



**UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

**DISEÑO E IMPLEMENTACIÓN DE UN CONTROL GPI EN UN  
FPGA PARA UN CONVERTIDOR CD-CD TIPO REDUCTOR**

**TESIS PARA OBTENER EL GRADO DE  
MAESTRO EN ELECTRÓNICA Y COMPUTACIÓN**

**PRESENTA**

**ING. ERIC WILLIAM ZURITA BUSTAMANTE**

**DIRECTOR DE TESIS**

**DR. JESÚS LINARES FLORES**

**CO-DIRECTOR DE TESIS**

**DR. ENRIQUE GUZMÁN RAMÍREZ**

**HUAJUAPAN DE LEÓN, OAXACA, SEPTIEMBRE DE 2011.**



Tesis presentada en Septiembre de 2011  
ante los siguiente sinodales:

Dr. Antonio Orantes Molina

Dr. José Aníbal Arias Aguilar

Dr. Marco Antonio Contreras Ordaz

**Director de tesis:**

Dr. Jesús Linares Flores

**Co-Director de tesis:**

Dr. Enrique Guzmán Ramírez



*A Jana y Analleli, mis dos tesoros más preciados.*



# Agradecimientos

En primer lugar agradezco a Dios por haberme permitido cumplir un sueño más en mi vida, por darme una oportunidad cada día de ser mejor persona.

Agradezco profundamente a mi esposa Analleli, por ser el motor que impulsa mi vida, por soportarme y amarme todos los días. Gracias Analleli por la paciencia que has tenido.

A mi hija Jana, mi princesita que con sus risas, besos y abrazos, ilumina mi vida a cada instante.

A mi padre Élfego Zurita Torres por todo el apoyo que me ha brindado durante toda mi vida, a mi madre Gema Bustamante Alcázar por todo el amor y cariño inagotable. A mi segunda madre Aurea Santiago Figueroa, por su amor incondicional, por enseñarme que realmente existe el amor. A mis hermanos Didyer, Moctezuma y Quetzalcoatl Zurita Bustamante, por su amistad, cariño y apoyo inquebrantable.

Al Dr. Jesús Linares Flores mi director de tesis, por todo el apoyo brindado en este trabajo. Gracias por su tiempo, por sus valiosos comentarios, por sus palabras en momentos difíciles y sobre todo por ser un amigo.

Al Dr. Enrique Guzmán Ramírez mi co-director de tesis, por su amistad, por el tiempo y apoyo brindado durante este trabajo, por sus valiosas clases y observaciones que hicieron mejorar este trabajo.

A mis sinodales Dr. Antonio Orantes García, Dr. José Aníbal Arias Aguilar y Dr. Marco Antonio Contreras Ordaz por su disposición en la revisión de esta tesis, así como los valiosos comentarios y observaciones que ayudaron a mejorar enormemente el documento.

A mis amigos Andrea (Ande), Ulises (Uli), Fidel (Pollo), Miguel (Miguelin), por todos esos momentos divertidos, gracias por todas las sonrisas, por los momentos de ocio en el laboratorio, por las retas de "Smash" (que al final no

pudieron vencer a "Yoshi") y como olvidar las tlayudas y los "gallos" que se escapaban en el karaoke.

A los encargados del laboratorio Miguelin, Vince, Magdiel y Juan Carlos, por brindarme su apoyo y amistad.

A mi amigo Dayfel, por el gran apoyo que siempre he recibido, su valioso tiempo de grandes pláticas y consejos y por la gran amistad que hemos conservado durante varios años.

A Carmen Martínez Torres, por la ayuda brindada durante mi estancia en el posgrado de la Universidad Tecnológica de la Mixteca, pero sobre todo por su amistad.

A la Universidad Tecnológica de la Mixteca por permitirme ser uno de sus alumnos y forjarme dentro de sus aulas. A mis profesores de la maestría de los que aprendí cosas nuevas e interesantes.

Agradezco al CONACYT por la beca otorgada (Becario 223662), apoyo con el cual pude concluir mis estudios de maestría.

Finalmente agradezco a los desarrolladores de la plantilla en latex  $\text{T}_{\text{E}}\text{X}_{\text{S}}$ , que fue una herramienta muy útil para realizar este documento en latex.



# Resumen

En este trabajo se presenta el diseño e implementación en una tarjeta NEXYS2 Spartan-3E FPGA *Board*, que incluye un FPGA Spartan-3E1200 de la compañía Xilinx, del controlador Proporcional Integral Generalizado mejor conocido en la literatura como GPI, para la regulación del voltaje de salida del convertidor CD-CD tipo reductor. Las características del desempeño que exhibe el controlador GPI en lazo cerrado son: rápida respuesta estática y dinámica ante una excitación constante, sobretiro porcentual menor al 2 %, tiempo de establecimiento corto, rápida recuperación de la señal de referencia ante perturbaciones del tipo constante y rampa desconocidas. Además, de que evita el uso de sensores en la medición de todos los estados del sistema, ya que a través de reconstructores integrales que miden la entrada y la salida se consigue estimar los estados que necesita el controlador para operar. Esta técnica de control se basa en la propiedad de controlabilidad de sistema, la cual implica que el sistema tiene la propiedad de ser diferencialmente plano y mediante la cual el controlador GPI reconstruye los estados no medidos, y demuestra la estabilidad del error del controlador en lazo cerrado. Dentro del mismo FPGA se describió e implementó un modulador de anchos de pulsos (PWM), esto para activar y desactivar el semiconductor del convertidor CD-CD. Para la implementación del controlador propuesto en el FPGA se diseñan y describen los módulos que lo integran bajo el formato de punto flotante en precisión simple del estándar IEEE-754 mediante el lenguaje descriptor de hardware VHDL, siguiendo una metodología descendente (Top-Down). Debido a que el FPGA utilizado en este trabajo no cuenta con un convertidor analógico digital, esto para convertir la señal del voltaje de salida del convertidor reductor a una señal digital, fue necesario utilizar un convertidor ADC0820 externo cuyo tiempo de conversión no afectó de manera significativa el desempeño del controlador en lazo cerrado.

Finalmente, en este trabajo se comparó el desempeño del controlador GPI contra un controlador PID clásico para la regulación del voltaje de salida del convertidor reductor. Resultó que el controlador GPI presenta una mejor respuesta estática y dinámica ante una señal de referencia constante, y una más rápida recuperación de la señal de referencia ante una perturbación súbita estática y dinámica de carga que el controlador PID clásico.



# Índice

<b>Agradecimientos</b>	<b>VII</b>
<b>Resumen</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Estado del Arte . . . . .	5
1.3. Planteamiento del Problema . . . . .	8
1.4. Justificación . . . . .	9
1.5. Hipótesis . . . . .	10
1.6. Objetivos . . . . .	11
1.6.1. Objetivo General . . . . .	11
1.6.2. Objetivos Específicos . . . . .	11
1.6.3. Objetivos Secundarios . . . . .	11
1.6.4. Metas . . . . .	12
1.7. Contenido del documento . . . . .	12
<b>2. Modelado Matemático</b>	<b>13</b>
2.1. Convertidores CD-CD . . . . .	13
2.1.1. Introducción . . . . .	13
2.1.2. Convertidor reductor “ <i>Buck</i> ” . . . . .	14
2.2. Modelo promedio del convertidor reductor . . . . .	14
2.2.1. Modelo promedio del convertidor reductor . . . . .	15
2.2.2. Análisis en estado estable del convertidor reductor . . . . .	17
2.3. Modulador por ancho de pulsos . . . . .	25
<b>3. Planitud Diferencial</b>	<b>27</b>
3.1. Introducción . . . . .	27
3.2. Planitud en sistemas SISO lineales invariantes con el tiempo . . . . .	28
3.3. Controlabilidad en Sistemas Lineales . . . . .	30
3.3.1. Controlabilidad mediante funciones de transferencia . . . . .	30
3.3.2. Controlabilidad mediante la representación de estados . . . . .	34

3.4. Controlador GPI . . . . .	42
<b>4. Métodos y Tecnologías Empleadas</b>	<b>47</b>
4.1. Representación de números en Punto flotante . . . . .	47
4.1.1. Aspectos Generales . . . . .	47
4.1.2. Estándar IEEE para punto flotante . . . . .	49
4.1.3. Suma y Resta en punto flotante . . . . .	51
4.1.4. Multiplicación en punto flotante . . . . .	52
4.2. VHDL . . . . .	53
4.2.1. Descripción de circuitos en VHDL . . . . .	55
4.2.2. Entidad y Arquitectura en VHDL . . . . .	55
4.2.3. Metodología de diseño . . . . .	56
4.3. Arreglo de Compuertas Programables en Campo <i>FPGA</i> . . . . .	57
4.3.1. Arquitectura de un FPGA . . . . .	58
4.3.2. Métodos de programación . . . . .	60
4.3.3. Ventajas de los FPGAs . . . . .	60
<b>5. Plataforma Experimental y Resultados</b>	<b>63</b>
5.1. Introducción . . . . .	63
5.2. Circuito convertidor reductor . . . . .	64
5.2.1. Semiconductores en el circuito reductor . . . . .	66
5.2.2. Circuito impulsor de compuerta . . . . .	68
5.3. Sistema de control GPI . . . . .	71
5.3.1. Módulo de control del ADC . . . . .	75
5.3.2. Módulo del control GPI . . . . .	78
5.3.3. Módulo del convertidor punto flotante a punto fijo. . . . .	88
5.3.4. Modulación por ancho de pulsos . . . . .	91
5.3.5. Simulación del control GPI . . . . .	95
5.4. Resultados Experimentales . . . . .	100
5.4.1. Convertidor reductor a lazo abierto . . . . .	101
5.4.2. Convertidor reductor a lazo cerrado con control GPI . . . . .	103
5.4.3. Comparativa del controlador GPI y PID. . . . .	106
<b>6. Conclusiones y Perspectivas</b>	<b>113</b>
6.1. Conclusiones . . . . .	113
6.2. Perspectivas . . . . .	115
<b>Apéndices</b>	<b>119</b>
<b>A. Cosimulación en Matlab/Simulink y ModelSim</b>	<b>119</b>
A.1. Señal triangular en VHDL . . . . .	119
A.2. Compilando el archivo VHDL . . . . .	120

---

A.3. Creando el Modelo en Simulink . . . . .	122
A.4. Configurando ModelSim para trabajar con Simulink . . . . .	127
A.5. Cargando instancias de la entidad VHDL para cosimulación con Simulink . . . . .	127
A.6. Ejecutando la simulación . . . . .	128
A.7. Terminando la simulación . . . . .	130
<b>B. Cosimulación en Matlab/Simulink y Psim</b>	<b>131</b>
B.1. Introducción . . . . .	131
B.2. Creación del circuito en Psim . . . . .	132
B.3. Creación del bloque en Matlab/Simulink . . . . .	132
<b>C. Diagramas de simulación Simulink</b>	<b>137</b>
<b>D. Circuito Impreso</b>	<b>139</b>
<b>E. Artículos publicados</b>	<b>141</b>
<b>Bibliografía</b>	<b>163</b>



# Índice de figuras

1.1.	Estructura básica de un sistema de potencia. . . . .	2
1.2.	Esquema de un convertidor CD-CD. . . . .	5
1.3.	Diagrama a bloques del control GPI en un FPGA. . . . .	9
2.1.	Circuito eléctrico del convertidor reductor. . . . .	15
2.2.	Estados del circuito eléctrico del convertidor reductor. . . . .	15
2.3.	Circuito eléctrico equivalente para el estado $u = 1$ , estado <i>ON</i> . . . . .	16
2.4.	Circuito eléctrico equivalente para el estado $u = 0$ , estado <i>OFF</i> . . . . .	16
2.5.	Forma de señales en estado estable del convertidor reductor . . . . .	19
2.6.	Circuito equivalente en la transformación de CD de un convertidor reductor. . . . .	21
2.7.	Formas de señal de corriente y voltaje en el capacitor. . . . .	23
2.8.	Modulador de ancho de pulsos PWM . . . . .	26
3.1.	Respuesta del convertidor reductor en lazo abierto. . . . .	45
3.2.	Respuesta del convertidor reductor con la técnica GPI. . . . .	46
4.1.	Formatos en punto flotante. (a) Precisión simple. (b) Precisión doble. . . . .	50
4.2.	Diagrama de flujo de la suma y resta en punto flotante. . . . .	52
4.3.	Diagrama de flujo de la multiplicación en punto flotante. . . . .	53
4.4.	Dupla Entidad-Arquitectura . . . . .	56
4.5.	Arquitectura de un FPGA . . . . .	59
4.6.	Métodos de programación de un FPGA: a) Conexiones SRAM y b) antifusible. . . . .	60
5.1.	Respuesta en lazo abierto de la corriente en la carga del convertidor. . . . .	66
5.2.	Respuesta en lazo abierto del voltaje de salida del convertidor, para una entrada de control $u_{av} = 0,5$ . . . . .	66
5.3.	Corriente de rizo en el inductor. . . . .	67
5.4.	Voltaje de rizo en el capacitor. . . . .	67

5.5. Configuración del circuito impulsor de compuerta y convertidor reductor. . . . .	68
5.6. Voltaje de estabilización compuerta-fuente del Mosfet IRF640. . . . .	71
5.7. Unidades modeladas para el controlador GPI basado en FPGA. . . . .	72
5.8. Tarjeta de desarrollo digital NEXYS-2. . . . .	74
5.9. Entorno de desarrollo Xilinx ISE 12.1i. . . . .	75
5.10. Tiempos necesarios para la conversión del ADC0820. . . . .	76
5.11. Símbolo del módulo de control del ADC. . . . .	78
5.12. Simulación del módulo de control del ADC. . . . .	78
5.13. Diagrama a bloques del control GPI. . . . .	79
5.14. Símbolo del módulo de suma en punto flotante. . . . .	82
5.15. Simulación del módulo de suma en punto flotante. . . . .	83
5.16. Símbolo del módulo de multiplicación en punto flotante. . . . .	85
5.17. Simulación del módulo de multiplicación en punto flotante. . . . .	86
5.18. Simulación de la fórmula directa de segundo orden de Adams-Bashforth. . . . .	87
5.19. Símbolo del módulo de integración en punto flotante. . . . .	88
5.20. Símbolo del módulo de conversión de punto flotante a punto fijo. . . . .	91
5.21. Símbolo del módulo de conversión ADC. . . . .	93
5.22. Simulación del módulo de conversión ADC. . . . .	93
5.23. Símbolo del módulo de onda triangular. . . . .	94
5.24. Simulación de la onda triangular. . . . .	95
5.25. Simulación de la onda triangular realizada en Matlab y Modelsim. . . . .	95
5.26. Símbolo del módulo de comparación. . . . .	96
5.27. Diagrama de bloques del control GPI y PWM implementado en el FPGA. . . . .	96
5.28. Cosimulación entre Matlab y ModelSim. . . . .	97
5.29. En cosimulación, Matlab funciona como servidor, mientras el simulador HDL funciona como cliente. . . . .	98
5.30. Comunicación entre el servidor Matlab y el cliente simulador HDL. . . . .	98
5.31. Circuito convertidor reductor para la cosimulación con Simulink. . . . .	99
5.32. Entradas y Salidas del circuito en Psim. . . . .	100
5.33. Bloque en Simulink con las entradas y salidas del circuito en Psim. . . . .	100
5.34. Voltaje de salida del convertidor reductor a un voltaje de referencia de 4 V en cosimulación. . . . .	101
5.35. Voltaje de salida del convertidor reductor a un voltaje de referencia de 18 V en cosimulación. . . . .	102



5.36. Respuesta en lazo abierto del convertidor reductor a un voltaje de referencia de 18 V. . . . .	102
5.37. Respuesta en lazo abierto del convertidor reductor a un voltaje de referencia de 12 V. . . . .	103
5.38. Respuesta del controlador GPI implementado en un FPGA. . . . .	104
5.39. Desempeño del controlador GPI basado en FPGA al conectar súbitamente una carga estática a la salida del convertidor. . . . .	104
5.40. Respuesta del controlador GPI implementado en un FPGA al conectar súbitamente un motor de CD. a) Voltaje de referencia b) Señal de control $u_{av}$ . . . . .	105
5.41. Desempeño del controlador GPI al conectar una carga dinámica y un freno mecánico. . . . .	106
5.42. Diagrama a bloques del control PID en lazo cerrado. . . . .	107
5.43. Desempeño del controlador PID para el convertidor reductor en cosimulación. . . . .	109
5.44. Diagrama a bloques del control PID implementado en FPGA. . . . .	109
5.45. Respuesta transitoria del convertidor reductor con los esquemas de control GPI y PID. . . . .	111
5.46. Respuesta de la señal de voltaje de salida al conectar súbitamente un motor de CD. . . . .	111
A.1. Entidad en VHDL de una onda triangular. . . . .	119
A.2. Consola en ModelSim. . . . .	122
A.3. Alineación de bloques para cosimulación. . . . .	122
A.4. Parámetros del bloque <i>Step</i> . . . . .	123
A.5. Propiedades del bloque <i>Data type conversion</i> . . . . .	124
A.6. Parámetros de la pestaña puertos del bloque <i>HDL Cosimulation</i> . . . . .	125
A.7. Parámetros de la pestaña Conexión del bloque <i>HDL Cosimulation</i> . . . . .	125
A.8. Parámetros de la pestaña reloj del bloque <i>HDL Cosimulation</i> . . . . .	126
A.9. Parámetros de la pestaña de escalas de tiempo del bloque <i>HDL Cosimulation</i> . . . . .	127
A.10. Conexión de los bloques del modelo en Simulink. . . . .	127
A.11. Configuración de parámetros de simulación. . . . .	128
A.12. Mensajes al cargar la entidad del modelo triangular. . . . .	128
A.13. Señales de la entidad triangular. . . . .	129
A.14. Señales simuladas en ModelSim. . . . .	129
A.15. Señal triangular obtenida en Simulink. . . . .	130
B.1. Circuito del convertidor reductor en Psim, utilizado para simulación. . . . .	132
B.2. Bloques utilizados en Simulink para cosimulación. . . . .	133

---

B.3. Cuadro de diálogo para configurar la ruta del archivo en Psim.	133
B.4. Nodos de entrada y salida del módulo <i>SimCoupler</i> .	133
B.5. Bloque <i>SimCoupler</i> conectado con Psim.	134
B.6. Conexión de bloques en el modelo.	134
B.7. Parametros de simulación del modelo.	135
B.8. Bloque de ganancia agregado para evitar problemas con el bloque de <i>SimCoupler</i> .	135
B.9. Resultado de la cosimulación entre Psim y Matlab/Simulink del convertidor reductor.	135
C.1. Diagrama en Simulink para simulación del control GPI con Psim.	137
C.2. Circuito del convertidor reductor en Psim, utilizado para co- simulación.	137
D.1. PCB del convertidor reductor y convertidor analógico digital.	139
D.2. Guía de montaje de la placa de circuito impreso ( <i>Top silk</i> ).	140
D.3. Vista 3D de la placa de circuito impreso	140

# Índice de Tablas

3.1. Parámetros para el controlador GPI. . . . .	45
4.1. Resumen de los parámetros de los formatos IEEE-754. . . . .	50
4.2. Casos especiales para formato en precisión simple del estándar IEEE-754. . . . .	51
5.1. Parámetros utilizados para el cálculo del capacitor <i>bootstrap</i> . . . . .	70
5.2. Parámetros utilizados para el cálculo de la resistencia de compuerta. . . . .	71
5.3. Descripción de tiempos para la conversión del ADC0820. . . . .	77
5.5. Descripción de señales del módulo de control del ADC. . . . .	77
5.4. Descripción de pines del convertidor analógico digital ADC0820. . . . .	77
5.6. Algoritmo para la suma en punto flotante. . . . .	81
5.7. Descripción de señales del módulo de suma en punto flotante. . . . .	82
5.8. Algoritmo para la multiplicación en punto flotante . . . . .	84
5.9. Descripción de señales del módulo de multiplicación en punto flotante. . . . .	85
5.10. Descripción de señales del módulo de integración en punto flotante. . . . .	88
5.11. Algoritmo para la conversión de punto flotante a punto fijo. . . . .	90
5.12. Descripción de señales del módulo de conversión de punto flotante a punto fijo. . . . .	92
5.13. Descripción de señales del módulo de conversión ADC. . . . .	93
5.14. Descripción de señales del módulo de onda triangular. . . . .	94
5.15. Descripción de señales del módulo de comparación. . . . .	96
5.16. Comparativa entre los controladores GPI y PID para el convertidor reductor. . . . .	112
A.1. Código en VHDL para la generación de una onda triangular . . . . .	121



# Capítulo 1

## Introducción

**RESUMEN:** En este capítulo se presenta un panorama general acerca de los convertidores CD-CD y del control GPI. Se muestra el estado del arte que hay detrás de este trabajo y que da hincapié a realizar esta tesis, por lo cual se muestra también el planteamiento general del problema, las hipótesis, y la justificación y los objetivos que se alcanzarán al terminar el presente trabajo.

### 1.1. Introducción

En la industria moderna nos encontramos con dos necesidades fundamentales que deben ser atendidas por la ingeniería electrónica. Por una parte se requiere disponer de sistemas electrónicos que permitan realizar la conversión entre las diferentes formas en las que se maneja la energía eléctrica y por otra, son necesarios sistemas electrónicos que permitan controlar los procesos industriales en los que se utiliza esta energía. La electrónica de potencia será aquella parte de la electrónica encargada del control y la conversión de la energía eléctrica (ver [1]). Como ejemplo de aplicación de la electrónica de potencia podemos pensar en una de las necesidades más frecuentes de la industria, el control de la velocidad de un motor de corriente directa. Para realizar este control normalmente se recurre a variar el voltaje de armadura con que se alimenta directamente el motor. Puesto que la energía eléctrica disponible en la red es de corriente alterna, se necesita disponer de un sistema electrónico que permita realizar la conversión de la energía eléctrica de corriente alterna (CA) a energía eléctrica de corriente directa (CD), además, que realice el control del voltaje de corriente directa de forma que permita regular eficientemente la velocidad del motor según las necesidades del proceso [2]. Pese a que los procesos de conversión y control de la energía eléctrica pueden ser realizados por el área de ingeniería eléctrica clásica (electrotecnia), lo cierto es que cada vez más se está imponiendo la utilización de las

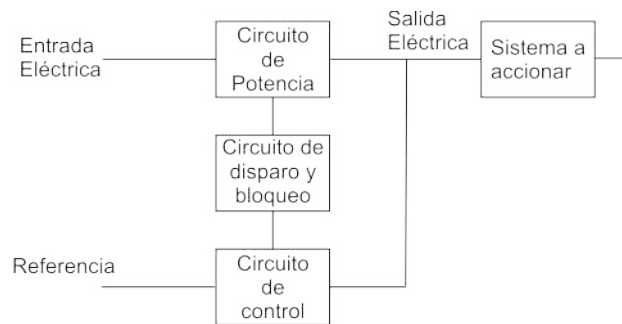


Figura 1.1: Estructura básica de un sistema de potencia.

soluciones aportadas por la electrónica de potencia. Esto se debe a que las ventajas aportadas por la electrónica superan con creces a los inconvenientes, los cuales, además, cada vez son menores debido al constante desarrollo de técnicas y componentes en el campo de la electrónica de potencia [3, 4]. A continuación se indican las ventajas e inconvenientes de la electrónica de potencia respecto a la electrotecnia clásica en el estado actual de la técnica.

#### VENTAJAS:

- a) Mejores características eléctricas (rapidez, etc.).
- b) Mayor fiabilidad y duración.
- c) Carencia casi total de mantenimiento (no existen partes móviles).
- d) Ausencia de vibraciones.
- e) No hay arco eléctrico, evitándose así inconvenientes tales como el desgaste de contactos, generación de ruido electromagnético.

#### DESVENTAJAS:

- a) Menor robustez eléctrica, es decir, menor capacidad para soportar sobretensiones y sobreintensidades.
- b) Algunos montajes son muy caros. Esta desventaja es cada vez menor, debido al avance en la fabricación de semiconductores de potencia.

Todos los sistemas electrónicos de potencia presentan una estructura básica similar formada por tres bloques: el circuito de potencia, el circuito de disparo y bloqueo y el circuito de control, tal y como se muestra en la figura 1.1.

El circuito de potencia comprende dispositivos de potencia que al agruparse forman diferentes topologías correspondientes a cada tipo de convertidor. Los circuitos de potencia se encargan de actuar sobre la energía eléctrica presente a la entrada del sistema para convertirla en energía eléctrica, con la forma deseada, y la cual está disponible a la salida. El circuito de control, como lo indica su nombre, se encarga de controlar el proceso de conversión de energía. Este control se realiza comparando la salida del sistema con la salida deseada, y conforme al resultado, se generan las señales necesarias para disparar y bloquear los semiconductores de potencia de forma adecuada [1, 5, 6].

Los circuitos de disparo y bloqueo se encargan de dar a las señales que provienen de los circuitos de control los niveles de voltaje y corriente adecuados para poder disparar y bloquear los semiconductores de potencia, además proporcionan aislamiento galvánico necesario entre la etapa de potencia y la de control [1, 5, 6]. Los sistemas electrónicos de potencia se clasifican según el tipo de conversión de energía que realizan. Por tanto se puede clasificar a los convertidores en la siguiente forma:

Rectificadores o convertidores CA–CD. Son convertidores de corriente alterna a corriente continua. Entre sus aplicaciones se encuentran:

- Alimentación de equipos electrónicos.
- Control de motores de CD.
- Transporte.
- Transporte de energía en CD y alta tensión.

Reguladores de corriente alterna o convertidores CA–CA. Estos convierten voltaje alterno de una determinada frecuencia y amplitud, en un voltaje alterno de amplitud variable a la misma frecuencia. Algunas aplicaciones son:

- Calentamiento industrial.
- Control de nivel de iluminación.
- Control de velocidad de motores de inducción.

Inversores o convertidores CD–CA. Son convertidores de corriente continua a corriente alterna. Posibles aplicaciones pueden ser:

- Conversión de CD a CA en líneas de transporte de CD.
- Accionadores de motores de CA en aplicaciones industriales.

- Suministros de aeronaves.
- Calentamiento por inducción.

Choppers o convertidores CD-CD. Este tipo de convertidores, transforman voltaje continuo en otro de amplitud variable. Pueden ser utilizados en:

- Alimentación y control de motores de CD a partir de baterías.
- Automóviles y demás vehículos eléctricos.
- Fuentes de alimentación de equipos de cómputo.
- Telefonía celular.
- Fuentes ininterrumpidas (UPS, *Uninterruptible Power Supply*).

Este trabajo de investigación abordará en específico el control en lazo cerrado del convertidor CD-CD tipo reductor, por ello es de interés el hacer un estudio a detalle de este tipo de convertidor, a continuación se proporcionará la definición exacta de este dispositivo y sus diferentes tipos de leyes de control que se han diseñado e implementado a lo largo del tiempo.

**DEFINICIÓN 1** . *Un convertidor CD-CD es un circuito electrónico de potencia que transforma, a partir de una fuente de energía constante (voltaje o corriente), a un voltaje o corriente constante regulado (que no depende de las variaciones en la fuente de entrada y de la carga) y de amplitud configurable [5].*

En los convertidores de potencia se distinguen dos etapas principales, las cuales se muestran en la figura 1.2. La etapa de potencia está constituida por interruptores electrónicos de potencia (diodos, MOSFETs, BJTs, etc.), componentes pasivos de filtrado y almacenamiento de energía (inductores y capacitores). Esta etapa es caracterizada por ser la que realiza la transferencia de energía entre la entrada y la salida, de forma que los componentes que la integran sean sometidos a voltajes y corrientes del mismo orden de magnitud que las existentes en la fuente de entrada y la carga.

La etapa de control constituye la parte principal de convertidor y es la encargada de decidir la activación y bloqueo de los interruptores a fin de conseguir una salida regulada. En esta etapa no se realizan operaciones de transferencia de energía y por tanto está constituida por elementos de pequeña potencia. En la siguiente sección se abordará el estado del arte de los diversos trabajos publicados en el diseño e implementación de diferentes leyes de control para el convertidor CD-CD tipo reductor.



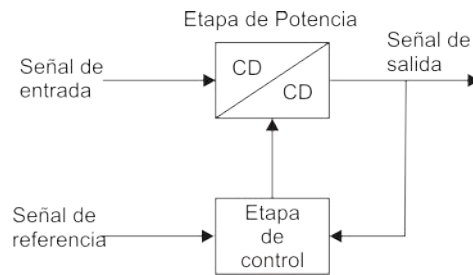


Figura 1.2: Esquema de un convertidor CD-CD.

## 1.2. Estado del Arte

En años recientes han sido reportados algunos trabajos relacionados con el convertidor CD-CD tipo reductor (Buck); la temática abordada por dichos trabajos está relacionada con cuestiones que van desde el modelado matemático, control y aplicaciones específicas tales como elevador eléctrico espacial alimentado vía paneles solares, cargadores eficientes de baterías a través de paneles solares, construcción de convertidores alimentados por celdas solares que extraen eficientemente la máxima potencia de salida de un panel solar bajo diferentes condiciones meteorológicas, etc., [7, 8, 9, 10].

Por otro lado, las diferentes técnicas de control que han sido diseñadas e implementadas para el convertidor reductor en diferentes aplicaciones van desde controladores discontinuos (control por histéresis, y control por modos deslizantes) hasta controladores continuos, como por ejemplo el control GPI (*Generalized Proportional Integral*), control por planitud diferencial, control PID lineal y no lineal y control basado en pasividad [11, 12, 13, 14].

En este trabajo nos enfocaremos a la técnica de control GPI propuesto por Fliess y colaboradores, esta técnica ha sido implementada para el control de convertidores de tipo CD-CD y para el control de sistemas electromecánicos. La técnica GPI presenta las siguientes características en su desempeño en lazo cerrado: rápida respuesta dinámica, robustez ante perturbaciones del tipo constante y rampa desconocidas y reducción en el uso de sensores en la medición de los estados del sistema. Para la reducción en el uso de sensores en su implementación, la técnica GPI se basa en reconstructores integrales de estado midiendo únicamente la entrada y la salida del sistema [15, 16]. (Ver además [17, 18]). En [19] se presenta un esquema de control GPI para controlar el torque de un brazo simple flexible donde la única variable medida del sistema es el torque. El esquema GPI propuesto es robusto con respecto a los efectos de perturbaciones desconocidas de fricción de Coulomb. Un trabajo similar se observa en [20], donde se presenta un controlador adaptable para un brazo flexible, el cual se basa en identificaciones algebraicas de parámetros y un control proporcional integral generalizado. Los autores

desarrollan un controlador GPI combinado con la identificación de parámetros no conocidos en lazo cerrado, el cual presenta una robustez con respecto a las no linealidades del motor del sistema. Por otra parte, en [21] se utiliza la técnica GPI para controlar un convertidor de potencia CD-CD tipo reductor (*Buck*) implementado únicamente mediante capacitores (no utiliza inductores de alta frecuencia), el control de corriente regula el voltaje de salida del convertidor reductor (CCS, *Current Control Scheme*). Este control GPI propuesto es robusto con respecto a variaciones súbitas de carga y solo requiere de la medida del voltaje de salida del convertidor para poder estimar la variable de corriente a través de un reconstructor integral. El desempeño del control GPI es evaluado mediante un simulador. Por otro lado, el trabajo publicado por Franco-González [17] presenta un convertidor multivariable del tipo CD-CD con topología elevadora-elevadora (Boost-Boost), constituido por dos convertidores elevadores (Boost) en cascada en modo de conducción continua, donde cada convertidor alimenta una carga resistiva. Desarrollan un controlador por modos deslizantes, basado en la técnica GPI para la tarea de regulación. El control por retroalimentación utiliza solo el monitoreo de los voltajes de salida en los capacitores del convertidor, así como señales de entrada representadas por la posición del interruptor. Además la robustez del sistema se prueba mediante variaciones súbitas de las resistencias de carga de cada uno de los convertidores. La implementación experimental del controlador fue realizada a través de una computadora y una tarjeta de adquisición de datos utilizada como interfaz. En la investigación realizada y publicada en [22] se presentan resultados a nivel de simulación del desempeño de un controlador GPI para un control de arranque suave para un motor de CD accionado mediante un convertidor de CD-CD tipo reductor (*Buck*). El modulador utilizado para la implementación del controlador fue un sigma-delta (Ver [15]). El esquema de regulación propuesto se basa en la propiedad de los sistemas diferencialmente planos para el control directo de velocidad en el eje del motor ([18]). Como consecuencia, el controlador de arranque suave por retroalimentación GPI no se basa en observadores asintóticos ni cálculos basados en muestreos para la estimación de estados, si no se basa en la reconstrucción mediante integradores de los estados midiendo la entrada y la salida del sistema.

En lo que se refiere a trabajos publicados sobre controladores robustos implementados en FPGA para la regulación del voltaje de salida del convertidor CD-CD tipo reductor (*Buck*), se tiene el trabajo reportado en [23] el cual presenta la comparativa entre dos controladores promedio del tipo PID. El primero corresponde a un PID clásico, mientras que el segundo corresponde a un controlador PID no lineal. Los resultados experimentales muestran que el controlador PID no-lineal presenta una mejor respuesta dinámica en la variable de salida y un menor tiempo de recuperación de la señal de referencia cuando se presenta una perturbación desconocida por el sistema. Por

otro lado, en el trabajo reportado en [24] proponen un diseño modular de un controlador PID por retroalimentación embebido en un arreglo de puertas programable en campo (FPGA, *Field Programmable Gate Array*). Proponen un sistema de aritmética distribuida, el cual es un algoritmo de cálculo de bits en serie que realiza la multiplicación mediante LUT (*Look Up Tables*), para controlar un sistema de temperatura. Los resultados obtenidos muestran una buena estabilidad y rendimiento del controlador en lazo cerrado. En [25] se presenta una aplicación digital de una estrategia para la observación de los voltajes en los capacitores flotados que conforman a los convertidores multicelda apilados. En este trabajo se diseña un observador de estados a través de la técnica por modos deslizantes (*sliding mode*) para observar los voltajes en los capacitores flotados, el observador es digitalizado e implementado en el FPGA. Los resultados muestran una buena estimación de los voltajes en los capacitores. Por otra parte, Naouar *et. al.* [26] muestran los beneficios de usar FPGAs en sistemas de control industrial, específicamente técnicas de control aplicados a máquinas de CA. Implementan un controlador de corriente ON-OFF, un controlador PI clásico, y un controlador de corriente predictivo. Para cada caso, diseñan una arquitectura dedicada en FPGA y se tienen resultados experimentales relevantes en cuanto al desempeño de cada uno de los controladores. Por otro lado, Yanxia Gao *et. al.* [27] presentan el diseño de un control PID clásico y un Modulador digital por ancho de pulso (DPWM) como módulo principal, para un convertidor CD-CD tipo reductor (Buck). Para verificar la eficiencia del DPWM, se implementa el DPWM de 11 bits en un FPGA. Los resultados experimentales muestran la funcionalidad del DPWM a una frecuencia fija de 2 MHz, donde también se presenta una rápida respuesta transitoria del voltaje de salida del convertidor y una robustez ante cambios súbitos en la carga. La implementación la realizan en un FPGA Virtex-II Pro XC2VP30. El artículo de Saggini [28] muestra la implementación de un estimador de eficiencia implementado con la técnica de control digital aplicado a un convertidor reductor (Buck), basado en una estimación indirecta de la razón entre la corriente de entrada y salida del convertidor. Para la implementación experimental del estimador se utiliza solamente un sensor de corriente para el monitoreo de la corriente que circula en el capacitor de salida del convertidor. Los resultados de la implementación muestran que el error máximo de estimación está alrededor del 5%. Con base al estudio sobre el control del convertidor CD-CD tipo reductor hecho previamente se formulan las siguientes preguntas de investigación.

