionamiento del sistema en función de las curvas del ciclo de trabajo calculadas, se han realizado simulaciones en *Matlab* para hacer un análisis de forma paramétrica. Debido a que la sincronización es un elemento crítico en el factor de potencia resultante, estudiaremos sus efectos en este capítulo, pero en el capítulo 6 se describe más detalladamente el funcionamiento del mecanismo de sincronización. El objetivo de este análisis es estudiar el efecto sobre el factor de potencia cuando variamos la potencia de salida, el valor de la bobina y la capacidad del condensador para distintos desfases. φ expresa el desfase en grados que existe entre las curvas del ciclo de trabajo y un semiciclo de tensión de entrada.

5.2.1. Efecto de la cuantificación y limitación del ciclo de trabajo

Antes de comprobar los efectos de la variación de la potencia de salida, o de los valores del condensador y la inductancia, se va a analizar el factor de potencia obtenido cuando las condiciones son las nominales y el desfase de sincronización es cero.

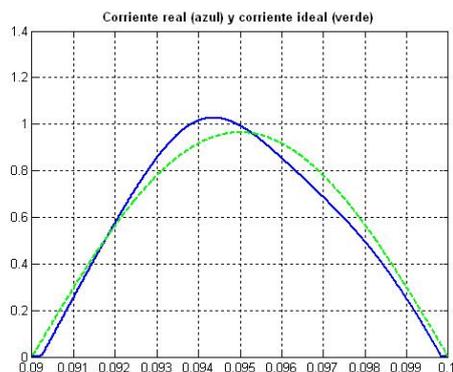


Figura 5.4: Efecto de la cuantificación y limitación del ciclo de trabajo en corriente de entrada

Los parámetros utilizados y factores de potencia resultantes se detallan en la siguiente tabla:

figura	$\alpha(^{\circ})$	$V_{in}(V)$	$V_{out}(V)$	$L(mH)$	$C(\mu F)$	$P(W)$	PF
5.4	0	55	100	5	100	37,5	0,9956

En la figura 5.4, obtenida con los parámetros anteriores, podemos comprobar cómo en las simulaciones, sin incluir desajustes en la potencia o en el desfase, ya se obtiene una curva de corriente que es distinta de la ideal (corriente ideal en verde y corriente real en azul) y por tanto el factor de potencia obtenido es menor que 1. Esto es debido a que en la simulación ya se incluyen dos efectos reales. El primero es que el ciclo de trabajo máximo está limitado a 0,95 ya que alcanzase el valor 1 la frecuencia de conmutación no sería constante, y el segundo es que los valores del ciclo de trabajo ajustados a los valores cuantificados de 10 bits que se pueden manejar en la FPGA. Por eso es imposible, debido a estos efectos, obtener un factor de potencia unidad en un sistema real.

5.2.2. Efecto de la potencia de salida

Analizamos el efecto en el factor de potencia de variar la potencia que consume la carga. Las condiciones nominales para las que se ha dimensionado el convertidor son 230 V de tensión de entrada y 300 W de salida. Debido a que las pruebas experimentales se han realizado inicialmente con tensiones más bajas (55 V), y por tanto las simulaciones que se presentan aquí también se han realizado con esa tensión de entrada, para poder compararlas con los resultados experimentales. Para esta tensión de entrada imponemos una potencia de salida de 75 W ya que eso implica que la corriente es la misma que la calculada para el caso nominal de 230 V de entrada (las siglas PF corresponden a *power factor*, o factor de potencia).

figura	$\alpha(^{\circ})$	$V_{in}(V)$	$V_{out}(V)$	$L(mH)$	$C(\mu F)$	$P(W)$	PF
5.5.b	0	55	100	5	200	75	0,9727
5.4	0	55	100	5	200	37,5	0,9945
5.5.a	0	55	100	5	200	18,75	0,8338

La figura 5.5 muestra las corrientes reales e ideales que se obtienen cuando los valores del ciclo de trabajo son calculados para una potencia nominal de 37.5 W y se consume

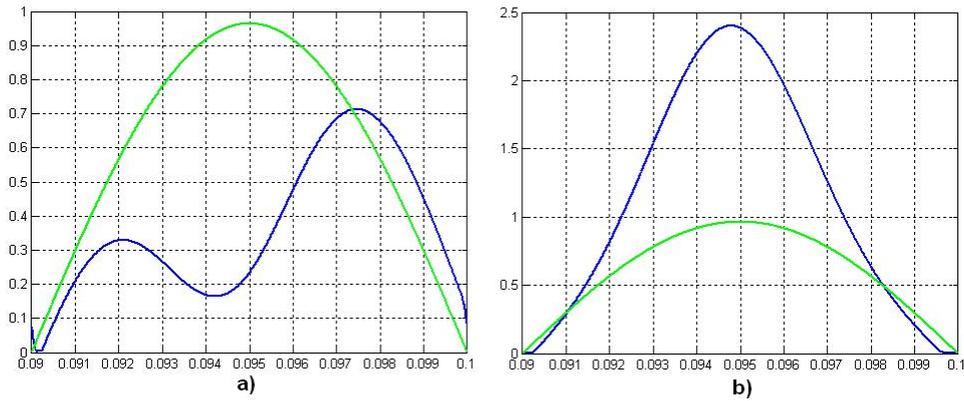


Figura 5.5: Efecto de la variación de la potencia entregada en la corriente de entrada

realmente esa potencia (figura 5.4) en comparación con las que se obtienen cuando los valores se calculan también para esa potencia, pero se consume el doble (figura 5.5.b) o la mitad de potencia (figura 5.5.a). Podemos comprobar que cuando la potencia consumida es distinta de la que se ha utilizado teóricamente para calcular el ciclo de trabajo, las curvas de corriente se distorsionan y el factor de potencia empeora, en mayor medida cuando la potencia real consumida es menor.

Podemos concluir que si se quiere obtener el máximo factor de potencia sólo a valores nominales (como exige la normativa EN16000-3-2) el caso óptimo es calcular los valores del ciclo de trabajo sólo para esa potencia. Sin embargo, si se quiere obtener un alto factor de potencia para un cierto rango de potencias, es más favorable calcular el ciclo de trabajo para una potencia baja de dicho rango, ya que el factor de potencia se deteriora mucho al disminuir la potencia consumida, pero no tanto al aumentarla.

5.2.3. Efecto de la bobina

Para comprobar cómo afectan distintos valores de la bobina en el factor de potencia realizamos simulaciones con distintos valores y para distintos desfases.

figura	$\alpha(^{\circ})$	$V_{in}(V)$	$V_{out}(V)$	L (mH)	C (μF)	P (W)	PF
5.6.a	+0,2	55	100	1,1	100	37,5	0,9609
5.6.b	+0,2	55	100	5	100	37,5	0,9667
-	0	55	100	1,1	100	37,5	0,9990
-	0	55	100	5	100	37,5	0,9956
5.6.c	-0,2	55	100	1,1	100	37,5	0,9753
5.6.d	-0,2	55	100	5	100	37,5	0,9804

De los datos anteriores y de las curvas obtenidas en las figura 5.6 podemos concluir que cuanto mayor es el valor de la bobina mejor es el factor de potencia, y que este es mejor si el desfase introducido es negativo.

5.2.4. Efecto de la variación del condensador en la corriente de entrada

Estudiamos ahora el efecto que tiene sobre el factor de potencia variar la capacidad del condensador en función de si se introduce un desfase negativo o positivo. Para ello

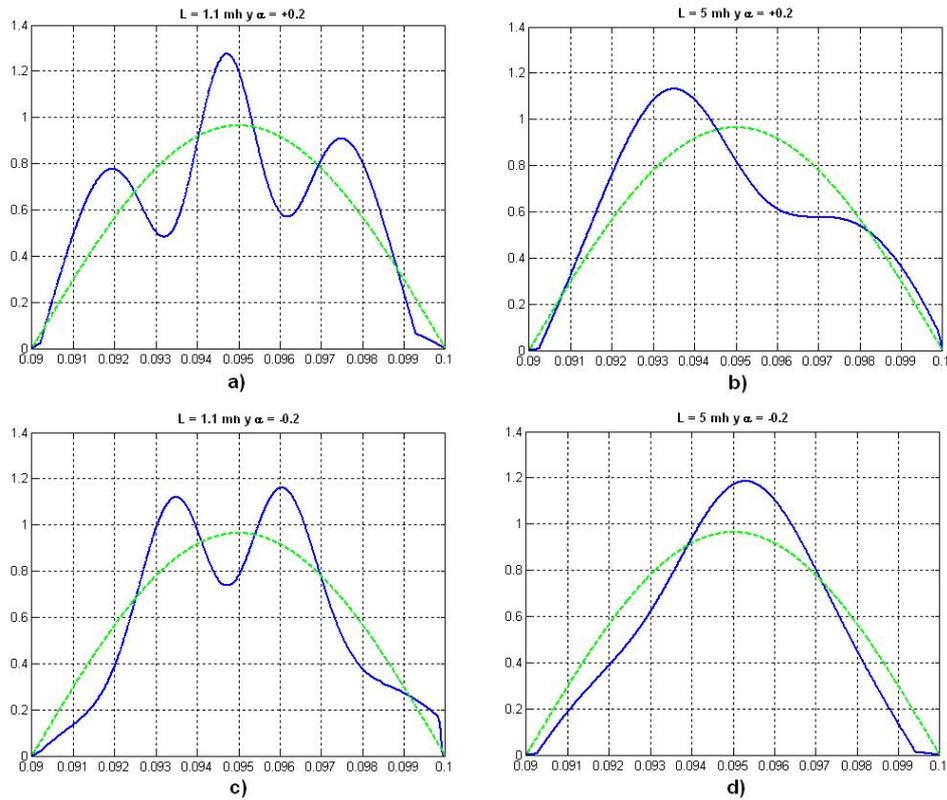


Figura 5.6: Efecto de la variación del valor de la bobina en la corriente de entrada

se han calculado los valores del ciclo de trabajo para distintos valores de capacidad y de desfase. Los parámetros utilizados para comprobar este efecto y el factor de potencia obtenido en cada uno de ellos son:

figura	$\alpha(^{\circ})$	$V_{in}(V)$	$V_{out}(V)$	L (mH)	C (μF)	P (W)	PF
5.7.a	+0,2	55	100	5	100	37,5	0,9667
5.7.b	+0,2	55	100	5	200	37,5	0,9720
-	0	55	100	5	100	37,5	0,9956
-	0	55	100	5	200	37,5	0,9945
5.7.c	-0,2	55	100	5	100	37,5	0,9804
5.7.d	-0,2	55	100	5	200	3,5	0,9789

Las curvas resultantes de estas simulaciones pueden verse en la figura 5.7. Se observa que si el desfase es negativo el factor mejora cuanto menor es la capacidad del condensador, y a la inversa cuando el desfase es positivo. Si bien el factor de potencia es menos sensible a la variación en el condensador que en la bobina, por lo que no es un elemento crítico del diseño.

5.3. Implementación del diseño en VHDL

La solución empleada para generar la señal de control es la siguiente: los valores del ciclo de trabajo se almacenan en una memoria interna de la FPGA, y a partir de estos valores se genera la señal PWM que se manda al MOSFET (figura 5.8).

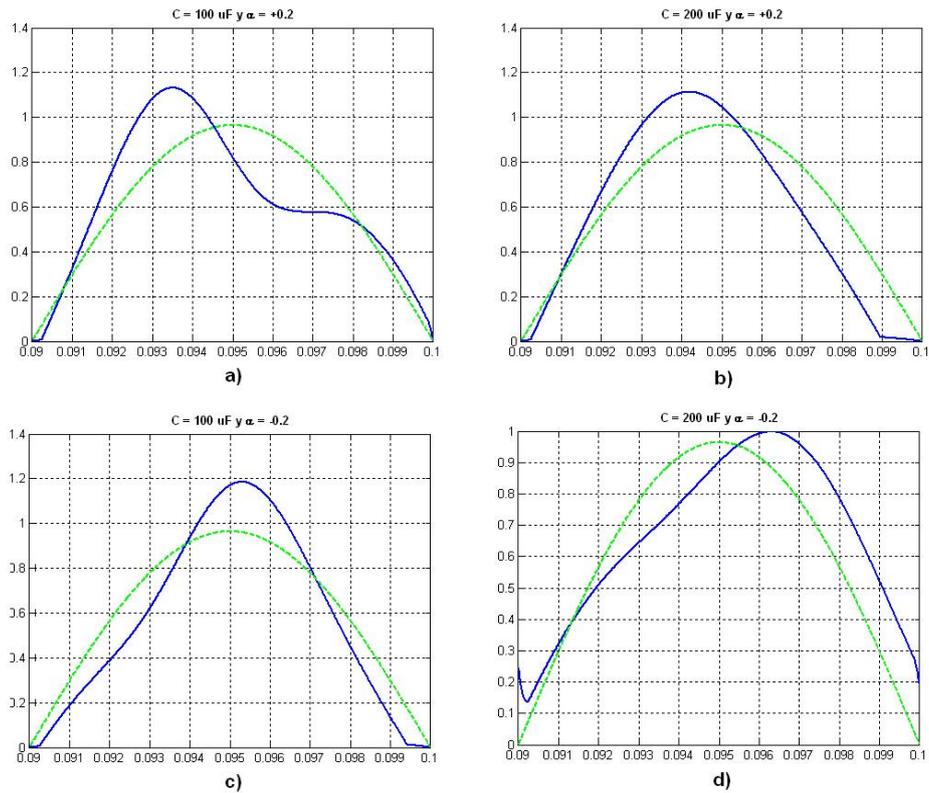


Figura 5.7: Efecto del condensador en la corriente de entrada

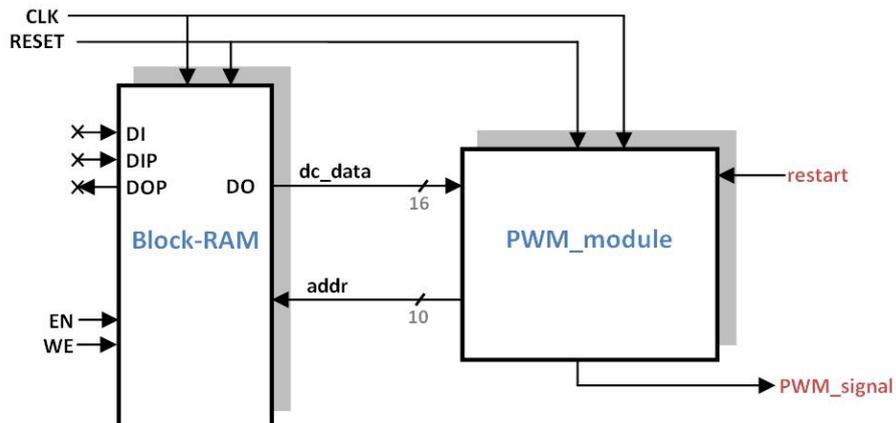


Figura 5.8: Diagrama de bloques: señal PWM

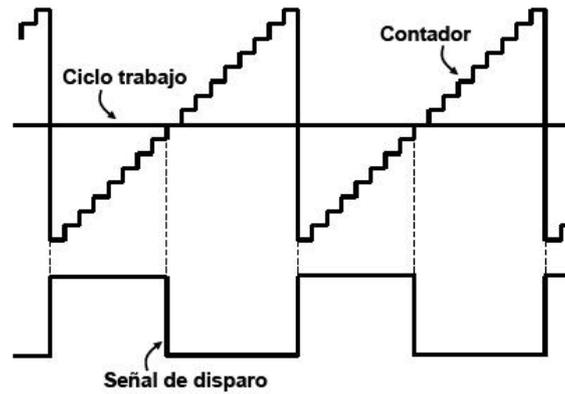


Figura 5.9: Generación de la señal PWM

La resolución utilizada para las curvas de ciclo de trabajo es de 1000 valores, es decir, por cada semiciclo de tensión, que dura 10 ms (frecuencia de red de 50 Hz), tenemos 1000 valores diferentes de ciclo de trabajo.

Al tiempo que dura cada uno de estos ciclos de trabajo se le conoce como periodo de conmutación (que son 10 μ s), en el cual se genera un ciclo de la señal PWM. La resolución de la señal PWM es a su vez de 1000 muestras por cada periodo de conmutación, por lo que los bits de esta señal de control se generan cada 10 ns, o lo que es lo mismo, la frecuencia de la actualización de la señal PWM es de 100 MHz. Por lo tanto, la frecuencia de reloj a la que debe funcionar la FPGA para obtener las resoluciones deseadas es de 100 MHz.

La placa de desarrollo empleada para este proyecto dispone de un cristal generador de reloj de 50 MHz, por lo que haciendo uso de un módulo DCM (*Digital Clock Manager*) podemos multiplicar por dos la frecuencia de esta señal y obtener la frecuencia de reloj deseada.

Para almacenar los valores del ciclo de trabajo utilizamos una de las memorias en configuración 1k x 16 + 2 bits de paridad con puerto único, lo cual quiere decir que tenemos una memoria con datos de 16 bits y un direccionamiento de 10 bits (1024 direcciones). Los bits de paridad no han sido utilizados. Como se ha explicado anteriormente, la resolución de la señal PWM es de 1000 valores por cada periodo de conmutación, por lo que realmente se utilizan los 10 bits menos significativos de los 16 de cada dirección, y las 1000 direcciones más bajas.

El módulo que genera la señal PWM recibe estos valores de la memoria y realiza la siguiente operación: mediante un contador que se extiende de 0 a 999, compara el valor de la señal que le entrega la memoria con dicho contador. Si la señal recibida es menor que el contador, la señal PWM generada (*pwm_signal*) es un '1' lógico. En el momento en el que el contador es igual o superior a la señal recibida la señal PWM pasa a ser un '0'. (Figura 5.9).

Como veremos posteriormente, la señal de sincronismo que indica el punto en el que acaba un semiciclo de tensión de red y empieza el siguiente se ha denominado *restart* y es un pulso que dura un ciclo de conmutación. Cuando el contador alcanza el último valor (999) mantiene el último valor del ciclo de trabajo a la espera del *restart*. Con esto se consigue evitar cambios abruptos en las curvas del ciclo de trabajo. De igual manera,

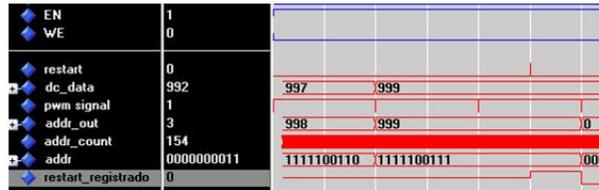


Figura 5.10: La ultima dirección se mantiene hasta que se activa la señal de restart

si el *restart* se produce antes de que finalice la cuenta, esta se reinicia. Podemos ver un ejemplo de simulación con *ModelSim* en la figura 5.10

5.3.1. Protecciones en el ciclo de trabajo

Otra de las ventajas de tener un control digital es la forma en la que se pueden crear restricciones y protecciones en distintos parámetros. Mientras que en un control analógico añadir una protección supone realizar cálculos más complejos, añadir más componentes o incluso rediseñar el circuito, en un sistema digital esto se resuelve añadiendo unas pocas líneas de código.

En cuanto a la señal PWM se han añadido dos protecciones. La primera de ellas es que el ciclo de trabajo nunca supere un determinado valor máximo, en este caso 0,95, para así evitar picos excesivos de corriente si se produce un fallo en la sincronización o mientras se espera a la señal de restart para reinicializar las curvas del ciclo de trabajo y además para que la frecuencia de conmutación siempre sea de 100 kHz, ya que si D fuese 1, no habría conmutación durante en ese ciclo. Esto se resuelve modificando los valores almacenados en la memoria de tal manera que nunca se supere este valor. Otra protección añadida es la de mantener un ciclo de trabajo mínimo. Debido a que el mecanismo de sincronización tarda varios ciclos de red en estabilizarse es necesario mantener unos valores de corriente mínimos para que cuando arranque el sistema de sincronización no se produzcan picos de corriente excesivos. En concreto el ciclo mínimo impuesto es 0,3, que se mantiene hasta que el mecanismo de sincronización activa una señal de control (*dc_enable*), y a partir de ese momento el valor del ciclo de trabajo se lee de la memoria.

Se han añadido otras protecciones relacionadas con las tensiones de entrada o con los valores del lazo de tensión, pero se estudian en sus capítulos correspondientes.

6

Sincronización

Para conseguir corrección del factor de potencia es necesario que las curvas del ciclo de trabajo y la tensión de entrada estén sincronizadas de la manera más exacta posible, pues es un factor crítico en el FP resultante. En este capítulo vamos a describir la importancia del mecanismo de sincronización y la forma en la que se ha diseñado e implementado.

6.1. Diseño del mecanismo de sincronización

Para poder sincronizar la señal con la tensión de entrada primeramente es necesario conocer los valores de esta tensión. Para ello se utiliza un conversor analógico digital sigma delta de los presentados anteriormente. La tensión de entrada se obtiene a través de un divisor de tensión que nos proporciona un rango adecuado para el conversor.

El procedimiento más obvio para sincronizar la señal de entrada sería detectar los puntos de paso por cero de dicha tensión, pero se ha de tener en cuenta que la tensión de entrada rectificadora no es ideal y realmente no presenta paso por cero, además de que los valores de la tensión en el punto mínimo no se dan en único instante (figura 6.1), por lo que tampoco sería suficiente con medir el instante en el que obtenemos la tensión mínima. Debido a estas razones se hace necesario un mecanismo más complejo que se explica a continuación.

El mecanismo realizado de sincronización se basa en la detección de flancos de bajada y subida con respecto a unos determinados umbrales, de manera que si una muestra del ADC es menor que el umbral y la siguiente muestra es igual o mayor se detecta un flanco de subida, y a la inversa, si la muestra actual es mayor que el umbral y la siguiente es menor o igual se detecta un flanco de bajada. Debido a que el ADC presenta oscilaciones definimos dos umbrales para evitar la detección errónea de flancos. Si estos umbrales están separados una distancia mayor que la oscilación del ADC no se producirán rebotes en los flancos.

A pesar de considerar los dos umbrales es posible que se detecten varios flancos de subida o bajada consecutivos por las oscilaciones o el ruido (puede verse una explicación

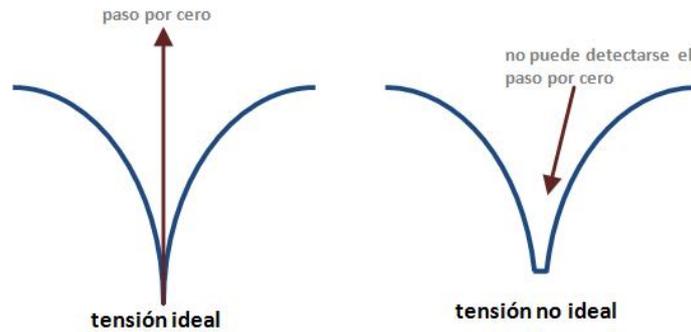


Figura 6.1: Caso ideal y real de la tensión de entrada rectificada

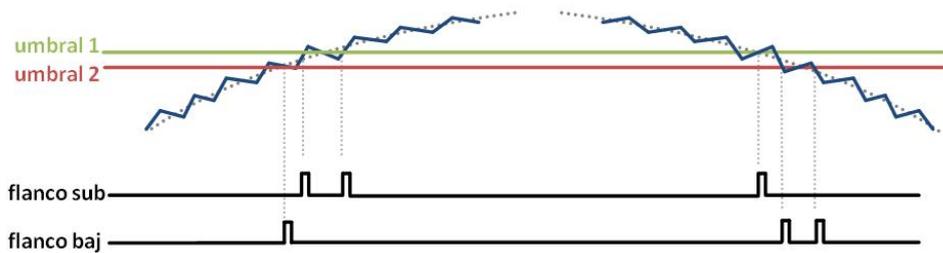


Figura 6.2: Detección de falsos flancos de subida o bajada

gráfica en la figura 6.2), por lo que de cada grupo de flancos de subida o bajada se considera solo el primero de ellos. De esta forma obtenemos por cada semiciclo de tensión un flanco de subida y uno de bajada.

A partir de los flancos que se detectan se calculan una serie de parámetros:

- N : es el número de ciclos de conmutación que transcurren desde que se detecta un flanco de bajada y el siguiente
- M : se define como el número de ciclos de conmutación entre un flanco de bajada y el flanco de subida posterior
- R : lo calculamos como $N + \frac{M}{2}$

De esta manera a partir del valor de R podemos crear una señal de sincronismo que se activa R ciclos de conmutación después de detectar un flanco de bajada y que debe coincidir con el paso por cero de la tensión de entrada. Para hacer una medida más precisa se obtienen valores medios de M y N a partir de 4 muestras anteriores:

$$N_{mean} = \sum_{i=1}^4 (N_i) / 4$$

$$M_{mean} = \sum_{i=1}^4 (M_i) / 4$$

$$R = N_{mean} + \frac{M_{mean}}{2}$$

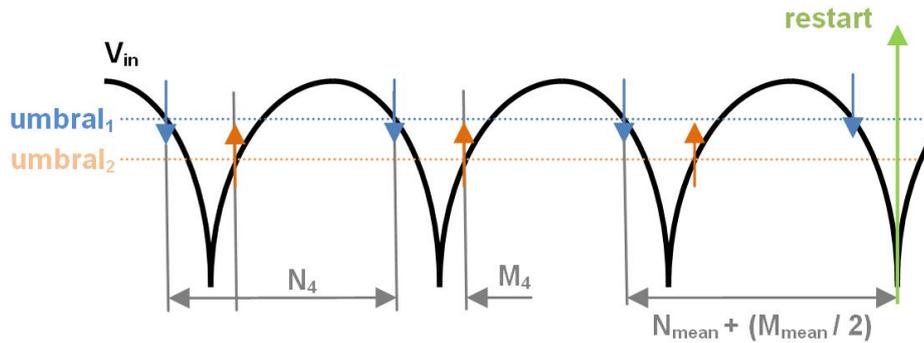


Figura 6.3: Esquema del mecanismo de sincronización

Se emplean 4 muestras porque se ha comprobado experimentalmente que con ellas se obtiene un funcionamiento correcto casi libre de ruido y que el emplear más muestras no produce mejoras significativas mientras que sí consume más recursos. Además elegir un número de muestras que sea potencia de dos permite realizar las divisiones para calcular las medias mediante operaciones de desplazamiento. En la figura 6.3 podemos ver un esquema de cómo se realiza la sincronización.

El resultado final es una señal que denominamos *restart* que se activa durante un ciclo de conmutación cada vez que la tensión de entrada pasa por cero.

En las pruebas experimentales se observó que esta señal presentaba una rápida oscilación alrededor de un valor central debido a los ruidos y oscilaciones del ADC que hacían que se obtuvieran valores ligeramente distintos de R cada vez. Para evitar cambios bruscos en la señal de sincronismo y por consiguiente en los valores del ciclo de trabajo se realizó una segunda versión de la sincronización. En esta versión sólo se permite a la señal de *restart* desplazarse un único ciclo de conmutación entre dos semiciclos de red consecutivos, de manera que si el valor de R en el semiciclo actual es mayor que en el anterior, el nuevo valor de R es $R + 1$ y a la inversa, si el valor de R actual es menor que el anterior es $R - 1$. Con estas mejoras obtenemos una señal de sincronismo más precisa y estable.

Dado que el mecanismo de sincronización necesita varios semiciclos de red para calcular los valores medios existe un periodo inicial al arrancar el sistema en el que no disponemos de señal de *restart*. Durante este periodo imponemos un ciclo de trabajo fijo de 0.3 durante todo el semiciclo ya que si se utilizasen los valores almacenados en la memoria inadecuadamente pueden llegar a producirse picos de corriente excesivos durante varios semiciclos de red. La señal *dc_enable* indica al resto de los módulos que las señales de *restart* son válidas, es decir, ya son estables.

6.2. Implementación y simulaciones

Siguiendo con la arquitectura modular del sistema de control se ha definido un bloque en lenguaje VHDL que contenga el mecanismo de sincronización llamado *dc_sync*. Como entradas tiene, aparte de la señal de reset y reloj comunes a todos los bloques, la señal de 10 bits procedente del ADC, V_{in} , que representa la tensión de entrada y una señal de 8 bits, *switches*, que indica el estado de los interruptores de la placa necesarios para la

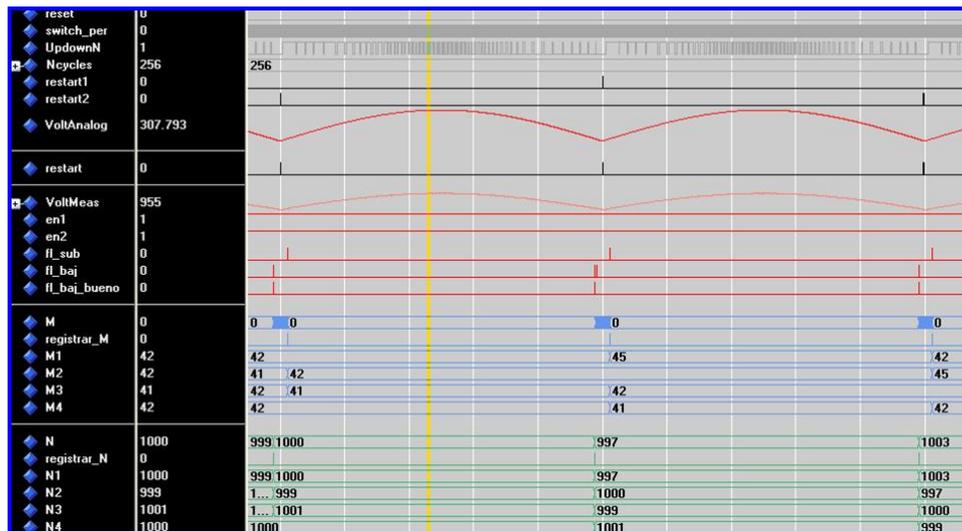


Figura 6.4: Simulación del sistema de sincronización

corrección del sincronismo, como veremos posteriormente. Como salidas tiene la señal de sincronismo *restart* y la señal *dc_enable*.

Como señales de control se han definido dos *enables*. El *enable_1* es '1' cuando los valores calculados de *N* se encuentran comprendidos entre 900 y 1100, indicando que la frecuencia de red es aproximadamente de 50 Hz (en caso contrario no se permite el funcionamiento) y la señal *enable_2* se activa cuando se han calculado 4 muestras de *M* y *N* y sus valores medios lo que indica que se pueden calcular valores válidos de *R*. La señal de *restart* sólo se activará si los dos *enables* están activos. De esta forma protegemos al sistema de una sincronización errónea. La señal *dc_enable* se activa cuando los dos *enables* son '1' y además cuando se ha activado el primer *restart*.

Como se ha comentado anteriormente la señal de *restart* se activa *R* ciclos de conmutación después de cada flanco de bajada. Para generarla se emplea un contador que comienza la cuenta al detectar un flanco de bajada y que se reinicia cuando el valor de la cuenta es igual a *R*. Teniendo en cuenta que en un sistema ideal *N* sería 1000, el valor de *R* será también mayor que 1000, por lo tanto siempre se detectará un nuevo flanco de bajada antes de que el contador haya sido reiniciado. Por lo tanto se hace necesario implementar dos contadores que se utilizan alternativamente cada semiciclo de red y que se activan mediante un *flag* de control.

Para comprobar el funcionamiento se han realizado simulaciones con *Modelsim*. Primeramente se simuló un sistema en el que la tensión de entrada se simuló almacenando sus valores en una memoria y añadiendo a estos un rizado para simular las oscilaciones del ADC. Una vez comprobado el correcto funcionamiento en este sistema se realizó una segunda simulación integrando un ADC sigma-delta. En la figura 6.4 podemos ver una captura de una simulación que incluye un conversor sigma-delta. Puede verse cómo se van calculando los valores de *M* y *N*, la activación de los *enables*, los flancos y las primeras señales de *restart*.

Para calcular los valores medios de *M* y *N* se emplean 3 registros, un sumador y un registro de desplazamiento, ya que los valores medios se actualizan cada semiciclo de red a partir de los 4 últimos valores (figura 6.5).