- ¿Es factible la implementación de un convertidor GPI para un convertidor reductor en un dispositivo lógico programable?
- ¿Qué tipo de aritmética será necesaria implementar en el dispositivo lógico programable para poder implementar el control GPI?

- ¿Cuál es el dispositivo lógico programable que se ajuste mejor a la necesidad de implementar la ley de control para regular el voltaje de salida del convertidor reductor?
- ¿Se puede implementar la ley de control y el modulador por ancho de pulsos en la misma plataforma sin necesidad de componentes externos?
- ¿El control GPI implementado en el dispositivo lógico programable seguirá siendo robusto ante perturbaciones desconocidas por el sistema?

### 1.3. Planteamiento del Problema

Para obtener un voltaje deseado a la salida del convertidor reductor, que sea menor al voltaje de su alimentación principal, es necesario diseñar un sistema de control que sea capaz de regular el voltaje a su salida en un tiempo muy corto sin causar sobretiro más allá del 2% en la respuesta transitoria. Además, otra característica importante a tomar en cuenta, es que al conectar cargas desconocidas a la salida del convertidor, éste pueda en primera instancia mantener el voltaje deseado y en segundo disminuir el efecto de la perturbación en el voltaje causado por la conexión de cargas desconocidas a su salida.

Muchas aplicaciones pueden resolverse mediante el uso de una computadora, sin embargo, ejecutar aplicaciones de medición o control en una computadora estándar con un sistema operativo de propósito general puede hacer que el sistema falle. Esto debido a que en cualquier instante el sistema operativo puede causar un retardo en la ejecución debido a múltiples causas: ejecutar un antivirus, actualizar gráficos, realizar tareas en segundo plano, entre otras; lo que resulta una desventaja de la computadora por el tiempo de procesamiento [29]. Así, las aplicaciones que se ejecutan en una computadora son programadas y se necesita cargar en memoria los datos y código a ejecutar en un procesador, con lo que se podría decir que la computadora permite ejecutar porciones de código a la vez, en contraste con el FPGA donde el sistema implementado es la aplicación final y además permite el paralelismo.

Para dar solución al problema planteado, es necesario utilizar una plataforma que permita implementar la ley de control GPI junto con su modulador para activar y desactivar al semiconductor de potencia del convertidor y así regular su voltaje de salida. Por tanto, se propone la utilización de un FPGA para implementar el control GPI y un modulador por ancho de pulsos modelados mediante lenguaje VHDL. Para poder representar el amplio rango de valores de los resultados de las operaciones del controlador GPI, se hace necesaria la utilización de aritmética de punto flotante. Por ello se propone implementar los componentes del controlador como unidades de punto flotante de precisión simple. Así mismo generar mediante el lenguaje des-

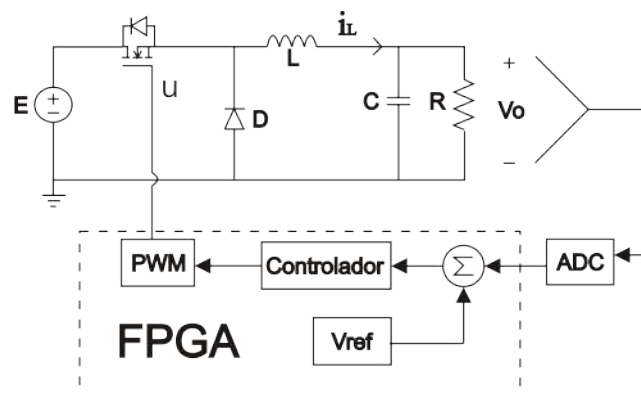


Figura 1.3: Diagrama a bloques del control GPI en un FPGA.

criptor de hardware VHDL, la señal portadora del modulador, además de la unidad de comparación que son componentes principales del modulador de ancho de pulsos. La adquisición de las muestras del voltaje a regular se obtendrá mediante un convertidor analógico digital (ADC, *Analogic Digital Converter*), el cual es el único hardware adicional al FPGA para controlar al convertidor reductor. En la figura 1.3 se muestra el diagrama a bloques del diseño del control GPI en un FPGA para la regulación del voltaje de salida del convertidor CD-CD tipo reductor.

## 1.4. Justificación

Desde que una computadora tiene que ejecutar múltiples tareas, no permite la implementación de una sola aplicación final. El tiempo de procesamiento es mayor que otras tecnologías donde se implementa únicamente la aplicación de control, es importante abrir el campo y buscar otras alternativas que den solución a problemas específicos como los que se presentan en el área de control automático aplicado a electrónica de potencia. Diseñar con base en un FPGA presenta las siguientes ventajas: rápido tiempo de desarrollo, facilidad en reconfiguración del sistema realizado, ejecución concurrente que repercute en bajos tiempos de procesamiento, uso de lenguajes descriptores de hardware estándares, etc.; esto permite implementar aplicaciones autónomas eficientes. Aunado a esto, en la literatura no se encuentran trabajos relacionados del control GPI implementado en un FPGA para el control del convertidor CD-CD tipo reductor. Con lo cual da pauta a desarrollar controladores lineales y no lineales dentro de un FPGA para el control de convertidores CD-CD y así explotar sus características de rendimiento en un margen más amplio.

## 1.5. Hipótesis

En base a las preguntas de investigación formuladas en el apartado anterior, se plantean las siguientes hipótesis:

- Los dispositivos lógicos programables son componentes que contienen conjuntos de elementos lógicos, los cuales pueden configurarse en cualquier función lógica que el usuario decida. Así con la lógica programable o lógica reconfigurable se pueden implementar simples funciones hasta sistemas complejos en un solo circuito integrado de este tipo de lógica. La programación de estos dispositivos generalmente se realizan a través de una computadora, sin embargo, para su funcionamiento no es necesario el uso de la misma. Por tanto, una implementación de un algoritmo GPI puede perfectamente realizarse dentro de un dispositivo lógico programable.
- Como el rango de valores que manejan las variables del controlador GPI es muy amplio, los cuales suelen tener valores muy pequeños (alrededor de  $10^{-6}$ ) y valores muy grandes (alrededor de  $10^{12}$ ). Esto implica que al emplear una aritmética en punto fijo implicaría el uso de muchos recursos dentro del dispositivo, por tanto sería más adecuado el uso de una aritmética de punto flotante de precisión simple.
- Los dispositivos lógicos programables como el FPGA contienen bloques lógicos cuya interconexión y funcionalidad se pueden programar. Aunado a esto, los FPGAs permiten la ejecución en paralelo de las unidades modeladas, con lo que es posible utilizar un FPGA para la implementación del controlador GPI.
- Como los FPGAs permiten la ejecución en paralelo de las unidades modeladas, es factible diseñar el control GPI y el modulador por ancho de pulsos en un mismo FPGA, sin necesidad de generar la señal portadora y el comparador de forma externa.
- La técnica de control GPI hace uso de reconstructores integrales para la estimación de las variables midiendo la entrada y la salida, lo cual evita medir más de una variable de estado. Los términos de integración del error de que está compuesto la ley de control GPI dentro del FPGA para la regulación del voltaje de salida del convertidor CD-CD tipo reductor, mantendrá la característica de robustez en la variable del voltaje de salida del convertidor.

## 1.6. Objetivos

### 1.6.1. Objetivo General

Diseñar un Controlador Proporcional Integral Generalizado (GPI) e implementarlo en un FPGA para la regulación del voltaje de salida del convertidor CD-CD tipo reductor (*Buck*).

### 1.6.2. Objetivos Específicos

- Implementar un control GPI en un Arreglo de Compuertas Programables en Campo, específicamente en una tarjeta NEXYS2 Spartan3E.
- Desarrollar unidades de aritmética necesarias en punto flotante, basada en el estándar IEEE 754, para una precisión simple (números de 32 bits).
- Implementar el control GPI, y el modulador por ancho de pulsos dentro de la misma tarjeta FPGA sin necesidad de componentes externos interfazados al FPGA.
- El control GPI implementado debe heredar la característica de robustez aún cuando este sea implementado dentro del FPGA, sometiendo al sistema a cambios bruscos de carga estática y dinámica.

### 1.6.3. Objetivos Secundarios

- Realizar un análisis matemático de un convertidor CD-CD tipo reductor en estado transitorio y en estado estable en modo conducción continuo (MCC), para obtener el modelo matemático del convertidor.
- Realizar una simulación del convertidor CD-CD tipo reductor para observar el comportamiento del circuito.
- Diseñar un control GPI basado en el modelo promedio del convertidor CD-CD tipo reductor con el fin de obtener las ecuaciones matemáticas y su posterior modelado en lenguaje VHDL.
- Diseñar una unidad de multiplicación de números en punto flotante, y modelarla mediante VHDL.
- Diseñar una unidad de suma de números en punto flotante, en VHDL.
- Realizar un módulo de integración en VHDL, en punto flotante.
- Simulación de la ley de control mediante herramientas como Matlab®, Psim y ModelSim®.

#### 1.6.4. Metas

- Implementación un sistema convertidor reductor.
- Implementación de un controlador GPI en un FPGA utilizando el lenguaje descriptor de Hardware VHDL para modelar el diseño del control.
- Comprobar el funcionamiento de la plataforma experimental en lazo abierto y cerrado.

### 1.7. Contenido del documento

El presente documento está estructurado de la siguiente forma:

El capítulo 1 presenta una breve introducción sobre convertidores CD-CD, el estado del arte que hay detrás de este trabajo, el planteamiento del problema, justificación, hipótesis y objetivos del presente trabajo de investigación

En el capítulo 2 se desarrolla el modelo matemático del convertidor reductor. Se obtienen las ecuaciones necesarias para diseñar un convertidor CD-CD tipo reductor, y el modulador por ancho de pulsos.

El capítulo 3 muestra la teoría relacionada con la planitud diferencial. Se obtiene la salida plana del sistema, y se hace un análisis del controlador GPI.

El capítulo 4 muestra los métodos utilizados para la implementación del controlador GPI y el PWM. Se describen las características del punto flotante y se muestra el lenguaje descriptor de hardware utilizado en la descripción.

En el capítulo 5 se presentan las simulaciones del sistema final, y los resultados obtenidos de la implementación del controlador GPI.

El capítulo 6 muestra las conclusiones y trabajos futuros obtenidos del presente trabajo de investigación.

Por último se muestran los apéndices y la bibliografía utilizada durante el transcurso de este proyecto.



## Capítulo 2

# Modelado Matemático

**RESUMEN:** En este capítulo se obtiene en forma dinámica el modelo promedio que describe el funcionamiento del convertidor eléctrico del tipo CD–CD de topología reductora, también se hace un análisis en estado estable para obtener las ecuaciones características del circuito en el modo de conducción continuo (MCC), posteriormente se diseña un convertidor en el modo de conducción continuo a una potencia de salida de 1.44W, frecuencia de conmutación de 48Khz y un rizo de voltaje de salida del 0.5 %, finalmente se simula el circuito en el programa PSIM.

### 2.1. Convertidores CD–CD

#### 2.1.1. Introducción

Una infinidad de aplicaciones industriales se hacen a través de los convertidores de potencia del tipo CD–CD, como por ejemplo en aplicaciones de fuentes de alimentación de potencia ininterrumpibles, en impulsores de motores CD para tracción eléctrica en los trolebuses, en sistemas de iluminación que usan balastos electrónicos, etc (véase [1, 5, 30]). Por otra parte, la evolución de los semiconductores de potencia (interruptores) han hecho posible que los ingenieros en electrónica de potencia diseñen fuentes de alimentación con mayor eficiencia comparadas con la fuentes de alimentación que utilizan reguladores lineales [5]. Desde que los transistores han sido usados como dispositivos de conmutación, las fuentes de alimentación son conocidas como transformadores de conversión de potencia de conmutación. En los últimos años, los convertidores de conmutación rápida han venido utilizándose debido a sus avances recientes en la tecnología de semiconductores de potencia [5]. Hoy en día los dispositivos de conmutación rápida están disponibles en el mercado con altas velocidades de conmutación y altas capacidades en el

manejo de potencia. Ahora es factible diseñar fuentes de alimentación de potencia por arriba del 90 % de eficiencia a un bajo costo y tamaño pequeño, además de un peso muy ligero. Por otro lado, el modelado matemático en forma promedio del convertidor eléctrico de topología reductora ha servido para entender mejor su aplicación dentro del área de ingeniería de control, ya que mediante el modelo promedio se ha podido analizar su estabilidad en lazo abierto y en lazo cerrado; esto último cuando se controla mediante un compensador (véase [31]). El uso del modelo promedio del convertidor reductor acoplado al modelo de un actuador electromecánico (motor de CD) ha servido para tener una mayor eficiencia en la transformación de potencia eléctrica a potencia mecánica, cuando a través del modelo se diseña un compensador de arranque suave de velocidad para el actuador electromecánico (véase [22]). En muchos trabajos de control de convertidores eléctricos del tipo CD-CD, se ha venido usando la técnica de modulación de ancho de pulso PWM para controlar el interruptor de potencia que hace funcionar a los convertidores cd-cd tanto en lazo abierto como en lazo cerrado [30, 31].

### 2.1.2. Convertidor reductor “*Buck*”

Recibe el nombre de convertidor reductor debido a que el voltaje de salida es menor que el voltaje de entrada, conservando la misma polaridad que el voltaje de entrada. La figura 2.1 muestra el circuito de un convertidor reductor, el cual consiste de una fuente de voltaje CD, un dispositivo de conmutación (interruptor) y un filtro pasabajos formado por una bobina y un capacitor, que alimentan una carga.

El dispositivo de conmutación es el encargado de conectar y desconectar la fuente de alimentación al inductor. En el tiempo en que el interruptor está encendido, se encuentra en estado ON, la corriente fluye a través del inductor alimentando la carga y el capacitor comienza a cargarse (Ver figura 2.2). Durante el intervalo que el interruptor está apagado, está en OFF, la corriente almacenada en el inductor empieza a fluir a través del diodo y alimenta la carga. Este tipo de trabajo recibe el nombre de modo de conducción continuo (MCC), cuando la corriente en la bobina es mayor a cero. Sin embargo, existe además el modo de conducción discontinuo (MCD), el cual ocurre cuando existe un cambio abrupto en la corriente del inductor, la cual es demasiado pequeña.

## 2.2. Modelo promedio del convertidor reductor

“*Un modelo matemático de un sistema dinámico se define como un conjunto de ecuaciones que representan la dinámica del sistema con precisión o, al menos, bastante bien*”, [31]. Por tanto, el modelo promedio del convertidor reductor permitirá entender mejor el funcionamiento en el modo de

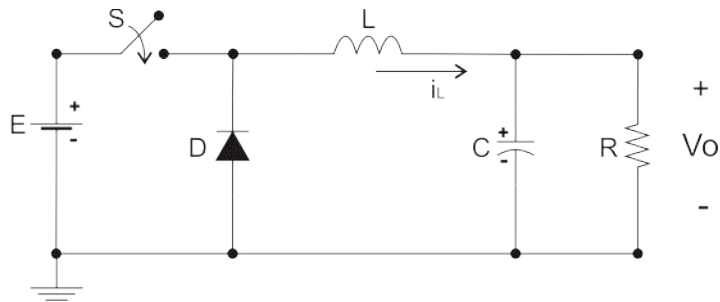


Figura 2.1: Circuito eléctrico del convertidor reductor.

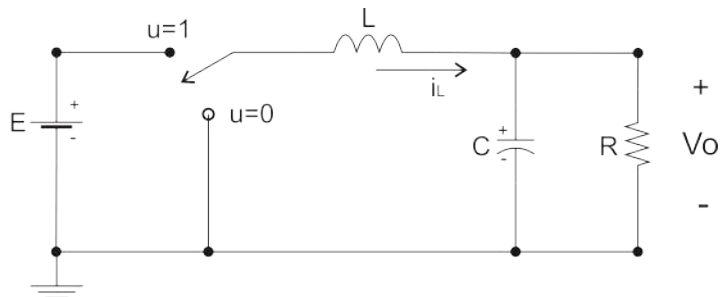


Figura 2.2: Estados del circuito eléctrico del convertidor reductor.

conducción continuo (MCC).

### 2.2.1. Modelo promedio del convertidor reductor

Teniendo en cuenta el circuito eléctrico del convertidor reductor mostrado en la figura 2.1 se procede a analizar el circuito en dos fases, la primera con el interruptor encendido y la segunda con el interruptor apagado. Tomando en cuenta que la variable de control  $u$ , es la que se encarga de activar (estado ON) y desactivar (estado OFF) al dispositivo de conmutación, se muestra en la figura 2.2 los posibles estados del circuito [21].

La figura 2.3 muestra el estado *ON* del interruptor, en este estado, el interruptor permanece encendido, haciendo que la energía de la alimentación se transfiera al resto del circuito.

Al aplicar la ley de voltajes de Kirchhoff (LVK) sobre el circuito de la figura 2.3 se obtiene la siguiente ecuación diferencial.

$$L \frac{di_L}{dt} = -V_o + E \quad (2.1)$$

Por otro lado, aplicando la ley de corrientes de Kirchhoff (LCK) sobre el

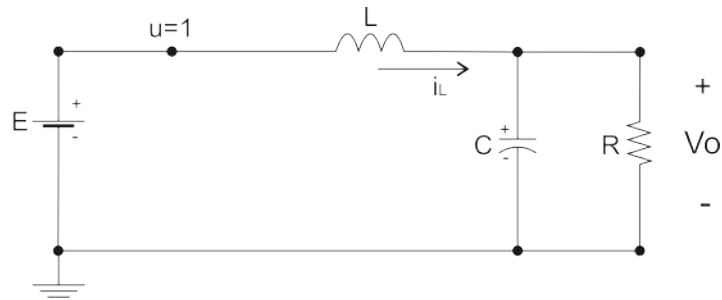


Figura 2.3: Circuito eléctrico equivalente para el estado  $u = 1$ , estado *ON*.

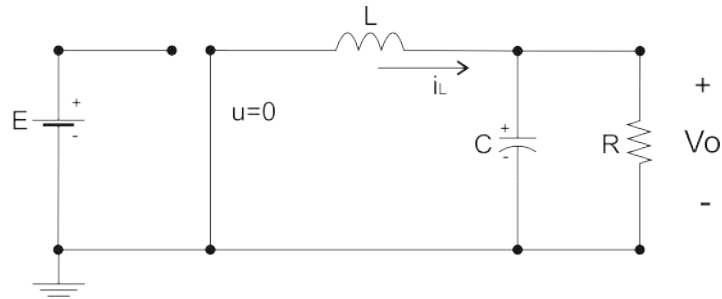


Figura 2.4: Circuito eléctrico equivalente para el estado  $u = 0$ , estado *OFF*.

circuito de la figura 2.3 se obtiene la ecuación diferencial.

$$C \frac{dV_o}{dt} = i_L - \left(\frac{1}{R}\right)V_o \quad (2.2)$$

Ahora, seleccionando el valor de  $u=0$ , se obtiene un circuito equivalente mostrado en la figura 2.4. Al aplicar LVK y LCK sobre el circuito de la figura 2.4 se obtienen las siguientes ecuaciones diferenciales.

$$L \frac{di_L}{dt} = -V_o \quad (2.3)$$

$$C \frac{dV_o}{dt} = i_L - \left(\frac{1}{R}\right)V_o \quad (2.4)$$

Las ecuaciones que unen los dos estados del convertidor y que corresponden con las ecuaciones (2.1) a (2.4), se muestran a continuación.

$$L \frac{di_L}{dt} = -V_o + Eu \quad (2.5)$$

$$C \frac{dV_o}{dt} = i_L - \left(\frac{1}{R}\right)V_o \quad (2.6)$$

Debido a que  $u \in \{0, 1\}$  representa un modelo en estado discreto, si es considerado un valor promedio de  $u$  en el tiempo en cada periodo de conmutación del interruptor, se obtiene  $u_{av} \in [0, 1]$  por tanto se obtienen las ecuaciones diferenciales del convertidor en su modelo promedio.

$$L \frac{di_L}{dt} = -V_o + E u_{av} \quad (2.7)$$

$$C \frac{dV_o}{dt} = i_L - \left(\frac{1}{R}\right)V_o \quad (2.8)$$

Desde que la variable a regular es el voltaje, el espacio de estados del convertidor *Buck* está representado de la siguiente forma:

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dV_o}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ V_o \end{bmatrix} + \begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix} u_{av} \quad (2.9)$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i_L \\ V_o \end{bmatrix} \quad (2.10)$$

### 2.2.2. Análisis en estado estable del convertidor reductor

El análisis que se hará a continuación se realizará en estado estable del convertidor, es decir, los valores de voltajes y corrientes tienden a tomar valores permanentes. La salida de voltaje en el convertidor reductor se denota por la función  $V_o(t)$ , compuesta por un nivel de CD deseado y por componentes de CA no deseados.

El rizo que se tiene a la salida del convertidor debido a la conmutación es muy pequeño (menor al 1%) comparado con el nivel de voltaje deseado de CD. Por ello, se asume que el voltaje de rizo de salida se desprecia por ser muy pequeño y por tanto, se tiene que  $V_o(t) = V_o$ . Es decir, el voltaje de rizo de salida se produce por la constante de tiempo de salida del filtro pasabajos compuesto por el capacitor con la resistencia de carga,  $RC$ , el cual es muy grande. Por tanto, para el análisis se tienen las siguientes suposiciones.

1. Debido a que se asume que se está trabajando con componentes ideales en el modelo matemático, es decir, no existen pérdidas en los componentes debido a la conmutación, la potencia de promedio de entrada  $P_{in}$ , y la potencia promedio de la salida  $P_o$ , son iguales, es decir:

$$P_{in} = P_o$$

2. Al estar los semiconductores del convertidor en conmutación a una frecuencia fija, se asume que el circuito eléctrico operando en estado

estable, la corriente en el inductor y el voltaje en el capacitor son periódicos en un ciclo de conmutación completo, por tanto se tiene:

$$\begin{aligned}i_L(t) &= i_L(t_0 + T) \\v_c(t) &= v_c(t_0 + T)\end{aligned}$$

3. Se asume que los capacitores e inductores son ideales, por tanto la corriente promedio en el capacitor y el voltaje promedio en el inductor son cero:

$$i_c = \frac{1}{T} \int_{t_0}^{T+t_0} i_c(t) dt = 0 \quad (2.11)$$

$$v_L(t) = \frac{1}{T} \int_{t_0}^{T+t_0} v_L(t) dt = 0 \quad (2.12)$$

4. Se define a  $D$  como el ciclo útil de trabajo en un periodo  $T$ , y el cual está definido como:

$$D = \frac{T_{encendido}}{T} = \frac{t_1 - t_0}{T}$$

El dispositivo de conmutación permanece en un estado de encendido (*ON*) durante un periodo de tiempo  $DT$ , y se encuentra en estado apagado (*OFF*) durante el tiempo restante del periodo, este tiempo es  $(1 - DT)T$ . Dependiendo del estado en que se encuentre el dispositivo de conmutación, ya sea encendido o apagado, la corriente en el inductor aumentará con  $E$ , y disminuirá a través del diodo. Por tanto, existe dos formas de operación del convertidor reductor. La primera es considerando el estado encendido ( $u = 1$ ), tal como se muestra en la figura 2.3.

Al integrar la ecuación (2.1) de  $t = 0$  a  $t$  y con la condición inicial de  $I_L(0)$ , se obtiene que:

$$i_L(t) = \frac{1}{L}(E - V_o)t + I_L(0) \quad (2.13)$$

La ecuación (2.13) indica un crecimiento en la corriente del inductor con pendiente positiva  $(E - V_o)/L$ , donde  $I_L(0)$  es la condición inicial en  $t = 0$  (véase figura 2.5(a)). Ahora bien, al llegar al tiempo  $t = DT$ , el dispositivo de conmutación pasa a un estado de apagado, con lo que se obtiene el circuito equivalente mostrado en la figura 2.4. Integrando la ecuación (2.3) de  $t = DT$  a  $t$  con condición inicial de  $I_L(DT)$ , se tiene que:

$$i_L(t) = -\frac{V_o}{L}(t - DT) + I_L(DT) \quad (2.14)$$

La ecuación anterior indica que la corriente en el inductor disminuirá con pendiente negativa  $-V_o/L$  a partir de  $t = DT$ . La gráfica de la figura 2.5(a)

muestra la forma de onda de la corriente en el inductor durante un periodo cuando el circuito se encuentra en estado estable, por lo tanto, el comportamiento de la corriente  $I_L$  en estado estable está dado por:

$$I_L(DT) = \frac{1}{L}(E - V_o)DT + I_L(0) \quad (2.15)$$

$$I_L(0) = -\frac{V_o}{L}(1 - D)T + I_L(DT) \quad (2.16)$$

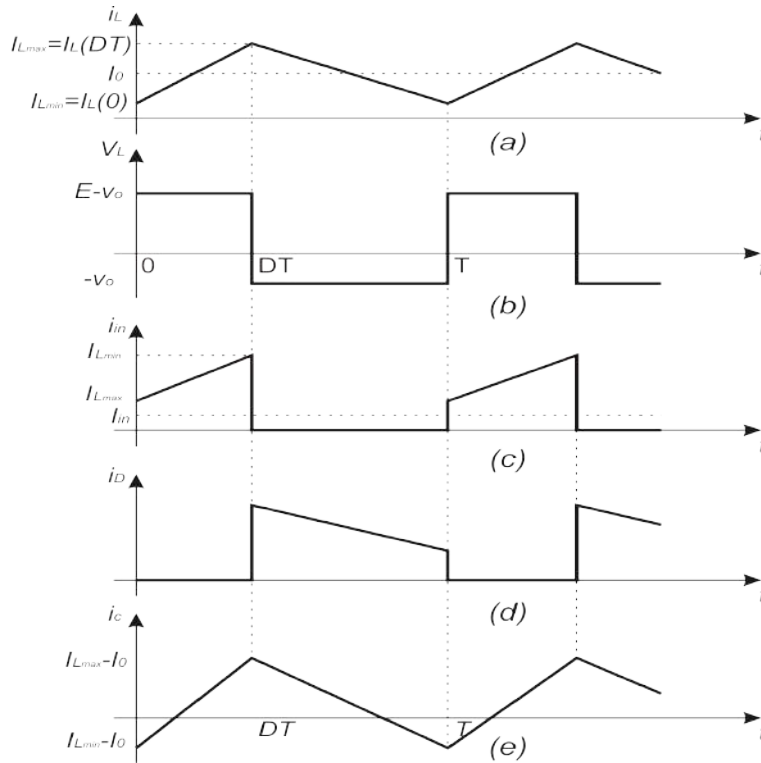


Figura 2.5: Forma de señales en estado estable del convertidor reductor. (a) Corriente en el inductor. (b) Voltaje en el inductor. (c) Corriente en el diodo. (d) Corriente en el capacitor.

Las formas de señal de corriente y voltaje en el inductor se pueden observar en la figura 2.5(a) y 2.5(b) respectivamente. En donde  $I_{L_{max}} = I_L(DT)$  e  $I_{L_{min}} = I_L(0)$  son los valores de la corriente del inductor en los instantes de tiempo en que el interruptor del convertidor conmuta entre el estado encendido y apagado para un periodo fijo.

Partiendo de las ecuaciones (2.15-2.16) se deriva la siguiente relación del voltaje de salida del convertidor reductor, y la cual está dada por:

$$\frac{V_o}{E} = D \quad (2.17)$$

Donde  $D$  pertenece al intervalo cerrado  $[0, 1]$ .

Por otro lado, el valor promedio de la corriente de entrada en el convertidor reductor se calcula de la siguiente manera (Ver figura 2.5):

$$I_{in} = \frac{1}{T} \int_0^T i_{in}(t) dt \quad (2.18)$$

Partiendo de que la corriente de entrada  $i_{in}(t) = i_L$ , al momento de que el interruptor se encuentra encendido, ( $u = 1$ ), se sustituye la ecuación (2.15) en (2.18) y se evalúa la integral en el tiempo de integración en que se encuentra el interruptor encendido, es decir, en el intervalo de  $t = 0$  al tiempo  $t = DT$ , con lo que se obtiene:

$$I_{in} = \frac{1}{2L}(E - V_o)D^2T + I_L(0)D \quad (2.19)$$

Sustituyendo la ecuación (2.16) en (2.19) se obtiene que:

$$I_{in} = \frac{1}{2}(I_L(DT) + I_L(0))D = \frac{1}{2}(I_{Lmax} + I_{Lmin})D \quad (2.20)$$

Mientras que la corriente de salida promedio del convertidor está dada como:

$$I_o = I_L = \frac{I_{Lmax} + I_{Lmin}}{2} = \frac{V_o}{R} \quad (2.21)$$

Las corrientes máxima y mínima del convertidor en estado estable se obtienen a partir de las ecuaciones (2.15-2.16) y (2.21), las cuales son:

$$I_{Lmax} = DE\left(\frac{1}{R} + \frac{(1-D)T}{2L}\right) \quad (2.22)$$

$$I_{Lmin} = DE\left(\frac{1}{R} - \frac{(1-D)T}{2L}\right) \quad (2.23)$$

Mientras, el valor de la corriente de entrada y salida promedio están dadas como:

$$I_{in} = \frac{D^2E}{R} \quad (2.24)$$

$$I_o = \frac{DE}{R} \quad (2.25)$$

Partiendo de (2.24) y (2.25), se calcula la ganancia de corriente del convertidor reductor, resultando:

$$\frac{I_o}{I_{in}} = \frac{1}{D} \quad (2.26)$$



Por tanto, el rendimiento de la potencia promedio de la entrada y la salida está dado por:

$$\begin{aligned} I_{in}E &= I_oV_o \\ \frac{I_{in}}{I_o} &= \frac{V_o}{E} = D \end{aligned} \quad (2.27)$$

Analizando las ecuaciones (2.17) y (2.26) se observa que las relaciones de entrada y salida entre la corriente y el voltaje del convertidor reductor, equivalen a un transformador de CD a una razón de reducción  $D$ , como se muestra en la figura 2.6. La señal de control  $u_{av}$  definida en 2.2.1 que toma valores

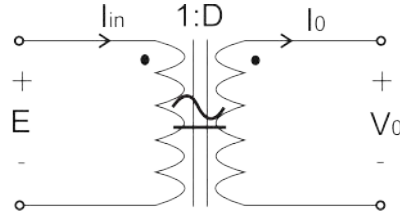


Figura 2.6: Circuito equivalente en la transformación de CD de un convertidor reductor.

en el intervalo cerrado  $[0, 1]$ , corresponde a la razón de reducción  $D$ , dada en (2.27).

### 2.2.2.1. Valor crítico de la inductancia del convertidor reductor

El valor crítico del inductor  $L$ , en el convertidor reductor determina que el convertidor trabaje o no en modo de conducción continuo (MCC), es decir que la corriente en el inductor no caiga bruscamente a cero. Para esto, el valor mínimo necesario para hacer funcionar al convertidor en MCC se establece a partir de igualar la ecuación (2.23) con cero, y se resuelve la ecuación para encontrar el valor crítico del inductor  $L$ .

$$I_{L_{min}} = DE\left(\frac{1}{R} - \frac{(1-D)T}{2L}\right) = 0 \quad (2.28)$$

Despejando el valor del inductor  $L$  de (2.28) se obtiene el valor crítico de la inductancia el cual está dado por:

$$L_{critico} = \frac{1-D}{2}TR \quad (2.29)$$

Donde  $L_{critico}$  es el valor mínimo que hace funcionar al convertidor reductor en MCC para valores de  $D$ ,  $T$  y  $R$  dadas para un diseño particular.

### 2.2.2.2. Voltaje de rizo de salida del convertidor reductor

En el análisis previo, se asumió que el capacitor era demasiado grande, por tanto  $V_o(t) = V_o$ . Asumiendo que la corriente de CA que sale del inductor entra al capacitor y se queda ahí, solo un voltaje de CD sin rizo llegará a la carga. Por tanto el valor de capacitor es muy importante a la hora de diseñar un convertidor reductor ya que de este dependerá la cantidad de rizo existente a la salida del convertidor. Es por ello que se selecciona generalmente un valor de capacitancia elevado para que limite el rizo de CA en la carga. Teóricamente se tiene que si el valor de  $C \rightarrow \infty$ , elimina completamente el rizo de CA y se tiene a la salida una señal sin componentes de CA. Sin embargo, por consideraciones prácticas se considera un valor de capacitancia teniendo un valor de rizo que se añade a la salida del convertidor. Para calcular el porcentaje de rizo del voltaje de salida, se obtiene una expresión de la corriente en el capacitor, la cual se calcula a través de la LCK de la siguiente manera:

$$i_c(t) = i_L(t) - I_0 \quad (2.30)$$

Evaluando (2.30) en  $t = 0$  se tiene:

$$\begin{aligned} I_c(0) &= I_L(0) - I_0 \\ &= -\frac{1}{2}(I_{L_{max}} - I_{L_{min}}) \end{aligned} \quad (2.31)$$

mientras que en  $t = DT$  se tiene que:

$$\begin{aligned} I_c(DT) &= I_L(DT) - I_0 \\ &= \frac{1}{2}(I_{L_{max}} + I_{L_{min}}) \end{aligned} \quad (2.32)$$

A partir de la figura 2.7 que representa el voltaje de rizo en el capacitor, es posible calcular la corriente instantánea que circula por el capacitor, esta se expresa en términos del error de corriente  $\Delta I = I_{L_{max}} - I_{L_{min}}$  mediante la ecuación dada como:

$$\begin{aligned} i_c(t) &= \frac{(I_{L_{max}} - I_{L_{min}})}{DT}t - \frac{(I_{L_{max}} - I_{L_{min}})}{2} \\ &= \frac{\Delta I}{DT}t - \frac{\Delta I}{2} \quad 0 \leq t \leq DT \end{aligned} \quad (2.33)$$

mientras que la corriente durante el resto del período está dado como:

$$\begin{aligned} i_c(t) &= \frac{(I_{L_{max}} - I_{L_{min}})}{DT}(t - DT) - \frac{(I_{L_{max}} - I_{L_{min}})}{2} \\ &= -\frac{\Delta I}{DT}(t - DT) + \frac{\Delta I}{2} \quad DT \leq t \leq T \end{aligned} \quad (2.34)$$

Donde

$$\Delta I = \frac{E(1-D)TD}{L} \quad (2.35)$$

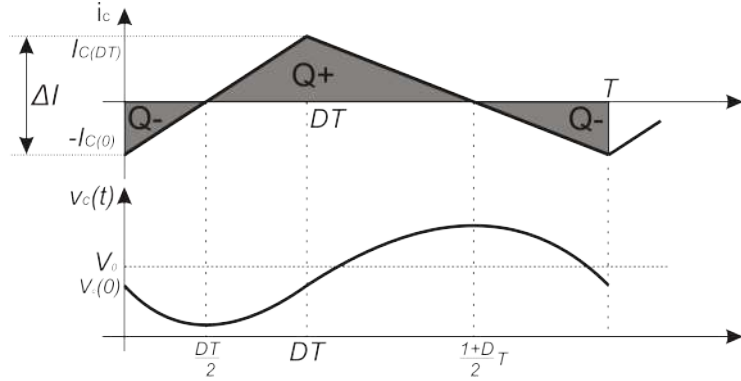


Figura 2.7: Formas de señal de corriente y voltaje en el capacitor.