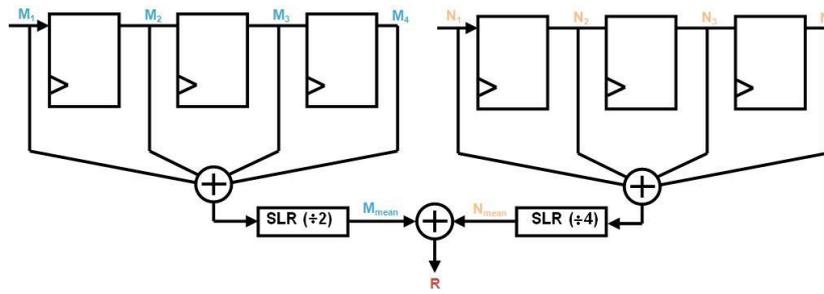


Figura 6.5: Esquema de las operaciones para el cálculo de R

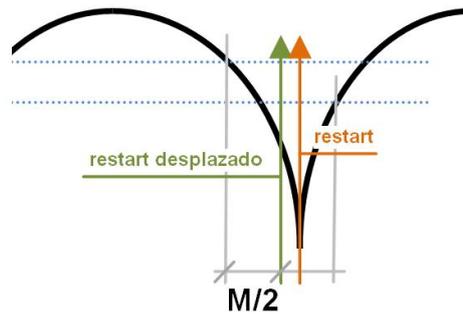


Figura 6.6: Representación de la desviación de la señal de sincronismo

6.3. Resultados experimentales

Para verificar el funcionamiento de la sincronización se han realizado pruebas experimentales comprobando con un osciloscopio las señales de flancos y *restart* obtenidas. Primeramente se realizó un ajuste experimental de los umbrales para que las señales de flancos de subida y bajada se calculasen correctamente, sin flancos repetidos o falsos flancos. Los flancos calculados necesarios para una tensión de entrada de entre 30 y 100 voltios es de 140 y 260 sobre 1000. Una vez obtenidos correctamente los flancos se observó que debido al efecto de los umbrales la señal de *restart* presenta una desviación respecto del paso por cero. Esto es debido a que los puntos de los flancos de subida y bajada no son simétricos respecto el paso por cero y el valor de $\frac{M}{2}$ no es exactamente el punto de sincronización. Podemos ver una explicación gráfica en la figura 6.6 y una captura de osciloscopio de este efecto en la figura 6.7. La señal de color verde es la señal de *restart* y la amarilla la tensión de entrada rectificadas.

Otro efecto observado es que al aumentar la tensión de la señal de entrada varía su forma y al permanecer fijos los valores de los umbrales las señales de los flancos no se detectan exactamente en la misma posición, es decir, que para distintas tensiones de entrada el punto de sincronización obtenido es ligeramente distinto. Para poder compensar este efecto y facilitar la realización de las pruebas con diferentes tensiones de entrada de forma más sencilla se introduce una variable que suma o resta una determinada cantidad a la variable R. El valor de esta variable se introduce mediante 8 interruptores disponibles en la placa de desarrollo utilizada. El bit más significativo indica el signo y los otros 7 representan en binario la cantidad a sumar o restar. De esta forma operando sobre los interruptores podemos obtener valores muy precisos de la señal de sincronismo como

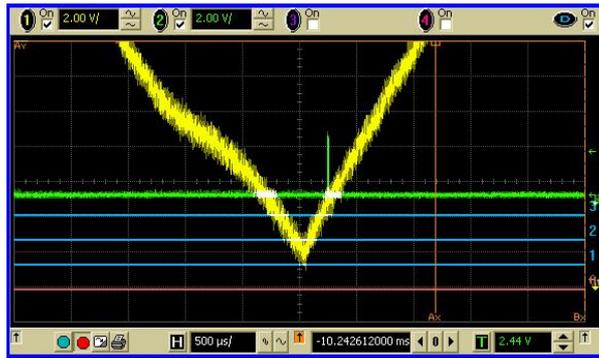


Figura 6.7: Captura del desvío de la señal de restart

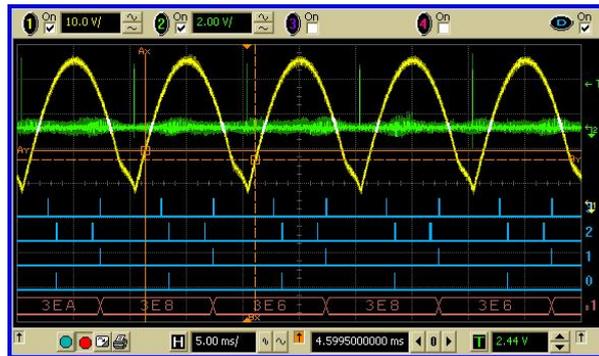


Figura 6.8: Captura de la señal de sincronización

puede verse en la figura 6.8. En el bus representado en rojo se observan los valores hexadecimales del valor de N y como puede verse oscilan un par de valores alrededor de 1000 (E38 en notación hexadecimal), que precisamente es el número de ciclos de conmutación de un semiciclo de red.

Gracias a estos *switches* que corrigen la desviación de la señal de sincronismo puede comprobarse lo crítica que es la sincronización en el sistema. Sólo con variar dos o tres partes sobre mil la señal de sincronismo se obtienen variaciones significativas del factor de potencia. Los *switches* sirven por tanto para comprobar este efecto y para calibrar la señal de sincronismo con distintas tensiones de entrada para obtener el mejor factor de potencia posible. El valor de los *switches* se tienen que cambiar manualmente de forma dinámica según varían los valores de la tensión de entrada, pero en un producto comercial este valor debería ser fijo, o ajustarse dinámicamente según la tensión de entrada medida.

Una pequeña mejora introducida es que para evitar cambios bruscos en la señal de sincronismo al activar por ejemplo los bits más significativos de los *switches* es utilizar uno de los pulsadores como señal de enable, dado que al desplazar la señal de sincronismo de manera incontrolada puede suponer picos de corriente de entrada excesivos. De esta forma se pueden manejar los interruptores y el valor de estos sólo se lee cuando se pulsa el botón de enable.

7

Sistema de control en lazo cerrado de tensión

En este capítulo se analiza el sistema de control en lazo cerrado creado para mantener valores estables de tensión en la salida. Primeramente se hace un repaso de los fundamentos teóricos necesarios para comprender el funcionamiento de los sistemas de control y seguidamente se presenta la forma de implementarlos en dispositivos digitales programables. Finalmente se plantea la solución escogida y su validación teórica.

7.1. Introducción a los sistemas de control: regulación automática

En este apartado se va a presentar un repaso de los conceptos y fundamentos matemáticos necesarios para comprender el funcionamiento de los sistemas de control.

La regulación automática consiste en el control de un proceso en un estado determinado, como por ejemplo el control de temperatura o, como en nuestro caso, el control de la tensión a la salida de un convertidor conmutado. Los sistemas de control son sistemas que al recibir una entrada realizan el control de las señales de salida de forma automática sin intervención humana.

Podemos distinguir entre dos tipos de sistemas de control:

- Sistema de control en lazo abierto

Son sistemas en los que la salida no tiene efecto sobre la acción de control, de modo que para cada entrada de referencia corresponde una condición de operación fija. No tienen control de la salida frente a perturbaciones externas.

- Sistemas de control en lazo cerrado

En estos la acción de control es dependiente del estado de la salida. Al hecho de medir las salidas para actuar en las entradas se le conoce como realimentación.

Los sistemas en lazo cerrado presentan la ventaja de que son relativamente insensibles a perturbaciones externas o a variaciones internas pero tienen como desventaja que pueden

ser inestables y presentar oscilaciones en la respuesta si no se realiza un diseño adecuado, además de ser más complejos y caros. En el método de control propuesto en este proyecto el lazo de corriente es en lazo abierto y el lazo de tensión de salida es en lazo cerrado.

7.1.1. Fundamentos matemáticos de los sistemas de control

Para el estudio y diseño de los sistemas de control se utilizan representaciones matemáticas de los mismos. Un sistema puede caracterizarse como un bloque que realiza una determinada función $g(t)$ cuando se aplica una función de entrada $x(t)$ y presenta una función de salida $y(t)$. Si el sistema es lineal y a la salida aplicamos un impulso unidad $\delta(t)$ a la función $g(t)$ la llamamos respuesta al impulso. Ésta función nos permite caracterizar el sistema para cualquier entrada mediante la **integral de convolución** de esta función con la entrada:

$$y(t) = x(t) * g(t) = \int_0^t g(t - \tau) \cdot x(\tau) \cdot d\tau$$

La operación de convolución no es matemáticamente sencilla y por eso se utilizan transformaciones sobre las señales del sistema. La **transformada de Laplace** para un sistema de control es una particularización de la transformada de *Fourier* y se define como:

$$F(s) = \int_0^{\infty} f(t) \cdot e^{-st} \cdot dt$$

donde s es una variable compleja. La ventaja de operar con las señales en el dominio transformado es que, debido a sus propiedades, la convolución es equivalente al producto en el dominio transformado. La particularización de la transformada de *Laplace* en el dominio discreto se conoce como **transformada Z** y debido a su naturaleza discreta es la que se utiliza para el diseño y estudio de los sistemas de control digitales. En los sistemas discretos las señales se representan mediante series en lugar de funciones continuas. La transformada Z de una señal en tiempo discreto x_k se define matemáticamente como:

$$X(z) = \sum_{k=-\infty}^{\infty} x_k \cdot z^{-k} = Z\{(x_k)\}$$

Por ejemplo, la transformada Z de una señal discreta $u_k = \{1, 1, 1, 1, \dots\}$ que representa la función escalón unidad $u(t)$ es igual a:

$$U(z) = \sum_{k=-\infty}^{\infty} u_k \cdot z^{-k} = \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

Como se ha comentado anteriormente la operación de convolución queda simplificada en el dominio transformador por una operación de multiplicación, por lo tanto:

$$Y(z) = X(z) \cdot G(z)$$

donde

$$G(z) = \frac{Y(z)}{X(z)}$$

es conocida como **función de transferencia** y representa la relación que existe entre la entrada y la salida, y en el dominio transformado queda caracterizado como un simple cociente.

7.1.2. Ecuación en diferencias

Sea la función de transferencia de un sistema:

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}}$$

reagrupando los términos se obtiene que:

$$(a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_n \cdot z^{-n}) \cdot Y(z) = (b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_m \cdot z^{-m}) \cdot X(z)$$

De la expresión anterior podemos obtener la relación entre los elementos de la señal de salida y la señal de de entrada conocida como **ecuación en diferencias**:

$$a_0 \cdot y_k + a_1 \cdot y_{k-1} + \dots + a_n \cdot y_{k-n} = b_0 \cdot x_k + b_1 \cdot x_{k-1} + \dots + b_m \cdot x_{k-m}$$

$$y_k = -\frac{a_1}{a_0}y_{k-1} - \dots - \frac{a_n}{a_0}y_{k-n} + \frac{b_0}{a_0}x_k + \dots + \frac{b_m}{a_0}x_{k-m}$$

Se puede implementar la función de transferencia en un sistema digital de manera sencilla transformando su función de transferencia en una ecuación en diferencias.

7.1.3. Sistemas de control en lazo cerrado

Podemos modelar un sistema mediante un diagrama de bloques, que es una representación gráfica de las funciones realizadas por cada componente del mismo. A cada componente del sistema lo caracterizamos por su función de transferencia. La señal de referencia es restada con la variable realimentada para proporcionar el error. El elemento de control analiza la señal de error y proporciona una entrada al actuador tras el cual se obtiene la salida. Todo sistema de control está sometido a perturbaciones que se pueden modelar como una señal que se suma a la variable manipulada.

Sea la señal de entrada $R(z)$, el error $E(z)$, la variable realimentada $B(z)$, y la salida $C(z)$ se cumple que:

$$C(z) = G(z) \cdot E(z)$$

y

$$E(z) = R(z) - H(z) \cdot C(z)$$

donde $G(z)$ es la función de transferencia del actuador y $H(z)$ es la función de transferencia del elemento realimentación. Por lo tanto la función de transferencia del sistema completo es:

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1 + G(z) \cdot H(z)}$$

Todo sistema de control está sometido a ruidos y perturbaciones externas y podemos caracterizarlas como una señal $N(s)$ que se suma a la variable manipulada, de esta forma si llamamos $C_D(s)$ y $C_R(s)$ a las señales obtenidas a la salida cuando la señal aplicada al sistema es una perturbación o la señal de entrada respectivamente obtenemos las expresiones:

$$\frac{C_D(z)}{D(z)} = \frac{G_2(z)}{1 + G_1(z) \cdot G_2(z) \cdot H(z)}$$

$$\frac{C_R(z)}{D(z)} = \frac{G_1(z) \cdot G_2(z)}{1 + G_1(z) \cdot G_2(z) \cdot H(z)}$$

que representan las funciones de transferencia del sistema al aplicar sólo una señal de ruido o una señal a la entrada. En este caso se denota como $G_1(z)$ a la función de transferencia del elemento de control y como $G_2(z)$ a la función de transferencia del actuador.

De la superposición de las dos expresiones anteriores podemos obtener la ecuación que representa la salida del sistema completo teniendo en cuenta el elemento de control, lazo de realimentación y perturbaciones:

$$C(z) = \frac{G_2(z)}{1 + G_1(z) \cdot G_2(z) \cdot H(z)} (G_1(z) \cdot R(z) + D(z))$$

En la ecuación anterior podemos comprobar que si se cumplen estas dos condiciones:

$$|G_1(z) \cdot H(z)| \gg 1$$

$$|G_1(z) \cdot G_2(z) \cdot H(z)| \gg 1$$

el efecto de la perturbación se anula ($C_N(z)/D(z) \rightarrow 0$).

7.1.4. Elementos de control

Podemos distinguir entre distintos tipos de control según las características de la función de transferencia $G_1(z)$:

- Control proporcional

- Control integral
- Control diferencial

El controlador proporcional es el más comúnmente empleado, el controlador de tipo integral elimina el error en régimen permanente y el diferencial sólo se suele añadir si hace falta mejorar la dinámica del lazo. En la práctica se suelen utilizar combinaciones del control proporcional y el integral (P-I) y es precisamente el sistema de control que se ha utilizado en este proyecto.

Otro aspecto importante a la hora de diseñar el elemento de control es la estabilidad del sistema. Se puede observar la estabilidad mediante la representación en el plano complejo de los polos y los ceros de la función de transferencia, de manera que añadiendo los elementos necesarios al elemento de control podemos proporcionar la estabilidad requerida.

En conclusión, podemos representar un sistema de la forma vista anteriormente caracterizando cada elemento del sistema mediante su función de transferencia, y podemos regular la estabilidad del sistema añadiendo un elemento de control. Además podemos implementar en un sistema digital las funciones de transferencia de una manera sencilla si las representamos mediante su ecuación en diferencias.

7.2. Implementación digital de funciones de transferencia

En este apartado vamos a ver una explicación más específica de cómo se puede implementar una función de transferencia en un sistema digital, concretamente en lenguaje de descripción VHDL.

Para la representación de los coeficientes de la ecuación en diferencias se utiliza la notación $QX.Y$. Esta notación representa un número en coma fija, donde X es el número de bits que corresponden a la representación de la parte entera e Y el número que corresponden a la parte decimal. Se emplea numeración en complemento a 2 y un bit extra para el signo. Por ejemplo el número decimal 0,6321 puede representarse en notación $Q0,7$ mediante el número binario 01010001 donde el primer bit representa el signo positivo y los 7 restantes la parte decimal. Es importante resaltar que el número representado no es exactamente el número original debido a errores de redondeo. En el caso concreto de este ejemplo el valor que representa 01010001 es 0,6328 en lugar de 0,6321. Se trata de un error pequeño pero es necesario tenerlo en cuenta y saber que este error se va propagando si se realizan operaciones sucesivas sobre ese número (como ocurre en los sistemas realimentados).

Utilizar la notación $QX.Y$ simplifica la realización de operaciones de suma y multiplicación. Para multiplicar dos números uno en notación $QA.B$ y otro en notación $QC.D$, la multiplicación resultante de los dos números será un número en notación $Q(A + C + 1).(B + D)$. Es importante resaltar que esta notación no es interpretada por el lenguaje de descripción hardware. A la hora de implementar las señales en VHDL un número en notación $QX.Y$ se representa mediante un vector con signo (*std_logic_signed*) de longitud $X + Y + 1$, y las operaciones de suma y producto se realizan sin tener en cuenta esta notación. Es labor del programador el interpretar el valor de las señales resultantes pero se tiene la ventaja de que es posible abstraerse de las operaciones matemáticas siempre que se definan los vectores con la longitud adecuada.