A partir de integrar en el intervalo  $0 \leq t \leq DT$  la relación entre el voltaje y corriente en el capacitor,  $i_c = C(dv_c/dt)$ , se tiene la expresión que denota el voltaje instantáneo en el capacitor:

$$v_{c1}(t) = \frac{1}{C} \int_0^{DT} i_c(t) dt + V_c(0) \quad (2.36)$$

Sustituyendo la ecuación (2.33) en (2.36) se tiene el valor del capacitor en el intervalo  $0 \leq t \leq DT$ ,

$$v_{c1}(t) = \frac{1}{C} \int_0^{DT} \left[ \frac{\Delta I}{DT} t - \frac{\Delta I}{2} \right] + V_c(0) \quad 0 \leq t \leq DT \quad (2.37)$$

Resolviendo la integral se tiene que:

$$v_{c1}(t) = \frac{1}{C} \frac{\Delta I}{DT} t^2 - \frac{\Delta I}{2} t + V_c(0) \quad 0 \leq t \leq DT \quad (2.38)$$

De forma similar que en el primer intervalo, se calcula el voltaje en el capacitor para el resto del intervalo, es decir,  $DT \leq t \leq T$ , y está dado por:

$$v_{c2}(t) = \frac{\Delta I(t-DT)^2}{2C(1-D)T} + \frac{\Delta I(t-DT)}{2C} + V_c(DT) \quad DT \leq t \leq T \quad (2.39)$$

A partir de la gráfica de la figura 2.7 se evalúa en estado estable el voltaje en el capacitor de la siguiente forma:

$$v_{c_1}(DT) = v_{c_2}(T) \quad (2.40)$$

El voltaje promedio de salida del capacitor está dado por:

$$v_o = \frac{1}{T} \left[ \int_0^{DT} v_{c_1}(t) dt + \int_{DT}^T v_{c_2}(t) dt \right] \quad (2.41)$$

Usando los resultados de las ecuaciones (2.38) y (2.39), y sustituyéndolas en (2.41) se obtiene:

$$v_o = \frac{\Delta I}{12C} (1 - 2D)T + v_c(0) \quad (2.42)$$

Sustituyendo la ecuación (2.35) en (2.42) se llega a que:

$$v_c(0) = DE \left[ 1 - \frac{(1-D)(1-2D)}{12CL} T^2 \right] \quad (2.43)$$

El voltaje de pico del capacitor sucede cuando la corriente en el inductor es cero, por tanto se tiene que el voltaje mínimo en el capacitor ocurre en  $t = DT/2$ , el cual se obtiene a partir de la ecuación (2.38) y es:

$$\begin{aligned} v_{c_{min}} &= \frac{1}{C} \frac{\Delta I}{DT} t^2 - \frac{\Delta I}{2} t + v_c(0) \\ &= -\frac{\Delta I}{8C} DT + v_c(0) \end{aligned} \quad (2.44)$$

mientras que el valor máximo del voltaje en el capacitor ocurre en  $t = (1 + D)T/2$ , el cual es obtenido a partir de la ecuación (2.39), y está dado por:

$$\begin{aligned} v_{c_{max}} &= -\frac{\Delta I}{2C(1-D)T} \left( \frac{(1+D)T}{2} - DT \right)^2 + \frac{\Delta I}{2C} \left( \frac{(1+D)T}{2} - DT \right) + v_c(DT) \\ &= \frac{\Delta I}{8C} (1 - D) + v_c(DT) \end{aligned} \quad (2.45)$$

Utilizando los resultados de (2.43) en las ecuaciones (2.44) y (2.45), además de (2.35), se obtienen los valores finales de los voltajes  $v_{c_{min}}$  y  $v_{c_{max}}$  los cuales están dados en la siguiente forma:

$$\begin{aligned} v_{c_{min}} &= V_o \left[ 1 - \frac{(1-D)(1-2D)}{24LC} T^2 \right] \\ v_{c_{max}} &= V_o \left[ 1 + \frac{(1-D^2)}{24LC} T^2 \right] \end{aligned}$$

Por lo tanto, la variación en el rizo del voltaje del capacitor está dado como:

$$\begin{aligned}\Delta v_c &= v_{c_{max}} - v_{c_{min}} \\ &= \frac{V_o}{8LCf^2}(1 - D)\end{aligned}$$

De esta última expresión podemos obtener la expresión de la razón de rizo de voltaje de salida, y es:

$$\frac{\Delta v_c}{V_o} = \frac{(1-D)}{(8LCf^2)} \quad (2.46)$$

La expresión anterior es conocida como rizo del voltaje de salida. Cuando el valor de la capacitancia y la frecuencia aumenta, el nivel de rizo disminuye considerablemente. Hasta aquí podemos decir que ya tenemos las ecuaciones necesarias para diseñar un convertidor reductor dadas ciertas características deseadas, como el porcentaje de rizo  $\Delta v_c/V_o$ , voltaje de alimentación  $E$ , potencia de salida  $P_o$ , frecuencia de conmutación  $f$  y valor del ciclo de trabajo  $D$  lo que equivaldría a la razón entre el voltaje deseado y el voltaje de alimentación.

### 2.3. Modulador por ancho de pulsos

En el control de convertidores CD-CD el voltaje promedio de salida es controlado por medio de un nivel deseado a través del dispositivo de conmutación. Uno de los métodos empleados para controlar el voltaje promedio de un convertidor CD-CD es el controlar el tiempo de encendido  $t_{ON}$  y tiempo de apagado  $t_{OFF}$  del dispositivo de conmutación, considerando una frecuencia fija de conmutación  $f_s$  (El periodo de conmutación es constante y es  $T_s = t_{ON} + t_{OFF}$ ). Por lo regular es ajustado el tiempo de encendido  $t_{ON}$  con el fin de obtener el voltaje de salida deseado. Este método es denominado Modulador por Ancho de pulsos, más conocido por sus siglas en ingles PWM (*Pulse Width Modulation*), en el cual se define a  $D$  como el tiempo de encendido  $t_{ON}$  en el periodo de conmutación [1].

En la conmutación PWM a una frecuencia constante, existe una señal de control  $V_c$  que controla los estados ( $ON$  y  $OFF$ ) del dispositivo de conmutación, es generada a partir de la comparación entre un nivel de voltaje  $V_d$  y una señal periódica de amplitud constante  $V_p$ . La señal de control generalmente es obtenida a partir de la diferencia entre la señal de salida  $V_o$  y la señal deseada  $V_e$  (error del sistema), tal como se muestra en la figura 2.8a. La frecuencia de la señal periódica es la que establece la frecuencia de conmutación la cual es constante y generalmente está en el orden de unos cuantos kilohertz hasta unos cientos de kilohertz.

Las señales obtenidas en el PWM se pueden observar en la figura 2.8b. Cuando la señal del nivel de voltaje  $V_d$  es mayor que la señal periódica  $V_p$ ,  $V_c$  toma un nivel de encendido (ON). En caso contrario, el valor del estado será OFF. Con esto se puede obtener una señal de control con una amplitud y frecuencia fija, pero de ancho variable dependiendo del valor de la comparación entre  $V_d$  y  $V_p$ .

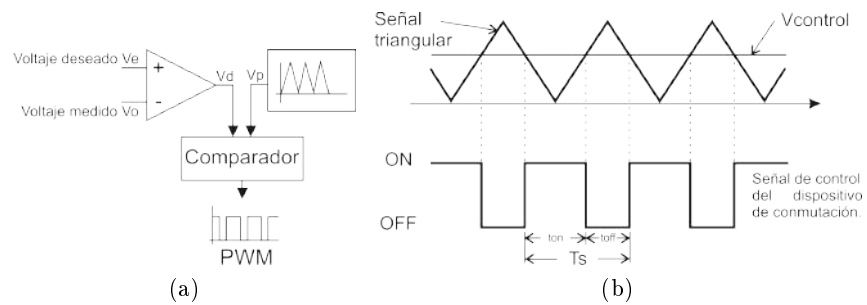


Figura 2.8: Modulador de ancho de pulsos PWM (a) Diagrama a bloques. (b) Señales del PWM

## Capítulo 3

# Planitud Diferencial

**RESUMEN:** En este capítulo se presenta la teoría relacionada con la planitud diferencial. Un sistema se dice que es plano si se puede encontrar un conjunto de variables igual al número de entradas, que son llamadas salidas planas, tal que las entradas y estados del sistema se puedan expresar en términos de éstas y sus derivadas. Se presenta el análisis matemático de la planitud diferencial realizado al sistema del convertidor reductor, así como el análisis del controlador GPI aplicado al convertidor.

### 3.1. Introducción

Actualmente, los sistemas de control son muy comunes en la tecnología moderna. El control por retroalimentación puede encontrarse en sistemas como control de temperatura, regulación de voltaje en fuentes de alimentación, etc. Con el paso del tiempo, nuevas tecnologías van surgiendo y en conjunto también la teoría de control avanza.

Un uso típico en teoría de control moderna en muchos sistemas es invertir la dinámica del sistema para calcular las entradas requeridas para realizar una tarea específica. Esta inversión puede implicar la búsqueda de entradas apropiadas para llevar al sistema de un estado a otro, o bien encontrar las entradas para seguimiento de trayectoria de alguna o de todas las variables de estado del sistema. En general, la solución de un sistema de control no será único, si es que existe, y uno tendrá que sacrificar el desempeño del sistema para la estabilidad y el esfuerzo de operación. A menudo, este sacrificio es descrito como una función de costos equilibrando el desempeño del sistema con estabilidad y desempeño, resultando en un problema de control óptimo.

En la práctica, siempre existe incertidumbre y ruido en el sistema, que debe tomarse en cuenta para tener un desempeño aceptable en el sistema.

Los controladores por retroalimentación permiten que el sistema responda a los errores y cambien las condiciones de operación en tiempo real, y pueden afectar de manera substancial la operatividad mediante la estabilización del sistema y se extienda su capacidad.

El paradigma utilizado en muchas de las técnicas de control, es estudiar la estructura matemática del sistema para obtener soluciones a la dinámica inversa y la regulación por retroalimentación. La estructura más común de estudiar es la estructura lineal, donde se linealiza al sistema y después se utiliza propiedades de sistemas lineales combinadas con funciones de costo para darle solución al problema. Una forma de estudiar y de resolver problemas de dinámica inversa es mediante el uso de la planitud diferencial.

La planitud diferencial fue introducida por Fliess *et.al.* [32], mediante el álgebra diferencial. En el álgebra diferencial, un sistema es visto como un campo diferencial generado por un conjunto de variables (entradas y salidas). El sistema se dice que es plano si se puede encontrar un conjunto de variables igual al número de entradas, que son llamadas salidas planas, tal que las entradas y estados del sistema se puedan expresar en términos de éstas y sus derivadas [33]. Matemáticamente, si el sistema tiene estados  $x \in \mathbb{R}^n$ , y entradas  $u \in \mathbb{R}^m$  entonces el sistema es plano si podemos encontrar salidas  $y \in \mathbb{R}^m$  de la forma:

$$y = h(x, u, \dot{u}, \dots, u^{(r)})$$

tal que

$$\begin{aligned} x &= \varphi(y, \dot{y}, \dots, y^{(q)}) \\ u &= \alpha(y, \dot{y}, \dots, y^{(q)}) \end{aligned}$$

En este capítulo solo abordaremos el uso de la planitud diferencial en sistemas lineales de una entrada una salida. Estos sistemas comúnmente son denominados sistemas SISO (*SISO, Simple Input, Simple Output*), debido a que la plataforma experimental presenta esta propiedad, y facilita el diseño del controlador GPI presentado más adelante.

### 3.2. Planitud en sistemas SISO lineales invariantes con el tiempo

Si en un sistema SISO existe una variable que sea expresada como una combinación lineal de la entrada, la salida y un número finito de sus derivadas, esta es llamada endógena. Una variable endógena se dice que es una función diferencial de las variables de entrada y salida. Un sistema SISO es plano (o diferencialmente plano), si existe una variable endógena, llamada



salida plana, tal que la entrada  $u$  y la salida  $y$  puedan ser expresadas como una combinación lineal de la salida plana y un número finito de sus derivadas. En forma matemática se dice que dado el sistema

$$\begin{aligned}\dot{x} &= f(x, u) \\ z &= h(x)\end{aligned}\tag{3.1}$$

donde  $x \in \mathbb{R}^n$  representa a los estados del sistema,  $u \in \mathbb{R}$  designa a la entrada y  $z \in \mathbb{R}$  es la salida del sistema. Se dice que el sistema (3.1) es plano si existe una salida

$$y = \theta(x, \dot{x}, \dots, x^{(n)})\tag{3.2}$$

tal que

$$\begin{aligned}x &= \varphi(y, \dot{y}, \dots, y^{(k)}) \\ u &= \alpha(y, \dot{y}, \dots, y^{(k+1)}) \\ z &= \psi(y, \dot{y}, \dots, y^{(k)})\end{aligned}\tag{3.3}$$

Al conjunto de ecuaciones (3.3) se denomina parametrización diferencial del sistema en función de la salida plana  $y$ .

### Parametrización diferencial del convertidor reductor

En la sección 2.2.1 se obtuvo el modelo promedio del convertidor reductor, de (2.9), se tienen las ecuaciones que describen el comportamiento dinámico del convertidor en el dominio del tiempo:

$$\begin{aligned}\frac{di_L}{dt} &= -\frac{1}{L}V_o(t) + \frac{E}{L}u_{av}(t) \\ \frac{dV_o}{dt} &= \frac{1}{C}i_L(t) - \frac{1}{RC}V_o(t)\end{aligned}\tag{3.4}$$

Utilizando la transformada de Laplace y considerando condiciones iniciales igual con cero sobre el sistema (3.4), se obtiene la siguiente representación en el dominio de la frecuencia.

$$sI_L(s) = -\frac{1}{L}V_o(s) + \frac{E}{L}U_{av}(s)\tag{3.5}$$

$$sV_o(s) = \frac{1}{C}I_L(s) - \frac{1}{RC}V_o(s)\tag{3.6}$$

Al despejar  $I_L(s)$  de (3.6) y sustituirla en (3.5) se obtiene la función de transferencia del circuito convertidor reductor, y es dada por:

$$\frac{V_o(s)}{U_{av}(s)} = \frac{E/LC}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad (3.7)$$

El sistema que rige al convertidor reductor, fácilmente se observa que es de segundo orden, además es fácil ver que las variables del sistema ( $i_L$  y  $u_{av}$ ) se pueden expresar en términos de la variable que representa al voltaje de salida ( $V_o$ ), con esto  $V_o$  es una salida plana del convertidor reductor. Renombrando a la variable  $V_o$  como  $y$ , tenemos la parametrización diferencial del sistema en función del voltaje de salida del convertidor y de sus derivadas, y es:

$$\begin{aligned} i_L &= \frac{y}{R} + C\dot{y} \\ u_{av} &= \frac{LC}{E}\ddot{y} + \frac{L}{ER}\dot{y} + \frac{1}{E}y \end{aligned} \quad (3.8)$$

Reescribiendo la relación entrada-salida del sistema, en función de la salida plana ( $y$ ) se tiene que:

$$\ddot{y} + \frac{1}{RC}\dot{y} + \frac{1}{LC}y = \frac{E}{LC}u_{av} \quad (3.9)$$

### 3.3. Controlabilidad en Sistemas Lineales

Considérese el siguiente sistema lineal e invariante con el tiempo:

$$\dot{x} = Ax + Bu \quad (3.10)$$

donde  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}$ ,  $A$  y  $B$  son matrices constantes de dimensión  $n \times n$ , y  $n \times 1$  respectivamente. Se dice que el sistema (3.10) es controlable en  $t = t_0$  si es posible tener señales de control sin restricción alguna que transfieran un estado inicial  $x(t_0)$ , a cualquier otro estado finito en un intervalo de tiempo finito. La propiedad de controlabilidad está íntimamente ligada con la planitud diferencial, de tal forma que se dice que un sistema que es controlable es a su vez plano y viceversa. (Véase [34, 35]).

#### 3.3.1. Controlabilidad mediante funciones de transferencia

La propiedad de controlabilidad completa de un sistema lineal que es representado mediante la función de transferencia, se cumple siempre y cuando

el numerador y el denominador sean coprimos entre sí, es decir, si no existen factores comunes no triviales que ocasionen una cancelación de términos dentro de la función de transferencia, (Ver [36]). Consideremos un sistema lineal SISO, cuya función de transferencia está dada por:

$$Y(s) = \frac{N(s)}{D(s)}U(s) \quad (3.11)$$

donde  $Y(s)$  es la salida,  $U(s)$  es la señal de entrada,  $N(s)$  y  $D(s)$  son polinomios en el dominio de la frecuencia, de tal forma que:

$$Y(s) = \frac{b_ms^m + b_{m-1}s^{m-1} + \dots + b_1s + b_0}{s^n + a_ns^{n-1} + a_{n-1}s^{n-2} + \dots + a_1s + a_0}U(s) \quad m < n \quad (3.12)$$

donde el grado del polinomio  $N(s)$  es menor que el polinomio  $D(s)$ . En [36] se establece que el sistema es controlable si y solo si los polinomios  $N(s)$  y  $D(s)$  son coprimos, es decir, que no tengan factores comunes no triviales. En tal caso, mediante el teorema de *Bezout*, existen polinomios  $A(s)$  y  $B(s)$ , llamados polinomios *Bezout*, tal que se satisface

$$A(s)N(s) + B(s)D(s) = 1 \quad (3.13)$$

Definamos a  $F(s)$  como una nueva variable,

$$F(s) = \frac{1}{D(s)}U(s) \quad (3.14)$$

Es claro que la entrada del sistema,  $U(s)$  y la salida del sistema  $Y(s)$  se escriben en términos de  $F(s)$  como,

$$Y(s) = N(s)F(s), \quad U(s) = D(s)F(s) \quad (3.15)$$

Multiplicando ambos lados de la identidad *Bezout* (3.13), por la variable  $F(s)$ , se obtiene:

$$\begin{aligned} F(s) &= A(s)N(s)F(s) + B(s)D(s)F(s) \\ &= A(s)Y(s) + B(s)U(s) \end{aligned} \quad (3.16)$$

La última ecuación muestra como se calcula una variable endógena,  $f$ , la cual depende únicamente de las entradas y las salidas del sistema, y tiene un número finito de sus derivadas, es decir, una variable la cual es una función

diferencial de las variables del sistema, tal que, todas las variables (entradas y salidas), se expresan como funciones diferenciales de la variable calculada,  $f$ . La variable  $f$ , claramente aprueba como una salida plana.

Por tanto, dando un sistema lineal controlable, una salida plana,  $f$ , puede ser calculada en términos de una relación endógena entre las variables del sistema,  $y$  y  $u$ . Todo esto se resume en la siguiente proposición.

**PROPOSICIÓN 1** [35]. *Dada una función de transferencia de un sistema SISO lineal e invariante con el tiempo, el sistema es plano si y solo si los polinomios del numerador y el denominador de la función de transferencia son coprimos. En otras palabras, un sistema lineal en la forma (3.11) es plano si y solo si el sistema es controlable.*

Suponiendo que un sistema lineal SISO de la forma (3.11), tiene un numerador  $N(s)$  el cual es una constante, mientras que el polinomio del denominador  $D(s)$  es de grado  $\geq 1$ . Se observa claramente que el sistema es controlable, y por lo tanto plano. Con esto, para todos los sistemas lineales SISO cuya función de transferencia contenga una constante en el numerador, una salida plana es dada como salida del sistema  $y$ , o por cualquier múltiplo constante de dicha salida.

### Controlabilidad mediante la función de transferencia del convertidor reductor

Para comprobar que el sistema del convertidor reductor es controlable, tenemos que corroborar que los polinomios del numerador y denominador de la función de transferencia sean coprimos. De la función de transferencia obtenida del convertidor reductor en (3.7) se tiene que,

$$\begin{aligned} N(s) &= \frac{E}{LC} \\ D(s) &= s^2 + \frac{1}{RC}s + \frac{1}{LC} \end{aligned} \quad (3.17)$$

Se puede observar fácilmente de (3.17) que los polinomios son coprimos, al ser el numerador una constante. Sin embargo, para poder comprobarlo mediante el teorema de *Bezout*, es necesario encontrar los polinomios  $A(s)$  y  $B(s)$  tal que satisfagan la ecuación (3.13), tal que

$$A(s)\left(\frac{E}{LC}\right) + B(s)\left(s^2 + \frac{1}{RC}s + \frac{1}{LC}\right) = 1 \quad (3.18)$$

De (3.18) se obtiene que la forma del polinomio de  $A$  es  $A(s) = ms^2 + ns + p$ , mientras que el polinomio  $B$  es  $B = k$ . Donde,  $m$ ,  $n$ ,  $p$  y  $k$  son coeficientes

constantes de los polinomios. Por tanto, la identidad de *Bezout* está dada por:

$$(ms^2 + ns + p)\left(\frac{E}{LC}\right) + k\left(s^2 + \frac{1}{RC}s + \frac{1}{LC}\right) = 1 \quad (3.19)$$

Al desarrollar (3.19) se tiene un sistema de tres ecuaciones con cuatro incógnitas, tal como se ve a continuación,

$$\begin{aligned} \frac{E}{LC}m + k &= 0 \\ \frac{E}{LC}n + \frac{1}{RC}k &= 0 \\ pE + k &= 1 \end{aligned}$$

Al analizar el sistema de ecuaciones anterior, se observa que el sistema tiene múltiples soluciones. Una solución en particular se obtiene al igualar el coeficiente  $k = 1$ , por tanto la solución al sistema es:

$$\begin{aligned} m &= -\frac{LC}{E} \\ n &= -\frac{L}{RE} \\ p &= \frac{LC - 1}{E} \end{aligned}$$

Los polinomios (A(s) y B(s)) que cumplen con la identidad *Bezout* son

$$\begin{aligned} A(s) &= -\frac{LC}{E}s^2 - \frac{L}{RE}s + \frac{LC - 1}{E} \\ B(s) &= 1 \end{aligned}$$

De aquí se tiene que el sistema es controlable y por lo tanto es plano, y la salida plana es dada por:

$$F(s) = \frac{U(s)}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad (3.20)$$

Tenemos que:

$$\begin{aligned} U(s) &= \left(s^2 + \frac{1}{RC}s + \frac{1}{LC}\right)F(s) \\ Y(s) &= \frac{E}{LC}F(s) \end{aligned}$$

La salida plana  $F(s)$  es dada, en términos de  $U$  y de  $Y$  mediante:

$$\begin{aligned} F(s) &= A(s)Y(s) + B(s)U(s) \\ F(s) &= \left(-\frac{LC}{E}s^2 - \frac{L}{RE}s + \frac{LC-1}{E}\right)Y(s) + U(s) \end{aligned} \quad (3.21)$$

### 3.3.2. Controlabilidad mediante la representación de estados

Considérese un sistema lineal SISO e invariante con el tiempo descrito por (3.12), con polinomios coprimos en el numerador y denominador. Entonces, una salida plana, está dada por,

$$F(s) = \frac{k}{s^n + a_{n-1}s^{n-1} + \dots + a_0}U(s) \quad (3.22)$$

Para cualquier constante arbitraria  $k$ , diferente de cero. La representación del sistema, en términos de una ecuación diferencial y una ecuación de salida escalar, es dada por:

$$\begin{aligned} \frac{d^n f}{dt^n} + a_{n-1}\frac{d^{n-1}f}{dt^{n-1}} + \dots + a_0 f &= ku \\ y &= \frac{1}{k}\left[b_m \frac{d^m f}{dt^m} + b_{m-1}\frac{d^{m-1}f}{dt^{m-1}} + \dots + b_0 f\right] \end{aligned} \quad (3.23)$$

Definiendo a  $x_1 = f$ ,  $x_2 = \dot{f}$ ,  $\dots$ ,  $x_n = f^{(n-1)}$ , se obtiene la representación en estados del sistema,

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = A \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + Bu, \quad y = C \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (3.24)$$

con

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & \dots & -a_{n-1} \end{bmatrix}, \quad b = k \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \\ C &= \frac{1}{k} [b_0 \quad \dots \quad b_m \quad 0 \quad \dots \quad 0] \end{aligned} \quad (3.25)$$

Para poder verificar la controlabilidad de sistema se hace uso del criterio de Kalman, el cual se expone a continuación.

### Criterio de Kalman

Consideremos ahora un sistema lineal general de la forma

$$\dot{x} = Ax + Bu \quad (3.26)$$

donde  $x \in \mathbb{R}^n$  es el vector de estados y  $u \in \mathbb{R}^m$  es el vector de entrada. Las matrices  $A$  y  $B$  son matrices constantes de dimensión  $n \times n$ , y  $n \times m$  respectivamente.

**DEFINICIÓN 2** [37]. Decimos que un sistema lineal es controlable, si y solo si, dado un intervalo de tiempo  $T > 0$  y dos puntos arbitrarios  $x_0$  y  $x_T$  en  $\mathbb{R}^n$ , existe una función continua en el intervalo de tiempo  $t \mapsto \bar{u}(t)$  de  $[0, T]$  en  $\mathbb{R}^m$  tal que la solución  $\bar{x}(t)$  de (3.26) generada por  $\bar{u}$  y con condiciones iniciales  $\bar{x}(0) = x_0$ , satisface  $\bar{x}(T) = x_T$ . En otras palabras:

$$e^{AT}x_0 + \int_0^T e^{A(T-t)}B\bar{u}(t)dt = x_T \quad (3.27)$$

Esta propiedad solo depende del par de matrices  $A$  y  $B$  como se observa en el siguiente criterio de controlabilidad obtenido por R. E. Kalman.

**TEOREMA 1**. Una condición necesaria y suficiente para que el sistema dado en (3.26) sea controlable es que el rango de la matriz

$$\mathcal{C} = (B \mid AB \mid \dots \mid A^{n-1}B) \quad (3.28)$$

sea igual a  $n$ .

La matriz  $\mathcal{C}$  es llamada matriz de controlabilidad de Kalman, y es de tamaño  $n \times nm$ . Ahora se define una transformación de coordenadas estática de la entrada,  $v$ , como:

$$v = ku - a_0x_1 - a_1x_2 - \dots - a_{n-1}x_n \quad (3.29)$$

El nuevo sistema, con la entrada redefinida, es reducida a un sistema en la forma canónica de *Brunovsky*.

$$\dot{x} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} v$$

Por lo tanto se dice que cualquier sistema lineal SISO plano que esté en su forma canónica controlable, a través de una transformación estática de la coordenada de entrada es equivalente al sistema en forma canónica controlable de *Brunovski*. Por otro lado, supongamos que el polinomio característico de la matriz constante  $A$ , escrito en el plano complejo de la variable  $s$ , es dado por

$$s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1s + \alpha_0$$

Si hacemos una transformación de coordenadas en el espacio de estados de la forma  $z = Tx$ , con  $T$  definido como la matriz inversa de la matriz de controlabilidad de Kalman,

$$T = [B, AB, \dots, A^{n-1}B]^{-1}$$

El nuevo sistema, con las nuevas coordenadas,  $z$ , es:

$$\dot{z} = \Lambda z + \gamma u, \quad \Lambda = TAT^{-1}, \quad \gamma = TB \quad (3.30)$$

donde,

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & -\alpha_0 \\ 1 & 0 & 0 & \cdots & 0 & -\alpha_1 \\ 0 & 1 & 0 & \cdots & 0 & -\alpha_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -\alpha_{n-2} \\ 0 & 0 & 0 & \cdots & 1 & -\alpha_{n-1} \end{bmatrix}, \quad \gamma = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

El sistema de coordenadas  $y = z_n$  completamente parametriza las variables de estado transformadas, y por lo tanto, las variables originales  $x$ , también como la entrada  $u$ . Así, las variables transformadas y la variable de entrada  $u$  pueden ser escritas en términos de  $y$  y un número finito de sus derivadas,

$$\begin{aligned} z_{n-1} &= \dot{y} + \alpha_{n-1}y \\ z_{n-2} &= \ddot{y} + \alpha_{n-1}\dot{y} + \alpha_{n-2}y \\ &\vdots \\ z_1 &= y^{(n-1)} + \alpha_{n-1}y^{(n-2)} + \cdots + \alpha_1y \\ u &= y^{(n)} + \alpha_{n-1}y^{(n-1)} + \cdots + \alpha_0y \end{aligned} \quad (3.31)$$

Como consecuencia, todos los estados originales del sistema  $x$  pueden ser parametrizados en términos de la salida  $y$ . Por lo tanto, el estado transformado  $y = z_n$  es entonces la salida plana. De ahí, se tiene la siguiente proposición.



PROPOSICIÓN 2 [35] *La salida plana del sistema lineal, controlable se denota en su forma de espacio de estado como:*

$$\dot{x} = Ax + Bu \quad (3.32)$$

la cual está dada por un módulo de factor constante y por una combinación de los estados obtenidos del último renglón de la matriz inversa de controlabilidad de Kalman (3.28), es decir:

$$y = [0 \ 0 \ \dots \ 1] [B, AB, \dots, A^{n-1}B]^{-1}x \quad (3.33)$$

Por otro lado, supongamos que la salida plana  $y$ , es una función sólo del vector  $x$ .

$$y = \lambda x = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (3.34)$$

Nótese que el vector  $\lambda$  es de dimensión  $1 \times n$ . Ahora el problema de encontrar la salida plana, se centra en como encontrar el vector  $\lambda$ . Para ello se debe encontrar primero, las  $n-1$  derivadas de la expresión (3.34) y sustituir (3.26). Por lo que se tiene:

$$\begin{aligned} y &= \lambda x \\ \dot{y} &= \lambda \dot{x} = \lambda Ax + \lambda Bu \\ \ddot{y} &= \lambda A^2 x + \lambda ABu + \lambda B\dot{u} \\ &\vdots \\ y^{(n-1)} &= \lambda A^{(n-1)}x + \lambda A^{(n-2)}Bu + \dots + \lambda Bu^{(n-2)} \end{aligned} \quad (3.35)$$

En forma matricial, las relaciones anteriores se escriben como:

$$\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda A \\ \lambda A^2 \\ \vdots \\ \lambda A^{(n-1)} \end{bmatrix} x + \begin{bmatrix} 0 & 0 & \dots & 0 \\ \lambda B & 0 & \dots & 0 \\ \lambda AB & \lambda B & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \lambda A^{(n-2)}B & \lambda A^{(n-3)}B & \dots & \lambda B \end{bmatrix} \begin{bmatrix} u \\ \dot{u} \\ \ddot{u} \\ \vdots \\ u^{(n-2)} \end{bmatrix} \quad (3.36)$$

Ya que se debe obtener  $x$  en términos solo de  $y$  y de sus derivadas, las entradas de  $u$  y sus derivadas deben desaparecer. Para ello debemos hacer que el vector  $\lambda$  satisfaga

$$\lambda B = 0, \lambda AB = 0, \dots, \lambda A^{(n-2)}B = 0 \quad (3.37)$$

Se observa que la matriz anterior corresponde con la matriz de controlabilidad, a excepción del término  $\lambda A^{(n-1)}B$ . Se debe asegurar que el término  $\lambda A^{(n-1)}B \neq 0$ , de lo contrario la única solución sería  $\lambda = 0$ . Por lo tanto,  $\lambda$  debe ser un vector ortogonal a todo vector columna de la matriz de controlabilidad, excepto el último vector, con el fin de obtener

$$\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = \begin{bmatrix} \lambda \\ \lambda A \\ \lambda A^2 \\ \vdots \\ \lambda A^{(n-1)} \end{bmatrix} x \quad (3.38)$$

Otro punto a considerar es que el vector que multiplica a  $x$ , debe ser invertible. Esto significa que la salida plana es una salida observable, para algún grado relativo  $n$ . De no ser cierto, esto significa que existe un vector  $\gamma$  tal que

$$\gamma \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(n-1)} \end{bmatrix} = \gamma \begin{bmatrix} \lambda \\ \lambda A \\ \lambda A^2 \\ \vdots \\ \lambda A^{(n-1)} \end{bmatrix} x = 0 \quad (3.39)$$

Entonces  $y$  estaría envuelta en sí misma y no estaría relacionada a ninguna otra variable en el sistema. Esto significa que  $y$  es una variable exógena, haciendo al sistema incontrolable. Para que esto no suceda y para que el sistema sea controlable se debe satisfacer que  $\lambda A^{(n-1)}B = k$ , donde  $k$  es una constante distinta de cero y con esto se puede proponer lo siguiente:

$$\lambda = k [0 \ 0 \ \dots \ 1] [B \ AB \ \dots \ A^{(n-1)}B]^{-1} \quad (3.40)$$

Donde  $\lambda$  es proporcional, solo al último renglón de la matriz de controlabilidad. Ahora despejando el vector  $x$  de (3.38) para la observabilidad de la salida plana

$$x = \begin{bmatrix} \lambda \\ \lambda A \\ \lambda A^2 \\ \vdots \\ \lambda A^{(n-1)} \end{bmatrix}^{-1} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ y^{(n-1)} \end{bmatrix} \quad (3.41)$$

se tiene de (3.38), lo siguiente

$$y^{(n-1)} = \lambda A^{(n-1)} x$$

derivando la última expresión y sustituyendo el modelo lineal se tiene:

$$y^{(n)} = \lambda A^{(n-1)} \dot{x} = \lambda A^n x + \lambda A^{(n-1)} B u \quad (3.42)$$

Utilizando el teorema de Cayley-Hamilton que se enuncia a continuación

**TEOREMA 2** . Si  $A$  es una matriz cuadrada de orden  $n$  y si

$$p(X) = \det(XI - A) = X^n + p_{(n-1)}X^{(n-1)} + \dots + p_1X + p_0 \quad (3.43)$$

es su polinomio característico, entonces al sustituir formalmente  $X$  por la matriz  $A$  en el polinomio, el resultado es la matriz nula:

$$p(A) = A^n + p_{(n-1)}A^{(n-1)} + \dots + p_1A + p_0I = 0 \quad (3.44)$$

Se puede decir que

$$A^n + a_{(n-1)}A^{(n-1)} + \dots + a_1A + a_0I = 0$$

despejando  $A^n$

$$A^n = -a_{(n-1)}A^{(n-1)} - \dots - a_1A - a_0I$$

Tomando el último resultado y sustituyéndolo en (3.42) se tiene que

$$\begin{aligned} y^{(n)} = & \lambda(-a_{(n-1)}A^{(n-1)} - \dots - a_1A - a_0I)\dot{x} \\ & + \lambda(-a_{(n-1)}A^{(n-2)} - \dots - a_1A - a_0I)Bu \end{aligned} \quad (3.45)$$

Tomando las condiciones en (3.37) y dado que el producto  $-a_0\lambda IB$  es una constante, la ecuación anterior puede reducirse a

$$y^{(n)} = \lambda(-a_{(n-1)}A^{(n-1)} - \dots - a_1A - a_0I)x + u \quad (3.46)$$

al desarrollar la ecuación anterior y despejando la variable de entrada  $u$  se tiene

$$u = y^{(n)} + a_{(n-1)}A^{(n-1)} + \dots + a_1\dot{y} + a_0y \quad (3.47)$$

con esto se demuestra que es posible obtener una salida endógena partiendo de la combinación lineal de los estados del sistema, basado en la forma (3.34) y la cual parametriza completamente al sistema de la forma (3.26).