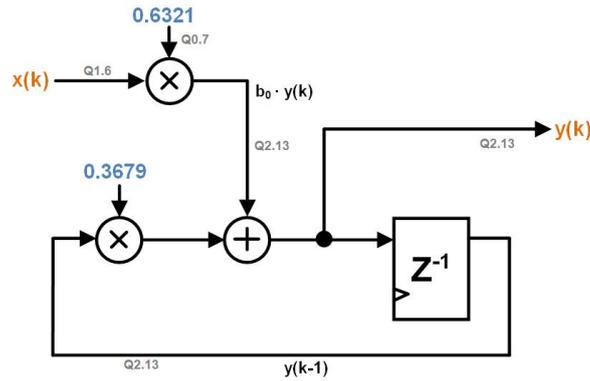


Figura 7.1: Digrama de bloques para un ejemplo de función de transferencia

Como se ha visto anteriormente es posible implementar una función de transferencia de manera sencilla en un sistema digital a partir de su ecuación en diferencias, de forma que sólo es necesario realizar sumas entre señales y multiplicaciones de las mismas por sus cocientes. Por ejemplo el término del numerador de la función de transferencia $a_1 \cdot z^{-1}$ corresponde en la ecuación en diferencias a $a_1 \cdot y_{k-1}$. Un retraso unitario en la señal se implementa en el sistema digital como un registro que se actualiza en cada ciclo del sistema discreto representado.

Veamos un ejemplo práctico en el que se quiere implementar la siguiente función de transferencia:

$$G(z) = \frac{0,6321 \cdot z}{1 - 0,3679 \cdot z^{-1}}$$

cuya ecuación en diferencias es:

$$y(k) = 0,3679 \cdot y(k - 1) + 0,6321 \cdot x(k) + 0 \cdot x(k - 1)$$

De manera que podemos implementarla en lenguaje VHDL de manera sencilla si seguimos el esquema que puede verse en la figura 7.1. Podemos ver que la señal de entrada se representa en formato Q1.6 y la salida en formato Q2.13. En la señal de realimentación es en la que se introduce un mayor error pues se realizan sucesivas multiplicaciones de la señal de salida en formato Q2.13 con el coeficiente 0,3679 en formato Q0.7 y, como es lógico, la longitud del vector que representa el producto de ambas es siempre de la misma longitud finita, por lo que se desprecian bits en la operación.

7.3. Diseño e implementación del lazo de tensión

Una vez explicados los conceptos y rudimentos matemáticos necesarios para implementar sistemas de control en dispositivos programables se va a estudiar el caso concreto del convertidor elevador. El objetivo es obtener la función de transferencia del elemento de control y para el cálculo de dicha función se ha empleado un *toolbox* del programa *MatLab* llamado *sistool*.

La función de transferencia en tiempo continuo $G_{vd}(s)$ de la planta (la del convertidor elevador) es conocida [8]:

$$G_{vd}(s) = G_{do} \cdot \frac{1 - \frac{s}{\omega_z}}{1 + \frac{s}{Q\omega_0} + \left(\frac{s}{\omega_0}\right)^2}$$

donde:

$$\omega_z = \frac{L}{(D')^2 \cdot R}$$

$$\omega_0 = \frac{D'}{\sqrt{L \cdot C}}$$

$$Q = \frac{1}{D' \cdot R \sqrt{\frac{C}{L}}}$$

$$D' = 1 - D$$

Sustituyendo estos parámetros en la función anterior obtenemos:

$$G_{vd}(s) = \frac{V_0}{D'} \cdot \frac{1 - \frac{L}{D'^2 \cdot R} \cdot s}{1 + \frac{L}{D'^2 \cdot R} \cdot s + \frac{L \cdot C}{D'^2} \cdot s^2}$$

Si sustituimos en la función anterior los valores de los distintos componentes empleados ($C = 100$, $L = 5mH$ y $R = 250\Omega$) y valores nominales de tensión de entrada $V_{inef} = 55V$ y de salida $V_{in} = 100V$ obtenemos la siguiente función de transferencia:

$$G_{vd}(s) = 181,818 \cdot \frac{1 - 6,61 \cdot 10^{-5}s}{1 + 6,61 \cdot 10^{-5}s + 1,65 \cdot 10^{-6}s^2}$$

donde $D' = 0,55$ ya que el ciclo de trabajo no es constante a lo largo de un semiciclo de tensión se escoge su valor medio.

Si definimos un periodo de muestreo $T = 10$ ms la función de transferencia en tiempo discreto del convertidor elevador es finalmente:

$$G_{vd}(z) = \frac{158,7z + 121,7}{z^2 - 0,1282 \cdot z + 0,6703}$$

Conocida esta función y utilizando el *toolbox sisotool* de *MatLab* para sistemas de control podemos obtener la función de transferencia del elemento de control. Se ha buscado un diseño estable para un amplio rango de cargas sin que su dinámica sea un factor importante, ya que el lazo de tensión ha de ser más lento que el de corriente en cualquier caso.

$$C(z) = 0,000244 \cdot \frac{z}{z - 1}$$

y su ecuación en diferencias es:

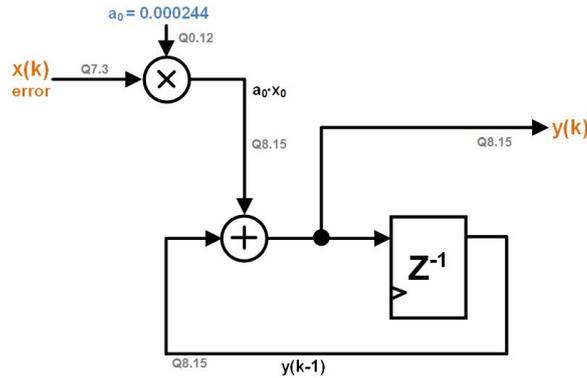


Figura 7.2: Esquema de la función de transferencia del elemento de control

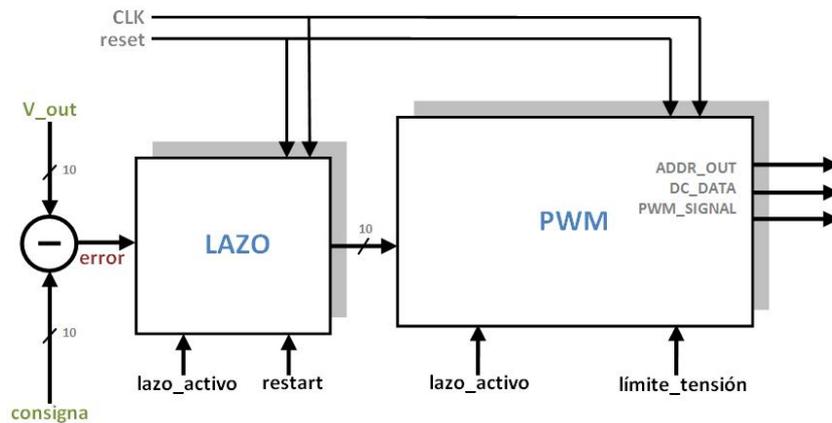


Figura 7.3: Diagrama de bloques del elemento de control y el bloque PWM

$$y(k) = y(k - 1) + 0,000244 \cdot x(k)$$

Dicha función es la que debe implementarse en la FPGA. La señal de entrada $x(k)$ es el error y se obtiene como la diferencia entre la consigna y la medida de tensión del ADC de salida, que para el caso nominal son 100 V. La consigna es el valor que se quiere alcanzar y eligiendo una representación en formato Q7.3 es igual a '01100100000' o $x'320'$ en hexadecimal. El valor que se obtiene a la salida del ADC para una determinada tensión depende de los valores del divisor resistivo. En este caso se ha escogido un divisor resistivo formado por $R_1 = 3,3K\Omega$ y $R_2 = 120K\Omega$ con el cual se obtiene la tensión de saturación para 128 V de salida.

En la figura 7.2 puede verse el esquema de la función de transferencia implementada. La señal de salida $y(k)$ representa la cantidad que ha de sumarse o restarse a los valores de ciclo de trabajo para mantener el error a cero y obtener el valor de tensión requerido en la salida. Como se ve en el esquema esta señal está en formato Q8.15 y para poder sumarla o restarla adecuadamente a los valores del ciclo de trabajo provenientes de la memoria (que son de 10 bits) se toman los bits del 14 al 5 y se concatena el bit de signo.

Se ha añadido una protección para evitar que debido al efecto del lazo no se supere una determinada tensión en la salida. La señal *limite_tensión* se activa cuando la tensión

de salida supera los 120 V e indica al módulo PWM que los valores provenientes del lazo no han de sumarse (o restarse) a los provenientes de la memoria. Igualmente, para poder realizar pruebas experimentales de manera sencilla y comprobar los efectos del lazo se define una señal (*lazo_ activo*) que se controla mediante un pulsador externo de la placa de desarrollo. La figura 7.3 representa un diagrama de bloques de como se interconecta el modulo de control con el módulo PWM y el ADC de salida.

Debido a que el funcionamiento del lazo no puede comprobarse sin el resto del sistema los resultados de las pruebas experimentales se analizan en profundidad en el capítulo 8.

8

Resultados experimentales. Sistema completo

En este capítulo se van a presentar los resultados experimentales obtenidos con el sistema completo, tanto con lazo de tensión como sin él, y un análisis de los efectos que tienen diferentes parámetros en la forma de la corriente de entrada y factor de potencia resultante.

8.1. Descripción de las pruebas

En la figura 8.1 puede verse un esquema de interconexión entre todos los elementos e instrumentos de medida con los que se ha realizado las pruebas experimentales. Los elementos que lo componen son la placa del convertidor elevador y la placa de los componentes analógicos de los dos convertidores sigma delta. La FPGA integra todos los bloques descritos en los capítulos previos, estos son, el mecanismo de sincronización, la memoria con los valores del ciclo de trabajo, dos convertidores sigma delta para las tensiones de entrada y salida, el módulo PWM, y el sistema de control del lazo de tensión. En los anexos puede verse el diagrama de bloques de los elementos que componen la FPGA para el sistema completo.

Como tensión de entrada puede emplearse cualquier toma de tensión europea, pues para ello ha sido diseñado este convertidor conmutado, pero a la hora de realizar las pruebas experimentales es más conveniente disponer de una tensión variable para realizar las pruebas comenzando por tensiones más bajas. Por ello se ha empleado una fuente de alimentación de tensión alterna, la PFC200, del fabricante *Adaptive Power Systems*, la cual ofrece una tensión de salida en un rango de 0 a 300 V eficaces, con una frecuencia de 50 o 60 Hz, y además nos permite establecer limitaciones de corriente, así como medidas de la potencia entregada y del factor de potencia.

Para realizar las medidas de la corriente de entrada se ha empleado una resistencia de $4,7 \Omega$ intercalada ente las pistas de tensión de entrada rectificadas. Colocando una sonda de osciloscopio en los terminales de la resistencia podemos obtener de una forma sencilla la forma de la corriente de entrada.

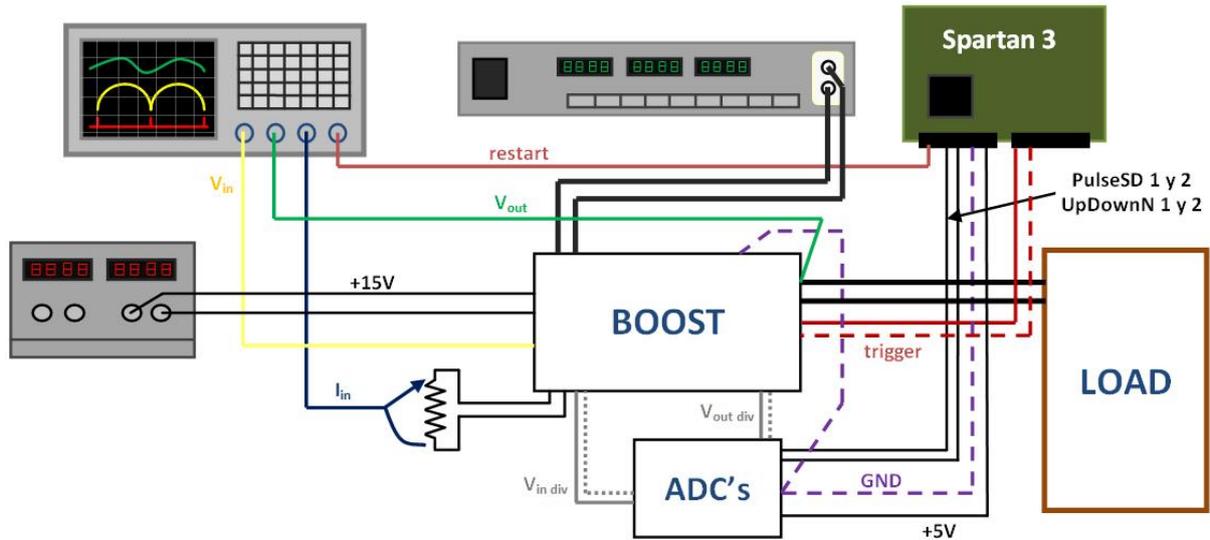


Figura 8.1: Esquema del montaje para las pruebas experimentales

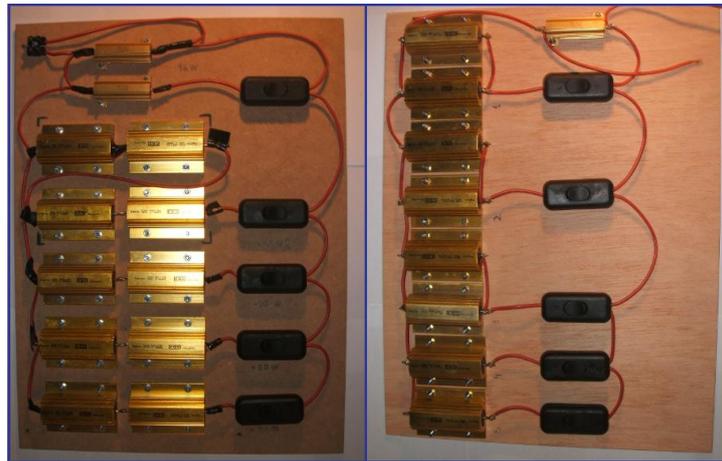


Figura 8.2: Fotografía de las cargas construidas y empleadas para las pruebas experimentales

Para la realización de las pruebas es necesario disponer de una carga en la salida del convertidor que consuma la potencia entregada por éste, ya que de lo contrario la tensión en el convertidor crecería indefinidamente. Por ello se ha diseñado un sistema de cargas creado con resistencias de potencia interconectadas entre sí, y que mediante unos interruptores se puede aumentar o disminuir la potencia total consumida a tensión de salida nominal. Se han diseñado dos sistemas de cargas diferentes, uno para tensiones de entrada bajas (de hasta 120 V aproximadamente) y otro para tensiones más altas (hasta los 230 V de entrada nominales). En la figura 8.2 podemos ver una fotografía de los estos elementos, los cuales se han construido manualmente por parte del alumno. Como puede verse siempre hay conectada una resistencia de 10 k Ω de 50 W, lo cual nos asegura un consumo de potencia mínimo manteniendo un nivel de seguridad. El resto de resistencias están fabricadas para consumir una potencia de hasta 100 W cada una.

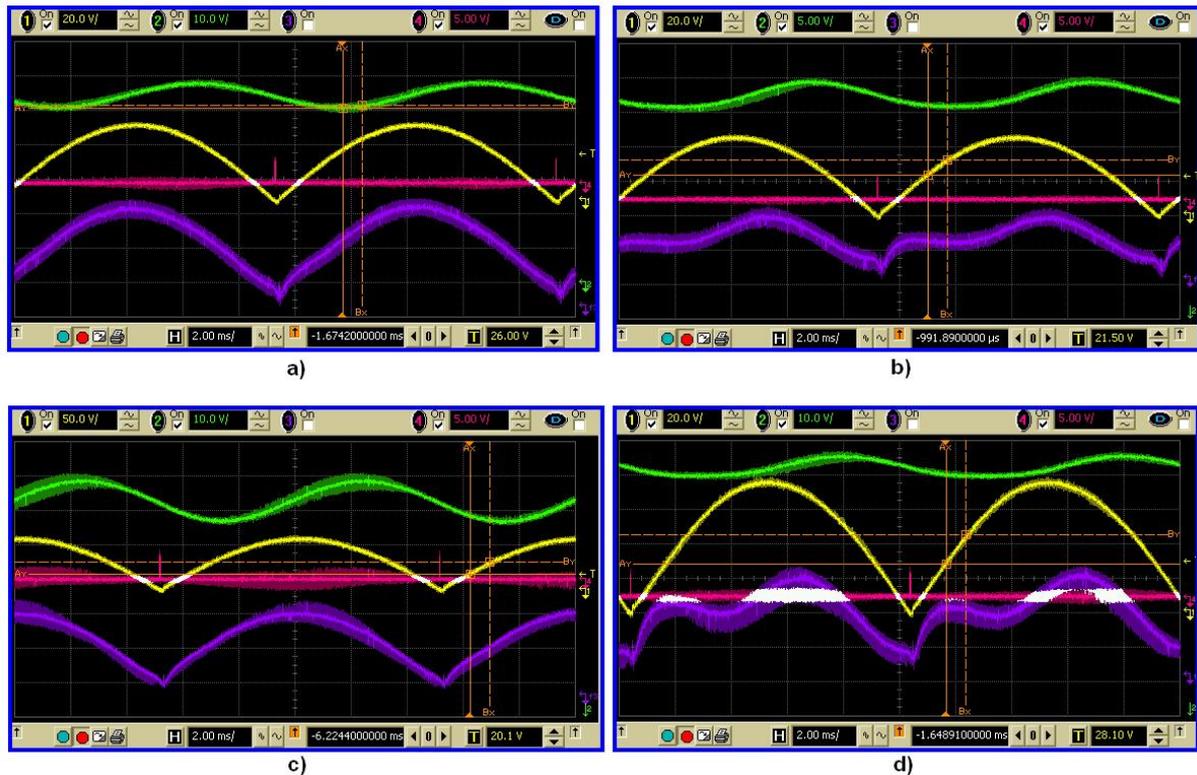


Figura 8.3: Capturas del sistema completo para tensiones de entrada de 33 V y 55 V. Tensión de entrada (amarillo), y de salida (verde) (50 V/div.), corriente de entrada (morado) (1.06 A/div.) y señal de sincronismo (rojo)

8.2. Sistema en lazo abierto

Las primeras pruebas realizadas sobre el convertidor se han hecho en lazo abierto. Para obtener medidas correctas de las tensiones de entrada y de salida se ha elegido como valores de las resistencias del divisor resistivo, 120 k Ω y 3,3 k Ω . Con este divisor resistivo, cuando la tensión medida sea 100 V obtendremos una tensión de 2,676 V, la cual da un amplio margen de seguridad para el comparador del ADC, el cual admite una tensión de entrada de hasta 5 V. En la siguiente tabla presentamos los resultados obtenidos para tensiones de entrada de 33 V y 55 V con distintas cargas de salida, de 250 Ω y 500 Ω :

figura	$V_{in}(V)$	$V_{out}(V)$	$P_{in}(W)$	Carga (Ω)	PF
8.3.a	33	53,9	13,9	250	0,998
8.3.b	33	57,18	7,8	500	0,980
8.3.c	55	91,54	39,0	250	0,997
8.3.d	55	96,4	21,6	500	0,974

En la figura 8.3 podemos ver capturas de osciloscopio de las pruebas anteriores. Las curvas de color amarillo (en el medio) corresponden a la tensión de entrada rectificadas, la de color verde (arriba) es la de salida, la roja es la señal de sincronismo (en el medio, con pulsos) y la azul (abajo) corresponde a la corriente de entrada. Como puede comprobarse la corriente de entrada tiene una forma aproximadamente proporcional a la tensión de entrada y los factores de potencia obtenidos son muy altos, mayores para cargas de 250

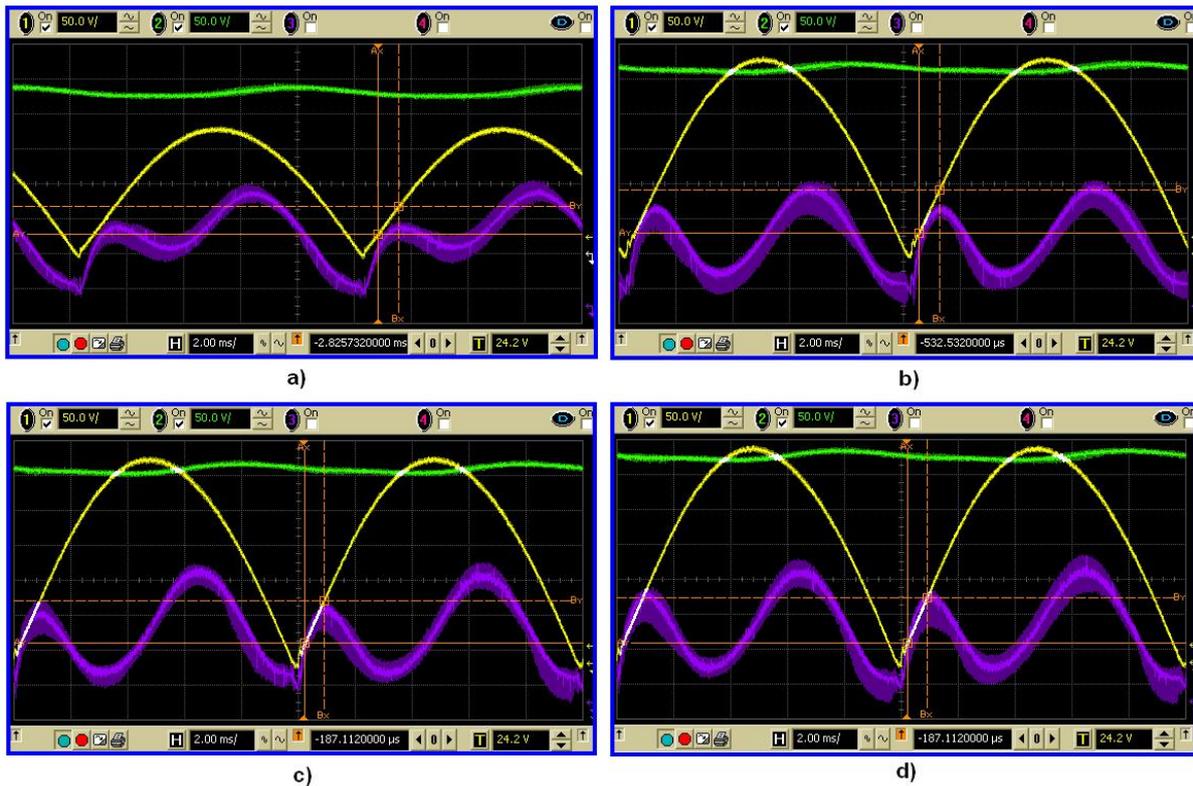


Figura 8.4: Capturas del sistema completo para tensiones de 130 V, 200 V, 210 V y 220 V. Tensión de entrada (amarillo), y de salida (verde) (50 V/div.), y corriente de entrada (morado) (1.06 A/div.)

Ω que para 500 Ω , además puede comprobarse observando las gráficas que para 250 Ω de carga las curvas tienen más aspecto de senoide que para el caso de 500 Ω .

La siguiente tabla muestra pruebas similares a las anteriores pero con mayores tensiones de entrada, de hasta 220 V. Para poder realizar estas pruebas los valores de los divisores resistivos del convertidor han de redimensionarse para que los comparadores del ADC no se saturen y funcionen con un margen de seguridad.

figura	$V_{in}(V)$	$V_{out}(V)$	$P_{in}(W)$	Carga (Ω)	PF
8.4.a	130	230,5	113,0	500	0,902
8.4.b	200	365,0	155,4	1000	0,885
8.4.c	210	384,1	171,0	1000	0,882
8.4.d	220	401,5	187,5	1000	0,884

En la figura 8.4 pueden verse las capturas de osciloscopio de las pruebas anteriores.

8.3. Efecto de la sincronización

Como se ha comentado anteriormente, las simulaciones muestran que el efecto de la sincronización es crítico en el factor de potencia resultante y esto se ha comprobado experimentalmente. Se han realizado pruebas ajustando la señal de sincronismo a su valor correcto (el que ofrece mayor factor de potencia) y después desfasando dicha señal para comprobar el efecto. En la figura 8.5 podemos ver el efecto de desfasar a la izquierda

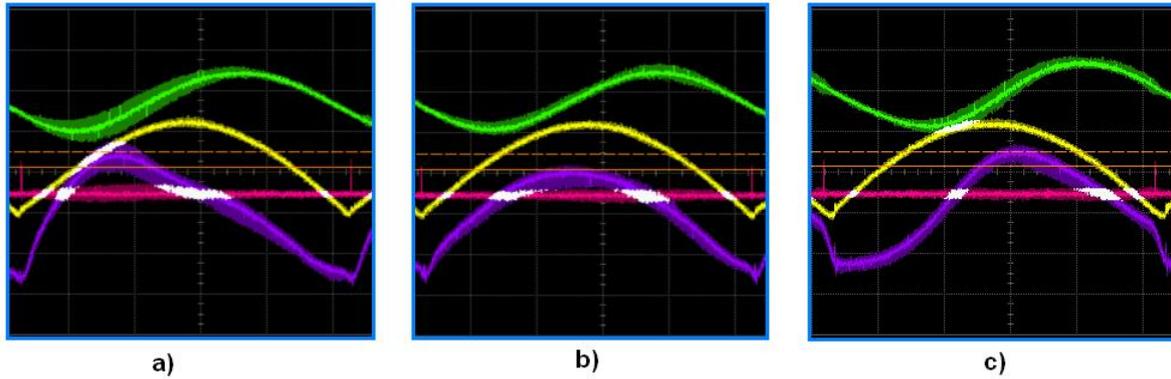


Figura 8.5: Efecto del desfase de sincronismo en la corriente de entrada

(figura a) y a la derecha (figura c) la señal de sincronismo $8 \cdot (180\text{°}/1000)$ grados. La figura b muestra el caso en el que la sincronización está ajustada correctamente. La tabla siguiente muestra los parámetros con los que se han realizado estas pruebas y los valores de factor de potencia obtenidos en cada una:

figura	$V_{in}(V)$	ϕ ($180\text{°}/1000$)	$V_{out} (V)$	$P_{in}(W)$	Carga (Ω)	PF
8.5.a	33	+8	54,6	13,8	250	0,952
8.5.c	33	-8	54,6	13,9	250	0,998
8.5.b	33	0	53,6	14,0	250	0,965
-	33	+16	53,7	14,0	250	0,885
-	33	-16	54,5	14,6	250	0,902

Con los mismos parámetros de entrada, pero con una sincronización correcta, el factor de potencia que se obtiene es 0,998, y podemos observar cómo para pequeños desfases, el factor de potencia se ve afectado negativamente, en mayor medida cuanto mayor es el desfase, y además, acorde con lo comprobado en las simulaciones, el efecto es más significativo cuando el desfase es positivo.

8.4. Sistema en lazo cerrado

De manera similar al primer apartado de este capítulo se han realizado medidas de sistema incluyendo el lazo de tensión. Los datos de las pruebas realizadas se resumen en la siguiente tabla:

figura	lazo activado	$V_{in}(V)$	$V_{out} (V)$	$P_{in}(W)$	Carga (Ω)	PF
8.6.a	X	55	96,4	21,6	500	0,974
8.6.b	SI	55	99,1	22,2	500	0,972
8.6.c	X	130	230,5	113	500	0,974
8.6.d	SI	130	232,4	115,2	500	0,955
8.6.e	X	210	384	171	500	0,922
8.6.f	SI	210	393	180	500	0,825

Las capturas de estas pruebas experimentales pueden verse en la figura 8.6. Puede comprobarse que cuando el lazo de tensión se encuentra activado el factor de potencia

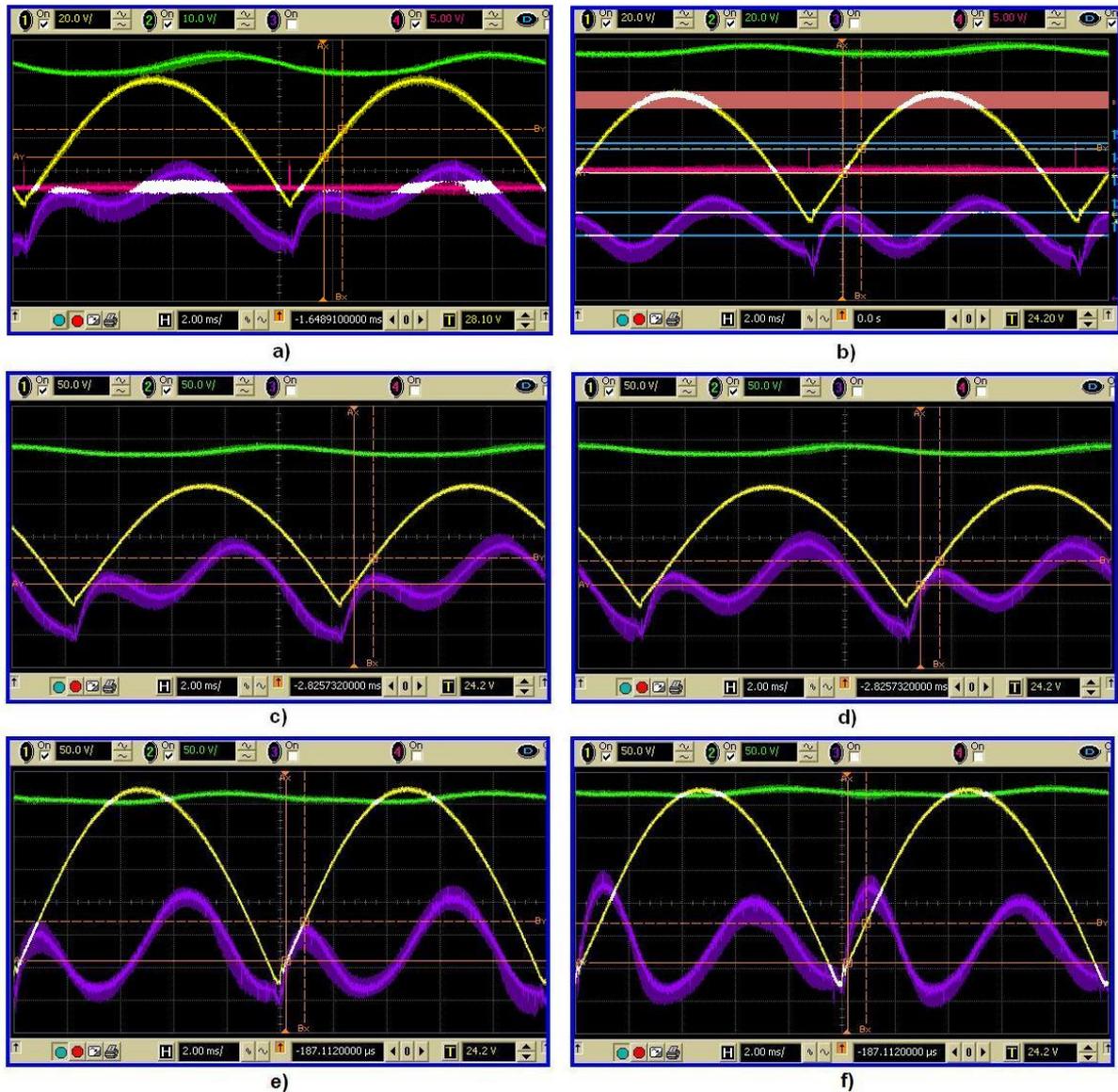


Figura 8.6: Capturas del sistema completo. Comparativa con el lazo activado y sin activar

empeora respecto a cuando no se encuentra activo, para los mismos parámetros de entrada. Esto es debido a que el lazo deteriora los valores del ciclo de trabajo, sumando o restando una cierta cantidad a dichos valores, lo cual implica que se obtiene el ajuste de la tensión de salida pero a costa de empeorar el factor de potencia. Podemos comprobar cómo la tensión de salida aumenta cuando se activa el lazo para compensar las pérdidas y mantener el valor requerido. Igualmente se observa que cuando la tensión de entrada aumenta, la potencia consumida es mayor.

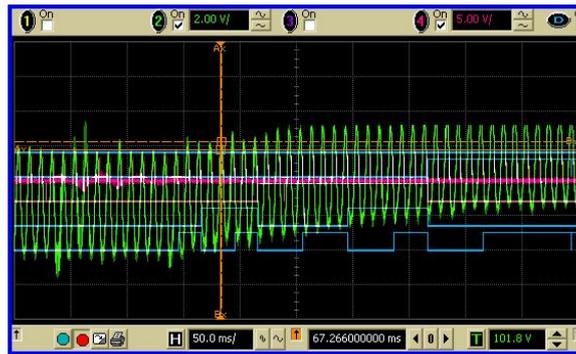


Figura 8.7: Catura del transitorio cuando se activa el lazo de tensión

En la figura 8.7 puede verse una captura que muestra la evolución de la tensión de salida cuando se activa el lazo de tensión. Puede verse el ajuste que sufre la tensión de salida debido al efecto del lazo.