### Obtención de la salida plana del sistema convertidor reductor

Renombrando las variables  $i_L$  y  $V_0$  como  $x_1$  y  $x_2$  respectivamente y en base a las ecuaciones diferenciales del convertidor (3.4) y a la función de transferencia (3.7), se obtiene su representación en variables de estados de la siguiente forma:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= x_2 \end{aligned}$$

donde,

$$\begin{aligned} A &= \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} & B &= \begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix} \\ x &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} & \dot{x} &= \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \end{aligned}$$

Con la matriz A y B se obtiene la matriz de controlabilidad del sistema dada por  $\mathcal{C} = [A \ AB]$ , la cual es:

$$\mathcal{C} = \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix} \quad (3.48)$$

donde el determinante de la matriz es

$$\frac{E^2}{L^2C} \neq 0$$

El rango de la matriz es 2, por lo tanto el sistema es controlable y a su vez plano. La obtención de la salida plana se realiza a través del Teorema (2), donde se tiene

$$\begin{aligned}
y &= [0 \ 1] \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
y &= \frac{LC}{E} x_2 = kx_2
\end{aligned} \tag{3.49}$$

donde  $k = \frac{LC}{E}$ . De aquí se obtiene como salida plana solamente a  $x_2$ . Definiendo el vector  $\lambda$ , como  $\lambda = [0 \ 1]$  y mediante (3.38), la salida plana y sus derivadas son:

$$\begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{x_1}{C} - \frac{x_2}{RC} \\ -\frac{1}{RC^2}x_1 + \frac{1}{C}\left(\frac{1}{R^2C} - \frac{1}{L}\right)x_2 \end{bmatrix}$$

Denotemos ahora a  $x_2$  como  $F$ . La planitud del sistema implica que las variables del sistema, incluyendo la variable de control, pueden ser parametrizadas en términos de  $F$  y un número finito de sus derivadas. Por tanto

$$\begin{aligned}
v_o &= F \\
i_L &= C\dot{F} + \frac{1}{R}F
\end{aligned} \tag{3.50}$$

y la entrada de control promedio se obtiene como

$$u_{av} = \frac{LC}{E} \left( \ddot{F} + \frac{1}{RC}\dot{F} + \frac{1}{LC}F \right) \tag{3.51}$$

Además, del modelo promedio dado en (2.7), se tiene que el sistema es observable desde la salida  $v_o$ , es decir, la matriz de observabilidad de Kalman, dada como,

$$\mathcal{O} = \begin{bmatrix} c^T \\ c^T A \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \tag{3.52}$$

tiene rango completo (rango=2). Por lo tanto, el modelo del sistema es observable desde la salida  $y = F = v_o$ . Este hecho establece la reconstructibilidad del sistema, es decir, todas las variables de estado son parametrizables en términos de la entrada, la salida, y un finito número de integrales de las variables de entrada y salida (Ver [38]). Esta parametrización de la integral de entrada-salida de las variables de estado, es dado, con la siguiente condición inicial:

$$\begin{aligned}
\hat{F} &= \left(\frac{E}{LC}\right) \int_0^t [u_{av}(\tau) - \frac{1}{E}F(\tau)]d\tau - \frac{1}{RC}F \\
F &= v_o
\end{aligned} \tag{3.53}$$

La primera expresión en (3.53) se obtiene mediante la integración directa de la expresión polinomial diferencial dada en (3.51). Nótese, que para que los estados iniciales diferentes de cero, la ecuación que relaciona el valor actual de la salida del convertidor reductor al valor estructural estimado en (3.53) este se da como:

$$\dot{F} = \widehat{F} + \dot{F}_0 \quad (3.54)$$

donde  $\dot{F}_0$  denota el valor inicial de la razón de cambio del voltaje de salida.

### 3.4. Controlador GPI

El control clásico PI (Proporcional Integral) tiende a eliminar el error en estado estable del sistema, sin embargo el tiempo de respuesta del sistema puede aumentar y causar picos en la señal controlada. Para ayudar a disminuir este efecto se hace uso del control PID. Sin embargo, el controlador PID presenta también algunas desventajas, como la amplificación de la señal de ruido o provocar efectos de saturación en el actuador del sistema. El control PI Generalizado o GPI, propuesto por Fliess *et al.* en [38], presenta la característica de eludir a los observadores asintóticos en términos de reconstructores estructurales de estado y compensación mediante integrales de error iteradas. La técnica GPI se ha venido utilizando en los últimos años para el control de convertidores de potencia CD-CD, debido a las siguientes características: rápida respuesta dinámica, robustez con respecto a perturbaciones constantes o de tipo rampa. Además, usando esta técnica, se reduce significativamente el uso de sensores en la medición de estados del sistema a controlar. Los controladores GPI se basan en reconstructores integrales de estado procesando únicamente las entradas y salidas del sistema [15], [16]. (Ver además: [17], [18]).

Para el diseño del control GPI se consideran los siguientes aspectos:

- El reconstructor integral se basa en el modelo dinámico de la planta.
- Las integrales de error de la salida se agregan a la ley de control para garantizar la robustez.
- Las integrales de la entrada que aparecen en la formulación de la ley de control por el método GPI, se agregan para facilitar la regulación del sistema.

### Diseño del Controlador GPI para el convertidor reductor

Sea el sistema del convertidor reductor dado por las siguientes ecuaciones en variables de estado

$$\begin{aligned} \dot{x}_1 &= -\frac{1}{L}x_2 + \frac{E}{L}u_{av} \\ \dot{x}_2 &= \frac{1}{C}x_1 - \frac{1}{RC}x_2 \\ y &= x_2 \end{aligned} \quad (3.55)$$

De (3.55), se diseña la ley de control para la estabilización de la salida del voltaje de convertidor reductor, para obtener un voltaje deseado, denotado como  $\bar{F}$ . Se propone una ley de control por retroalimentación de estados como:

$$\begin{aligned} u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\dot{F} + \frac{1}{E}F \\ v &= -k_3\dot{F} - k_2(F - \bar{F}) \end{aligned} \quad (3.56)$$

En la ley de control por retroalimentación (3.56), se sustituye la variable de estado  $\dot{F}$ , por la variable estimada de la derivada de la salida plana  $\hat{\dot{F}}$ . Esto implica que la ley de control dada en (3.56) se vea afectado por valores constantes, ocasionados por la incorporación del estimador de  $\hat{\dot{F}}$ , y por perturbaciones externas desconocidas por el sistema. Para corregir adecuadamente este efecto de desestabilización, ocasionados por los errores de estimación y por las perturbaciones externas, se utiliza la compensación mediante integrales del error de la salida plana. El diseño de la ley de control que incorpora el estimador de  $\hat{\dot{F}}$ , y la compensación a través de las integrales del error,  $F - \bar{F}$ , recibe el nombre de control Proporcional Integral Generalizado, GPI, el cual es dado como:

$$\begin{aligned} u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\hat{\dot{F}} + \frac{1}{E}F \\ v &= -k_3(\hat{\dot{F}}) - k_2(F - \bar{F}) - k_1\gamma - k_0\eta \\ \dot{\gamma} &= F - \bar{F} \\ \dot{\eta} &= \gamma \end{aligned} \quad (3.57)$$

Sea  $e = F - \bar{F}$  el error de estabilización. La dinámica del error se obtiene mediante la sustitución de (3.53) y las ecuaciones del controlador (3.57), en la entrada de control promedio dada en (3.51). Entonces, se obtiene el controlador GPI como:

$$\begin{aligned} \ddot{F} &= -k_3(\dot{F} - \dot{F}_0) - k_2(F - \bar{F}) - k_1 \int_0^t (F(\tau) - \bar{F})d\tau \\ &\quad - k_0 \int_0^t \int_0^\tau (F(\lambda) - \bar{F})d\lambda d\tau \end{aligned} \quad (3.58)$$

La ecuación característica de la relación integro-diferencial (3.58), en términos de la estabilización de error, es dado por:

$$e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e = 0$$

Los valores de los parámetros  $\{k_3, k_2, k_1, k_0\}$  deben calcularse de tal forma que el polinomio característico en lazo cerrado,

$$p(s) = s^4 + k_3 s^3 + k_2 s^2 + k_1 s + k_0 \quad (3.59)$$

tenga todas sus raíces con parte real negativa. Los parámetros del controlador se seleccionan a través del siguiente polinomio Hurwitz,

$$p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)^2$$

Por tanto, las ganancias del controlador son:

$$\begin{aligned} k_3 &= 4\zeta\omega_n \\ k_2 &= 4\zeta^2\omega_n^2 + 2\omega_n^2 \\ k_1 &= 4\zeta\omega_n^3 \\ k_0 &= \omega_n^4 \end{aligned}$$

y se obtienen con  $\zeta = 0,7071$  y  $\omega_n > 0$

### Simulación del control GPI

Antes de implementar el diseño del control GPI es importante realizar algunas pruebas de simulación para observar el comportamiento del controlador. Para ello se realiza una simulación del controlador utilizando las herramientas Matlab/Simulink y Psim. El convertidor reductor se diseña bajo los siguientes parámetros:  $E = 24V$ ,  $R = 100\Omega$ ,  $C = 470\mu F$ ,  $L = 1mH$  a una frecuencia de conmutación fija de 48 KHz. Los valores de diseño de los coeficientes  $\{k_3, k_2, k_1, k_0\}$  se muestran en la tabla 3.1.

En primer lugar se hizo una simulación del convertidor reductor a lazo abierto, sin la intervención de ningún controlador. En esta parte de la simulación se realizó la prueba con un voltaje de referencia de 12 V. Así en la figura 3.1 se muestra la respuesta en salida de voltaje del convertidor reductor con el voltaje de referencia deseado. En esta figura se observa que el tiempo de establecimiento es de aproximadamente 45 ms, sin embargo, en la respuesta transitoria se observa un sobretiro de 97.7%, lo cual es uno de los problemas que debe de corregir el controlador GPI propuesto.

Por otra parte, se hizo una simulación en lazo cerrado, donde interviene el controlador GPI para regular el voltaje de salida del convertidor reductor. Este controlador tendrá la tarea de obtener una rápida respuesta transitoria y evitar sobretiro, es decir, mejorar el desempeño del sistema. Así, en la figura



Tabla 3.1: Parámetros para el controlador GPI.

Parámetro	Valor
$k_0$	$3,91 * 10^{13}$
$k_1$	$4,42 * 10^{10}$
$k_2$	$2,50 * 10^7$
$k_3$	$7,07 * 10^3$

3.2 se muestra la respuesta del voltaje de salida del convertidor reductor debido a la acción del controlador GPI. En contraste con la respuesta a lazo abierto, se observa que el tiempo de establecimiento es de aproximadamente 16 ms, sin embargo esta respuesta no presenta sobretiro alguno, por lo que se deduce que el control GPI es eficiente y logra su objetivo que es alcanzar el nivel de voltaje deseado, mejorando el desempeño de la salida de voltaje obtenida a lazo abierto.

Una vez obtenidas estas respuestas se procede a la parte de modelado en VHDL, para su posterior implementación, tal como se verá en los siguientes capítulos.

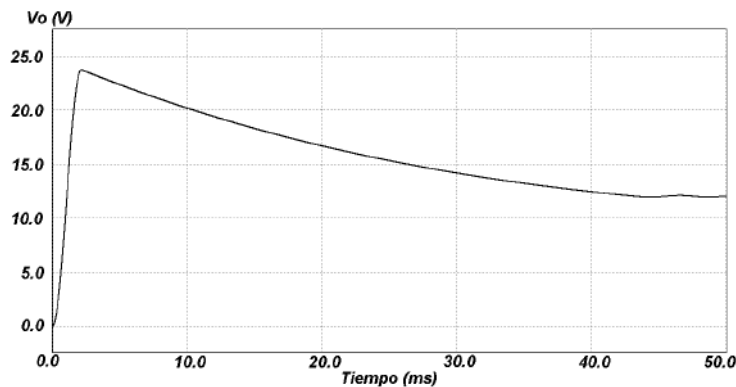


Figura 3.1: Respuesta del convertidor reductor en lazo abierto.

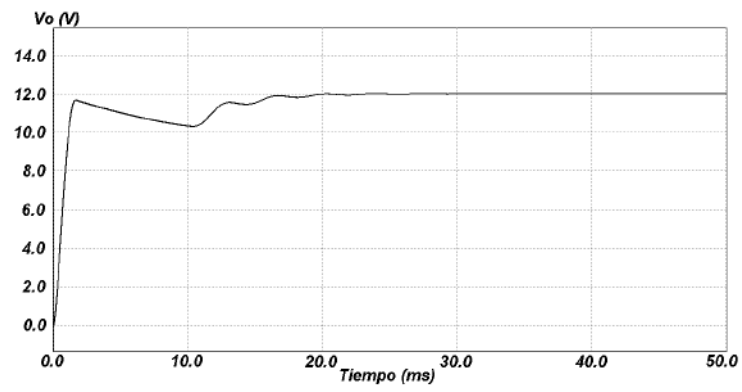


Figura 3.2: Respuesta del convertidor reductor con la técnica GPI.

## Capítulo 4

# Métodos y Tecnologías Empleadas

**RESUMEN:** En este capítulo se muestran aspectos teóricos importantes que permitirán implementar el control GPI mediante punto flotante. El punto flotante que se ocupa corresponde a un punto flotante de precisión simple, sobre el cual se realizan las operaciones básicas de suma, resta y multiplicación. Una vez que se obtienen los conocimientos necesarios sobre punto flotante, se muestra aspectos sobre el lenguaje descriptor de hardware (VHDL) utilizado y aspectos generales sobre los arreglos de compuertas programables en campo, FPGA.

### 4.1. Representación de números en Punto flotante

#### 4.1.1. Aspectos Generales

Los números reales pueden ser representados generalmente mediante una línea. Cada punto en la línea corresponde a un número. Los números racionales pueden ser representados por dos números enteros, el numerador y el denominador. Esto puede representar la ventaja de precisión, sin embargo la principal desventaja es que no es muy conveniente para usarlo en aritmética, los sistemas que representan números racionales en esta forma (numerador y denominador) se dice que son simbólicos más que numéricos. Sin embargo para propósitos de aritmética computacional, los números reales, racionales o irracionales, son aproximados mediante la representación binaria del número. Existen dos formas de representación de números racionales, punto fijo y punto flotante.

La representación en punto fijo está compuesta por tres campos, el primer campo formado por un bit, conocido como bit de signo, indica el signo del

número, el segundo campo, formado por  $n$  bits, representa la parte entera y el tercer campo, formado por  $m$  bits, representa la parte fraccionaria, [39, 40]. Por ejemplo, la representación en 32 bits con campos de 15 y 16 bits para la parte entera y decimal respectivamente, del número 4.75 sería de la siguiente forma

0	000000000000100	1100000000000000
---	-----------------	------------------

La principal desventaja que presenta el punto fijo es la limitación en el número de datos que puede representar. En la representación de 32 bits vista anteriormente, el rango de los datos estaría aproximadamente entre  $2^{-16}$  y  $2^{15}$ , lo cual no es muy conveniente en muchas aplicaciones, por lo que no es muy común que se utilice este tipo de representación en cálculos numéricos, [39].

Sin embargo, existe otra forma de representar los números reales basada en la notación científica o notación exponencial, la cual es llamada representación en punto flotante. En la notación decimal un número  $x$  diferente de cero se expresa en forma exponencial de la siguiente manera:

$$x = \pm S \times 10^E, \quad (4.1)$$

Donde  $1 \leq S < 10$  y  $E$  es un entero.  $S$  y  $E$  son denominados la mantisa y el exponente respectivamente. Sin embargo en la computadora la representación de un número es en forma binaria, así, un número diferente de cero en la computadora en forma exponencial está expresado de la siguiente manera:

$$x = \pm S \times 2^E, \quad (4.2)$$

Donde  $1 \leq S < 2$  y  $E$  es un entero. Como resultado de esto, es necesario que  $S$  esté en forma binaria:

$$S = (b_0.b_1b_2b_3\dots)_2 \quad \text{con } b_0 = 1 \quad (4.3)$$

Por ejemplo, el número decimal 4.75 en punto fijo quedó expresado en la forma 100,11, sin embargo en la computadora de la forma exponencial sería:

$$4,75 = (1,0011)_2 \times 2^2$$

La representación mostrada en (4.2) y (4.3) se dice que está normalizada debido a que el punto decimal se encuentra después del primer bit diferente de cero. El proceso de obtener esta forma de representación se denomina normalización. Cabe mencionar que solo números normalizados son los que se pueden almacenar en la computadora.

Para almacenar números flotantes en la computadora, dividimos la cadena de bits en tres campos principales que representan el signo, el exponente

$E$  y la mantisa  $S$ . Una representación de 32 bits puede ser dividida en los tres campos antes mencionados de la manera siguiente: un bit de signo, 8 bits para el exponente y 23 para la mantisa. El signo del número dependerá del valor del bit de signo, si el bit es 0 indica un número positivo, en cambio un valor de 1 indica un valor negativo. El valor del exponente  $E$ , al ser de 8 bits, podría representar un intervalo de 256 números, sin embargo en la siguiente sección se verán más detalles sobre la representación de los exponentes. Finalmente los 23 bits de la mantisa almacenarán los primeros 23 bits después del punto en la representación binaria normalizada ( $b_1b_2b_3 \dots b_{23}$ ). Un punto importante de mencionar es que no es necesario almacenar el bit  $b_0$  debido a que siempre tendrá el valor de 1, por ello podemos decir que  $b_0$  es un bit escondido.

#### 4.1.2. Estándar IEEE para punto flotante

Todavía a finales de los años 70's, cada fabricante de computadora tenía su propio formato de punto flotante, por lo que cada uno desarrollaba su propia aritmética para su propio formato, y muchos de ellos tenían errores en la aritmética desarrollada. Por ello, a principios de los años 80 se conformó un comité del IEEE (*Institute of Electrical and Electronics Engineers*) para estandarizar la aritmética de punto flotante, con el fin de poder intercambiar datos entre diferentes tipos de arquitecturas de computadoras, y además proporcionar un diseño correcto a los diseñadores [41]. El resultado fue el estándar IEEE 754 [42]. El estándar define cuatro precisiones: precisión simple (32 bits), precisión doble (64 bits), precisión simple extendida ( $\geq 43$  bits) y precisión doble extendida (generalmente implementada con 80 bits). La tabla 4.1 muestra un resumen de los parámetros de los formatos mencionados. El estándar solo requiere la implementación de los valores de 32 bits, los demás son opcionales.

Los formatos de precisión simple y precisión doble utilizan la base 2 para su representación para la parte fraccionaria y la notación en exceso para los exponentes. La figura 4.1 muestra la representación de estos formatos. Ambos formatos utilizan un bit de signo (0 para un número positivo, y 1 para un número negativo), enseguida el exponente el cual hace uso del exceso 127 y 1023 para la precisión simple y doble respectivamente [40]. Por último se encuentra la parte fraccionaria con un total de 23 bits para la precisión simple y 52 bits para la precisión doble.

Una fracción consiste en un bit implícito (1), punto binario implícito seguido de 23 o 52 bits. A fin de evitar confusiones con las fracciones convencionales, la combinación del bit '1' implícito, el punto decimal binario, y los 23 o 52 bits consecutivos se llama **significando** en vez de fracción o mantisa.

El estándar cuenta con cuatro tipos numéricos además de los números normalizados, los cuales se muestran en la tabla 4.2.

Tabla 4.1: Resumen de los parámetros de los formatos IEEE-754.

Parámetros	Formato			
	Simple	Simple Ex- tendida	Doble	Doble Ex- tendida
Tamaño del significando (bits)	24	$\geq 32$	53	$\geq 64$
Exponente máximo	+127	$\geq +1023$	+1023	$\geq +16383$
Exponente mínimo	-127	$\leq -1022$	-1022	$\leq -16382$
Tamaño del exponente (bits)	8	$\geq 11$	11	$\geq 15$
Ancho de formato (bits)	32	$\geq 43$	64	$\geq 79$

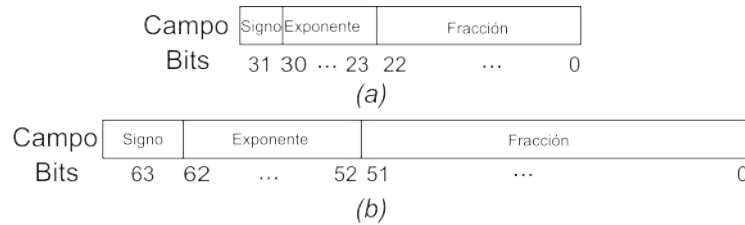


Figura 4.1: Formatos en punto flotante. (a) Precisión simple. (b) Precisión doble.

Existe un problema cuando un número presenta una magnitud menor que el número más pequeño normalizado que puede representarse en el sistema. Para solucionar este problema, anteriormente se adoptaban dos enfoques, el primero era igualar a cero el número y continuar, y el segundo era causar un subdesbordamiento. Sin embargo el IEEE inventó los números **desnormalizados**. Estos números tienen exponente cero y una fracción diferente de cero dada por los siguientes 23 o 53 bits siguientes. El bit implícito ahora se convierte en '0'. La diferencia entre los números desnormalizados y los normalizados, es el exponente, en los números normalizados el exponente es diferente de cero.

El esquema manejado por el IEEE, maneja dos ceros, positivo y negativo, determinados por el bit de signo. Ambos tienen un exponente y fracción de ceros. Además el bit implícito también es '0' en lugar de '1'. El caso especial infinito también es considerado por el estándar, el cual consiste en un exponente solamente de unos ( caso que no está permitido en los casos normalizados) y una fracción compuesta por puros ceros. Esta representación puede usarse como número y se comporta igual que las reglas matemáticas para infinito. Sin embargo existe otro caso especial, el resultado de dividir

Tabla 4.2: Casos especiales para formato en precisión simple del estándar IEEE-754.

Exponente en exceso	Mantisa	Valor
0	0	Cero
0	$\langle \rangle 0$	Número desnormalizado (0.+Mantisa) $\times 2^{-126}$
1... 254	Diferente de cero	(1.+Mantisa) $\times 2^{exp-127}$
255	0	Infinito
255	$\langle \rangle 0$	<i>Nan (Not a Number)</i>

infinito entre infinito. El resultado obviamente no está definido, y se maneja proporcionando otro formato especial, llamado **ningún número** (NaN, *not a number*), que también puede usarse como operando con resultados predecibles.

#### 4.1.3. Suma y Resta en punto flotante

Antes de hacer mención del algoritmo de suma y resta en punto flotante es necesario hacer las siguientes notaciones. El algoritmo hace la suma y/o resta de los números  $A_i$ , donde  $S_i$  y  $E_i$  representan el significando y el exponente de cada número respectivamente.  $S_i$  representa la mantisa con el bit explícito delante. La figura 4.2 muestra el diagrama de flujo del proceso de la suma o resta, el proceso de sumar dos números  $A_1$  y  $A_2$  consta de los siguientes pasos [40, 43]:

1. Si el exponente 1 es menor al exponente 2, cambiar los operadores para que la diferencia de estos sea mayor a cero, de manera que se satisfaga  $d = E_1 - E_2 \geq 0$ . Igualar el exponente menor ( $E_2$ ) al mayor ( $E_1$ ), para ello es necesario también desplazar a la derecha el significando ( $S_2$ ) tantas veces como sea la diferencia de los exponentes.
2. Sumar o restar los significandos  $S = S_1 \pm S_2$ .
3. Normalizar el resultado de la suma o resta desplazando a la derecha e incrementando el exponente mayor  $E_1$  o bien desplazando a la izquierda y decrementando el exponente. En cada desplazamiento checar si existe un desbordamiento a infinito (*overflow*) o desbordamiento a cero (*underflow*).
4. Redondear el significando  $S$ , con un número apropiado de bits, en caso de no quedar redondeado, el número está desnormalizado y se pasa al paso anterior.

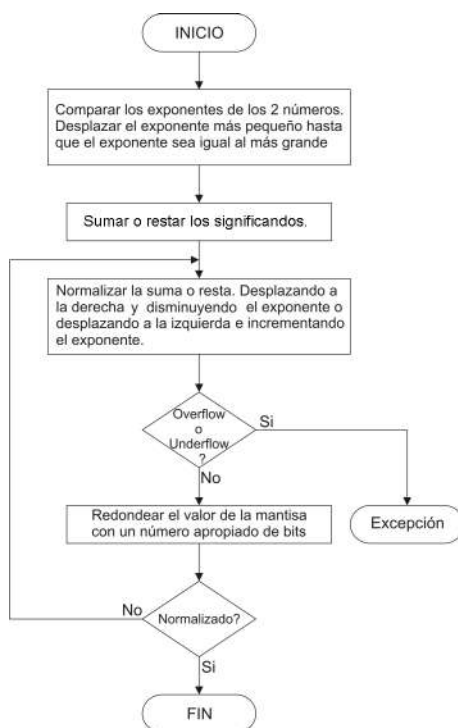


Figura 4.2: Diagrama de flujo de la suma y resta en punto flotante.

#### 4.1.4. Multiplicación en punto flotante

El proceso de multiplicar dos número  $A_1$  y  $A_2$  es más sencillo que el de suma y resta. La figura 4.3 muestra el diagrama de flujo del proceso de multiplicar dos números. Los pasos a seguir para el algoritmo de la multiplicación son los siguientes [40, 43]:

1. En primer lugar se checa si alguno de los operandos es 0, el resultado será 0.
2. Sumar los exponentes ( $d = E_1 + E_2$ ) en forma sesgada, y restar al resultado el sesgo. (En el caso de 32 bits, el sesgo es 127).
3. Multiplicar los significandos ( $S = S_1 * S_2$ ). En este caso la multiplicación se realiza como en el caso de los enteros. El resultado, tendrá el doble de tamaño en bits que el multiplicando o el multiplicador. Sin embargo al momento de redondear, los bits excesivos se perderán.
4. Normalizar el resultado de la multiplicación desplazando a la derecha e incrementando el exponente  $d$ , o bien desplazando a la izquierda y decrementando el exponente. En cada desplazamiento checar si exis-



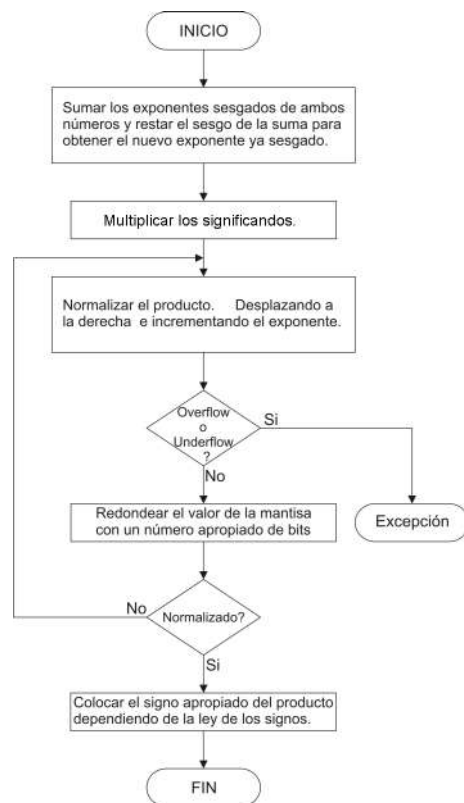


Figura 4.3: Diagrama de flujo de la multiplicación en punto flotante.

te un desbordamiento a infinito (*overflow*) o desbordamiento a cero (*underflow*).

5. Redondear la mantisa con un número apropiado de bits, en caso de no quedar redondeado, el número está desnormalizado y se pasa al paso anterior.
6. Por último es necesario colocar el signo correcto. Si ambos números son positivos o negativos, el resultado es positivo, y el bit de signo para el resultado es 0. Si los números tienen signos contrarios, es decir, ambos números tienen signos diferentes, el resultado es negativo, y por lo tanto el bit a colocar en el resultado es 1.

## 4.2. VHDL

VHDL (*VHSIC, Hardware Description Language*) es un lenguaje descriptor de Hardware (*HDL, Hardware Descriptor Language*) patrocinado por el

Departamento de Defensa de los Estados Unidos dentro del programa VHSIC (*Very High Speed Integrated Circuits*). En diciembre de 1987 fue aceptado como estándar IEEE bajo la referencia 1076-87. A partir de esa fecha empieza a consolidarse como estándar internacional. En 1993, surgió una nueva versión del estándar que incluía nuevas características no disponibles en la versión de 1987, [44]. VHDL es un lenguaje de descripción y modelado, diseñado con el fin de que tanto los humanos como las máquinas puedan entender la funcionalidad y organización de sistemas hardware digitales [45].

VHDL es un lenguaje flexible y con una amplia sintaxis que permite el modelado estructural, en flujo de datos o de comportamiento hardware. Entre las características más importantes que presenta son [46]:

1. Lenguaje de simulación de modelos, con lo que puede ser utilizado para construir circuitos muy extensos usando esquemáticos o en texto, con el fin de simularlos.
2. Lenguaje de entrada de diseño, debido a que es un lenguaje de alto nivel permite el diseño como programas de computadoras.
3. Puede ser utilizado como lenguaje *Netlist* de esta forma suele ser útil como manera de comunicación a bajo nivel entre herramientas de diseño basadas en computadoras.
4. Lenguaje estándar, que permite la portabilidad hacia diferentes arquitecturas o herramientas de diseño más recientes.

Entre las ventajas que se tienen de utilizar el lenguaje VHDL para la descripción de hardware se tienen [45]:

- Permite diseñar, modelar y comprobar (en simulación) cualquier sistema digital desde un nivel de abstracción alto hasta un nivel de abstracción de compuertas y biestables.
- Debido a que los módulos en VHDL pueden ser usados en diferentes diseños, es factible la reutilización de código. Además esa misma descripción puede ser utilizada en diferentes tecnologías sin la necesidad de rediseñar todo el circuito.
- Los errores de comunicación y compatibilidad son minimizados debido a que está basado en el estándar IEEE 1076-1987 y IEEE 1076-1993. Además cualquier usuario puede diseñar alguna aplicación en VHDL y comercializarla sin problemas al no ser un lenguaje de propietario.
- Permite el diseño *Top-Down*, es decir, permite la descripción y modelado del comportamiento de un circuito a un alto nivel.
- VHDL permite la modularidad, esto es, permite dividir o descomponer los diseños hardware y su descripción en unidades más pequeñas.

### 4.2.1. Descripción de circuitos en VHDL

En VHDL existen tres formas de describir un circuito, dependiendo del nivel de abstracción que se maneje, estos son: comportamental, flujo de datos y estructural.

El diseño comportamental es el más alto nivel de abstracción que permite VHDL. Cuando se ocupa esta forma de describir un circuito, el diseñador describe solamente el comportamiento del sistema sin importarle los componentes ocupados para llevar a cabo el diseño. Utilizando este nivel de abstracción se pueden describir las operaciones de un circuito a nivel de registros. La forma de diseñar en este nivel de abstracción es muy similar a un lenguaje de alto nivel, en el que se escriben programas que operan de manera secuencial y se comunican mediante sus interfaces [46, 47].

El diseño mediante flujo de datos es un nivel intermedio en el que se indica la forma en que los datos se pueden transferir de una señal a otra sin necesidad de utilizar declaraciones secuenciales [48]. Comúnmente es denominada Transferencia Lógica de Registros (RTL, *Register Transfer Logic*). En este tipo de descripción se pueden utilizar ecuaciones lógicas o sentencias de asignación, permitiendo que la lógica combinacional sea simplificada, mientras que las partes más importantes de un circuito, los registros, son especificados con mas detalle de acuerdo al circuito a modelar.

El tercer nivel de abstracción es el nivel estructural, o también llamado nivel lógico. Basa su comportamiento en términos de sus modelos lógicos establecidos (compuertas, sumadores, etc.) [48]. Se especifican los bloques que componen un circuito y sus interconexiones. Cada bloque ya debe contar con su descripción de tal forma que se construya una jerarquía de descripciones donde las inferiores dan lugar a un sistema de mayor complejidad.

### 4.2.2. Entidad y Arquitectura en VHDL

En un diseño VHDL existe forzosamente una entidad y al menos una arquitectura. Una entidad identifica claramente las entradas y salidas del diseño, es análogo a un símbolo en un esquemático, sirve para definir la visión externa de un diseño, es decir, la interfaz con su entorno. En cambio la arquitectura complementa el diseño de un sistema, la cual describe el comportamiento o estructura de la entidad que describe.

La entidad es una abstracción de un circuito, la cual puede verse como una “caja negra”, de la cual únicamente se conocen las entradas y las salidas. Un ejemplo de una entidad de un sumador puede verse en la figura 4.4a. Una arquitectura es el complemento de una entidad, es decir describe el comportamiento de un sistema. Describe el conjunto de operaciones que se realizan con las entradas y que definen a las salidas, modelando de esta manera el funcionamiento del circuito diseñado. La figura 4.4b muestra la estructura interna del sumador a nivel de compuertas lógicas.

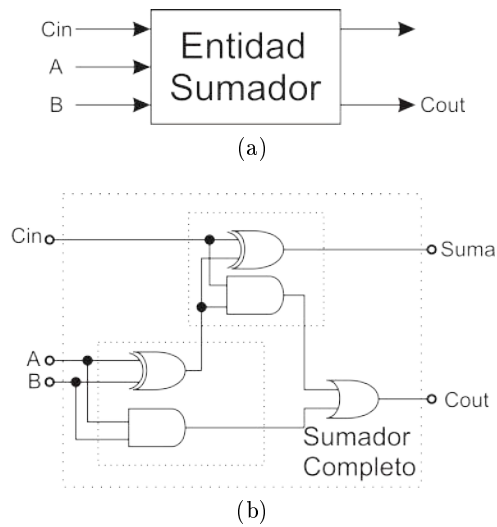


Figura 4.4: Dupla Entidad-Arquitectura (a) Símbolo funcional de una entidad. (b) Arquitectura de un sumador.

### 4.2.3. Metodología de diseño

Existen diversas metodologías de diseño de circuitos o sistemas digitales. Estas metodologías suelen ser independientes de las herramientas de Automatización del Diseño Electrónico<sup>1</sup> (*EDA, Electronic Design Automation*) y del HDL utilizado. Entre las metodologías más conocidas se encuentran la metodología Ascendente (*Bottom-Up*) y la metodología Descendente (*Top-Down*) [45, 46, 49].

#### 4.2.3.1. Diseño *Bottom-Up*

El diseño *Bottom-Up* (de abajo hacia arriba) de un circuito o sistema electrónico comienza con la descripción de los componentes más básicos, generalmente las primitivas, que se agrupan en diversos módulos, y estos a su vez se agrupan para formar módulos de mayor complejidad y así sucesivamente para formar un solo bloque que describa completamente al sistema [45, 49]. Esta metodología no implica una estructuración jerárquica de los elementos que conforman al sistema, debido a que la estructuración se realiza después de la descripción del sistema, resultando no necesaria. Esta metodología hace uso de los componentes más básicos (de menor nivel), que generalmente se encuentran en una biblioteca, conteniendo chips, resistencias, capacitores y otras unidades funcionales.

<sup>1</sup>Conjunto de herramientas de hardware y software que ayudan en el proceso de diseño de sistemas electrónicos.

Por la naturaleza de esta metodología, el diseño de circuitos utilizando esta metodología resulta ser ineficiente, por lo que generalmente no se usa, y más en la actualidad donde los diseños son bastante grandes, uniendo miles de componentes, es fácil cometer errores y por ende, hacer que el sistema no funcione adecuadamente; además el análisis de sistemas grandes resulta ser complejo, lo que resulta ser poco útil a la hora de detectar errores en el sistema.

#### 4.2.3.2. Diseño *Top-Down*

La metodología *Top-Down* (de arriba hacia abajo) es un proceso contrario a la metodología anterior. Inicia visualizando al sistema a diseñar con un nivel de abstracción alto y posteriormente desglosarlo en módulos jerárquicamente inferiores, los cuales pueden ser también divididos; los niveles de detalle de los módulos de menor jerarquía dependen directamente de la herramienta de diseño [45, 49].

Con el avance de las herramientas de diseño electrónico, es posible en la actualidad contar con herramientas que realicen de forma automática la metodología de diseño descendente; las herramientas de síntesis permiten la implementación automática de un diseño a partir de la idea abstracta del diseñador, sin necesidad de que el diseñador descomponga la idea abstracta.

Entre las ventajas que presenta esta metodología están:

- Incrementa la productividad del diseño, permite especificar un diseño en un nivel de abstracción alto sin la necesidad de implementar el mismo a un nivel lógico bajo, como el uso de compuertas.
- Incrementa la reutilización de código, en el diseño se utilizan tecnologías genéricas, definiéndose posteriormente la tecnología a utilizar. Esto permite la reutilización de código entre diferentes tecnologías.
- Detección rápida de errores, al centrarse más en la definición y diseño del sistema, los errores son más fáciles de detectar.

### 4.3. Arreglo de Compuertas Programables en Campo *FPGA*

En el año de 1984, la compañía Xilinx introdujo al mercado los *FPGAs* (*Field Programmable Gate Array*) como una evolución de los *CPLDs*, además de una idea nueva resultante de combinar el control de usuario y el tiempo de lanzamiento al mercado de los *PLDs* con la densidad y ventaja de los arreglos de compuertas, surgiendo así el *FPGA*. Un *FPGA* es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad se puede configurar. Puede ser usado para implementar casi cualquier

diseño hardware, desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

Tanto los CPLDs como las FPGAs contienen un gran número de elementos lógicos programables. Si medimos la densidad de los elementos lógicos programables en puertas lógicas equivalentes (número de puertas NAND equivalentes que podríamos programar en un dispositivo) podríamos decir que en un CPLD hallaríamos del orden de decenas de miles de puertas lógicas equivalentes y en un FPGA del orden de cientos de miles hasta millones de ellas. Aparte de las diferencias en densidad entre ambos tipos de dispositivos, la diferencia fundamental entre las FPGAs y los CPLDs es su arquitectura. El paradigma de los CPLDs se basa en que se puede implementar cualquier función lógica mediante suma de productos, mientras que un FPGA hace uso de Tablas de búsqueda. La enorme libertad disponible en la interconexión de dichos bloques confiere a las FPGAs una gran flexibilidad.

Otra diferencia importante entre FPGAs y CPLDs es que en la mayoría de las FPGAs se pueden encontrar funciones de alto nivel (como sumadores y multiplicadores) embebidas en la propia matriz de interconexiones, así como bloques de memoria.

#### 4.3.1. Arquitectura de un FPGA

La arquitectura básica de un FPGA consiste de una matriz de bloques lógicos que incluyen flip-flops con formas para que el usuario configure a) la función en cada bloque lógico, b) las entradas y salidas y, c) la interconexión entre los bloques (ver figura 4.5). Cada familia de FPGA difiere una de otra en la estructura y funcionalidad de los bloques lógicos, el sistema de interconexión, y los recursos de arquitectura fija de aplicación específica que incluye.

**Bloques lógicos configurables.** Estos bloques lógicos configurables CLBs (Configurable Logic Blocks) están distribuidos en forma matricial en el dispositivo y son recursos que permiten al usuario implementar funciones lógicas. Cuando estos dispositivos son de complejidad baja, es decir, las funciones que se pueden implementar son sencillas y además, existe un número considerable de ellos, se dice que el FPGA es de granularidad fina. En cambio, cuando los recursos lógicos están formados por memorias de acceso aleatorio llamadas Tablas de Búsqueda (LUT, Look-up Tables), flip-flops para almacenamiento de elementos que dependen de la señal de reloj, multiplexores que permiten la selección, reset y puesta a uno lógico de elementos, se dice que el FPGA es de granularidad gruesa; dentro de un FPGA este tipo de CLBs son reducidos pero tienen la característica de poder implementar funciones con mayor complejidad.

**Bloques configurables de entrada/salida.** La matriz de CLBs está

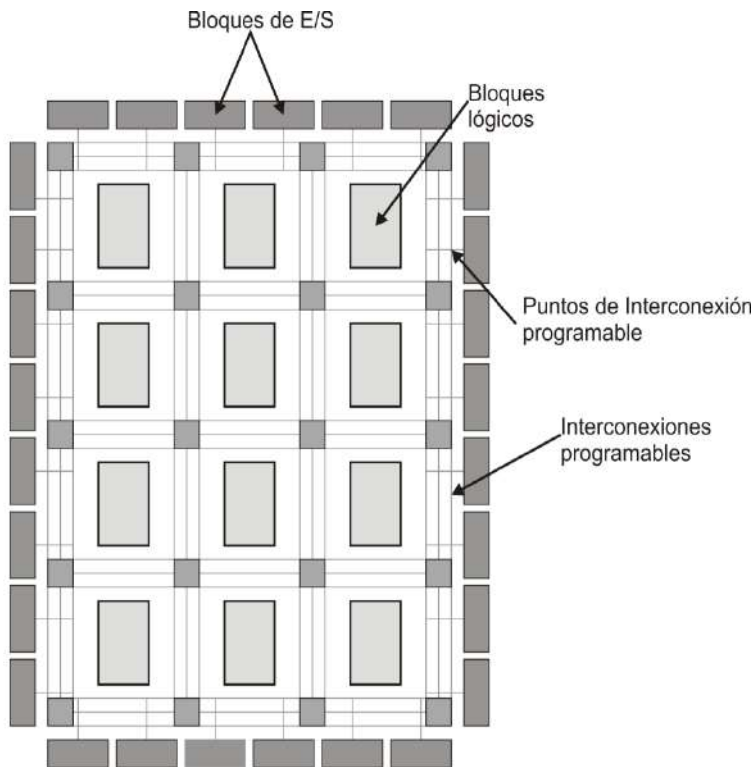


Figura 4.5: Arquitectura de un FPGA

rodeada de por un anillo de bloques de interfaz, denominada bloques configurables de entrada/salida. Dichos bloques tiene la tarea de proporcionar conectividad al FPGA y su ambiente, es decir, controlan las entradas y salidas de datos entre los pines de entrada/salida y la lógica interna. Cada bloque es bidireccional y soporta operaciones de tercer estado, para conseguir estas características un bloque está formado de flip-flops, latches y buffers que soportan el tercer estado. En ocasiones, suelen incluir resistores pull-up y/o pull-down en la salida. La polaridad de la señal de salida es programable.

**Interconexiones programables.** Están formados por: recursos de interconexión, conjunto de líneas y/o interruptores programables que permiten transmitir las señales entre los bloques lógicos internos y entre estos y los bloques de entrada/salida, y de matriz de interconexión, elementos lógicos que facilitan la comunicación entre los buses de comunicación (recursos de interconexión).

**Recursos de Arquitectura fija de aplicación específica.** En la actualidad muchos FPGAs incluyen en su arquitectura recursos de propósito específico. Permiten configurar una función óptimas dentro del FPGA, tal como módulos RAM, sumadores, multiplicadores, etc. Mediante estos bloques

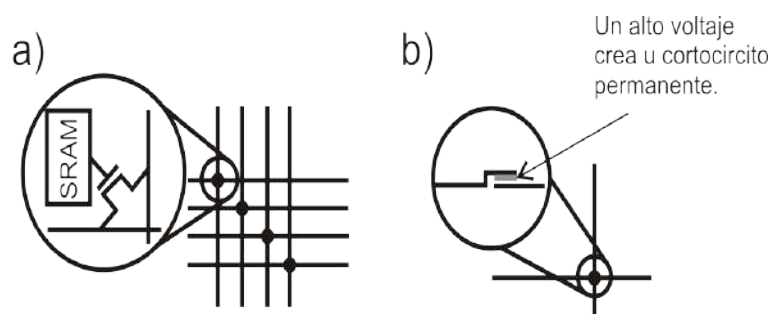


Figura 4.6: Métodos de programación de un FPGA: a) Conexiones SRAM y b) antifusible.

de aplicación específica, el diseño de un sistema digital se acelera.

#### 4.3.2. Métodos de programación

Existen tres tipos principales de programación [50]:

**Basado en SRAM.** Las conexiones en el FPGA se consiguen mediante el uso de transistores, compuertas de transmisión, o multiplexores que son controlados por células SRAM (Figura 4.6). Esta tecnología permite rapidez en la reconfiguración del circuito. Las principales desventajas que presenta es el tamaño del chip, requerido por la tecnología RAM, y las necesidades de recursos externos (usualmente chips de memoria no volátil) para cargar la configuración. El FPGA puede ser programado un número ilimitado de veces.

**Tecnología antifusible.** Un antifusible permanece en un estado de alta impedancia, hasta que se programa en una de baja impedancia o estado "fundido" (Figura 4.6). Mediante esta tecnología, los dispositivos se pueden programar solo una vez; y es menos cara que la tecnología RAM.

**Tecnología Flash/EEPROM.** Una alternativa a la metodología SRAM es el uso de memorias flash o EEPROM. Estas memorias son no-volátiles, por lo que no pierden su información aún cuando el sistema esté apagado. Esta característica elimina la necesidad de elementos externos requeridos para almacenar y cargar la configuración del FPGA.

#### 4.3.3. Ventajas de los FPGAs

Entre las ventajas que ofrecen los FPGAs se encuentran:

- Reducción de tamaño, peso y disipación de potencia.
- Mayor desempeño.
- Mejor seguridad contra copias no autorizadas.



- 
- Dispositivos y costos de inventarios reducidos.
  - Costos de tarjetas de prueba reducidos.
  - Una reducción considerable en los tiempos de desarrollo (prototipado rápido).
  - Reconfigurabilidad del circuito aún en sistema (ISP).
  - Costos bajos de ingeniería no-recurrente, resultando en diseños más económicos para soluciones que requieren de menos de 1000 unidades.



## Capítulo 5

# Plataforma Experimental y Resultados

**RESUMEN:** En este capítulo se presenta el diseño e implementación del sistema convertidor CD–CD tipo reductor, el cual consiste de la implementación de un convertidor reductor en el modo de conducción continuo a 1.44W y un controlador GPI implementado en FPGA. Finalmente, se muestran los resultados del desempeño del controlador en lazo abierto y en lazo cerrado, obtenidos a nivel simulación mediante MatLab/Simulink y PSIM, y a nivel experimental a través de la plataforma.

### 5.1. Introducción

En el presente capítulo se describe el diseño e implementación del sistema convertidor CD–CD tipo reductor, así como de la herramienta que permitirá desarrollar el controlador GPI dentro del dispositivo FPGA. El proceso de diseño e implementación del convertidor reductor y el control GPI basado en FPGA, está dividido en las siguientes etapas:

1. Diseño del circuito convertidor reductor.
  2. Modelado de unidades de operaciones aritméticas en punto flotante.
  3. Diseño, simulación e implementación del controlador GPI en VHDL.
- **Etap 1: Diseño del circuito convertidor reductor.** En esta etapa se diseña el circuito convertidor reductor, conforme a las características obtenidas del diseño del convertidor en el modo de conducción continuo del capítulo 2. Para diseñar el circuito físicamente es necesario tomar

en cuenta un circuito adicional que se encargará de activar y desactivar al interruptor, este circuito se conoce como circuito excitador de compuerta.

- **Etapa 2: Modelado de unidades de operaciones aritméticas en punto flotante.** Siguiendo una metodología descendente se realiza el diseño y modelado de unidades aritméticas en punto flotante de 32 bits. Estas unidades aritméticas son: unidad de suma y unidad de multiplicación. A partir de estas unidades se diseñará una unidad de integración en punto flotante.
- **Etapa 3. Diseño, simulación e implementación del controlador GPI en VHDL.** Conforme a las unidades diseñadas con anterioridad, y al diseño del controlador GPI del capítulo 3, se diseña el controlador GPI en lenguaje VHDL. Para su implementación física de este controlador, es necesario diseñar una unidad PWM en VHDL, la cual consta básicamente de un circuito generador de onda triangular y un comparador de señales del tipo analógico. Se presentan las simulaciones del diseño en VHDL mediante las herramientas MATLAB, Psim y ModelSim. Posteriormente, se implementa el control GPI basado en FPGA.

## 5.2. Circuito convertidor reductor

El convertidor reductor puede transformar un voltaje de entrada  $E$  en un voltaje de salida  $V_o$  de menor amplitud, conservando la misma polaridad. Esto se logra mediante la conmutación del interruptor (Transistor MOSFET), a través de una señal PWM cuya función es controlar el ciclo de trabajo  $D$  del transistor, lo que permite obtener a la salida diferentes amplitudes de voltaje.

Para el diseño del convertidor reductor, es necesario obtener los valores de la inductancia ( $L$ ), la capacitancia ( $C$ ), a partir de características propuestas como el voltaje de entrada, el voltaje de salida, la frecuencia de conmutación, la potencia de salida, y el porcentaje de rizo de voltaje de salida.

De acuerdo al diseño propuesto, el voltaje de entrada  $E$  es de  $24V$ . El voltaje de salida  $V_o$ , deseado para el convertidor es  $12V$ . El porcentaje de rizo  $\Delta V_o/V_o = 0,006\%$ , mientras que la frecuencia fija de conmutación se propone de  $f = 48KHz$ .

Para el diseño del convertidor con las características mencionadas anteriormente, se necesita el cálculo de los valores de R, L y C. El ciclo útil de trabajo ( $D$ ) se calcula mediante (2.17).

$$D = \frac{V_o}{E} = \frac{12V}{24V} = 0,5$$

Mediante el voltaje de salida y la potencia, se obtiene la corriente de salida, dada como:

$$I_o = \frac{P_o}{V_o} = \frac{1,44W}{12V} = 0,12A$$

Mientras la resistencia de carga se calcula a partir del voltaje y corriente de salida, y es:

$$R = \frac{V_o}{I_o} = \frac{12V}{0,12A} = 100\Omega$$

El cálculo de la inductancia crítica en el modo de conducción continuo (MCC) se hace a través de (2.29), y está es dada por:

$$L_{critico} = \frac{1-D}{2}TR = \left(\frac{1-0,75}{2}\right)\frac{1}{48E^3}100 = 520\mu H$$

De este último valor, se selecciona un valor de 1.92 veces mayor al valor crítico, esto para asegurar que el convertidor trabaje en el MCC, y se tenga un error de corriente más pequeño. Por lo que el valor de L es:

$$L = 2L_{critico} \approx 1mH$$

Una vez obtenido el valor de la inductancia, se calcula los valores máximo y mínimo de la corriente a través de (2.22) y (2.23), y están dadas como:

$$I_{Lmax} = DE\left(\frac{1}{R} + \frac{(1-D)T}{2L}\right) = (0,5)(24)\left(\frac{1}{100} + 2,60 * 10^{-8}\right) = 0,1825A$$

$$I_{Lmin} = DE\left(\frac{1}{R} - \frac{(1-D)T}{2L}\right) = (0,5)(24)\left(\frac{1}{100} - 2,60 * 10^{-8}\right) = 0,0575A$$

El voltaje de rizo se calcula a través de (2.46), y está dado por:

$$\frac{\Delta v_c}{V_o} = \frac{(1-D)}{(8LCf^2)} = 0,000060$$

Despejando la variable  $C$ , se tiene:

$$C = \frac{1-D}{8Lf^2(6,0 * 10^{-5})}$$

$$C \approx 470\mu F$$

La figura 5.1 y figura 5.2 muestran las respuestas en simulación obtenidas de corriente y voltaje en la carga de salida del convertidor en lazo abierto.

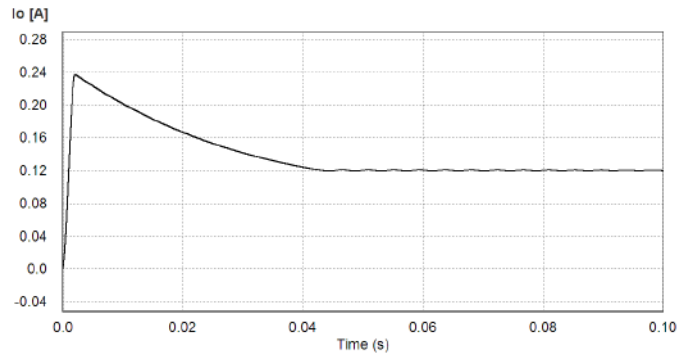


Figura 5.1: Respuesta en lazo abierto de la corriente en la carga del convertidor.

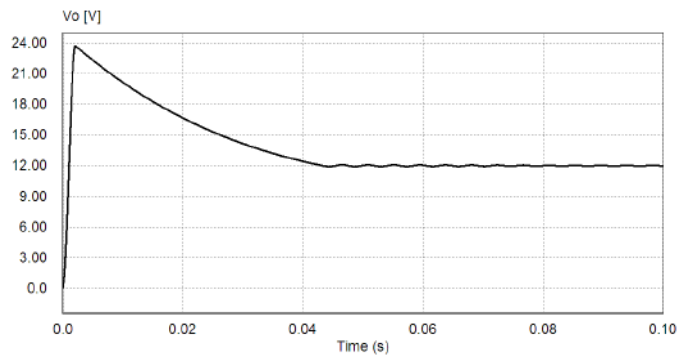


Figura 5.2: Respuesta en lazo abierto del voltaje de salida del convertidor, para una entrada de control  $u_{av} = 0,5$ .

La respuesta del voltaje de salida de la figura 5.2, muestra una disminución del 50 % con respecto al voltaje de entrada de alimentación del convertidor.

En la figura 5.3 se muestra el rizo de corriente que existe en el inductor cuando el convertidor se encuentra en estado estable. Se corrobora que se adecua a los cálculos obtenidos previamente. Así mismo, la figura 5.4 muestra el voltaje de rizo presente en el capacitor del convertidor. En ella se observa un voltaje máximo de 12.0004 V y un voltaje mínimo de 11.9996 V, correspondiente al voltaje de rizo propuesto en la presente sección.

### 5.2.1. Semiconductores en el circuito reductor

En la selección de componentes a utilizar está claro que al momento de hacerlo se debe tener cuidado con las especificaciones de cada elemento,

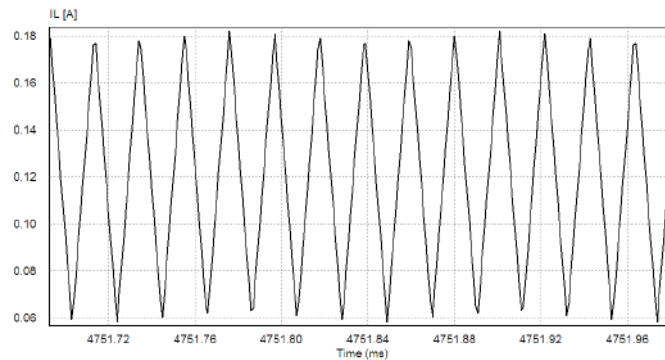


Figura 5.3: Corriente de rizo en el inductor.

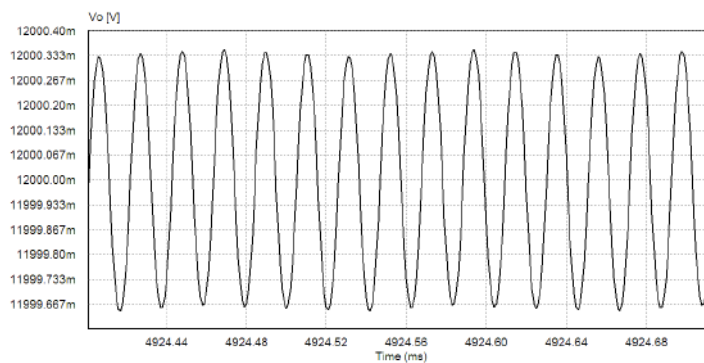


Figura 5.4: Voltaje de rizo en el capacitor.

para obtener la mayor eficiencia posible en el convertidor, evitando así las pérdidas que existen en la conmutación. Por ello se elige un transistor MOSFET (IRF640) de baja resistencia de encendido  $R_{DSon} = 0,18\Omega$ , además de conducir una corriente de 3,5A.

El transistor MOSFET necesita ser activado mediante un voltaje positivo aplicado en su compuerta  $V_{GS} > 0$ , así se tiene que el voltaje del MOSFET de su compuerta a tierra debe ser mayor que el voltaje de entrada, para que el MOSFET esté en estado de conducción. Para activarlo, se utiliza un dispositivo llamado impulsor de compuerta, con el cual se consiguen los niveles adecuados para activar la compuerta del MOSFET, además con este dispositivo realiza disparos de subida y bajada en tiempos cortos de conmutación. El dispositivo utilizado es el circuito integrado IR2117.

La selección del diodo de libre circulación (*free-wheeling diode*) del convertidor reductor,  $D_{fw}$ , corresponde al diodo 1N5817, el cual es un diodo *Schottky* con capacidad de corriente de 1 A y un voltaje de polarización pequeño ( $V_f = 0,45V$ ).

### 5.2.2. Circuito impulsor de compuerta

La configuración utilizada para el circuito impulsor de compuerta se muestra en la figura 5.5. Los elementos que son indispensables para el funcionamiento del impulsor de compuerta son: el diodo  $D_{bs}$ , el capacitor  $C_{bs}$ , y la resistencia de compuerta  $R_g$ . Estos elementos son configurados a partir de las características de la aplicación.

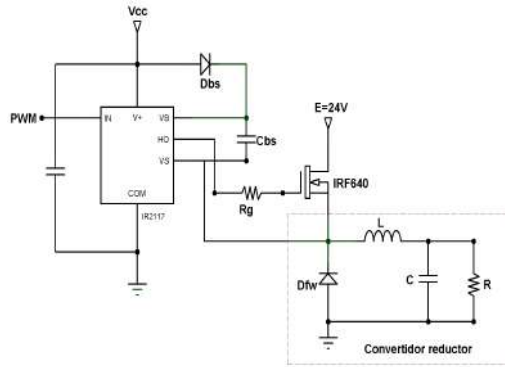


Figura 5.5: Configuración del circuito impulsor de compuerta y convertidor reductor.

#### 5.2.2.1. Circuito *bootstrap*

El driver que impulsa el voltaje de compuerta del MOSFET requiere para su funcionamiento un circuito conocido como *bootstrap* conformado por un capacitor  $C_{bs}$  y un diodo  $D_{bs}$  conectado de tal forma como se observa en la figura 5.5.

Para obtener el valor del capacitor *bootstrap*, el primer paso es obtener la caída de voltaje mínima ( $\Delta V_{BS}$ ) que se debe tener para garantizar que el MOSFET esté activado. (ON).

Si  $V_{GE_{min}}$  es el voltaje de compuerta-emisor mínimo, para mantener encendido al MOSFET, entonces la caída de voltaje es:

$$\Delta V_{BS} = V_{cc} - V_F - V_{GE_{min}} - V_L \quad (5.1)$$

bajo la condición:

$$V_{GE_{min}} > V_{BSUV-}$$

Donde  $V_{cc}$  es el voltaje de alimentación del driver,  $V_F$  es el voltaje de recuperación del diodo, y  $V_{BSUV-}$  es la caída de voltaje negativo de la fuente del lado alto.



El valor mínimo del capacitor bootstrap es [51]:

$$\begin{aligned} C_{bsmin} &\geq \frac{Q_{tot}}{\Delta V_{BS}} \\ Q_{tot} &= Q_g + Q_{ls} + (I_{GSIk} + I_{QBS} + I_{Dlk} + I_{Clk} + I_{DS-})t_{hon} \end{aligned} \quad (5.2)$$

donde

- $Q_g$  es la carga de encendido requerida por el MOSFET.
- $I_{GSIk}$  es la corriente de fuga de la compuerta-fuente del MOSFET.
- $I_{QBS}$  es la corriente de reposo (driver).
- $I_{Dlk}$  es la corriente de fuga (driver).
- $I_{Clk}^1$  es la corriente de fuga del diodo *bootstrap*.
- $I_{DS-}$  es la corriente de desaturación de entrada (driver).
- $t_{hon}$  es el tiempo de encendido del MOSFET.
- $Q_{ls}$  es la carga requerida por los elevadores de nivel internos (driver).

El tiempo de encendido  $t_{hon}$  se obtiene a partir de multiplicar el ciclo de trabajo  $D$  por la frecuencia de conmutación, como sigue:

$$t_{hon} = DT = D\left(\frac{1}{f}\right)$$

La tabla 5.1 resume los valores de las ecuaciones (5.2), así como el valor del capacitor obtenido para el circuito *bootstrap*.

El diodo  $D_{bs}$  para el circuito *bootstrap* necesita tener un tiempo de recuperación inverso menor al tiempo de propagación del driver IR2117, por lo que se propone que el tiempo de recuperación inverso debe cumplir que,  $t_{rr} < 100ns$ . Además este diodo debe soportar un voltaje de polarización inverso  $V_r > E$ , por lo que se propone el diodo 1N4148 que tiene un tiempo de recuperación inverso  $t_{rr} = 4ns$  y un voltaje inverso de 75 V [51, 52].

---

<sup>1</sup>  $I_{Clk}$  es solo relevante si se utilizan capacitores electrolíticos y puede ser ignorado si se utilizan otro tipo de capacitores [52].

Tabla 5.1: Parámetros utilizados para el cálculo del capacitor *bootstrap*.

Símbolo	Parámetro	Valor	Unidad
$Q_q$	Carga de encendido (Mosfet)	72	
$Q_{Is}$	Carga requerida por los elevadores de nivel internos (Mosfet)	5	nC
$I_{GSIk}$	Corriente pulsada de corto circuito en la salida alta (driver)	0.1	
$I_{QBS}$	Corriente de reposo (driver)	55	
$I_{DIk}$	Corriente de fuga (driver)	50	$\mu A$
$I_{QBS}$	Corriente de fuga del diodo <i>bootstrap</i>	50	
$I_{QBS}$	Corriente de saturación de entrada (driver)	0	
$t_{hon}$	Tiempo de encendido del Mosfet	10.42	$\mu s$
$V_{cc}$	Voltaje de alimentación del driver	15	
$V_F$	Voltaje de recuperación del diodo	1	
$V_{BSUV-}$	Caída de voltaje negativo de la fuente del lado alto	9	V
$V_{GEmin}$	Voltaje de compuerta-emisor mínimo $> V_{BSUV-}$	9.2	
$V_L$	Voltaje en la carga del diodo $D_{fw}$	-0.45	
$Cbs_{min}$	Valor mínimo para el capacitor <i>bootstrap</i>	14.97	
$Cbs$	Valor del capacitor <i>bootstrap</i>	220	nF

### 5.2.2.2. Resistencia de compuerta

Para poder calcular la resistencia de compuerta en el circuito impulsor de compuerta, se necesitan las ecuaciones dadas en [53].

$$I_{avg} = \frac{Q_{gd} + Q_{gs}}{t_{sw}} \quad (5.3)$$

y

$$R_g = \frac{V_{cc} - V_{gs*}}{I_{avg}} - \frac{V_{cc}}{I_{O+}} \quad (5.4)$$

Donde

- $Q_{gd}$  es la carga de compuerta-drenador que existe en el Mosfet.
- $Q_{gs}$  es la carga de compuerta-fuente que existe en el Mosfet.
- $I_{O+}$  es la corriente pulsante de corto circuito en la salida alta del driver.
- $t_{sw}$  es el tiempo de conmutación propuesto del driver.
- $V_{cc}$  es el voltaje de alimentación del driver.
- $V_{gs*}$  es el voltaje de estabilización del mosfet.
- $R_g$  es la resistencia de compuerta.

El voltaje de estabilización del mosfet, se obtiene a partir de la hoja de datos dada por el fabricante [54], mediante la curva característica de la gráfica de voltaje compuerta-fuente contra carga de compuerta ( Ver figura 5.6). Mientras que el tiempo de conmutación propuesto  $t_{sw}$  es de  $200ns$ , que es el tiempo de propagación máximo del driver IR2117.

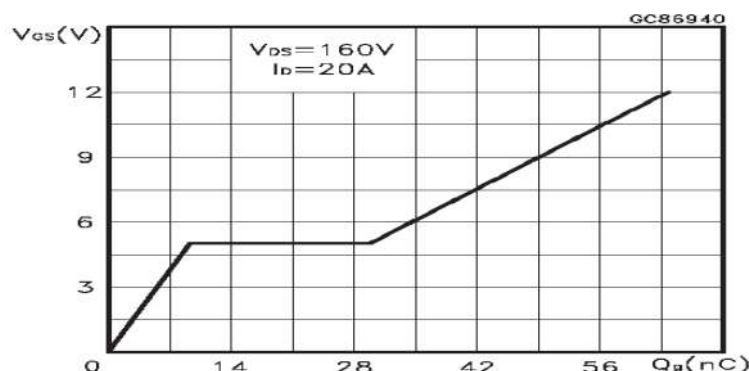


Figura 5.6: Voltaje de estabilización compuerta-fuente del Mosfet IRF640.

Tabla 5.2: Parámetros utilizados para el cálculo de la resistencia de compuerta.

Símbolo	Parámetro	Valor	Unidad
$Q_{gd}$	Carga de compuerta-drenador (Mosfet)	21	nC
$Q_{gs}$	Carga de compuerta-fuente (Mosfet)	10	nC
$I_{O+}$	Corriente pulsada de corto circuito en la salida alta (driver)	250	mA
$t_{sw}$	Tiempo de conmutación propuesto	200	ns
$V_{cc}$	Voltaje de alimentación del driver	15	V
$V_{gs*}$	Voltaje de estabilización (Plateau voltage) (Mosfet)	5	V
$R_g$	Resistencia de compuerta (Mosfet)	$\approx 4,7$	$\Omega$

La tabla 5.2 muestra el resumen de los parámetros obtenidos de las hojas de datos del Mosfet y del driver utilizados, así como el valor final de la resistencia de compuerta obtenido y adecuado a valores comerciales.

### 5.3. Sistema de control GPI

Con la finalidad de realizar las operaciones necesarias para el diseño del controlador GPI que regulará el voltaje de salida del convertidor reductor, se diseñarán unidades aritméticas, específicamente, unidades de aritmética en punto flotante de 32 bits, con el fin de abarcar un amplio rango de números para que se puedan realizar las operaciones necesarias.

Las computadoras y la mayoría de los DSPs cuentan ya, con operaciones de suma, resta, multiplicación y división, en punto flotante, las cuales se realizan en tiempos muy pequeños, sin embargo en el caso de las computadoras hacen uso del procesador que tienen, el cual posee altas velocidades de procesamiento. Sin embargo, los tiempos de los procesadores son compartidos con otras aplicaciones de software, por lo que el procesamiento no es una aplicación en tiempo real, y además no es la aplicación final.

Otra opción que se ha venido utilizando en el campo del control automático es el modelado de procesadores de aplicación específica en un circuito

digital configurable (CDC). Un procesador de aplicación específica es aquel procesador que se diseña considerando la aplicación o el área de la ingeniería en donde será utilizado. Esta opción presenta las siguientes ventajas: 1. Optimización de recursos, permite modelar únicamente las funciones necesarias en la aplicación. 2. Modelado de arquitecturas paralelas, repercutiendo en altas velocidades de procesamiento. 3. Uso de HDLs estandarizados, lo cual genera diseños portables entre diversas tecnologías, la portabilidad se extiende hacia otras herramientas CAD (*Computer-Aided Design*, diseño asistido por computadora) compatibles.

En esta sección se describe el modelado del controlador basado en FPGA, de la estructura de las unidades aritméticas de punto flotante, que incluyen las unidades de suma, multiplicación, y a partir de estas unidades principales, se diseña una unidad de integración numérica. Además, en esta sección se incluye la descripción de unidades que nos permitirán comunicarnos con el convertidor analógico digital, y por último la unidad del modulador por ancho de pulsos. En la figura 5.7 se observa a detalle, las unidades que conforman esta aplicación específica del controlador GPI, las cuales se describen a continuación.

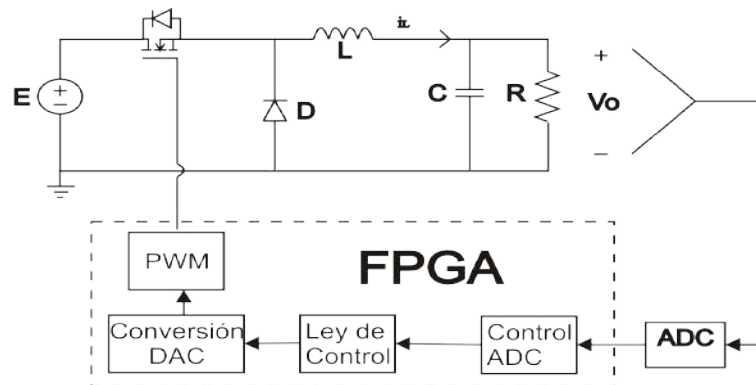


Figura 5.7: Unidades modeladas para el controlador GPI basado en FPGA.

- **Módulo de control del ADC.** Este módulo es el encargado de activar y desactivar las señales que permitirán activar al ADC, así como efectuar el inicio de las operaciones aritméticas para obtener una salida de la ley de control GPI. También tendrá a su cargo la activación de los módulos del convertidor digital analógico.
- **Módulo del control GPI.** El sistema hardware para la ley de control GPI es conformado por operaciones de sumas, multiplicaciones, integraciones en punto flotante. Para tal propósito se diseñan en primer instancia las operaciones de suma y multiplicación en punto flotante. A partir de estas operaciones primarias, se diseña la unidad de integración.

- **Módulo del convertidor punto flotante a punto fijo.** La función de este módulo es el convertir las magnitudes generadas por el control GPI, a unidades en punto fijo y adecuarlas a un nivel apropiado para poder hacer su comparación en el módulo del PWM.
- **Módulo del PWM.** Este módulo es el encargado de generar la señal PWM. Está conformado por dos bloques principales, el primero es el generador de una onda triangular a una frecuencia fija de 48 kHz que servirá como señal portadora. El segundo bloque es el de comparación que se encargará de comparar la señal proveniente del controlador GPI con la señal triangular, para obtener la señal de salida PWM.