9

Conclusiones y trabajo futuro

9.1. Conclusiones

En este proyecto se ha diseñado, implementado y validado experimentalmente un convertidor elevador con **corrección de factor de potencia sin medida de corriente** controlado por un dispositivo digital programable (FPGA). El método de control habitual es utilizar dos lazos de control, uno de corriente, de dinámica rápida, y otro de tensión, de dinámica más lenta, que actúa sobre el primero. Con el método planteado en este proyecto el lazo de corriente no es necesario y se evita así la medida de corriente, el cual es un proceso complejo y costoso.

Para la medida de las tensiones de entrada y salida se ha diseñado un **convertor analógico digital sigma delta** integrado en la FPGA que permite la utilización de un número mínimo de componentes analógicos externos, ofreciendo así una mayor reproducibilidad y disminuyendo el coste y complejidad de la integración del sistema de control en un ASIC.

A fin de obtener el mejor factor de potencia posible se ha diseñado un mecanismo de sincronización que permite estimar los puntos de paso por cero de la señal de tensión rectificadas. Se han diseñado diferentes mejoras en este mecanismo hasta obtener una medida estable y precisa del paso por cero. Además se ha podido comprobar mediante simulaciones y posteriormente de manera experimental que la sincronización es un elemento crítico del diseño.

Para mantener los valores requeridos de la tensión de salida se ha diseñado un lazo cerrado de tensión, implementado digitalmente.

Con el objetivo de validar experimentalmente el sistema se han diseñado y construido las placas del convertidor elevador y de los componentes necesarios para los convertidores analógico digitales.

En resumen, se ha demostrado que es posible realizar un sistema de control digital basado en un dispositivo lógico programable (**FPGA**) para un convertidor elevador sin necesidad de realizar medida de corriente ni la utilización de lazo cerrado de cor-

riente, obteniéndose unos valores elevados del factor de potencia que cumplen con la normativas europeas de las emisiones de armónicos de corriente para los convertidores alterna-continua.

9.2. Trabajo futuro

Siguiendo con esta línea de investigación surgen diferentes posibles mejoras que se pueden aplicar a este proyecto:

- Para mejorar las prestaciones del sistema frente a ruidos e interferencias y obtener un diseño más compacto, una posible mejora sería diseñar y construir una única placa de circuito impreso que incluyese la FPGA (en lugar de utilizar una placa de desarrollo como la utilizada en este proyecto), los dispositivos necesarios para su funcionamiento (tales como fuente de alimentación o condensadores de desacoplo), y los componentes analógicos de los conversores sigma delta, utilizando componentes SMD (de montaje superficial). De esta manera se podría crear un sistema de control de reducidas dimensiones y compacto con un mejor funcionamiento a altas velocidades cuando las interferencias producidas por las corrientes del convertidor sean mayores.
- Otra posible mejora sería refinar el sistema de sincronización. Como se ha comentado, para realizar una sincronización correcta se ha añadido una botonera que corrige la desviación que se produce en la señal de sincronismo por las variaciones de la tensión de entrada. Para poder tener un convertidor conmutado que opere con distintas tensiones de entrada, se podría realizar un mecanismo que, en función de la tensión de entrada medida, ajustase de manera dinámica y automática la señal de sincronismo.
- Otro punto de estudio propuesto es la mejora del lazo de tensión. En el proyecto realizado el ajuste de los valores del ciclo de trabajo se hace sumando o restando una misma cantidad fija durante un semiciclo de red a los valores precalculados. Otra posibilidad para mejorar el factor de potencia cuando el lazo de tensión está activado es almacenar diferentes curvas del ciclo de trabajo precalculadas para distintos valores de tensión y seleccionarlas dinámicamente en función de la corrección impuesta por el lazo.
- Otra mejora planteada es realizar una versión multifase del convertidor monofásico propuesto, realizando una paralelización del diseño y dividiendo la corriente de entrada entre las distintas fases.

9.3. Publicaciones derivadas de este proyecto

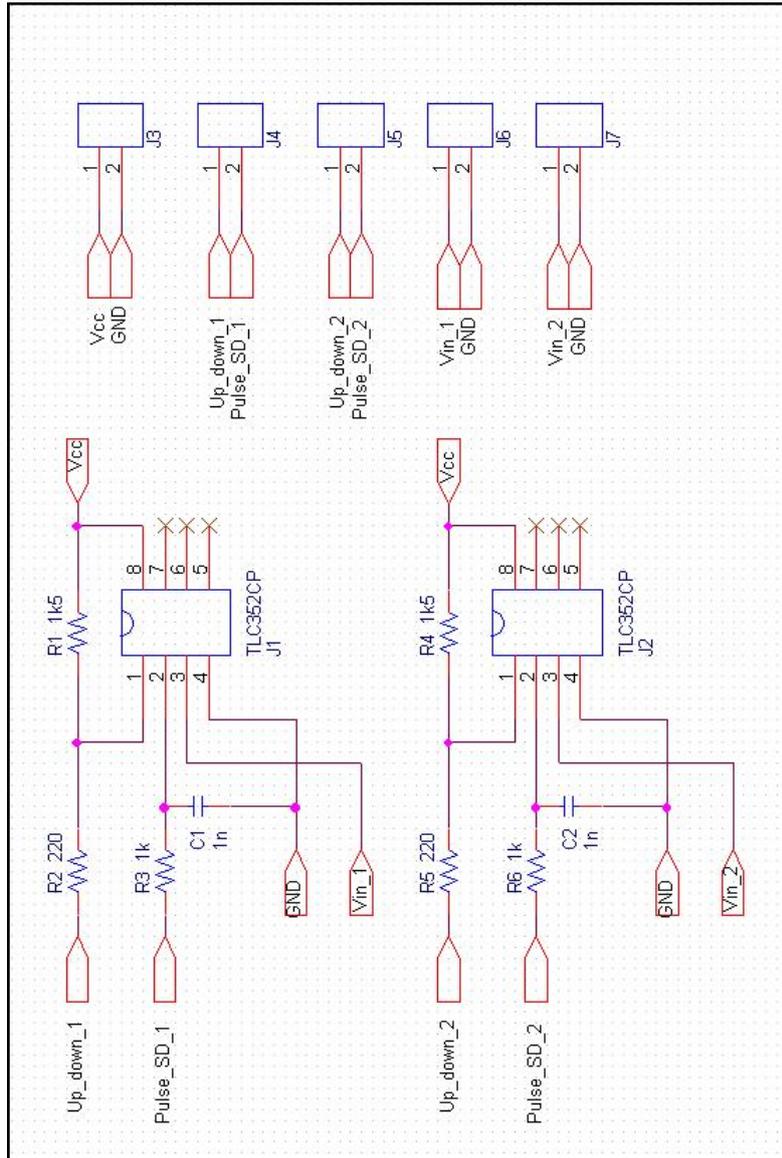
Del trabajo y conclusiones obtenidas de la elaboración este proyecto fin de carrera se ha realizado un artículo [1] titulado *Pre-Calculated Duty Cycle Control Implemented in FPGA for Power Factor Correction*, el cual ha sido enviado para su publicación para el congreso internacional DCIS (*Design of Circuits and Integrated Systems*) del cual todavía no se ha recibido respuesta. Se ha previsto que con las mejoras propuestas en el apartado de trabajo futuro, se realizará una segunda publicación que se enviará a la revista *IEEE Transactions on Power Electronics* para su posible publicación.

Bibliografía

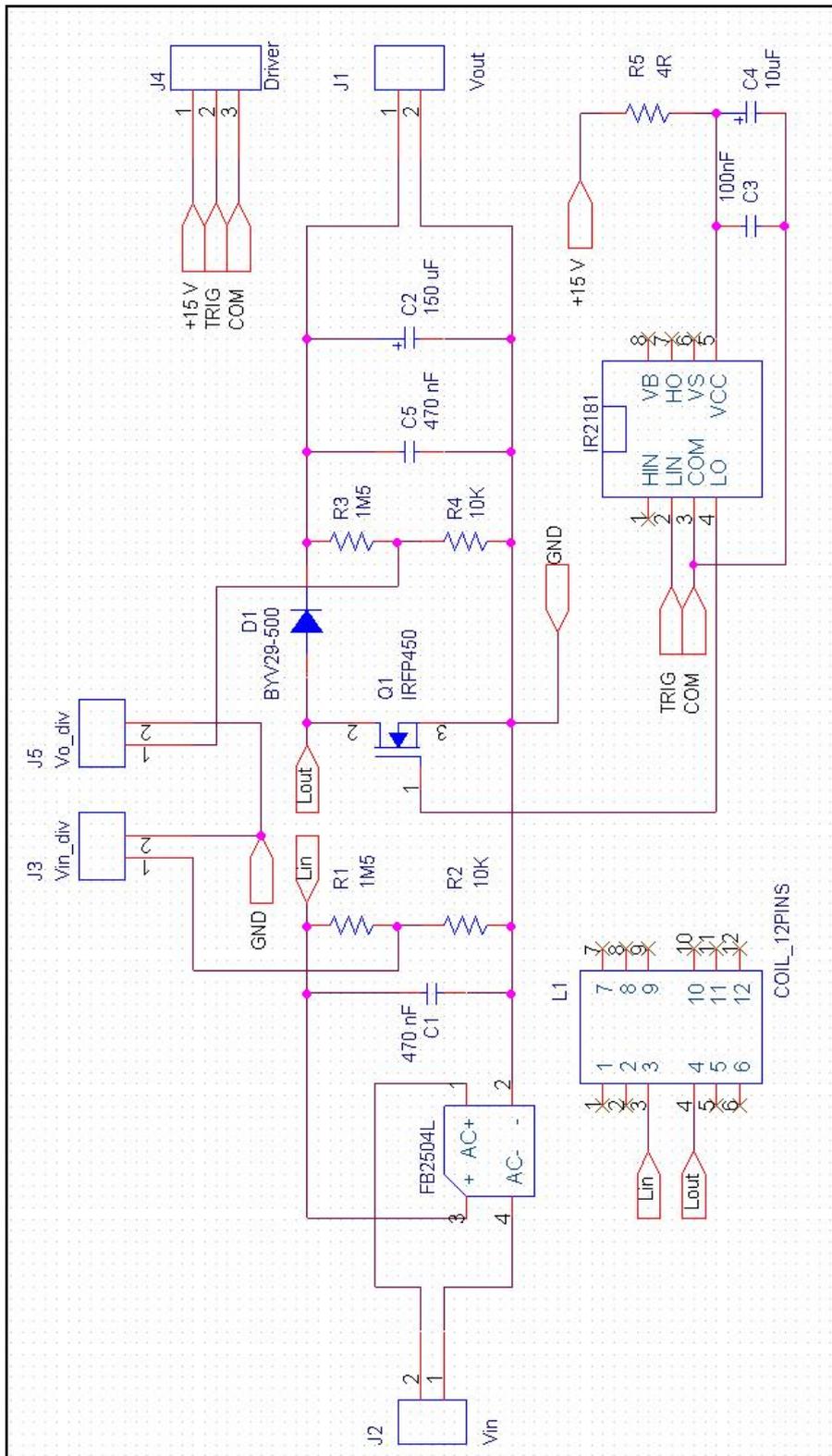
- [1] Alejandro Garcia, Angel de Castro, Oscar Garcia, Francisco J. Azcondo, "*Pre-Calculated Duty Cycle Control Implemented in FPGA for Power Factor Correction*". *Design of Circuits and Integrated Systems (DCIS 2009)*, Pendiente de publicación.
- [2] Francisco J. Azcondo, Angel de Castro, F. Javier Diaz, Oscar Garcia, "*Current Sensorless Power Factor Correction based on Digital Current Rebuilding*". *Proc. Appl. Power Electronics Conf. (APEC)*, Feb. 2009.
- [3] Francisco J. Azcondo, Angel de Castro, "*Power factor correction controllers based on current rebuilding technique implemented on FPGA*". *Proc. Appl. Power Electronics Conf. (APEC)*, Feb. 2009.
- [4] Igor Merfert, "*Analysis and application of a new Control method for continuous-moode boost converters in power factor correction circuits*". *Proc. Power Electron. Speciaists Conference (PESC)*, vol. 1:pp. 69–102, Jun. 1997.
- [5] Igor Wolfram Merfert, "*Stored duty ratio control for power factor correction*". *Proc. Appl. Power Electronics Conf. (APEC)*, vol. 2:pp. 1123–1129, Mar. 1999.
- [6] Wanfeng Zhang, Guang Feng, Yan-Fei Liu, Bin Wu, "*A digital power factor correction (PFC) control strategy optimized for DSP*". *IEEE, Trans. Power Electron.*, vol. 19:pp. 1474–1585, Nov. 2004.
- [7] Wanfeng Zhang, Yan-Fei Liu, Bin Wu, "*A new duty cycle control strategy for power factor correction and FPGA implementation*". *IEEE, Trans. Power Electron.*, vol. 21:pp. 1745–1753, Nov. 2006.
- [8] A. de Castro. *Aplicación del Control Digital basado en Hardware Específico para Convertidores de Potencia Conmutados. Tesis doctoral.* 2003.
- [9] Dragan Maksimovic R.W. Erikson. *Fundamentals of Power Electronics*. Ed. Chapman & Hall, 2 edition, 2001.
- [10] Katsuhiko Ogata. *Ingeniería de control moderna*. Prentice Hall, 3 edition, 1998.
- [11] Katsuhiko Ogata. *Sistemas de control en tiempo discreto*. Pearson Educación, 2 edition, 1996.
- [12] <http://www.fairchildsemi.com/ds/IR/IRFP450B.pdf>.
- [13] <http://www.irf.com/product-info/datasheets/data/ir2181.pdf>.
- [14] <http://www.fagorelectronica.com/semi/pdf/producto/fb25.pdf>.
- [15] <http://focus.ti.com/lit/ds/symlink/tlc352.pdf>.
- [16] http://www.xilinx.com/support/documentation/data_sheets/ds529.pdf.
- [17] http://www.epsma.org/pdf/PFC\%20Guide_April\%202005.pdf.

Esquemáticos

Esquema eléctrico del conversor sigma delta



Esquema eléctrico del convertidor elevador



Código VHDL. Sistema de control

Mecanismo de sincronización:

```
library WORK, IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use IEEE.std_logic_arith.all;
-- use WORK.UAM_UC_Pack.all;

entity adc_sd is
generic (NBits : in integer := 10);
port( Clk : in std_logic;
Reset : in std_logic;
UpDownN : in std_logic;
NCycles : in std_logic_vector(NBits-1 downto 0);
PulseSD : out std_logic;
VoltMeas : out std_logic_vector(NBits-1 downto 0)
--VoltMeasAux : out std_logic_vector(NBits-1 downto 0);
--aux_switch : out std_logic_vector (1 downto 0)--señal para utilizar switches adicionales
);
end adc_sd;
```

```
-----
-- Significado de los puertos y genericos:
-- NBits: numero de bits del ADC, el filtro RC debe filtrar la frec
-- de reloj dividida entre 2**NBits
-- Clk: reloj
-- Reset: reset asincrono activo a nivel alto
-- UpDownN: la entrada que viene del comparador, si esta a '1' hay
-- que incrementar el contador y a '0' disminuirlo
-- NCycles: entrada que indica cada cuantos ciclos cambia el contador
-- PulseSD: la salida del sigma delta que va al filtro RC
-- VoltMeas: la tension medida en NBits. El fondo de escala es la
-- tension de los pines digitales, ya que la tension en filtro RC
-- es la fraccion correspondiente de la tension maxima en el pin
-----
```

```
architecture Behavioral of adc_sd is

--signal UpDownNReg : std_logic;
-- Contadores para el numero de ciclos, el numero de 1s y de 0s
signal CounterCycles : std_logic_vector(NBits-1 downto 0);
signal CounterOnes, CounterZeroes : std_logic_vector(NBits downto 0);
signal Counter : std_logic_vector(NBits-1 downto 0);
-- Se puede limitar el contador entre dos valores, que no tienen por que
-- ser todo unos y todo ceros
constant CountMax : std_logic_vector(NBits-1 downto 0) := (others => '1');
constant CountMin : std_logic_vector(NBits-1 downto 0) := (others => '0');
signal AccSD : std_logic_vector(NBits-1 downto 0); -- Acumulador sigma delta
signal AddSD : std_logic_vector(NBits downto 0); -- Suma sigma delta
signal PulseSDNoReg : std_logic; -- PulseSD sin registrar
```

```
begin

-- Como UpDownN viene del comparador, la registramos al entrar
--UPDOWNNREGISTER: process(Clk, Reset)
--begin
--if Reset = '1' then
--UpDownNReg <= '0';
--elsif Clk='1' and Clk'event then
--UpDownNReg <= UpDownN;
--end if;
--end process;

-- PARA NO RENOVAR EL CONTADOR CADA NCICLOS
CONTADORCICLOS: process(Clk, Reset)
begin
if Reset = '1' then
CounterCycles <= (others => '0');
elsif Clk='1' and Clk'event then
if CounterCycles = NCycles then
CounterCycles <= (others => '0');
else
CounterCycles <= CounterCycles + 1;
end if;
end if;
end process;

CONTADORUNOS: process(Clk, Reset)
begin
if Reset = '1' then
CounterOnes <= (others => '0');
elsif Clk='1' and Clk'event then
if CounterCycles = NCycles then
if UpDownN = '1' then
CounterOnes <= conv_std_logic_vector(1, NBits+1);
else
CounterOnes <= (others => '0');
end if;
else -- CounterCycles < NCycles
if UpDownN = '1' then
CounterOnes <= CounterOnes + 1;
end if;
end if;
end if;
end process;

CONTADORCEROS: process(Clk, Reset)
begin
if Reset = '1' then
CounterZeroes <= (others => '0');
elsif Clk='1' and Clk'event then
if CounterCycles = NCycles then
if UpDownN = '0' then
CounterZeroes <= conv_std_logic_vector(1, NBits+1);
else
CounterZeroes <= (others => '0');
end if;
else -- CounterCycles < NCycles
if UpDownN = '0' then
CounterZeroes <= CounterZeroes + 1;
end if;
end if;
end if;
```

```
end if;
end process;

CONTADORMEDIDA: process(Clk, Reset)
begin
if Reset = '1' then
Counter <= (others => '0');
elsif Clk='1' and Clk'event then
-- Solo cambia si Counter = NCycles
if CounterCycles = NCycles then
if CounterOnes > CounterZeroes then
if Counter < CountMax then
Counter <= Counter + 1;
end if;
elsif CounterOnes < CounterZeroes then
if Counter > CountMin then
Counter <= Counter - 1;
end if;
-- Si CounterOnes = CounterZeroes, ni sube ni baja
end if;
end if;
end if;
end process;

VoltMeas <= Counter;
--VoltMeasAux <= Counter;

--aux_switch <= (others => '1');

-- El acumulador del sigma delta tiene NBits, pero el sumador NBits+1,
-- siendo el mas alto el que sale hacia fuera como PulseSD

AddSD <= ('0'&Counter) + ('0'&AccSD);
PulseSDNoReg <= AddSD(NBits);

ACCUMULATOR: process(Clk, Reset)
begin
if Reset = '1' then
AccSD <= (others => '0');
elsif Clk='1' and Clk'event then
AccSD <= AddSD(NBits-1 downto 0);
end if;
end process;

-- Para que PulseSD salga sin glitches y en ciclos completos de reloj,
-- se registra antes de salir al filtro RC
PULSESDREGISTER: process(Clk, Reset)
begin
if Reset = '1' then
PulseSD <= '0';
elsif Clk='1' and Clk'event then
PulseSD <= PulseSDNoReg;
end if;
end process;

end Behavioral;
```

Conversor analógico digital sigma-delta:

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity DC_SINC is
Port (Clk : in STD_LOGIC;
reset : in STD_LOGIC; --Reset asincrono
ADC : in STD_LOGIC_VECTOR (9 downto 0); --Tension de entrada del ADC
restart : out STD_LOGIC; --Señal de sincronizacion para PWM_module
switch_per : in STD_LOGIC; --Señal de periodo de conmutacion
--restart_inter_aux : out STD_LOGIC;
flanco_sub_bueno_aux : out STD_LOGIC;
flanco_baj_bueno_aux : out STD_LOGIC;
switches : in STD_LOGIC_VECTOR (7 downto 0); --ajustar el valor de M mediante switches
button : in STD_LOGIC; --pulsador para actualizar los switches
dc_enable : out STD_LOGIC);

end DC_SINC;

architecture Behavioral of DC_SINC is

signal en1 : std_logic ; --señal de enable 1
signal en2 : std_logic ; --señal de enable 1
signal fl_sub : std_logic ; --flanco de subida
signal fl_baj : std_logic ; --flanco de bajada
signal fl_baj_bueno : std_logic ; --flancos sin rebotes
signal fl_sub_bueno : std_logic ;
signal umbral1 : std_logic_vector(9 downto 0); --umbral del paso por cero
signal umbral2 : std_logic_vector(9 downto 0); --2 umbrales para considerar histeresis
signal primer_restart : std_logic ;

signal contar_N : std_logic ; --activador de contador de N
signal registrar_N : std_logic ; --indica actualizar nuevo valor de M
signal N : integer range 0 to 2000;
signal N_aux : integer range 0 to 2000;
signal N1 : integer range 0 to 2000; --Señales registradas de M.
signal N2 : integer range 0 to 2000;
signal N3 : integer range 0 to 2000;
signal N4 : integer range 0 to 2000;
signal N_media : integer range 0 to 2000;
signal N_suma : integer range 0 to 8000;
signal N_retocado : integer range 0 to 2000;
signal N_iniciales : integer range 0 to 6 := 0;
signal cont_N : integer range 0 to 2000;

signal contar_M : std_logic ; --activador de contador de M
signal registrar_M : std_logic ; --indica actualizar nuevo valor de M
signal M : integer range 0 to 2000;
signal M1 : integer range 0 to 2000; --Señales registradas de M.
signal M2 : integer range 0 to 2000;
signal M3 : integer range 0 to 2000;
signal M4 : integer range 0 to 2000;
signal M_media : integer range 0 to 2000;
signal M_suma : integer range 0 to 8000;
signal M_iniciales : integer range 0 to 5 := 0;

signal R : integer range 0 to 2000 := 0; --R=(M/2)+N
signal R_reg : integer range 0 to 2000 := 0;
signal R_act : integer range 0 to 2000 := 0;
```

```

signal R_aux : integer range 0 to 2000 := 0;
signal R_2 : integer range 0 to 2000 := 0;
signal ADC_reg : std_logic_vector (9 downto 0);
signal ADC_10 : std_logic_vector (9 downto 0);

signal cont1_activo : std_logic := '0';
signal cont1 : integer range 0 to 2000 := 0;
signal cont2 : integer range 0 to 2000 := 0;
signal restart1 : std_logic;
signal restart2 : std_logic;
signal restart_inter: std_logic;
signal restart_aux : std_logic;
signal restart_aux_aux : std_logic;
signal flancos : integer range 0 to 5 := 0;

signal sw_estable : std_logic_vector (7 downto 0);
-----
--Descripcion de las señales
--en1: enable que controla que M no sea mayor que un cierto valor para arrancar el sistema
-- de sincronizacion. Inicia la cuenta de contM
--en2: enable que se activa despues de disponer de suficientes valores de M con los que
-- realizar la sincronizacion.
--M : numero de ciclos por dos pasos consecutivos por cero
--N : numero de ciclos entre dos puntos iguales de un semiciclo
--Restart: Señal utilizada para reiniciar la memoria del controlador PWM
-----

begin
-----
umbral1 <= "0001100100"; -- 100
umbral2 <= "0010100000"; -- 160
-----

--Detector de flancos de subida y bajada con histeresis
-----
det_flancos: process(clk, reset)
begin
if reset = '1' then
ADC_reg <= (others => '0');
fl_sub <= '0';
fl_baj <= '0';
elsif clk = '1' and clk' event then
ADC_reg <= ADC;
--Flanco de bajada
if ADC < umbral1 and ADC_reg >= umbral1 then
fl_baj <= '1';
else
fl_baj <= '0';
end if;
--Flanco de subida
if ADC > umbral2 and ADC_reg <= umbral2 then
fl_sub <= '1';
else
fl_sub <= '0';
end if;
end if;
end process det_flancos;

--Calculo de M
-----
calcM: process(clk,reset)
begin

```

```
if reset = '1' then --Caso de RESET
M <= 0;
contar_M <= '0';
M_iniciales <= 0;
elsif clk = '1' and clk' event then
if fl_baj = '1' then
--Señal de disparo que inicia la cuenta
contar_M <= '1';
end if;
--La señal registrar_M se activa un unico flanco de reloj cada vez que se obtiene un
--nuevo valor de M.
if fl_sub_bueno = '1' and contar_M = '1' then

--indicador de que se han calculado 4 valores de M iniciales, para enable2
if M_iniciales < 5 then
M_iniciales <= M_iniciales+1;
end if;

contar_M <= '0';
end if;
if registrar_M = '1' then
M <= 0;
end if;
--solo incrementamos cada ciclo de conmutacion
if contar_M = '1' and registrar_M = '0' and switch_per = '1' then
M <= M + 1;
end if;
end if;
end process calcM;

-----

registrar_M <= '1' when fl_sub_bueno = '1' and contar_M = '1' else '0';
fl_baj_bueno <= '1' when fl_baj = '1' and contar_M = '0' else '0';
fl_sub_bueno <= '1' when fl_sub = '1' and M > 0 else '0';

flanco_sub_bueno_aux <= fl_sub_bueno;
flanco_baj_bueno_aux <= fl_baj_bueno;
--N_bin <= CONV_STD_LOGIC_VECTOR(N, 2000);

--Calculo de N
-----
calcN: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
N_aux <= 0;
N <= 0;
N_iniciales <= 0;
flancos <= 0;
registrar_N <= '0';
elsif clk = '1' and clk' event then

--detector de segundo paso por flanco de bajada para obtener un valor correcto de N
if fl_baj_bueno = '1' and flancos < 3 then
flancos <= flancos + 1;
end if;

if flancos >= 1 then --
if fl_baj_bueno = '1' then --flanco bueno
N <= N_aux; -- El valor de N queda registrado hasta un nuevo valor de N
N_aux <= 0;
```

```
registrar_N <= '1';
if N_iniciales < 6 then
N_iniciales <= N_iniciales+1;
end if;
else -- no flanco o falso flanco
registrar_N <= '0';
if switch_per = '1' and N_aux < 1200 then
--el contador de N se incremanta cada periodo de conmutacion
N_aux <= N_aux + 1;
end if;
end if;
end if;
end if;
end process calcN;

--registrar_N <= '1' when fl_baj_bueno = '1' and flancos >= 2 else '0';

--Proceso de activacion de Enables
-----
--En1: Comprobar que se ha calculado un primer valor correcto de N
--En2: Comprobar que se han calculado 4 valores de M iniciales para empezar a realizar la media
enable: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
en1 <= '0';
en2 <= '0';
elsif clk='1' and clk' event then
--proteccion contra frecuencias malas y falta de flancos
if (N > 900 and N < 1100) and N_aux < 1200 then --valores razonables de N
en1 <= '1';
else
en1 <= '0';
end if;
--cuando se ha calculados valores medios de N y M validos
if M_iniciales > 4 and N_iniciales > 4 and en1 = '1' then
en2 <= '1';
else
en2 <= '0';
end if;
end if;
end process enable;

--dc enable es una proteccion cuando el sistema no esta sincronizado, imponiendo
--ciclo de trabajo cero.
--registramos dc_enable durante un periodo de conmutacion
prim_restart: process (clk,reset)
begin
if reset = '1' then
primer_restart <= '0';
elsif clk='1' and clk' event then
if (restart1 or restart2) = '1' then
primer_restart <= '1';
end if;
if en2 = '0' then
primer_restart <= '0';
end if;
end if;
end process prim_restart;

--dc_enable <= en2;
dc_enable <= '1' when en2 = '1' and primer_restart = '1' else '0';
```

```
-----
--Proceso para calcular el valor medio de 4 valores consecutivos de M
M_mean: process(clk,reset,registrar_M)
begin
if reset = '1' then --Caso de RESET
M_media <= 0;
M_suma <= 0;
M1 <= 0;
M2 <= 0;
M3 <= 0;
M4 <= 0;
elsif clk ='1' and clk' event then
if registrar_M = '1' then --cada nuevo valor de M hacemos la media
M1 <= M;
M2 <= M1;
M3 <= M2;
M4 <= M3;
M_suma <= (M1 + M2) + (M3 + M4);
M_media <= M_suma / 4;
end if;
end if;
end process M_mean;
```

```
-----
--Proceso para calcular el valor medio de 4 valores consecutivos de N
N_mean: process(clk,reset,registrar_N)
begin
if reset = '1' then --Caso de RESET
N_media <= 0;
N1 <= 0;
N2 <= 0;
N3 <= 0;
N4 <= 0;
elsif clk ='1' and clk' event then
if registrar_N = '1' then --cada nuevo valor de N hacemos la media
N1 <= N;
N2 <= N1;
N3 <= N2;
N4 <= N3;
N_media <= ((N1 + N2) + (N3 + N4))/4;
--N_media <= N_suma / 4 ;
end if;
end if;
end process N_mean;
```

--Calculo de R

```
-----
R_aux <= (M_media/2) + N_media - 20;
--NOTA: M ya esta dividido por dos en el proceso anterior
-----
```

```
r_filt: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
R_reg <= 0;
R <= 0;
R_act <= 0;
elsif clk ='1' and clk' event then
R_act <= R_aux;
R_reg <= R_act;
if R_reg < R_act then
R <= R_act - 1;
```

```
elsif R_reg > R_act then
R <= R_act + 1;
else
R <= R_act;
end if;
end if;
end process r_filt;

--Proceso para ajustar R con los switches
-----

r_ajust: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
R_2 <= 0;
sw_estable <= (others => '0');
elsif clk='1' and clk' event then
if button = '1' then
sw_estable <= switches;
end if;

if sw_estable(7) = '0' then
R_2 <= R + CONV_INTEGER ('0' & sw_estable(6 downto 0));
else
R_2 <= R - CONV_INTEGER ('0' & sw_estable(6 downto 0));
end if;
end if;
end process r_ajust;

-----

--Sincronizar el restart en funcion de los parametros anteriores
--Hay que introducir dos contadores simultaneos que se solapan y se van alternando.
sincro: process(clk,reset)
begin
if reset = '1' then
cont1_activo <= '0';
cont1 <= 0;
cont2 <= 0;
elsif clk='1' and clk' event then

if en2 = '1' then
--Activar uno u otro contador
if fl_baj_bueno = '1' then
cont1_activo <= not cont1_activo;
end if;

--contador 1
if switch_per = '1' then
if cont1 = R_2 then
cont1 <= 0;
elsif cont1 = 0 and cont1_activo = '1' then
cont1 <= cont1 + 1;
elsif cont1 > 0 then
cont1 <= cont1 + 1;
-- no hay else, para en 0 si no esta cont1_activo
end if;

--contador 2
if cont2 = R_2 then
cont2 <= 0;
elsif cont2 = 0 and cont1_activo = '0' then
cont2 <= cont2 + 1;
```

```
elsif cont2 > 0 then
cont2 <= cont2 + 1;
-- no hay else, para en 0 si no esta cont1_activo
end if;
end if;
end if;
end if;
end process sincro;

restart1 <= '1' when cont1 = R_2 and cont1 > 0 else '0';
restart2 <= '1' when cont2 = R_2 and cont2 > 0 else '0';
restart_inter <= restart1 or restart2;
restart_aux <= restart1 or restart2;
--restart_inter_aux <= restart_inter;

-----
--segundo ajuste del restart
sincro2: process(clk,reset)
begin
if reset = '1' then
cont_N <= 0;
N_retocado <= 1000;
restart <= '0';

elsif clk = '1' and clk' event then
if switch_per = '1' then --sp
if en2 = '1' then
if cont_N >= N_retocado then
cont_N <= 1;
restart <= '1';
--restart_bueno_aux <= '1';
else
cont_N <= cont_N + 1;
restart <= '0';
--restart_bueno_aux <= '0';
end if;

if restart_aux = '1' then
if cont_N = N_retocado then
N_retocado <= N_media;
elsif cont_N > 500 then
N_retocado <= N_media - 1;
elsif cont_N <= 500 then
N_retocado <= N_media + 1;
end if;
end if;
end if;
end if;
end if; --sp
end if;
end process;

end Behavioral;
```

Lazo de tensión:

```
library WORK, IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all; -- conv_integer
use IEEE.std_logic_arith.all; -- conv_std_logic_vector
```

```

entity lazo_tension is
  port( Clk      : in  std_logic;
        Reset   : in  std_logic;
restart : in  std_logic;
        X       : in  std_logic_vector(10 downto 0);
Ypwm : out std_logic_vector(10 downto 0);
lazo_activo : in std_logic
  );
end lazo_tension;

-----
-- Significado de los puertos
-----

-- Clk: reloj del sistema
-- Reset: reset asíncrono del sistema (activo a nivel alto)
-- X: secuencia de entrada en formato Q1.6
-- Y: secuencia de salida en formato Q2.6
-----

-----
-- Formato Q
-----
-- En formato QX.Y se representa el número en coma fija. Se utiliza
-- complemento a 2, con X bits para la parte entera e Y bits para la
-- parte decimal (en total X+Y+1 bits con el de signo).
-----

architecture Behavioral of lazo_tension is

  signal X1 : std_logic_vector(10 downto 0); -- x(k) en formato Q7.3
  signal Y1 : std_logic_vector(23 downto 0); -- y(k-1) en formato Q8.15
  signal Yaux : std_logic_vector(23 downto 0); -- y(k) en formato Q8.15,
signal Yaux_aux : std_logic_vector(23 downto 0); -- y(k) en formato Q8.15,
  constant A0 : std_logic_vector(12 downto 0) := "0000000000001"; -- en formato Q0.7
  signal A0X1 : std_logic_vector(23 downto 0); -- A0*x(k) en formato Q8.15
signal restart_reg : std_logic;
signal restart_flanco : std_logic;

begin

  PIPELINEX: process(Clk, Reset)
  -- Cada z**-1 equivale a un registro
  begin
    if Reset = '1' then
      X1 <= (others => '0');
    elsif Clk = '1' and Clk'event then
X1 <= X;
    end if;
  end process PIPELINEX;

  REGISTROS: process(Clk, Reset)
  -- Cada z**-1 equivale a un registro
  begin
    if Reset = '1' then
      Y1 <= (others => '0');
    elsif Clk = '1' and Clk'event then
if restart = '1' then
if lazo_activo = '1' then
Y1 <= Yaux;
else
Y1 <= (others => '0');
end if;

```

```

end if;
    end if;
end process REGISTROS;

Ypwm <= Yaux (23) & Yaux (14 downto 5);

-----
-- Implementacion de la FDT:
--
--          0.000244z
-- C(z) = -----
--          z-1
--
-- AO: 0.000244*(2**12)=1="000000000001"
--
-----

AOX1 <= AO * X1; -- Q7.3 * Q0.12 => Q8.15
Yaux_aux <= AOX1 + Y1; -- Todo en formato Q2.13
regist_restart: process(Clk, Reset)
begin
if Reset = '1' then
restart_reg <= '0';
restart_flanco <= '0';
elsif Clk = '1' and Clk'event then
restart_reg <= restart;
restart_flanco <= restart and (not restart_reg);
end if;
end process regist_restart;

--Protecciones en la correccion del ciclo de trabajo
proteccion_dc: process(Clk, Reset)
begin
if Reset = '1' then
Yaux <= (others => '0');
elsif Clk = '1' and Clk'event then
if restart_flanco = '1' then --restart un semiciclo posterior
if Yaux_aux (23) = '0' then --signo positivo
if Yaux_aux (22 downto 10) /= "000000000000" then
Yaux <= '0' & "000000000000" & "1111111111";
else
Yaux <= Yaux_aux;
end if;
end if;
else --signo negativo
if Yaux_aux (22 downto 10) /= "111111111111" then
Yaux <= '1' & "111111111111" & "0000000000";
else
Yaux <= Yaux_aux;
end if;
end if;
end if;
end if;
end process proteccion_dc;

end Behavioral;

```

Módulo PWM:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

```

```
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity PWM_module is
port(
restart : in std_logic; --para direccionar la memoria a la posicion inicial
clk : in std_logic; --Reloj de 100 Mhz
dc_data : in std_logic_vector(15 downto 0); --Señal que contiene la informacion PWM
Ypwm : in std_logic_vector(10 downto 0); --señal que contiene la correccion del lazo
button : in std_logic; --pulsador de enable de lazo y ajuste de sincro
aux_switch : in std_logic; --activar/desactivar lazo
pwm_signal : out std_logic; --Señal codificada PWM
pwm_signal_aux : out std_logic; --Señal codificada PWM
addr_out : out std_logic_vector (9 downto 0); --Direccionamiento de la memoria
reset : in std_logic; --señal de reset
switch_per : out std_logic;
dc_enable : in std_logic;
limite_tension : in std_logic;
lazo_activo : out std_logic
);
end PWM_module;

architecture Behavioral of PWM_module is

signal addr_count : integer range 0 to 1000 := 0; --contador para direccionar la memoria
signal addr_aux : integer range 0 to 1023 := 0; --auxiliar de addr en integer
signal addr : std_logic_vector (10 downto 0); --sobra un bit, es el de signo
signal restart_reg : std_logic; --Registrar el restart que se produce en medio de un ciclo de conmutacion
signal dc_data_10 : std_logic_vector (10 downto 0); --10 LSB de la señal de la memoria con bit de signo
signal dc_data_lazo : std_logic_vector (10 downto 0); --señal corregida por el lazo de tension
signal lazo_activo_aux: std_logic;

begin

--10 LSB de los datos de la memoria
dc_data_10 <= '0' & dc_data(9 downto 0);

-----
--Proceso para la activacion del lazo
enable_lazo: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
lazo_activo_aux <= '0';
elsif clk = '1' and clk' event then
if button = '1' then
lazo_activo_aux <= aux_switch;
end if;
end if;
end process enable_lazo;

lazo_activo <= lazo_activo_aux;

data_lazo: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
dc_data_lazo <= (others => '0');
elsif clk = '1' and clk' event then
if lazo_activo_aux = '1' then
dc_data_lazo <= dc_data_10 + Ypwm;
else
```

```
dc_data_lazo <= dc_data_10;
end if;
end if;
end process data_lazo;

--dc_data_lazo <= dc_data_10 + Ypwm when lazo_activo_aux = '1' else dc_data_10;

-----
--Proceso para direccionar la memoria cada 1000 ciclos de reloj.
ADDRESSING: process(clk,reset)
begin
if reset = '1' then --Caso de RESET
addr_count <= 0;
addr_aux <= 0;--(others => '0');
pwm_signal <= '0';
pwm_signal_aux <= '0';
restart_reg <= '1';
elsif clk ='1' and clk' event then

if restart = '1' then
restart_reg <= '1';
end if;
if addr_count = 998 then --cuando llegamos a 999 ponemos la siguiente direccion
para disponer del dato en el siguiente ciclo
addr_count <= addr_count + 1;
if restart_reg = '1' then --si se alcanza la ultima posicion o hay un restart se reinicia
addr_aux <= 0;--(others => '0');
restart_reg <= '0';
else
--si se llega a la ultima posicion no se reinicia, se espera al restart
if addr_aux < 999 then
addr_aux <= addr_aux + 1;
end if;
end if;
end if;

if addr_count = 999 then --cuando llegamos a 1000 reiniciamos
switch_per <= '1';--señal que se activa cada 1000 ciclos para el resto de modulos
addr_count <= 0;
else
addr_count <= addr_count + 1;
switch_per <= '0';
end if;

--limitacion de la tension de salida.
if limite_tension = '1' then
pwm_signal <= '0';
else
if dc_enable = '0' then --ponemos un ciclo de trabajo fijo 0.3
if addr_count < 300 then --generacion de la señal PWM
pwm_signal <= '1';
else
pwm_signal <= '0';
end if;
elsif addr_count < dc_data_lazo then --generacion de la señal PWM
pwm_signal <= '1';
else
pwm_signal <= '0';
end if;
end if;

--señal pwm auxiliar de ciclo 0.3
```

```
-----  
if addr_count < 300 then --generacion de la señal PWM  
pwm_signal_aux <= '1';  
--end if;  
else  
pwm_signal_aux <= '0';  
end if;  
-----  
end if;  
  
end process ADDRESSING;  
  
addr <= conv_std_logic_vector(addr_aux,11);  
addr_out <= addr (9 downto 0);  
  
end Behavioral;
```


Código VHDL. Modelos digitales

Modelo del convertidor elevador:

```
library IEEE,WORK;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use WORK.UAM_UC_Pack.all;
library UNISIM;
use UNISIM.Vcomponents.ALL;

entity Boost_Model is
  port( Vg      : in  real;
        R      : in  real;
        Ir      : in  real;
        Resist  : in  boolean;
        OnOff   : in  std_logic; -- On = '1', off = '0'
        Iin     : out real;
        Vo      : out real
  );
end Boost_Model;

-----
-- Significado de los puertos:
-- Vg: tension de entrada
-- R: resistencia de la carga
-- Ir: intensidad demandada por la carga
-- Resist: indica si el resultado se calculara teniendo en cuenta R
-- (true) o Ir (false), es decir, si la carga se comporta como una
-- resistencia o una fuente de corriente
-- OnOff: interruptor ON a '1' y OFF a '0'
-- Iin: intensidad de entrada
-- Vo: tension de salida
-----

architecture Behavioral of Boost_Model is

  constant Il_init : real := 0.0; -- Valor inicial de Il
  constant Vo_init : real := 400.0; --200.0;-- Valor inicial de Vo
  signal Il : real := Il_init; -- Corriente por la bobina
  signal VoAux : real := Vo_init; -- Auxiliar de Vo para poder leerse
  constant L : real := 5.0e-3; -- Inductancia de L
  constant C : real := 100.0e-6; -- Capacidad de C
  signal OnOffDelay : std_logic := '0';
  --constant Noffon : integer := 20; -- Retraso medido en numero de ciclos del driver de Off a On
  --constant Nonoff : integer := 40; -- Retraso medido en numero de ciclos del driver de On a Off
  --signal ContOffOn : integer := 20; -- Contador de Off a On
  --signal ContOnOff : integer := 40; -- Contador de On a Off

begin
```

```
Iin <= Il;
Vo <= VoAux;

--DELAY: process
--begin
--
-- if OnOff = '1' then -- Le estoy metiendo un retraso de 1 en ambas partes si o si
--if OnOffDelay = '0' then
--if ContOffOn > 0 then
--ContOffOn <= ContOffOn - 1;
--OnOffDelay <='0';
--else
--OnOffDelay <= '1';
--ContOffOn <= Noffon;
--end if;
--end if;
--else
--if OnOffDelay = '1' then
--if ContOnOff > 0 then
--ContOnOff <= ContOnOff - 1;
--OnOffDelay <='1';
--else
--OnOffDelay <= '0';
--ContOnOff <= Nonoff;
--end if;
--end if;
--end if;
-- wait for Cycle;
-- end process DELAY;

CALC: process
-- Cada periodo de integracion se actualizan las variables de estado
begin
--OnOffDelay
if OnOff = '1' then -- interruptor cerrado
  Il <= Il + ( (Vg*dt) / L );
  if Resist then -- Carga resistiva
    VoAux <= VoAux - ( (VoAux/R)*dt / C );
  else
    VoAux <= VoAux - ( (Ir*dt) / C );
  end if;
else -- interruptor abierto
  if Il > 0.0 then -- diodo conduce, CCM
    Il <= Il + ( (Vg-VoAux)*dt / L );
    if Resist then -- Carga resistiva
      VoAux <= VoAux + ( (Il-(VoAux/R))*dt / C );
    else
      VoAux <= VoAux + ( (Il-Ir)*dt / C );
    end if;
  else -- diodo no conduce, DCM
    Il <= 0.0;
    if Resist then -- Carga resistiva
      VoAux <= VoAux - ( (VoAux/R)*dt / C );
    else
      VoAux <= VoAux - ( (Ir*dt) / C );
    end if;
  end if;
end if;
wait for Cycle;
end process CALC;
```

```
end Behavioral;
```

Modelo de la parte analógica del ADC:

```
library IEEE, WORK;
  use IEEE.std_logic_1164.all;
  use IEEE.std_logic_unsigned.all;
  use IEEE.std_logic_arith.all;
  use WORK.UAM_UC_Pack.all;
  use WORK.DWMath.all;

entity ADC_model is
Port(
PulseSD : in STD_LOGIC;
  UpDownN : out STD_LOGIC;
Vg : in real
);
end ADC_model;

architecture Behavioral of ADC_model is

-----
  constant Ndelay : integer := 600; --(delay = N * 1ns)
  constant Hist : real := 0.09; --histeresis del comparador
-----
  signal delayChain : std_logic_vector (1 to Ndelay) := (others => '0');
  constant R : real := 1.0e3; -- Valores comerciales
  constant C : real := 220.0e-12; -- Valores comerciales
  constant f : real := 50.0; --Valor de la frecuencia de la señal analogica en Hz
  constant VMaxPin : real := 3.3; -- Tension con '1' en el pin
  constant VMinPin : real := 0.0; -- Tension con '0' en el pin
  signal Ic : real := 0.0; -- Corriente comun de R y C
  signal Vc : real := 0.0; -- Tension en el condensador del filtro,
  signal UpDownNaux : std_logic := '0';
  signal UpDownNRetr : std_logic := '0';

  signal EndSim : boolean := false;
  signal VoltAnalog : real := 0.0; -- Tension analogica medida
  signal VoltAnalog_aux : real := 0.0; -- Tension analogica medida
  constant ResDiv : real := 0.0066;--0.0066; -- Divisor resistivo
  signal VoltDiv : real := 0.0; -- Tension tras el divisor resistivo

begin

  VoltDiv <= ResDiv * Vg;
  Ic <= (VMaxPin-Vc)/R when PulseSD='1' else (VMinPin-Vc)/R;

  --Simular el comportamiento del condensador
  -----
  CAPACITOR: process
  begin
  while not EndSim loop
  wait for Cycle2; -- Paso de integracion analogico
  Vc <= Vc + ((Ic*dt2) / C); -- Incremento de t analogico
  end loop;
  wait;
  end process CAPACITOR;

  --Simular histeresis en el comparador
```

```
-----  
-- Comparador sin histeresis  
-- UpDownN <= '1' when VoltDiv > Vc else '0';-- after 10 ns;  
-- UpDownNRetr <= UpDownN after 200 ns;  
COMPARADOR: process  
begin  
wait for Cycle;  
if VoltDiv - Vc > Hist then  
UpDownNaux <= '1';  
elsif VoltDiv - Vc < -Hist then  
UpDownNaux <= '0';  
end if;  
end process COMPARADOR;  
  
--Simular retardo del comparador. Se hace como una cadena de retardos.  
--La funcion after funciona mal en este caso.  
-----  
DELAY: process  
begin  
wait for Cycle2;  
delayChain(1) <= UpDownNaux;  
for i in 1 to Ndelay-1 loop  
delayChain(i+1) <= delayChain (i);  
end loop;  
UpDownN <= delayChain (Ndelay);  
end process DELAY;  
  
end Behavioral;
```

Presupuesto

1) Ejecución Material	
▪ Compra de ordenador personal	1.000 €
▪ Software necesario (<i>Xilinx ISE, ModelSim, Orcad Pspice y Matlab</i>)	2.450 €
▪ Material de oficina	150 €
▪ Total de ejecución material	3.600 €
2) Beneficio Industrial	
▪ 6 % sobre Ejecución Material	216 €
3) Gastos generales	
▪ 16 % sobre Ejecución Material	576 €
4) Honorarios Proyecto	
▪ 900 horas a 15 €/ hora	13.500 €
5) Material fungible	
▪ Gastos de impresión	100 €
▪ Encuadernación	150 €
6) Subtotal del presupuesto	
▪ Subtotal Presupuesto	18.142 €
7) I.V.A. aplicable	
▪ 16 % Subtotal Presupuesto	2.902,72 €
8) Total presupuesto	
▪ Total Presupuesto	21.044,72 €

Madrid, Abril 2009

El Ingeniero Jefe de Proyecto

Fdo.: Alejandro García Talón

Ingeniero Superior de Telecomunicación

Pliego de condiciones

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un *Control digital de fuentes de Alimentación*. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales.

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.
8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras,

- o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
 13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
 14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
 15. La garantía definitiva será del 4
 16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
 17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.
 18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
 19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
 20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
 21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
 22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
 23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrataz anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares.

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.