Es importante hacer mención, antes de iniciar con la descripción de las unidades en VHDL, sobre las herramientas utilizadas, así como la metodología a utilizar en el control GPI. Entre las herramientas utilizadas se definen las herramientas de automatización de diseño electrónico (EDA, *Electronic Design Automation*).

El objetivo de las herramientas EDA, es facilitar el proceso de diseño electrónico, y están divididas en herramientas EDA-hardware y EDA-software (EDA-CAD) [46]. Para el diseño de los módulos vistos previamente se ha elegido como herramienta hardware a la tarjeta de desarrollo Nexys-2 de la compañía *Digilent*, y como herramienta EDA-CAD al programa *ISE Foundation*, en su versión 12.1i, de la compañía Xilinx.

La tarjeta Nexys-2 es una plataforma de diseño de sistemas digitales que tiene como principal elemento al FPGA 3E-1200 FG320 de la familia Spartan de la compañía Xilinx (Ver figura 5.8). Las características más relevantes de esta plataforma de desarrollo son las siguientes:

- El FPGA 3E-1200 tiene una capacidad de 1200,000 compuertas (*System Gates*).
- Utiliza el puerto USB, ver. 2.0, para configurar al FPGA y establecer una transferencia de datos de alta velocidad entre el FPGA y una PC.
- Compatible con la herramienta ISE/Webpack de Xilinx.
- Incluye 16 MBytes de PSDRAM de la compañía de Micron y 16 Mbytes de StrataFlash ROM de Intel.
- Plataforma flash no volátil para configuración del FPGA.
- Fuente de alimentación conmutada de alta eficiencia.
- Oscilador integrado de 50MHz y socket para un oscilador extra.
- Conectores de expansión con 60 entradas/salidas al FPGA.

- Periféricos elementales: 8 leds, 4 *push-buttons*, 8 interruptores tipo *slide* y 4 *displays* de 7 segmentos.



Figura 5.8: Tarjeta de desarrollo digital NEXYS-2.

Por otro lado, la herramienta *ISE Foundation*, la cual fue creada por la compañía Xilinx para configurar sus respectivos CDCs, es un entorno de desarrollo integrado (IDE), cuyo objetivo es el de facilitar en el proceso de diseño de los sistemas digitales, mediante la automatización del tránsito por las diferentes etapas que integran dicho proceso, desde el modelado, simulación y la síntesis lógica hasta llegar a la implementación del sistema en la arquitectura configurable elegida. En la figura 5.9 se observa el entorno de desarrollo ISE que incluye diferentes formas de modelado de sistemas digitales, captura de esquemáticos, editor de diagramas de estado y editor de HDLs. Esta última forma de modelado permite elegir entre las normas IEEE 1076 (VHDL) [55] y la IEEE 1364 (Verilog) [56]; en el caso particular del trabajo presente se ha elegido el estándar VHDL.

Una vez elegidas las herramientas EDA y el lenguaje HDL a utilizar, es necesario definir el tipo de metodología que se utilizará, para el diseño conceptual de las unidades mencionadas anteriormente y que tipo de modelado se empleará para la descripción de las mismas.

En la sección 4.2.3 se describieron los diferentes tipos de metodologías que se utilizarán en el diseño de un sistema digital, las cuales son ascendente y descendente. Esta última es la elegida para llevar a cabo el diseño conceptual de las unidades que conformaran al sistema de control, debido a que en la actualidad es ampliamente utilizada en el diseño de arquitecturas VLSI o de complejidad elevada.

Los tipos de descripción de circuitos que permite VHDL se describieron en la sección 4.2.1, estos tipos son estructural, flujo de datos y comportamental. La elección del tipo de modelado depende básicamente del nivel de abstracción que se desea manejar. Para el modelado del sistema de control se ha elegido el estilo comportamental para generar los diferentes módu-

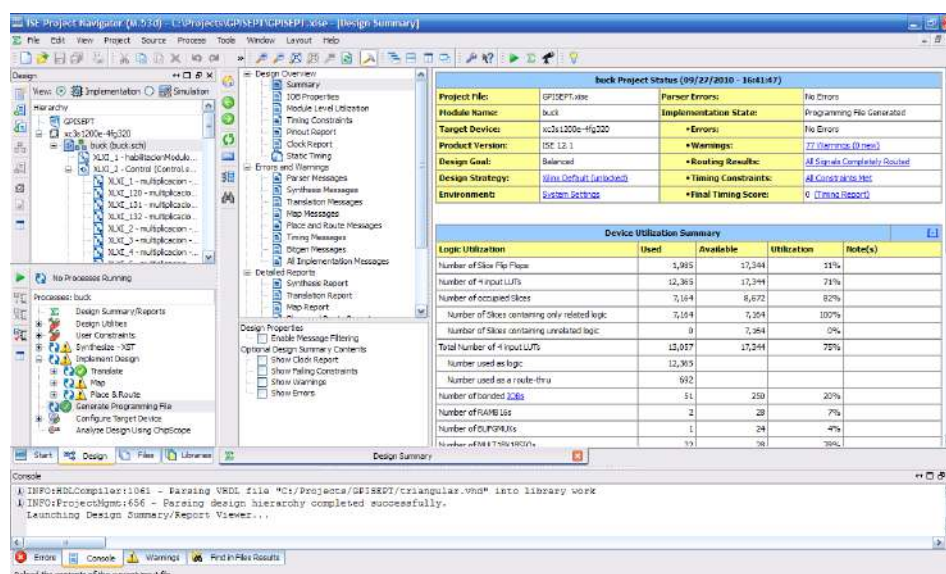


Figura 5.9: Entorno de desarrollo Xilinx ISE 12.1i.

los del sistema de control, y el estilo estructural para unirlos mediante un esquemático.

### 5.3.1. Módulo de control del ADC

La tarjeta NEXYS2 no cuenta con un convertidor analógico digital, por lo que es necesario realizar una interfaz entre la tarjeta y el convertidor analógico digital para poder obtener los datos del voltaje de salida del convertidor reductor. El tiempo de conversión del ADC es un factor importante a tomar en cuenta al momento de elegir el dispositivo para que no afecte y retarde al sistema. Por tal motivo el ADC elegido es el ADC0820 [57], el cual es controlado solo a través de una señal y su tiempo conversión máximo es de  $2.5 \mu\text{s}$ .

Las características más importantes del ADC son las siguientes:

- Resolución de 8 bits.
- No necesita un reloj externo.
- Voltaje de alimentación de 5V.
- Fácil interfaz con microprocesadores u operación *stand-alone*.
- No existen códigos perdidos.
- Salida de 3 estados.
- Rango de voltaje de entrada de 0 a 5V.

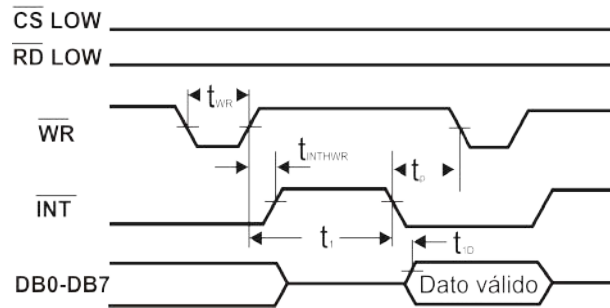


Figura 5.10: Tiempos necesarios para la conversión del ADC0820.

- No se requiere ajuste a 0 o a escala completa.
- Bajo consumo de potencia.

El ADC0820 es un circuito integrado de 20 pines. El modo de operación configurado es *stand-alone* por lo que los pines configurados se describen en la tabla 5.4. Una descripción más detallada de los modos de operación del ADC0820 se puede ver en [57].

El módulo de control del ADC se diseña con base en los diagramas de tiempo del ADC, figura 5.10 y tabla 5.4. El modo de operación del ADC es configurado como operación *stand-alone*, donde únicamente se controla el tiempo del pin 6 ( $\overline{WR}$ ). La conversión iniciará al desactivar la señal del pin 6 ( $\overline{WR} = 0$ ) durante al menos 600ns ( $t_{WR}$ ). La conversión estará lista aproximadamente 800ns después del flanco de subida de la señal del pin 6 ( $\overline{WR} = 1$ ).

Por otro lado, una segunda tarea de este módulo es activar las señales necesarias para habilitar el inicio de las operaciones del control GPI. Debido a que las operaciones se ejecutan en escalera para obtener un resultado final, este módulo activa las señales conforme terminan las operaciones del nivel anterior. Por tanto este módulo está conformado por una máquina de estados que controla al convertidor ADC y las operaciones del control. La figura 5.11 muestra el símbolo del módulo de control del ADC obtenido de la descripción utilizando VHDL. Además en la tabla 5.5 se detallan las funciones de cada una de las señales de este módulo.

La figura 5.12 muestra una simulación del módulo de control del ADC. Se observa que la primera señal que se activa es *wr* que controla al ADC. Una vez que el ADC0820 ha terminado la conversión, se activa la señal *datovalido* que indica al módulo de control que un dato se encuentra listo para ser procesado, lo cual da inicio a la activación de las señales para administrar la operación del control.



Tabla 5.3: Descripción de tiempos para la conversión del ADC0820.

Símbolo	Parámetro	Valor			Unidad
		Mínimo	Típico	Máximo	
$t_{WR}$	Tiempo de escritura	Max		50	$\mu s$
		Min	600		ns
$t_{INTHWR}$	Retardo del flanco de subida de $\overline{WR}$ al flanco de caída de $\overline{INT}$		175	270	
$t_I$	Tiempo interno de comparación		800	1300	ns
$t_P$	Retardo del fin de conversión a la siguiente conversión	500			
$t_{ID}$	Retardo de $\overline{INT}$ a datos válidos		20	50	

Tabla 5.5: Descripción de señales del módulo de control del ADC.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
datovalido	Salida	Indica al módulo de control que un dato válido se encuentra en el ADC0820
wr	Salida	Señal que activa la conversión del ADC0820
sal1-sal13	Salida	Activan el inicio de operaciones del módulo de control

Tabla 5.4: Descripción de pines del convertidor analógico digital ADC0820.

Pin	Nombre	Descripción
1	$V_{IN}$	Voltaje análogo de entrada. Rango= $GND \leq V_{IN} \leq V_{CC}$
2	$DB_0$	Dato de salida de tres estados. Bit 0 (LSB)
3	$DB_1$	Dato de salida de tres estados. Bit 1
4	$DB_2$	Dato de salida de tres estados. Bit 2
5	$DB_3$	Dato de salida de tres estados. Bit 3
6	$\overline{WR}/RDY$	<b>Modo WR-RD.</b> $\overline{WR}$ . Con $\overline{CS}$ en bajo, la conversión empieza en el flanco de caída de $\overline{WR}$ . Aproximadamente 800 ns después del flanco de subida de $\overline{WR}$ el resultado estará en el latch de salida.
7	$MODE$	Entrada de selección del modo a operar. <b>Modo RD</b> cuando toma un valor bajo. <b>Modo WR-RD</b> cuando toma un valor alto.
8	$\overline{RD}$	<b>Modo WR-RD.</b> Los datos de salida ( $DB_0$ - $DB_7$ ) estarán disponibles cuando $\overline{CS}$ toma un valor bajo.
9	$\overline{INT}$	<b>Modo WR-RD.</b> $\overline{INT}$ indica el fin de la conversión y la presencia de los datos en la salida. $\overline{INT}$ tomará un valor bajo aproximadamente 800 ns después del flanco de subida de $\overline{WR}$ .
10	$GND$	Tierra.
11	$V_{REF(-)}$	El nivel inferior de conversión. Rango= $GND \leq V_{REF(-)} \leq V_{REF(+)}$
12	$V_{REF(+)}$	El nivel superior de conversión. Rango= $V_{REF(-)} \leq V_{REF(+)} \leq V_{CC}$
13	$\overline{CS}$	$\overline{CS}$ debe estar en bajo para que $\overline{RD}$ y $\overline{WR}$ sean reconocidos por el convertidor.
14	$DB_4$	Dato de salida de tres estados. Bit 4
15	$DB_5$	Dato de salida de tres estados. Bit 5
16	$DB_6$	Dato de salida de tres estados. Bit 6
17	$DB_7$	Dato de salida de tres estados. Bit 7 (MSB)
18	$\overline{OFL}$	Salida <i>overflow</i> . Si la entrada análoga es mayor que $V_{REF(+)}$ , $\overline{OFL}$ tendrá un valor bajo al final de la conversión.
19	$NC$	No conectado.
20	$V_{CC}$	Voltaje de alimentación.

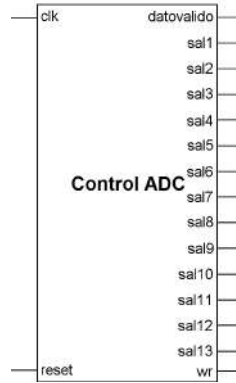


Figura 5.11: Símbolo del módulo de control del ADC.

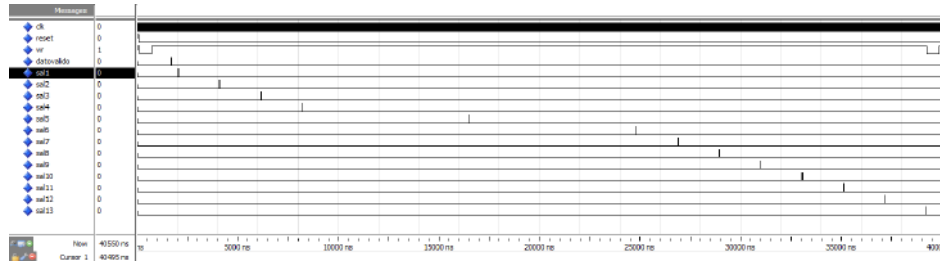


Figura 5.12: Simulación del módulo de control del ADC.

### 5.3.2. Módulo del control GPI

El objetivo principal de este módulo es el realizar las operaciones en punto flotante del control GPI. Con el valor obtenido mediante el ADC realizará las operaciones concernientes a la ley de control y obtendrá a su salida el valor de  $u_{av}$  para que posteriormente este valor pueda ser convertido a punto fijo y pase al módulo del PWM.

La ley de control a programar dentro del FPGA se puede observar en la ecuación (3.57). Se pueden reescribir las ecuaciones del controlador GPI como:

$$\begin{aligned}
 u_{av} &= a_1 v + a_2 \hat{F} + a_3 F \\
 v &= -k_3(\hat{F}) - k_2(F - \bar{F}) - k_1 \int_0^t (F(\tau) - \bar{F}(\tau)) d\tau \\
 &\quad - k_0 \int_0^t \int_0^\tau (F(\lambda) - \bar{F}(\lambda)) d\lambda d\tau \\
 \hat{F} &= \int_0^t (a_4 u - a_5 F) dt - a_6 F
 \end{aligned} \tag{5.5}$$

donde,

$$a_1 = \frac{LC}{E}, a_2 = \frac{L}{ER}, a_3 = \frac{1}{E},$$

$$a_4 = \frac{E}{LC}, a_5 = \frac{1}{LC}, a_6 = \frac{1}{RC}$$

Para mayor facilidad en el diseño del control GPI, a partir (5.5) se realiza un diagrama a bloques para observar las operaciones a realizar. Dicho diagrama a bloques puede ser visto en la figura 5.13.

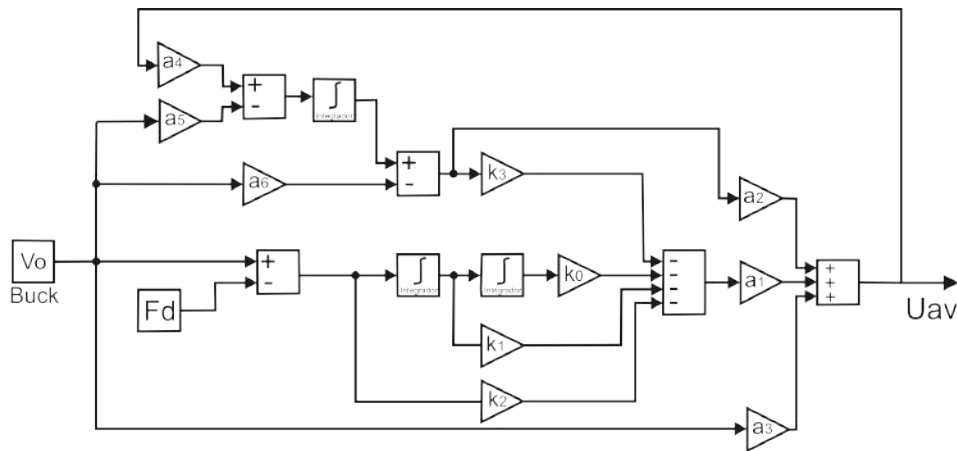


Figura 5.13: Diagrama a bloques del control GPI.

Del diagrama a bloques de la figura 5.13 se observa que es necesario diseñar unidades de suma, multiplicación e integración en punto flotante. Es por ello que estas tres unidades se diseñan en punto flotante y posteriormente se unirán mediante un esquemático para formar el control GPI.

### 5.3.2.1. Unidad de suma en punto flotante

En la sección 4.1.3 se mostró un diagrama de flujo para realizar la suma en punto flotante, sin embargo es necesario realizar un algoritmo para poder facilitar el traslado al diseño en VHDL.

Antes de realizar el algoritmo, es necesario definir algunos parámetros que serán ocupados durante la realización del algoritmo y del diseño en VHDL, por tanto se tiene:

- La operación se realiza entre dos operandos A y B.
- El signo del operando A se definirá como *signA*, mientras que el signo del operando B será *signB*.
- El exponente del operando A está definido como *expa*, así el exponente del operando B es *expb*.

- La mantisa del operando A, ya con el bit implícito será definido como  $sa$ , por otro lado, la mantisa del operando B con el bit implícito es  $sb$ .

En la tabla 5.6 se muestra un algoritmo para realizar la suma en punto flotante y que está compuesto principalmente de 4 etapas [43]:

1. **Swap.** Debido a que la operación se realiza entre dos números, esta etapa se encarga de realizar un intercambio entre los operadores con el fin de intercambiar y dejar en el operando A al número más grande, esto se lleva a cabo comparando sus exponentes, es decir, si el operando A es más grande que el operando B no se hace ningún cambio, en caso contrario se hará un intercambio y el operando B será el nuevo operando A y el operando A pasará a ser el operando B.
2. **Alineación.** El objetivo de este paso es igualar los exponentes, para poder realizar posteriormente la suma. Si los exponentes son iguales, no es necesario realizar ninguna operación. En el paso anterior se intercambiaron los operandos, por tanto en el operando A quedó el número con el exponente mayor y en el operando B el número con el exponente menor. Si los exponentes son diferentes, el exponente  $expb$  será igualado a  $expa$  y se realizará un corrimiento a la izquierda de la mantisa  $sb$  tantas veces como la diferencia entre  $expa$  y  $expb$ .
3. **Suma.** Esta etapa realiza la suma o resta entre los operandos. Si el signo de los dos operandos es el mismo entonces se suman los dos números y el signo permanece igual, en caso contrario se resta el número mayor del menor y el signo del resultado será el del número más grande.
4. **Normalización.** Esta es la etapa final del algoritmo y consiste en normalizar el resultado de la operación, para que el número obtenido se encuentre en el formato adecuado de 32 bits. El signo del resultado estará en  $signa$ , y el exponente del resultado permanecerá en  $expa$ .

Conforme al algoritmo visto previamente se diseña una unidad en VHDL siguiendo la metodología descendente debido a la complejidad del algoritmo, la entidad resultante de este módulo puede ser vista en el símbolo obtenido en la figura 5.14, además en la tabla 5.7 se observa a detalle la descripción de las entradas y salidas de este módulo.

La unidad modelada corresponde con una máquina de 3 estados que funciona de la siguiente manera; en el primer estado se espera el inicio de operación, este será detectado con la señal  $Ini\_Op='1'$  y con esto se iniciará el algoritmo de la suma y se realizarán los pasos 1,2 y 3, en donde se realizan el intercambio (*swap*) entre los operadores en caso de ser necesario, también

Tabla 5.6: Algoritmo para la suma en punto flotante.

```

-Entrada Operando A= signa,expa,sa;
Operando B= signb,expb,sb;
-Salida Resultado=signr,expr,sr; Inicio_suma
PASO1:
  -SWAP
  if expb<expa then
    swap(signa, signb);
    swap(sa, sb);
    swap (expa, expb);
  end if;
PASO2:
  -ALINEACION
  if expa!=expb then
    diferencia=expa-expb;
    expb=expa;
    sb=shift_right(signb,diferencia)
  end if;
PASO3:
  -SUMA
  if signa=signb then
    resultado=sa+sb;
  else
    if sa>sb then
      resultado=sa-sb;
    else
      resultado=sb-sa;
      signa=signb
    end if;
  end if;
PASO4:
  -NORMALIZACION
  if resultado=0 then
    expa=0;
    signa=0;
    IR A PASO 5;
  else
    if resultado(24)='1'then
      resultado=shift_right(resultado,1);
      expa:=expa+1;
      IR A PASO 4;
    elsif resultado(23)='1'then
      IR A PASO 5;
    else
      resultado:=shift_left(resultado,1);
      expa:=expa-1;
      IR A PASO 4;
    end if;
  end if;
PASO 5:
  //Mantisa sr=resultado(22 downto 0);
  //Signo signr=signa;
  //Exponente expr=expa;
Fin_suma;

```

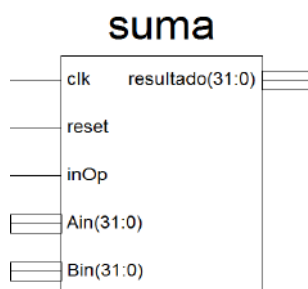


Figura 5.14: Símbolo del módulo de suma en punto flotante.

Tabla 5.7: Descripción de señales del módulo de suma en punto flotante.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
inOp	Entrada	Indica el inicio de la operación de suma en punto flotante
Ain(31:0)	Entrada	Vector de entrada que representa el operador A en punto flotante de precisión simple
Bin(31:0)	Entrada	Vector de entrada que representa el operador B en punto flotante de precisión simple
resultado(31:0)	Salida	Vector de salida que representa la suma de los operadores A y B en punto flotante

se realiza la alineación del operando B y por último la suma de las mantisas. El segundo estado corresponde al paso 4 donde se realiza la normalización del resultado, para ello se checa en primer lugar si el resultado de la suma es cero, en ese caso el resultado final será un vector de 32 bits de ceros; en caso contrario se checa los dos últimos bits más significativos del resultado de la suma de las mantisas, si el bit más significativo corresponde con un '1', se realiza un corrimiento a la derecha y el exponente se aumenta en uno, si el segundo bit más significativo es '1', el resultado está normalizado y se pasa al siguiente estado; si no se cumplen las condiciones anteriores se realiza un corrimiento a la derecha del resultado de la suma, y el exponente de A se decrementa en 1. El último estado de la máquina es mostrar el resultado en su estándar de 32 bits, si el resultado es diferente de cero, el resultado final estará compuesto del bit de signo del operando A, 8 bits del exponente de A, y los 23 bits menos significativos del resultado de la operación después de haber sido normalizado.

La figura 5.15 muestra una suma entre dos operadores A y B con valores de 23.466 y 43.98, enseguida se realiza la suma entre -23.25 y 30.57. La señal *InOp* indica el inicio de la operación. Tres ciclos de reloj mas tarde el resultado se tiene en la señal de salida *resultado*.

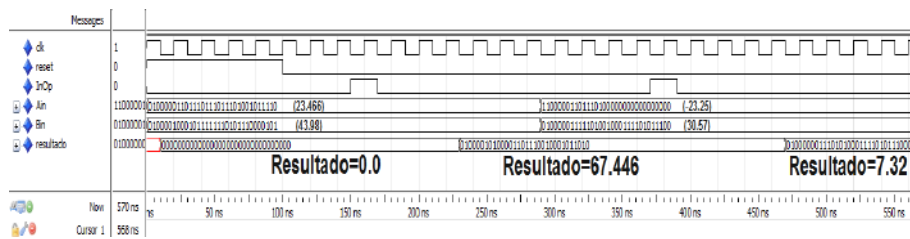


Figura 5.15: Simulación del módulo de suma en punto flotante.

### 5.3.2.2. Unidad de multiplicación en punto flotante

De igual manera que la suma en punto flotante, en la sección 4.1.4 se puede observar un diagrama de flujo de la multiplicación en punto flotante. Partiendo de dicho diagrama se procede a realizar un algoritmo para su posterior diseño en VHDL. La multiplicación en punto flotante es más sencillo que la suma en punto flotante, sin embargo ocupa un poco más de recursos debido a que se realiza una multiplicación de significandos de 24 bits.

Al igual que la suma, antes de realizar el algoritmo es necesario definir algunos parámetros que nos ayudarán a entender de mejor forma el algoritmo de la multiplicación, por tanto se consideran los siguientes aspectos:

- La multiplicación se realiza entre dos operandos A y B.
- El signo del operando A se definirá como  $signA$ , mientras que el signo del operando B será  $signB$ .
- El exponente del operando A está definido como  $expa$ , así el exponente del operando B es  $expb$ .
- La mantisa del operando A, ya con el bit implícito será definido como  $sa$ , por otro lado, la mantisa del operando B con el bit implícito es  $sb$ .

En la tabla 5.8 se observa el algoritmo de la multiplicación en punto flotante que consta de tres etapas que se describen a continuación:

1. *Suma.* Esta etapa consiste en obtener los exponentes de los operandos A y B y sumarlos, teniendo en cuenta el sesgo, que en este caso al ser exponentes de 8 bits, el sesgo es 127. Además también se encarga de verificar si existiese un *overflow* o *underflow* al momento de realizar la multiplicación.
2. *Multiplicación.* Debido a que el FPGA utilizado incluye bloques de aplicación específica como lo son los multiplicadores, sólo se toman las mantisas de los operandos,  $sa$  y  $sb$ , y se multiplican, guardando el resultado en una variable temporal para después normalizarlo.

Tabla 5.8: Algoritmo para la multiplicación en punto flotante

```

-Entrada Operando A = signa,expa,sa;
Operando B = signb,expb,sb;
-Salida Resultado = signr,expr,sr; Inicio_multiplicación
PASO1:
  -SUMA
  auxEx = expa + expb;
  if auxEx > 382 then
    -overflow;
  else
    if auxEx < 127 then
      -underflow;
    end if;
  end if;
PASO2:
  -MULTIPLICACIÓN
  auxMan = sa * sb;
PASO3:
  -NORMALIZACIÓN
  if (auxMan(47) = '1') then
    auxMan := shift_rigth(auxMan,1);
    auxEx := auxEx + 1;
  end if;
  sign = signa xor signb;
PASO 4:
  //Mantisa sr = auxMan(45 downto 23);
  //Signo signr = sign;
  //Exponente expr = auxEx;
Fin_multiplicación;

```

3. *Normalización.* El objetivo de esta etapa es el normalizar el resultado de la multiplicación para poder obtener el resultado en el formato adecuado de 32 bits.

Del algoritmo de la multiplicación en punto flotante, se diseña en VHDL mediante la metodología descendente la unidad de multiplicación en punto flotante. En la figura 5.16 se observa el símbolo de la unidad de multiplicación en punto flotante; además, en la tabla 5.9 se observa a detalle la descripción de las entradas y salidas de este módulo. Este diseño corresponde con un máquina de 2 estados cuyo funcionamiento es como sigue: En el primer estado se espera la señal para el inicio de la operación, esta iniciará con la señal *Ini\_Op* = '1'. Al momento de detectar el inicio de operación, se realizarán los pasos 1,2 y 3. Los exponentes de los operadores A y B se sumarán, como estos exponentes están sesgados, a la suma se le restará el sesgo. Además se checa si existe un *overflow* o bien *underflow* en cuyo caso solo se limitará al número más grande o a cero según sea el caso. Una vez realizado esto se procede a multiplicar las mantisas ya con el bit implícito. El resultado de esta multiplicación será normalizado observando únicamente los dos bits más significativos del resultado de la operación. Si el bit más significativo contiene un '1', se realiza un corrimiento a la derecha decrementando además el exponente en 1 y se procede al paso 4. El signo de la operación corres-



Tabla 5.9: Descripción de señales del módulo de multiplicación en punto flotante.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
inOp	Entrada	Indica el inicio de la operación de suma en punto flotante
Ain(31:0)	Entrada	Vector de entrada que representa el operador A en punto flotante de precisión simple
Bin(31:0)	Entrada	Vector de entrada que representa el operador B en punto flotante de precisión simple
resultado(31:0)	Salida	Vector de salida que representa la multiplicación de los operadores A y B en punto flotante

ponde con la ley de los signos para la multiplicación, que en este caso se realizará con una compuerta de tipo xor entre los signos de los operandos A y B, que en este caso son *signa* y *signb* respectivamente. El segundo estado corresponde con la finalización de la operación y adecuar la salida de la operación de acuerdo a los resultados obtenidos en el estado 1, por lo que el resultado final estará compuesto de 32 bits, el más significativo corresponde con el signo calculado a partir de la compuerta xor, enseguida 8 bits de la suma del exponente y por último los 23 bits siguientes del resultado de la multiplicación de significandos, eliminando los dos bits más significativos.

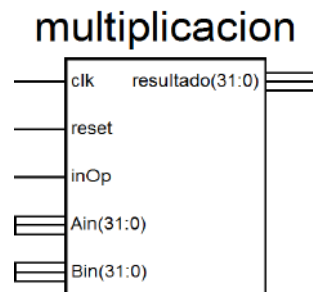


Figura 5.16: Símbolo del módulo de multiplicación en punto flotante.

La figura 5.17 muestra la multiplicación entre dos operandos, A y B, con valores de 30.5 y 10.75, enseguida se realiza la multiplicación entre -5035 y 100000. La señal *InOp* indica el inicio de la operación. El resultado de la operación se tiene en la señal de salida *resultado*.

### 5.3.2.3. Módulo de integración en punto flotante

Una de las características principales del control GPI es la robustez que presenta. Esta característica es debida a los reconstructores integrales que

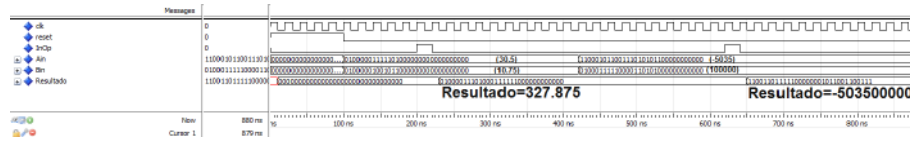


Figura 5.17: Simulación del módulo de multiplicación en punto flotante.

presenta en su diseño, por lo que la integración pasa a ser una operación fundamental en la implementación de dicho control. Una forma de obtener el resultado de una integral en un sistema dinámico como el control GPI es mediante métodos de integración numérica. El sistema dinámico puede ser expresado en forma de ecuaciones diferenciales que tendrán que ser resueltas por algún método numérico.

La forma de ecuación diferencial más sencilla que se puede tener es:

$$\frac{dy}{dx} = f(x, y) = \Psi(x) \quad (5.6)$$

Donde  $f(x_i, y_i)$  es la ecuación diferencial evaluada en  $x_i$  y  $y_i$ . Para resolver esta ecuación se necesita encontrar una función cuya derivada sea  $f(x)$ , es decir, encontrar una integral indefinida de  $f(x)$ . Entre los métodos más conocidos para la resolución de ecuaciones diferenciales se encuentran los métodos de paso simple y los métodos multipaso. Los métodos de paso simple utilizan información en un solo punto  $x_i$  para predecir un valor de la variable dependiente  $y_{i+1}$  en un punto futuro  $x_{i+1}$ .

**Método de Adams-Bashforth.** Los métodos de Adams utilizan información en múltiples pasos de la solución para volver a llegar a ella en el siguiente punto  $x$ . Así un ejemplo de dos pasos utiliza la información  $x_i$ , y  $x_{i-1}$  para generar la solución en el nuevo punto  $x_{i+1}$ . Los métodos de Adams pueden clasificarse en dos tipos: los métodos explícitos de Adams-Bashforth y los métodos implícitos de Adams-Moulton. Ambos pueden combinarse para generar los métodos predictor-corrector de Adams-Bashforth-Moulton [58].

El principal objetivo del método de Adams-Bashforth de  $n$  pasos es utilizar un polinomio de interpolación con base en información en la localización actual  $x_i$  y en ubicaciones previas  $(x_{i-1}, x_{i-2}, \dots, x_{i-n+1})$  para poder expresar  $(x)$ . La solución en  $x_{i+1}$  se encuentra a partir de

$$y_{i+1} - y_i = \int_{x_i}^{x_{i+1}} \Psi(x) dx \quad (5.7)$$

Para la deducción del método de Adams-Bashforth se toma el caso de dos pasos. Para ello se supone que se tiene un tamaño de paso constante  $h$  desde el punto  $(i - 1)$  hasta el punto  $i$  y desde el punto  $i$  hasta el punto  $(i + 1)$ , y además que existe una solución hasta el punto  $i$ .

Se puede utilizar un polinomio de interpolación de Lagrange [58] con información en los dos pasos  $x_i$  y  $x_{i-1}$ , de tal manera que se obtenga:

$$\psi(x) \approx = [(x_i - x)\psi(x_{i-1}) + (x - x_{i-1})\psi(x_i)]/h \quad (5.8)$$

El resultado de la integración de  $\psi(x)$  visto en (5.7) es:

$$y_{i+1} = y_i + (h/2)[3\psi(x_i) - \psi(x_{i-1})] \quad (5.9)$$

Donde esta última ecuación corresponde a la fórmula directa del método de Adams-Bashforth de segundo orden que nos permite obtener la solución a una ecuación diferencial o bien realizar la integral que se necesita en el control GPI.

La figura 5.18 muestra la simulación del método de Adams-Bashforth de segundo orden obtenida mediante Matlab. En dicha figura se observa prácticamente una misma señal que corresponde al bloque de integración incluido en Simulink, y a la implementación de la fórmula directa del método de Adams-Bashforth. En esta figura se puede observar que las respuestas son muy similares, existiendo solo un pequeño error entre las gráficas, esto debido al orden de la fórmula directa.

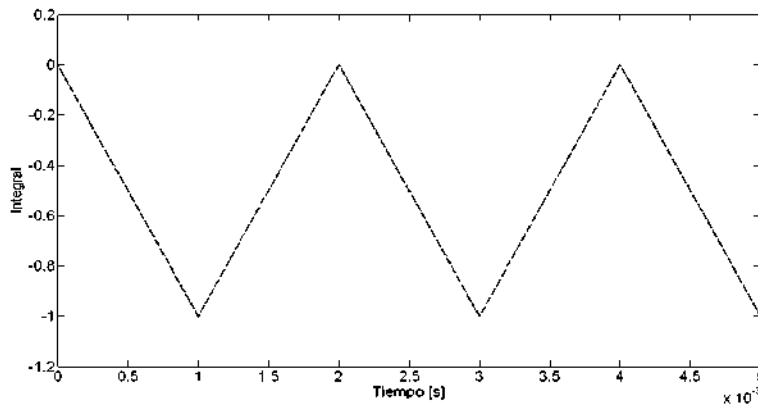


Figura 5.18: Simulación de la fórmula directa de segundo orden de Adams-Bashforth.

**Módulo de Integración en VHDL.** El módulo de integración en el sistema de control permitirá obtener la integral del valor de entrada que se tenga en el módulo. De acuerdo al método de Adams-Bashforth se tiene una fórmula directa para poder obtener el valor de la integral la cual se puede escribir como:

$$y_{i+1} = y_i + (h/2)[3\dot{y}_i - \dot{y}_{i-1}] \quad (5.10)$$

Tabla 5.10: Descripción de señales del módulo de integración en punto flotante.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
Sing_in(31:0)	Entrada	Vector de entrada en punto flotante que representa la función a integrar
ini_op2	Entrada	El valor $y_{i-1}$ se actualiza cuando esta señal se activa
ini_op	Entrada	Señal que indica el inicio de la operación de integración
integral(31:0)	Salida	Vector de salida en punto flotante que representa la integral de la señal de entrada

En la ecuación (5.10) se observa que para obtener el valor de  $y_{i+1}$  solo se necesitan hacer operaciones con el valor actual y anterior de la entrada  $\dot{y}$  y  $\dot{y}_{i-1}$ , además del último valor calculado  $y_i$ . Por tanto se realiza el módulo de integración a partir de las operaciones de suma y multiplicación antes modeladas. El paso de tiempo  $h$  se obtiene a partir de medir el tiempo de procesamiento en que tarda el módulo de control en obtener una salida y su valor es aproximado a  $40\mu s$ . La figura 5.19 muestra el símbolo obtenido para el cálculo de la integral, además la tabla 5.10 describe cada una de las señales de entrada y salida de este módulo.

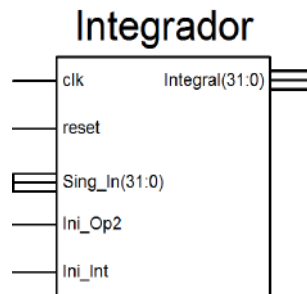


Figura 5.19: Símbolo del módulo de integración en punto flotante.

### 5.3.3. Módulo del convertidor punto flotante a punto fijo.

La ley de control diseñada en VHDL presenta los resultados en formato de punto flotante en 32 bits, sin embargo el módulo del PWM maneja números de punto fijo en 8 bits, por lo que es necesario una etapa intermedia entre la ley de control y el módulo PWM. De esta manera, este módulo es el encargado de traducir el formato de punto flotante a punto fijo.

Por otra parte, al momento de realizar la comparación de este valor con la señal triangular es necesario realizar una conversión del valor en punto fijo

a un valor adecuado que se encuentre en los niveles de la señal triangular y poder compararlos sin ningún inconveniente.

Antes de realizar el módulo en VHDL, es necesario definir algunos aspectos que facilitarán el diseño de este módulo, así como ciertas limitaciones que hay que tomar en cuenta, tales cuestiones se muestran a continuación:

- El valor máximo de la señal triangular es 5, por lo que la señal  $u_{av}$  obtenida de la ley de control será limitada entre los valores 0 y 5.
- La señal triangular generada está en el rango de 0 a 5 V, representada mediante un vector de 8 bits, por lo que su rango dentro del FPGA será de 0 a 255, donde 0 representa su valor mínimo y 255 representa su valor máximo (5V).
- Si el número en punto flotante es un vector de ceros, la salida en punto fijo será de cero.
- Para poder convertir un número en punto flotante a punto fijo, es necesario que el exponente corresponda con el sesgo, en este caso que el exponente tenga un valor de 127. Si el exponente es mayor o menor se disminuirá o aumentará su valor en uno, tantas veces como sea necesario, además de realizar el mismo número de corrimientos con la mantisa a la izquierda o derecha según sea el caso.
- La parte entera de la representación en punto fijo será de 12 bits, mientras que la parte decimal estará representada por un vector de 8 bits.
- Debido a que la parte entera es de 12 bits, el número mayor que se podrá convertir será 4095. Si existiese un número mayor a este se limitará al valor 4095 con lo que la salida estará saturada. Por otro lado su límite inferior será 0.
- Para adecuar los valores del punto fijo para su comparación con la señal triangular se tiene que convertir el valor a niveles de 0 a 255. Debido a que el punto fijo estará limitado entre 0 y 5 V, este valor en punto fijo se multiplica por 51 para obtener los valores de 0 a 255.

En la tabla 5.11 se observa el algoritmo para convertir de punto flotante a punto fijo, el cual consta básicamente de 4 pasos.

El diseño en VHDL corresponde con una máquina de 4 estados que se describen a continuación.

- *ESTADO 1.* Se espera el inicio de la operación mediante la señal ( $Ini\_Op='1'$ ). Una vez que se inicia la operación se procede verificar los valores del vector de entrada en punto flotante. Si es cero, el valor en punto fijo es cero y se pasa al estado 4.

Tabla 5.11: Algoritmo para la conversión de punto flotante a punto fijo.

```

-Entrada
Número en punto flotante A= signa,expa,sa;
-Salida
Parte entera en punto fijo= puntofijo(8bits);
Inicio conversión
PASO1:
    If A=0 then
        puntofijo=0;
        Ir a Paso ;
        end if;
PASO2:
    If A>4095 then
        puntofijo=4095;
        Ir a Paso 4;
        end if;
    If A<0.020 then
        puntofijo=0;
        Ir a Paso 4;
        end if;
PASO3:
    if expa>127 then
        -Decrementar el exponente;
        expa=expa-1;
        -Desplazar a la Izquierda la mantisa;
        sa=shift_left(sa,1);
        -Ir a Paso 3;
    elsif exponente<127 then
        -Aumentar el exponente;
        expa=expa+1;
        -Desplazar a la Derecha la mantisa;
        sa=shift_right(sa,1);
        -Ir a Paso 3;
    elsif exp=127 then
        -Guardar parte entera;
        -Guardar parte decimal;
        -Ir a Paso 4;
    end if;
PASO 4:
    -Limitar el punto fijo entre 0 y 5
    -Guardar en un vector de 12 bits la parte entera (limitada)
    -y la parte decimal en 8 bits
    pf=entera & decimal;

Fin conversión;

```

- *ESTADO 2.* Se verifican los límites de la conversión. Debido a que el punto fijo es de 12 bits el número más grande podrá ser 4095, si es mayor el número en punto flotante se limita y se pasa al estado 4. Por otro lado se checa el límite inferior, si es menor a 0.020 el valor del punto fijo es 0 y se pasa al estado 4.
- *ESTADO 3.* En este estado se verifica el valor del exponente. Si es mayor a 127 se decrementa el exponente y se hace un corrimiento a la izquierda de la mantisa. Si el exponente es menor a 127, se incrementa su valor y se hace un corrimiento a la derecha de la mantisa. Si el valor del exponente es 127, se guarda la parte entera y decimal y se pasa al estado 3. Una vez que el exponente es 127 se almacena en un vector de 12 bits la parte entera y en otro de 8 bits la parte decimal. Se pasa al estado 4.
- *ESTADO 4.* Se guardan los valores convertidos en punto fijo, en las señales de salida y se regresa al estado 1.

En la figura 5.20 se observa el símbolo generado en este módulo, además en la tabla 5.12 se observan a detalle cada uno de las señales de entrada y salida de este módulo.

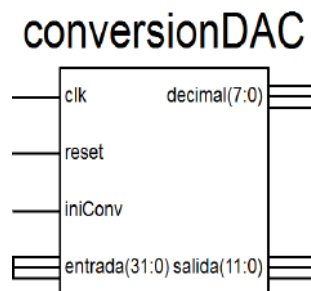


Figura 5.20: Símbolo del módulo de conversión de punto flotante a punto fijo.

#### 5.3.4. Modulación por ancho de pulsos

El último bloque del sistema corresponde al PWM. Este bloque está conformado por tres módulos diseñados en VHDL los cuales son:

- Módulo de conversión ADC. Convierte la señal de punto fijo a una señal de 8 bits para ser comparado con la señal triangular y obtener la modulación por ancho de pulsos.

Tabla 5.12: Descripción de señales del módulo de conversión de punto flotante a punto fijo.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
inConv	Entrada	Indica el inicio de la conversión de punto flotante a punto fijo
entrada(31:0)	Entrada	Vector de entrada que representa el operador en punto flotante a convertir
decimal(7:0)	Salida	Vector de salida que representa la parte decimal de la representación en punto fijo
salida(11:0)	Salida	Vector de salida que representa la parte entera de la representación en punto fijo

- Módulo de onda triangular. Genera la onda triangular a una frecuencia de 48 kHz, la cual será generada con un vector de 8 bits.
- Módulo de comparación. Compara la señal obtenida del control  $u_{av}$ , con la onda triangular, para obtener una señal cuadrada con un ancho de pulso variable que será la salida del sistema PWM y que activará y desactivará al transistor MOSFET.

#### 5.3.4.1. Módulo de conversión ADC

La señal  $u_{av}$  de la ley de control oscilará entre valores de 0 a 5V, dependiendo del voltaje de salida requerido en el convertidor reductor. Para obtener la señal de PWM es necesario comparar esta señal con la onda triangular, sin embargo antes de poder compararlo se deben solucionar dos problemas el primero es limitar la señal proveniente del módulo de control entre los valores 0 y 5; y el segundo problema es cambiar el formato de punto fijo a un formato de 8 bits para poder ser comparado de manera fácil con la onda triangular. El nuevo formato de esta señal oscilará entre los valores 0 y 255 que corresponderán a los valores de 0V y 5V respectivamente, similar a convertir una señal analógica a una señal digital en un convertidor analógico digital, donde el valor de referencia será 5V.

La conversión se realiza de manera sencilla mediante una regla de tres, donde el valor digital máximo de salida será 255 correspondiente al valor analógico 5, y el valor digital mínimo será 0 correspondiente al valor analógico 0. En la figura 5.21 se observa el símbolo obtenido de este módulo, mientras que la tabla 5.13 describe cada una de las señales de entrada y salida de este módulo.

Para verificar el correcto funcionamiento de éste módulo se realizaron simulaciones como las que se muestran en la figura 5.22 donde se convierte un valor de 3.75 al valor correspondiente en un vector de 8 bits. De igual manera se convierte el valor 12.65 donde se observa que este valor sobrepasa



Tabla 5.13: Descripción de señales del módulo de conversión ADC.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
iniop	Entrada	Indica el inicio de la conversión de punto fijo a vector de 8 bits
entera(11:0)	Entrada	Vector de entrada que representa la parte entera del número en punto fijo
decimal(7:0)	Entrada	Vector de entrada que representa la parte decimal de la representación en punto fijo
salidaadc(7:0)	Salida	Vector de salida que representa la conversión del punto fijo a compararse con la onda triangular

el máximo valor (5) y se limita al máximo valor posible en el vector de 8 bits correspondiente al valor 255.

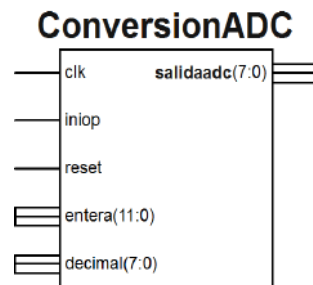


Figura 5.21: Símbolo del módulo de conversión ADC.

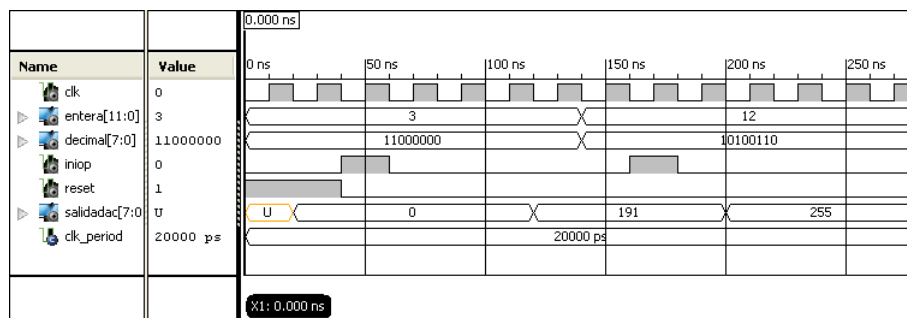


Figura 5.22: Simulación del módulo de conversión ADC.

### 5.3.4.2. Módulo de onda triangular

Este módulo del PWM se encarga de generar una señal triangular a una frecuencia de 48 kHz. La señal triangular generada oscilará entre los valores

Tabla 5.14: Descripción de señales del módulo de onda triangular.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
sTriangular(7:0)	Salida	Vector de salida que representa la onda triangular en un vector de 8 bits

0 y 5V, representado por un vector de 8 bits. El valor 0 de la señal triangular corresponderá al valor 0 del vector de 8 bits, en cambio el valor 5V de la onda triangular corresponde al valor 255 del vector de salida.

Para generar la frecuencia deseada, es importante tomar en cuenta la frecuencia del reloj de entrada. En este caso la frecuencia del reloj de entrada corresponde al valor de 50 MHz, por lo que el ciclo de reloj corresponde a 20ns. El periodo de la onda triangular es el inverso de la frecuencia, lo que resulta aproximadamente  $20.83 \mu\text{s}$ . De esto se obtiene que el tiempo en que la señal triangular tiene pendiente positiva corresponde a  $10.42 \mu\text{s}$ , y de igual forma el tiempo en que la onda triangular tiene pendiente negativa. Al dividir estos tiempos entre el ciclo de reloj de 50 MHz el resultado corresponde al valor 521 que es el valor entero mas cercano al resultado. Por tanto, la señal triangular estará compuesta por 1024 ciclos del reloj de 50 MHz, 512 ciclos cuando la pendiente es positiva y 512 cuando la pendiente es negativa, por lo que la frecuencia se verá modificada en su valor y corresponderá a 48.83 kHz.

La figura 5.23 muestra el símbolo obtenido en este módulo y sus señales correspondientes son descritas en la tabla 5.14. La simulación de este módulo se observa en la figura 5.24, donde se observa que el valor de la señal va aumentando hasta llegar al nivel máximo que es 255 y después empieza a disminuir, lo que corresponde con el nivel máximo de la onda triangular. Otra forma de ver la onda triangular es mediante la simulación en Matlab, donde se pueden obtener simulaciones del módulo en VHDL. La figura 5.25 muestra la onda triangular simulada en Matlab donde se observa claramente la forma de onda.



Figura 5.23: Símbolo del módulo de onda triangular.

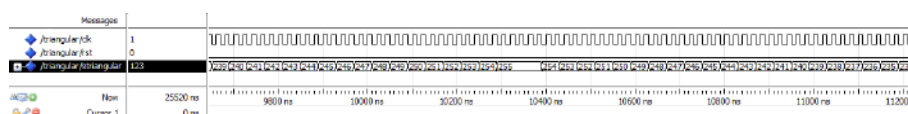


Figura 5.24: Simulación de la onda triangular.

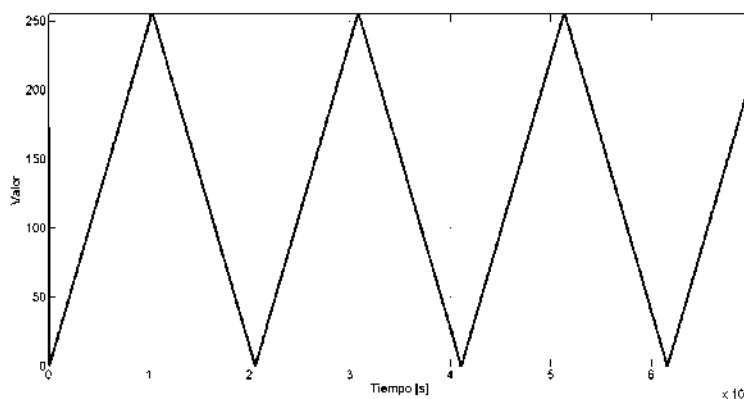


Figura 5.25: Simulación de la onda triangular realizada en Matlab y Modelsim.

### 5.3.4.3. Módulo de comparación

Este módulo es el más sencillo del bloque del modulador por ancho de pulsos. Su tarea consiste básicamente en comparar las señales de entrada que provienen de la ley de control y la onda triangular. Las señales a comparar tienen un tamaño de 8 bits. Este módulo solo realiza la comparación entre estas señales, si la señal proveniente de la ley de control es mayor que la onda triangular se tendrá como salida un nivel alto ('1'), en cambio si la señal de la onda triangular es mayor que la señal de la ley de control, la salida será un nivel bajo ('0') que activará o desactivará al *switch* del convertidor reductor para obtener el voltaje deseado a su salida.

La figura 5.26 muestra el símbolo correspondiente al módulo de comparación. Consta de dos entradas asociadas a la señal proveniente de la ley de control y la onda triangular respectivamente, y una señal de salida correspondiente a la modulación PWM. Estas señales son descritas en la tabla 5.15.

### 5.3.5. Simulación del control GPI

Una vez simulado cada uno de los bloques diseñados en VHDL se procede a realizar el sistema final que estará conformado por el módulo de control

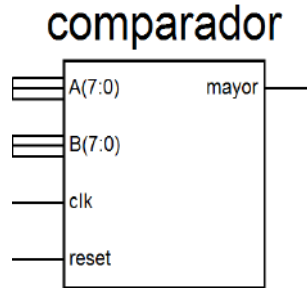


Figura 5.26: Símbolo del módulo de comparación.

Tabla 5.15: Descripción de señales del módulo de comparación.

Señal	Tipo	Descripción
clk	Entrada	Reloj de 50 MHz que sincroniza las salidas de este módulo
reset	Entrada	Señal global que activa el <i>reset</i> del sistema
sTriangular(7:0)	Salida	Vector de salida que representa la onda triangular en un vector de 8 bits

del ADC, el control GPI y el módulo de PWM. En la figura 5.27 se observa el sistema final que será implementado en el FPGA.

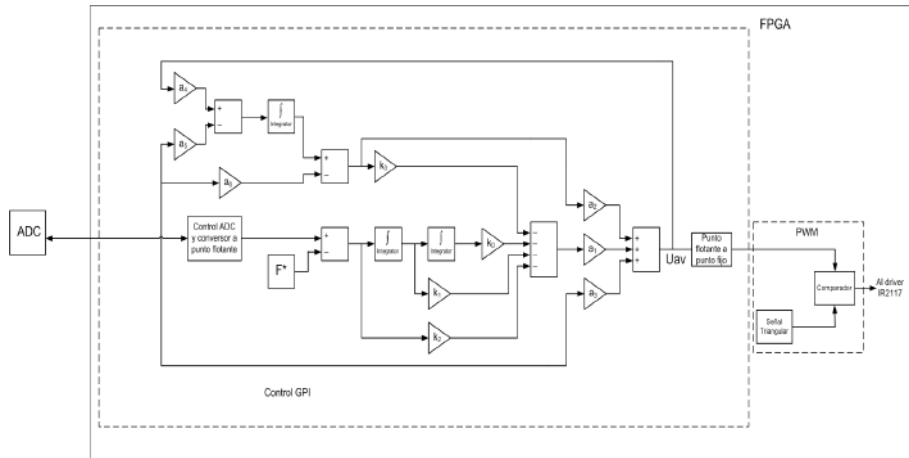


Figura 5.27: Diagrama de bloques del control GPI y PWM implementado en el FPGA.

Antes de implementar el diseño final, es importante poder simular el sistema antes de poder implementarlo, para ello es posible realizar una simulación del sistema final junto con la simulación del circuito convertidor reductor. Para ello se procede a realizar la cosimulación entre tres programas: Matlab/Simulink, ModelSim y Psim. Matlab/Simulink será la herramienta que permita la simulación con los otros programas, así en Matlab Simulink

se realizarán los módulos en VHDL y creará la conexión con el programa ModelSim donde se realizarán las simulaciones de los bloques VHDL. Por otro lado, Matlab/Simulink también estará conectado con el programa Psim que permite realizar la simulación del circuito convertidor reductor. Por tanto, Matlab/Simulink será la herramienta que permita la simulación de los bloques VHDL con la simulación del circuito reductor, mediante la cosimulación.

### 5.3.5.1. Cosimulación en Matlab y ModelSim

La cosimulación en Matlab/Simulink crea una interfaz entre Matlab y un programa externo a Matlab, es decir, la cosimulación permite la interacción de algún simulador externo con herramientas propias de Matlab, así se tiene la interacción bidireccional de instancias de módulos de VHDL simulados en ModelSim y Matlab/Simulink [59].

La figura 5.28 muestra la manera en que ModelSim y Matlab trabajan en conjunto para crear la cosimulación. El bloque que permite la interacción con el simulador HDL se denomina “*EDA Simulator Link MQ*”.

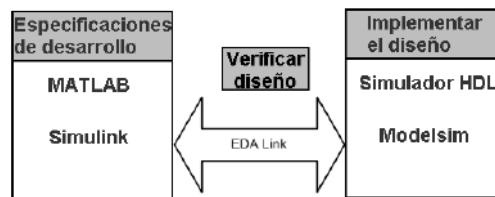


Figura 5.28: Cosimulación entre Matlab y ModelSim.

Cuando se realiza una cosimulación en Matlab, el Simulador (ModelSim) funciona como cliente, tal como se muestra en la figura 5.29. En este escenario, una función del servidor Matlab espera las solicitudes de servicio que recibe del simulador de HDL. Después de recibir una solicitud, el servidor establece un enlace de comunicación e invoca una determinada función de Matlab que calcula los datos para la verificación o visualización del módulo de VHDL que está bajo simulación en ModelSim. Cuando el servidor está corriendo en Matlab, se puede iniciar y configurar el simulador con la función *vsim* (Véase Apéndice A) suministrada con el bloque “*EDA Simulator Link MQ*”.

La figura 5.30 muestra como una función de vector de prueba en Matlab se invoca y comunica con el simulador durante una sesión de simulación. Así se tiene que ModelSim realizará las simulaciones de la entidad en VHDL de acuerdo a los vectores de prueba enviados como parámetros desde Matlab, y asimismo, ModelSim, genera los vectores de salida que serán las respuestas y parámetros de entrada de las funciones en Matlab.

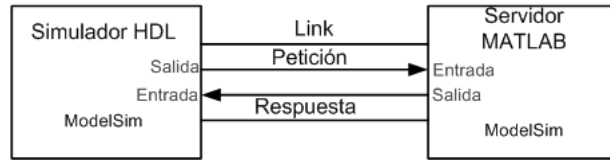


Figura 5.29: En cosimulación, Matlab funciona como servidor, mientras el simulador HDL funciona como cliente.

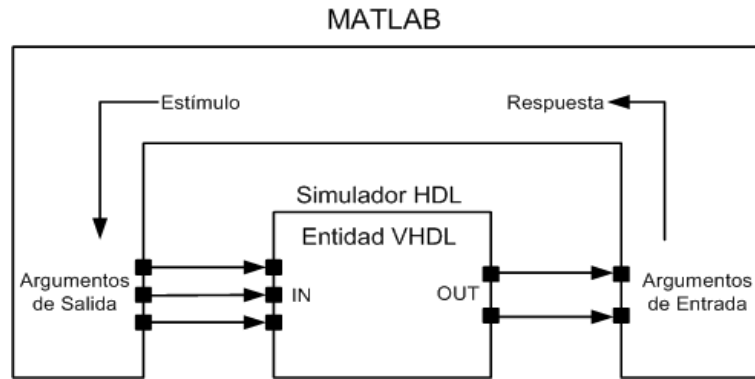


Figura 5.30: Comunicación entre el servidor Matlab y el cliente simulador HDL.

Usando los parámetros del bloque “*EDA Simulator Link MQ*” en Matlab Simulink para una simulación en VHDL se puede configurar lo siguiente:

- Puertos de entrada y salida que corresponden a las señales de un módulo en VHDL.
- Configuración del tipo de comunicación utilizados para intercambiar datos entre las herramientas de simulación.
- Tipo de reloj usado para la aplicación (flanco de caída o flanco de subida). El reloj para cada módulo puede ser independiente.
- Comandos TCL que pueden ejecutarse antes o después de la simulación.

### 5.3.5.2. Cosimulación en Matlab y Psim

El software Psim contiene una aplicación denominada *SimCoupler* que presenta una interfaz entre el mismo PSIM y Matlab Simulink para cosimulación. Con el módulo *SimCoupler* parte del sistema puede ser implementado y simulado en Psim, y el resto del sistema en Simulink. Con esto se puede tener acceso a las amplias características de simulación de Psim, y la capacidad de simulación de Simulink en una forma complementaria [60].

El módulo *SimCoupler* consiste de dos partes: los nodos de enlace en Psim, y el modelo SimCoupler en Simulink. En el presente trabajo se hace uso del módulo *SimCoupler* para poder simular en Psim el convertidor reductor, mientras que en matlab y mediante otra cosimulación la parte del control GPI. La figura 5.31 muestra el circuito convertidor reductor en Psim, en dicha figura se puede observar que la entrada del circuito es la señal PWM que proviene del control en Simulink, esta señal de entrada se conecta utilizando la herramienta *In link node*. Debido a que el sistema es retroalimentado, y la señal de retroalimentación es el voltaje de salida del convertidor reductor ( $V_o$ ), se usa la herramienta *Out link node* para que esta señal pueda ser obtenida en Simulink y utilizarse en el control GPI.

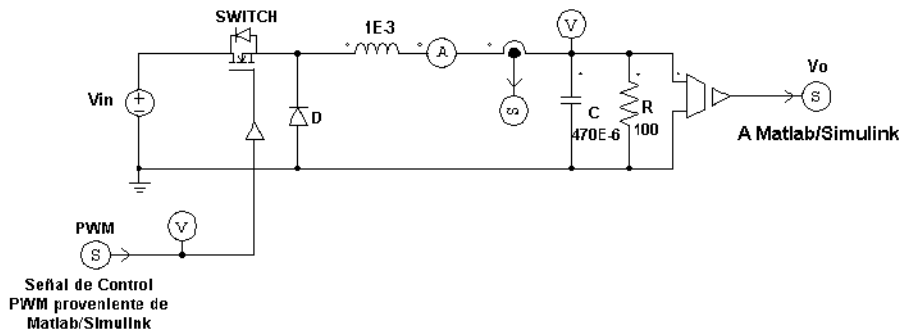


Figura 5.31: Circuito convertidor reductor para la cosimulación con Simulink.

En Simulink se elige la librería S-function SimCoupler y se agrega el bloque SimCoupler al diseño. Una vez agregado el bloque, en sus propiedades se elige la ruta del archivo a cosimular con Psim, automáticamente aparecerá una ventana como la mostrada en la figura 5.32 donde se muestran todas las entradas y salidas a cosimular del circuito en Psim, en este caso, de acuerdo al diagrama elaborado se tienen dos señales de salida del circuito que son la señal de corriente del inductor  $i_L$  y el voltaje de salida del convertidor reductor  $V_o$ . Por otra parte se tienen las entradas al circuito que es la señal de PWM. Una vez configurado el bloque, aparecen automáticamente las señales de entrada y salida en el bloque para su posterior cosimulación, tal como se muestra en la figura 5.33.

### 5.3.5.3. Resultados en Simulación

Una vez diseñado el control GPI en Simulink de acuerdo a la figura 5.27 y mediante el circuito convertidor reductor diseñado en Psim, se procede a la cosimulación. Se hicieron dos pruebas ya con el diseño final del control GPI, esto debido a que el tiempo de cosimulación entre los tres programas es demasiado elevado, para poder simular 5 ms del control GPI tiene que pasar alrededor de 1 hora en simulación, esto en una computadora con las

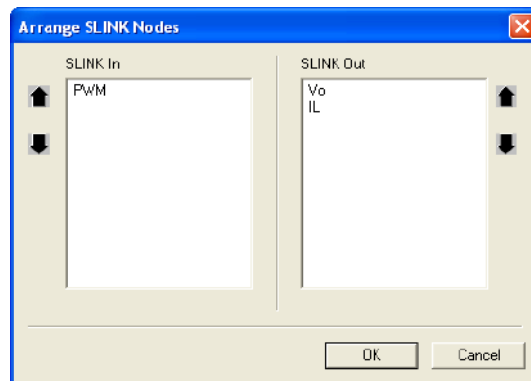


Figura 5.32: Entradas y Salidas del circuito en Psim.

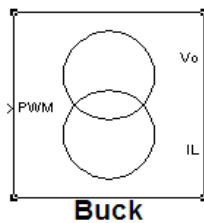


Figura 5.33: Bloque en Simulink con las entradas y salidas del circuito en Psim.

siguientes características:

- Procesador Intel Centrino Duo a 1.6 GHz.
- 2 Gb en Memoria Ram.
- 128 Mb en memoria de video compartida.

La figura 5.34 muestra una simulación del sistema final para un voltaje de referencia de 4V, se observa que en la respuesta transitoria no existe sobretiro, más sin embargo el tiempo de establecimiento es alrededor de 23 ms, cuestión que se pretende mejorar en los resultados experimentales. Asimismo se cosimuló el mismo sistema para un voltaje de referencia de 18 V (figura 5.35), donde se observa que se tiene un sobretiro máximo de 7.6 %, así como un error de 0.15 V. De acuerdo a estas simulaciones se procede a implementar el diseño en un FPGA con la tarjeta NEXYS2.

## 5.4. Resultados Experimentales

De acuerdo al diseño del convertidor realizado en 5.2, se tienen las siguientes características, voltaje de entrada  $E = 24V$ , inductor  $L = 1mH$ ,



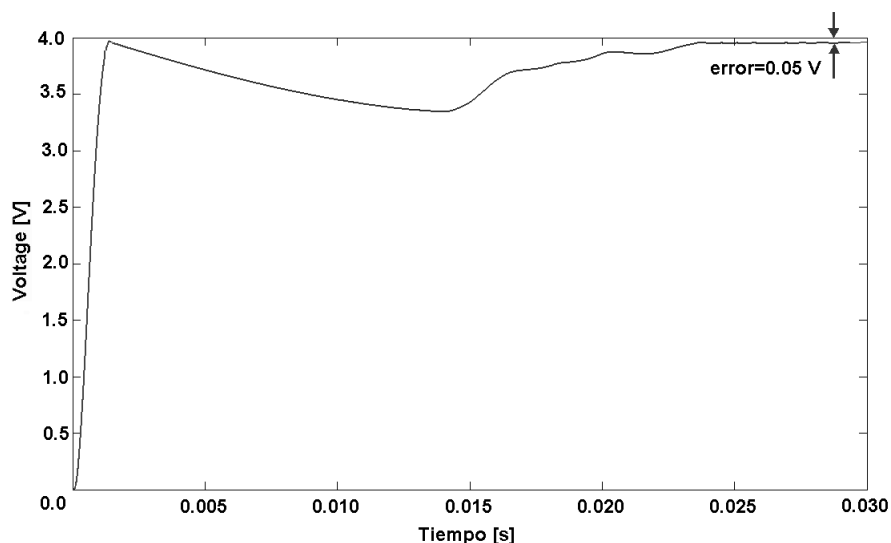


Figura 5.34: Voltaje de salida del convertidor reductor a un voltaje de referencia de 4 V en cosimulación.

capacitor  $C = 470\mu F$ , resistencia de salida  $R = 100\Omega$ , frecuencia de conmutación fija  $f = 48,828kHz$ , corriente de rizo en el inductor  $\Delta i_L = 0,092$ , voltaje de rizo en la salida del convertidor reductor  $\Delta V_o/V_o = 0,013$ , todo esto a un ciclo de trabajo  $D = 0,75$ .

Los resultados aquí mostrados se dividen en dos fases. La primera fase consta de observar el tiempo de establecimiento del convertidor GPI ante un voltaje deseado. La segunda fase consta de tres pruebas para observar el desempeño del controlador GPI propuesto, implementado en el FPGA. En las tres pruebas se somete al controlador a cambios bruscos en la carga de salida del convertidor reductor. La primera prueba consiste en conectar súbitamente una carga estática a la salida del convertidor reductor. La segunda prueba consiste en conectar repentinamente una carga dinámica a la carga de salida del convertidor, esta carga está representada por un motor de CD. Por último se realiza la tercera prueba donde al motor de CD se conecta además un freno mecánico con el fin de observar la robustez del convertidor implementado.

#### 5.4.1. Convertidor reductor a lazo abierto

Una vez diseñado el convertidor reductor se procede a observar el comportamiento del convertidor reductor a lazo abierto, es decir, introducir una señal cuadrada al MOSFET para obtener un voltaje de salida en la resistencia de carga, proporcional al ancho de pulso de la señal cuadrada.

En la figura 5.36 se observa la respuesta del convertidor reductor con una

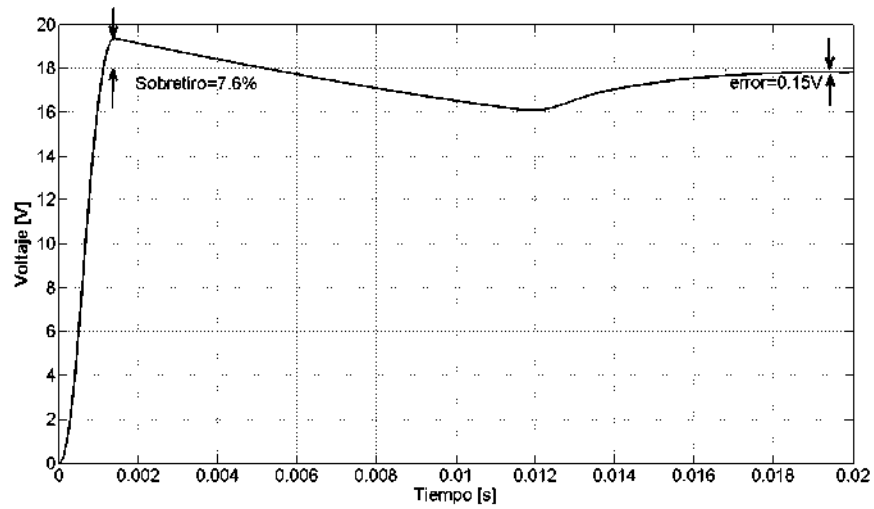


Figura 5.35: Voltaje de salida del convertidor reductor a un voltaje de referencia de 18 V en cosimulación.

señal de entrada con un ciclo de trabajo de 75 % lo que corresponde a tener un voltaje de salida de 18V, sin embargo se observa que la salida obtenida es inferior al voltaje de referencia por lo que se tiene un error de 1 V en estado estable. No existe un sobretiro significativo en la respuesta a lazo abierto, sin embargo no se llega al voltaje deseado. Asimismo se observa en la misma figura (5.36) un tiempo de establecimiento de 11.5 ms.

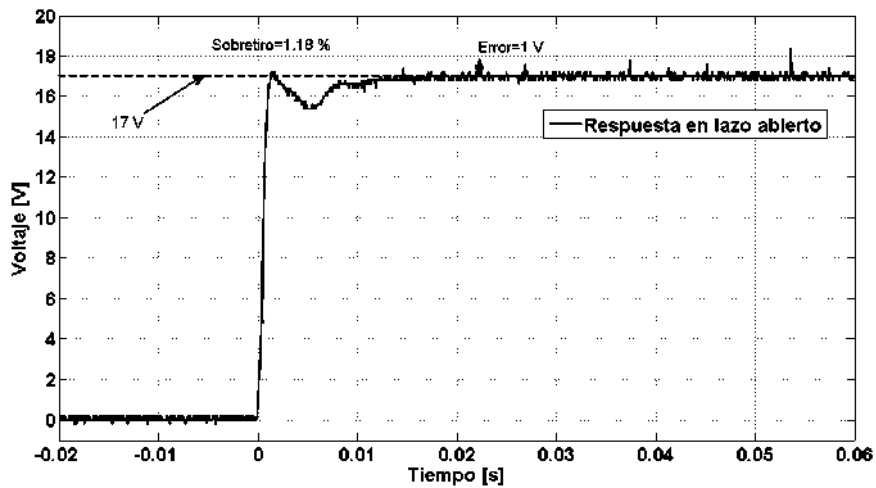


Figura 5.36: Respuesta en lazo abierto del convertidor reductor a un voltaje de referencia de 18 V.

Por otra parte se hizo otra prueba a lazo abierto, esta vez se configuró una señal cuadrada para obtener a la salida un voltaje de 12V, lo que corresponde a tener un ciclo de trabajo de 50 %. La gráfica de la figura 5.37 muestra la señal del voltaje de salida obtenida en el convertidor reductor, al igual que la primera prueba no se llega al voltaje deseado. Se observa claramente que en la gráfica obtenida se obtiene un sobretiro del 50 % y su valor máximo alcanzado es 10.4 V por lo que el error que se tiene en estado estable es de 1.6 V. El tiempo de establecimiento de esta respuesta es de 22.5 ms.

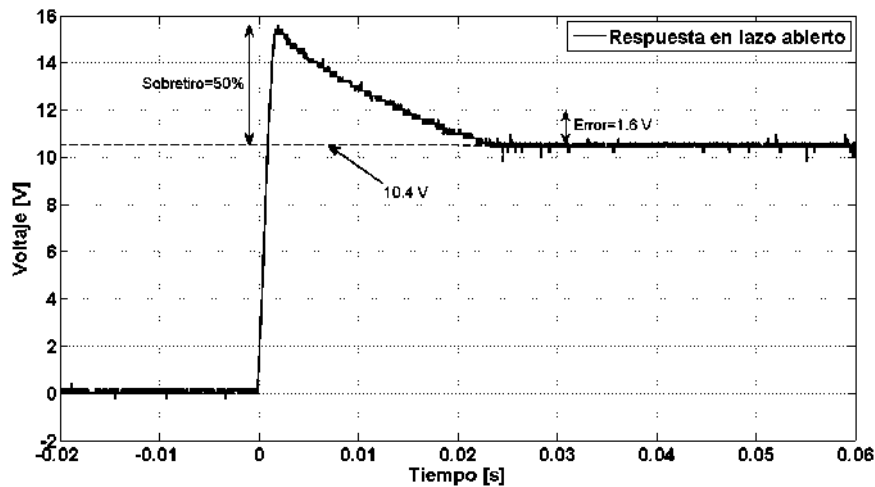


Figura 5.37: Respuesta en lazo abierto del convertidor reductor a un voltaje de referencia de 12 V.

En ambas respuestas a lazo abierto se observó que la señal en estado estable no alcanza el nivel de referencia deseado, el controlador diseñado debe ser capaz de corregir este problema y además de tener un tiempo de establecimiento menor, aunado a un sobretiro nulo o muy pequeño.

#### 5.4.2. Convertidor reductor a lazo cerrado con control GPI

La figura 5.38 muestra dos señales de salida del convertidor reductor. La primera gráfica muestra el voltaje de salida del convertidor reductor a un voltaje de referencia de 18V, donde se muestra que el tiempo de establecimiento de esta señal es de aproximadamente 4.64 ms, además esta señal presenta poco sobretiro. En la segunda respuesta se aprecia la señal  $u_{av}$  que es el voltaje de control que pasa al módulo de PWM para activar y desactivar al MOSFET y obtener a la salida el voltaje de referencia.

Una de las primeras pruebas que se hizo con el controlador diseñado e implementado en el FPGA consiste en conectar súbitamente una carga del mismo valor que la carga de salida del convertidor reductor, es decir, reducir

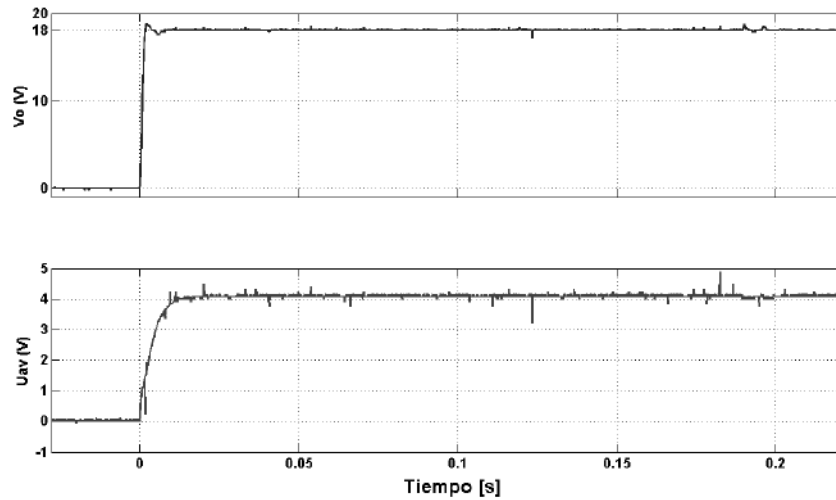


Figura 5.38: Respuesta del controlador GPI implementado en un FPGA.

la carga de salida en un 50%. Al hacer esto, el voltaje deseado debe de mantenerse para probar la robustez del sistema. En la figura 5.39 se muestra la respuesta del controlador al conectar súbitamente la carga estática; se observa que existe una pequeña perturbación en la salida del convertidor reductor, pero debido a la acción del controlador, el voltaje de salida se mantiene en el nivel de referencia sin presentar cambios significativos en su nivel.

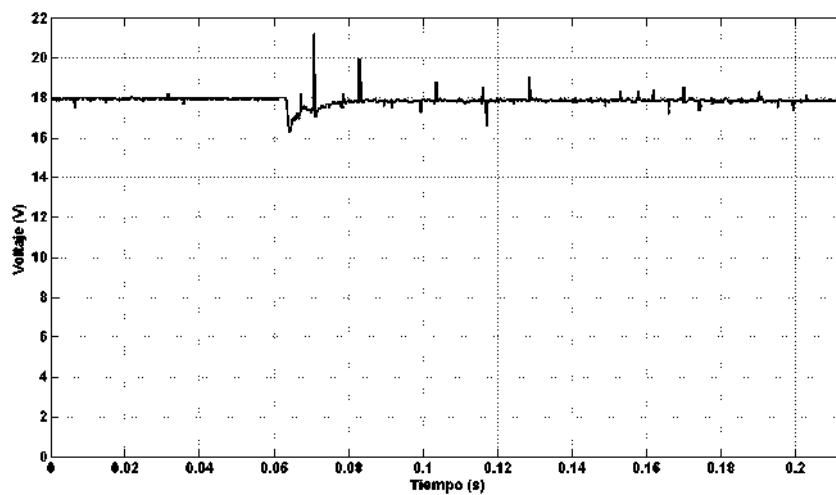


Figura 5.39: Desempeño del controlador GPI basado en FPGA al conectar súbitamente una carga estática a la salida del convertidor.

Una segunda prueba que se realizó para probar el rendimiento del controlador es conectar de manera súbita una carga dinámica. Esta carga dinámica está representada por un motor de CD que perturba al voltaje de salida del convertidor reductor. En la figura 5.40a se observa claramente que al conectar el motor en paralelo con la carga de salida, el voltaje de referencia es perturbado de manera significativa, sin embargo, la acción del control GPI es capaz de solventar esta perturbación y recuperar el voltaje de referencia deseado. Se observa en dicha figura que el tiempo de recuperación es de aproximadamente 50 ms. Esta recuperación de la señal se debe a que el control GPI incorpora valores estimados de la primera derivada del voltaje de salida ( $F$ ). Además se observa en la figura 5.40b el voltaje de la señal de control, que tiene como valor máximo 5V, permitiendo al control GPI recuperar la señal de referencia.

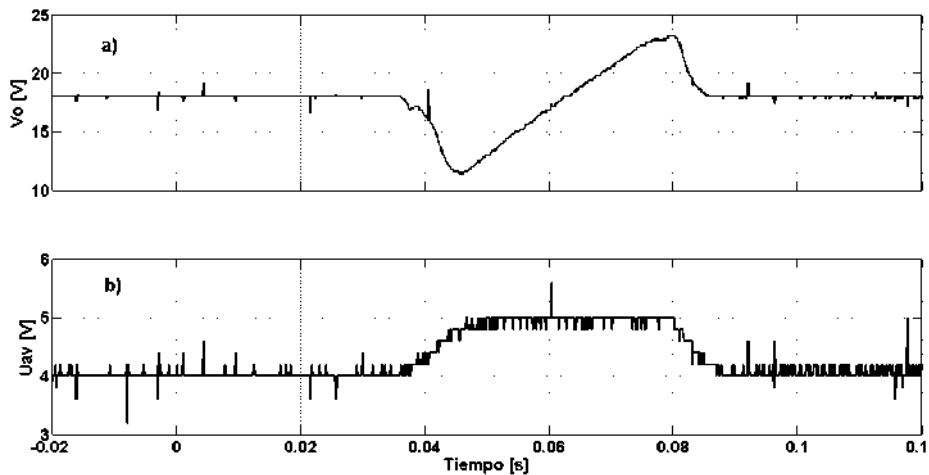


Figura 5.40: Respuesta del controlador GPI implementado en un FPGA al conectar súbitamente un motor de CD. a) Voltaje de referencia b) Señal de control  $u_{av}$ .

Una última prueba realizada para observar el desempeño del controlador GPI fue conectar el motor de CD en conjunto con un freno mecánico. En la figura 5.41 se observa una respuesta en el tiempo del voltaje de salida del convertidor reductor. Esta respuesta está dividida en 4 etapas, en primer lugar se activa el control GPI para obtener un voltaje de referencia de 18 V, motivo por lo cual el sistema demanda aproximadamente 180 mA. En la segunda etapa se conecta súbitamente una carga dinámica representada por el motor de CD, el sistema es perturbado pero la acción de control GPI recupera el voltaje de referencia. En esta etapa, el sistema demanda una corriente de 420 mA. En la tercera etapa se activa el freno mecánico para demandar mayor energía al sistema en la etapa de salida. Se observa

que el sistema demanda más corriente pero el voltaje de salida mantiene el nivel de referencia, la corriente demandada por el freno mecánico es de 620 mA, teniendo picos que superan los 2 amperes. Por último se apaga el freno mecánico y el motor con una perturbación despreciable en la variable de salida del convertidor.

El controlador GPI implementado se observa que es robusto ante perturbaciones al sistema, además de presentar una rápida respuesta dinámica y un tiempo de recuperación rápido. En la última respuesta se observa que el sistema solo controla el voltaje de salida del convertidor reductor, no sucede así con la señal de corriente, la cual se observa no es controlada con lo que su valor se incrementa o decrecienta según el valor de la carga conectada en la salida del convertidor reductor.

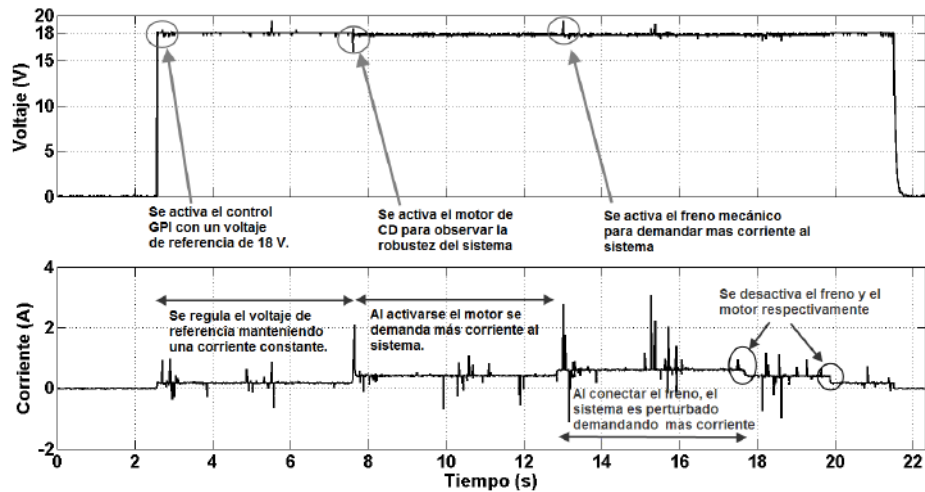


Figura 5.41: Desempeño del controlador GPI al conectar una carga dinámica y un freno mecánico.

### 5.4.3. Comparativa del controlador GPI y PID.

Una vez mostrado el funcionamiento del controlador GPI, es bueno comprobar su desempeño del sistema con otro tipo de controlador. En este caso se hará la comparativa con un controlador del tipo Proporcional Integral Derivativo (PID). El controlador PID es ampliamente utilizado en la industria, debido a las ventajas de costo/beneficio que ofrece este controlador. En la actualidad más del 95% de los controladores utilizados son del tipo PID. Estos controles son recomendados para procesos de control de bajo a medio orden [61]. Este controlador está conformado por tres tipos de acción, la parte proporcional al error, una parte proporcional a la integral del error que se basa en errores anteriores medidos, y la parte derivativa del error obtenido

que se encarga de valores predecidos del error, en [62, 63] se puede ver con más detalles las características de este controlador.

### 5.4.3.1. Estructura del controlador PID

La combinación de las acciones proporcional, integral y derivativa provee la siguiente función de transferencia del controlador PID.

$$F_{PID}(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (5.11)$$

donde  $K_p$  es la ganancia proporcional,  $T_i$  es la constante de tiempo de integración, y  $T_d$  es la constante de tiempo de derivación. Este controlador es aplicado al sistema convertidor reductor, tal como se muestra en el diagrama a bloques de la figura 5.42.

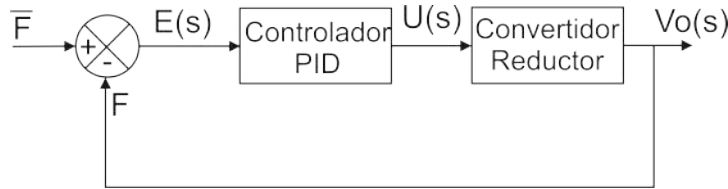


Figura 5.42: Diagrama a bloques del control PID en lazo cerrado.

La función de transferencia del sistema que se obtiene por la retroalimentación es:

$$H(s) = \frac{(K_p T_d T_i s^2 + K_p T_i s + K_p) \left(\frac{E}{LC}\right)}{s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right) s^2 + \frac{(1+EK_p)}{LC} s + \frac{EK_p}{LCT_i}} \quad (5.12)$$

De acuerdo a (5.12), el polinomio característico de la función de transferencia en lazo cerrado es:

$$s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right) s^2 + \frac{(1+EK_p)}{LC} s + \frac{EK_p}{LCT_i} = 0 \quad (5.13)$$

Los parámetros a sintonizar son  $K_p$ ,  $T_i$  y  $T_d$  de tal forma que el sistema sea óptimo. Una forma de sintonizarlo es que el polinomio característico (5.13) sea un polinomio Hurwitz de tercer orden de la forma:

$$p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + \alpha) \quad (5.14)$$

tomando en cuenta que  $\zeta$ ,  $\omega_n$  y  $\alpha$  son coeficientes positivos. Igualando los coeficientes del polinomio característico (5.13) con el polinomio Hurwitz de tercer orden (5.14), se obtienen los siguientes parámetros del controlador PID:

$$\begin{aligned}
K_p &= \frac{2\zeta\omega_n\alpha LC + \omega_n^2 LC - 1}{E} \\
T_i &= \frac{EK_p}{LC\alpha\omega_n^2} \\
T_d &= \frac{LC}{EK_p} \left( \alpha + 2\zeta\omega_n - \frac{1}{RC} \right)
\end{aligned} \tag{5.15}$$

De acuerdo a los requerimientos del convertidor reductor, se sintonizaron las variables del controlador PID de acuerdo a (5.15), con los valores de coeficientes  $\zeta = 0,707$ ,  $\alpha = 1000$  y una frecuencia natural de  $\omega_n = 2500$ . Con esto se tienen las siguientes cantidades:

$$\begin{aligned}
K_p &= 0,15 \\
T_i &= 1,2 \times 10^{-3} \\
T_d &= 5,9 \times 10^{-4}
\end{aligned}$$

#### 5.4.3.2. Comparación del controlador GPI y PID.

Una vez obtenido los parámetros del controlador PID, en primer lugar se simula el controlador mediante cosimulación utilizando las herramientas Matlab/Simulink y Psim. La figura 5.43 muestra el desempeño del controlador PID en simulación para un voltaje de referencia de 18 V. Se observa en dicha gráfica que no existe sobretiro, el tiempo de establecimiento es de aproximadamente 15 ms y el error en estado estable es cero. El control PID propuesto resulta ser adecuado y presenta un buen desempeño por lo que se procede a su implementación para la comparación con el control GPI.

Para la implementación del controlador PID se procede de manera similar que se hizo con el controlador GPI. El controlador PID junto con el módulo PWM se implementaron en el FPGA. La figura 5.44 muestra el diagrama a bloques del control PID implementado en el FPGA. Para las operaciones se utilizaron las mismas unidades en punto flotante utilizadas en el control GPI.

Antes de realizar la comparativa entre los dos controladores implementados es necesario definir que las comparaciones se hicieron entre los dos controladores sobre el desempeño en la respuesta transitoria del convertidor reductor y el desempeño de ambos controladores al momento de conectar cargas súbitamente a la salida del convertidor reductor, específicamente un motor de CD.

En la respuesta transitoria se compararon los siguientes aspectos:

1. **Máximo sobretiro.** Asíumase que  $v(t)$  es la señal de respuesta transitoria del convertidor reductor, y que  $v_{max}$  es el valor máximo de esta



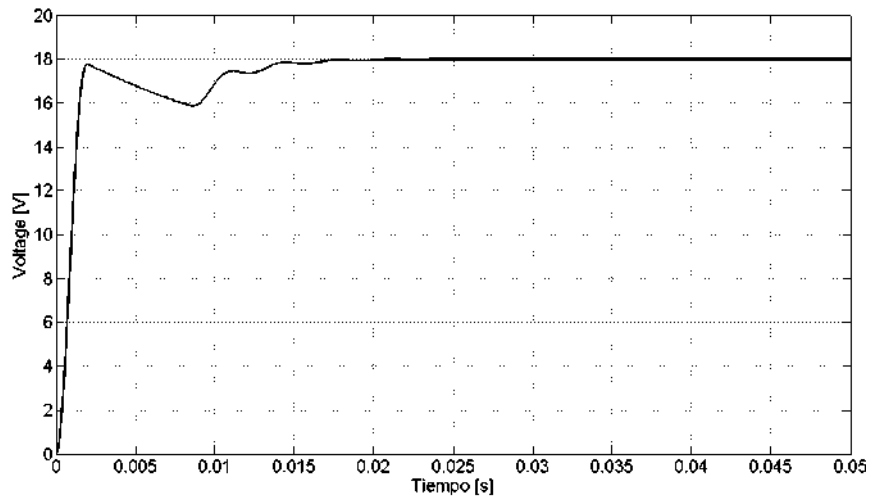


Figura 5.43: Desempeño del controlador PID para el convertidor reductor en cosimulación.

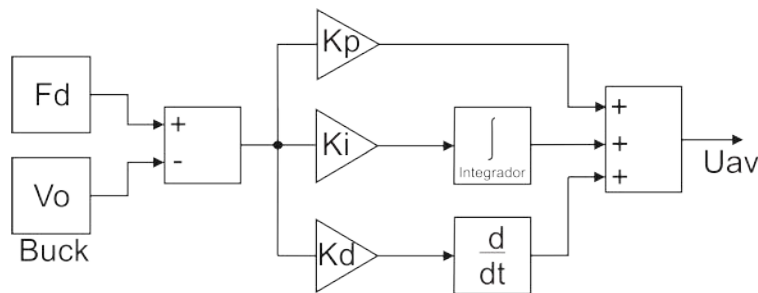


Figura 5.44: Diagrama a bloques del control PID implementado en FPGA.

respuesta  $v(t)$ , y  $v_o$  es el valor deseado o valor de referencia, además que  $v_{max} \geq v_o$ . El máximo sobretiro de la señal  $v(t)$  es:

$$\text{m\u00e1ximo sobretiro} = v_{max} - v_o$$

El m\u00e1ximo sobretiro generalmente se representa como porcentaje de la se\u00f1al en estado estable, esto es:

$$\text{porcentaje sobretiro} = M_p = \frac{\text{m\u00e1ximo sobretiro}}{v_o} * 100 \%$$

2. **Tiempo de retardo.** El tiempo de retardo  $t_d$  se define como el tiempo requerido para que la respuesta transitoria alcance el 50 % de su valor final.

3. **Tiempo de levantamiento.** El tiempo de levantamiento  $t_r$  es el tiempo requerido para que la señal de la respuesta transitoria se eleve del 10 % al 90 % de su valor final.
4. **Tiempo de establecimiento.** El tiempo de establecimiento  $t_s$  es el tiempo requerido para que la señal transitoria permanezca en un porcentaje pequeño de su valor final. Generalmente es del 2 %.
5. **Tiempo de pico.** El tiempo de pico  $t_p$ , es el tiempo en que la señal alcanza su valor máximo de la respuesta transitoria.

Una vez definido los parámetros a evaluar en la comparación se procede a revisar el desempeño de ambos controladores. Así en la figura 5.45 se muestra el desempeño de los controladores GPI y PID en la respuesta transitoria de la señal de voltaje de salida del convertidor reductor. Para este caso se utilizó un voltaje de referencia de 18 V.

La figura 5.45 muestra el desempeño de los controladores GPI y PID en la estabilización del voltaje de salida del convertidor reductor, la línea punteada muestra la señal escalón con un voltaje de 18 V. La gráfica en negro corresponde con la señal obtenida mediante el controlador GPI, mientras que la gráfica en color muestra la respuesta obtenida en el convertidor mediante el uso del control PID. El tiempo de establecimiento de la salida de voltaje de convertidor reductor a través del controlador PID es tres veces mayor que la señal obtenida con el control GPI. La salida de voltaje del convertidor reductor muestra un desempeño diferente que la dinámica obtenida por el controlador PID debido a la estructura del controlador, ya que el controlador GPI hace uso de una doble integral del error para la regulación. Como consecuencia, el controlador GPI fue sintonizado a través de un polinomio Hurwitz de cuarto orden, mientras que el esquema PID utilizó solamente un polinomio Hurwitz de tercer orden.

Por otro lado, para ilustrar la robustez del controlador GPI en comparación con el esquema PID, se realizó una prueba sometiendo al convertidor reductor a un cambio brusco en la salida mediante un cambio repentino de la carga. Esto se hizo mediante la conexión de un motor de CD. La figura 5.46 muestra el comportamiento del voltaje de salida del convertidor al conectar el motor de CD, cabe aclarar que en la salida del convertidor ya se tiene estabilizado el voltaje a 18 V. La señal en negro muestra la rápida recuperación de la salida de voltaje en la señal de referencia deseada, cuando el convertidor se controla mediante el esquema GPI. El controlador GPI además reduce el ruido que afecta la señal de voltaje en el convertidor. La reducción de ruido en la respuesta debido al controlador GPI es por la incorporación de valores estimados de la primera derivada de  $F$ , mientras que el controlador PID usa valores anteriores de  $F$  para obtener el valor de la primera derivada. En resumen, en la tabla 5.16 se muestran los puntos que

se consideraron para comparación de ambos esquemas de controladores GPI y PID.

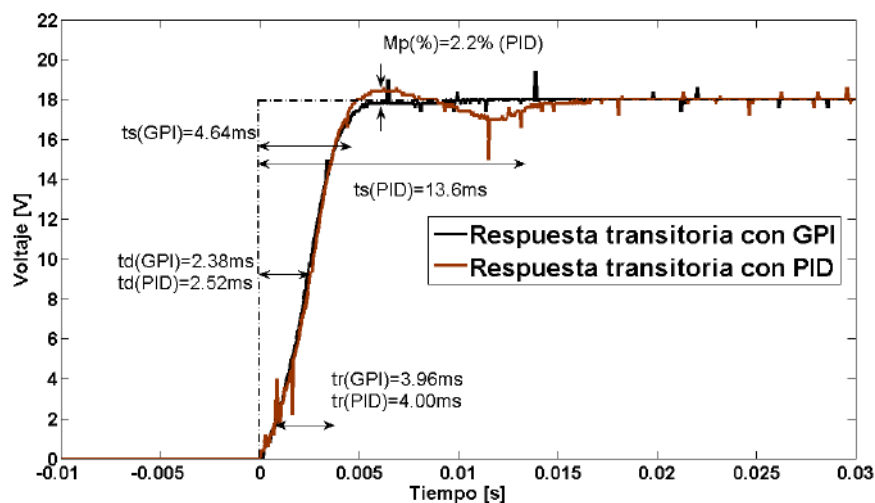


Figura 5.45: Respuesta transitoria del convertidor reductor con los esquemas de control GPI y PID.

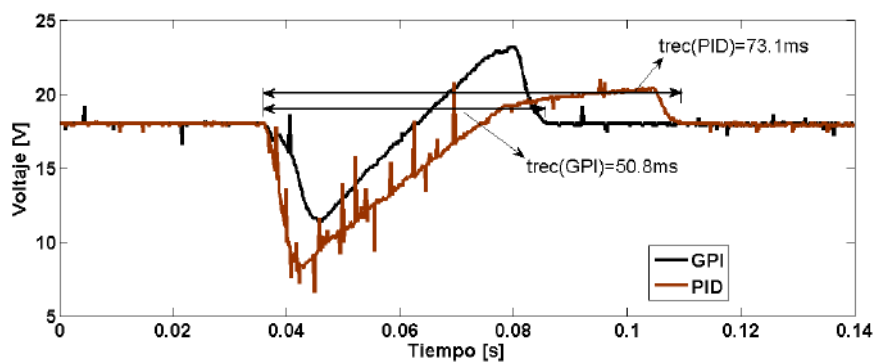


Figura 5.46: Respuesta de la señal de voltaje de salida al conectar súbitamente un motor de CD.

Tabla 5.16: Comparativa entre los controladores GPI y PID para el convertidor reductor.

Especificaciones de la respuesta transitoria	Controladores	
	GPI	PID
<b>Tiempo de retardo</b> $t_d$	2.38 ms	2.52 ms
<b>Tiempo de subida</b> $t_r$	3.96 ms	4 ms
<b>Tiempo de pico</b> $t_p$	–	6.24 ms
<b>Porcentaje de sobretiro</b> $M_p$	0 %	2.2 %
<b>Tiempo de establecimiento</b> $t_s$	4.64 ms	13.64 ms
<b>Tiempo de establecimiento</b> $t_{rec}$	50.8 ms	73.1 ms
<b>Ancho de banda (Apéndice E)</b> $\omega_{BW}$	1219.5 Hz	414.85 Hz

## Capítulo 6

# Conclusiones y Perspectivas

**RESUMEN:** En el presente proyecto se realizó el diseño de un controlador GPI en un FPGA para un convertidor CD-CD tipo reductor. El sistema fue implementado y probado en un FPGA NEXYS 2. En este capítulo se presentan las conclusiones obtenidas del desarrollo de este trabajo. Finalmente se presentan las perspectivas o trabajos futuros que darán seguimiento al trabajo.

### 6.1. Conclusiones

- Un estudio del estado del arte referente a controladores GPI y PID, diseñados e implementados para convertidores CD-CD, permitió ver el estado del conocimiento sobre controladores robustos que han sido implementados dentro de un FPGA, y con ello este trabajo planteó un controlador basado en FPGA de una técnica de control llamada GPI, que por vez primera ha sido implementada en un dispositivo FPGA.
- Se diseñaron y modelaron unidades en punto flotante para las operaciones básicas de suma, resta, multiplicación e integración, debido al amplio rango de valores requeridos para el controlador, y que se puede cubrir fácilmente con la utilización del punto flotante. Las unidades modeladas fueron simuladas, y se observó su rápida respuesta con diferentes valores de operandos.
- El uso de la metodología descendente (*Top-down*) para el diseño y modelado de las unidades de punto flotante, el controlador GPI y el módulo de PWM, permitió generar código reutilizable debido a su arquitectura modular. Con esto, el código se puede modificar fácilmente y le permite crecer añadiendo módulos de manera sencilla.

- Debido al uso de un lenguaje descriptor de hardware estandarizado como lo es VHDL en la descripción del controlador, permite que esta descripción sea portable, es decir, compatible con otras tecnologías de FPGAs y distintas herramientas CAD.
- Una de las características sobre la implementación del controlador GPI en el FPGA es que no se necesitaron muchos componentes externos. Debido a que el FPGA utilizado no cuenta con un convertidor analógico digital que permita medir el voltaje de salida del convertidor reductor, el convertidor ADC fue el único componente externo utilizado para interfazar al FPGA con la salida de voltaje del convertidor reductor.
- La herramienta utilizadas en la simulación y que permitió corregir muchos errores sobre el diseño de las unidades de punto flotante fue la cosimulación entre ModelSim y Matlab/Simulink, tal como se explica en el apéndice A. Con esta herramienta se tiene un mejor control y supervisión de las variables de las unidades modeladas, con lo cual se pueden observar y corregir errores en un menor tiempo. La única desventaja en este procedimiento es el tiempo de simulación requerido, cuando se tienen múltiples unidades en simulación.
- Otra herramienta de cosimulación que permitió observar en simulación la respuesta del controlador en el FPGA fue la cosimulación entre Psim y Matlab/Simulink, de manera que en Matlab/Simulink se realizó la simulación del controlador GPI y en Psim la simulación del circuito convertidor reductor. Una vez hecha esta cosimulación se procedió a simular las unidades modeladas en Modelsim y Matlab/Simulink y el circuito simulado en Psim, con lo cual se simuló el sistema completamente, obteniendo resultados similares a los obtenidos en la implementación.
- Ya que las unidades en punto flotante modeladas son portables, es fácil la implementación de otros tipos de control sobre el FPGA, tales controles pueden ser: control proporcional, proporcional-integral o incluso un control proporcional integral derivativo (PID).
- La implementación del controlador GPI permitió observar su desempeño en la respuesta transitoria del convertidor reductor para diferentes valores deseados del voltaje de salida. La respuesta transitoria no presentó demasiado sobretiro y se obtuvo una rápida estabilización de la señal de salida. Además se sometió al convertidor a cambios bruscos de carga para observar la robustez del controlador GPI. La primera prueba consistió en conectar súbitamente una carga estática para reducir la carga de salida del convertidor en un 50 %. La segunda prueba consistió en conectar una carga dinámica a la salida de convertidor, mediante un motor de CD. En ambos casos, el controlador GPI resultó

ser robusto, teniendo una buena respuesta para mantener la salida del convertidor reductor en el valor deseado.

- Con la implementación de un control del tipo PID se logra estabilizar la señal de salida del convertidor reductor con una muy buena respuesta transitoria obtenida mediante la sintonización de las variables del controlador, a través de un polinomio Hurwitz de tercer orden. La implementación de este control llevó poco tiempo debido a que se utilizaron las mismas unidades diseñadas y modeladas para el control GPI.
- La comparativa realizada entre los controladores GPI y PID implementados, permitió observar los desempeños de ambos controladores para regular la señal de salida del convertidor reductor. Mediante esta comparativa se observó que el controlador GPI presenta una mejor respuesta estática y dinámica, para estabilizar la señal de voltaje de salida del convertidor reductor y, presenta también una mejor respuesta cuando se conectan cargas dinámicas en la salida del convertidor. Esto se logra gracias a las características propias del controlador GPI, al presentar una estimación de los valores de la primera derivada del error.
- Este trabajo resultó ser novedoso debido a que no se había implementado este tipo de control en un FPGA. Es así, que de este trabajo surgen hasta el momento dos artículos ya publicados. La primer publicación tiene como título "Una comparativa en FPGA entre los controladores GPI y PID para la regulación del voltaje de salida en un convertidor CD-CD tipo reductor", publicado en el Congreso Internacional sobre Innovación y Desarrollo Tecnológico celebrado en la ciudad de Cuernavaca, Morelos, con ISBN:978-607-95255-2-1. La segunda publicación lleva por título "*A comparison between the GPI and the PID controllers for the stabilization of a DC-DC "buck" converter: A Field Programmable Gate Array implementation*", publicada en la revista *IEEE Transactions on Industrial Electronics*.

## 6.2. Perspectivas

Con la experiencia obtenida en el desarrollo del presente trabajo se proponen los siguientes trabajos futuros:

- Debido a que los valores de los parámetros del controlador GPI son calculados externamente para posteriormente ser introducidos en la herramienta CAD y sintetizar el diseño, se propone realizar un sistema de comunicación entre una computadora y el FPGA mediante

un módulo de comunicación USB, para actualizar los valores de los parámetros del controlador GPI de manera eficiente y más rápida.

- Modelado e implementación del controlador GPI para otras topologías de convertidores CD-CD.
- Se propone como trabajo futuro el análisis, diseño e implementación de un control GPI digital para la topología reductora del convertidor CD-CD. Así como también, la comparativa entre el control digital GPI con el control GPI diseñado de forma continua.



# Apéndice



## Apéndice A

# Cosimulación en Matlab/Simulink y ModelSim

**RESUMEN:** En este apéndice se muestran los pasos para poder realizar la cosimulación entre Matlab/Simulink y ModelSim. Para ello se realiza la simulación de una onda triangular generada en VHDL que posteriormente se simulará en ModelSim y se podrá ver la señal en un bloque osciloscopio de Simulink.

### A.1. Señal triangular en VHDL

El primer paso para poder cosimular en Matlab y ModelSim es obtener un diseño hardware modelado en VHDL. Para ello se hará uso del programa ModelSim para desarrollar el modelo en VHDL. Para este ejemplo se desarrollará un modelo que representa una onda triangular a una frecuencia aproximada de 48 kHz (figura A.1). La señal triangular estará representada por 8 bits, por lo tanto solo tendrá valores positivos de 0 a 255 y viceversa.

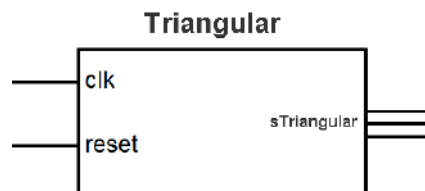


Figura A.1: Entidad en VHDL de una onda triangular.

La entidad del modelo VHDL que se genera contiene 2 puertos de entrada de tipo *STD\_LOGIC* que corresponden a un reloj de 50 MHz y la señal de reset. Asimismo contiene un puerto de salida de tipo *STD\_LOGIC\_VECTOR* con un tamaño de 8 bits.

Los pasos necesarios para poder generar el archivo VHDL son:

1. Ejecutar ModelSim.
2. Cambiar la ruta a una carpeta que se encuentre en la raíz, por ejemplo "C:\test". Si no existe la carpeta es necesario haberla creado antes  
*ModelSim>cd C:\test*
3. Abrir un nuevo archivo VHDL en la ventana de edición.
4. Copiar el código VHDL que aparece en la tabla A.1.
5. Guardar el archivo como *triangular.vhd*.

## A.2. Compilando el archivo VHDL

En esta parte se explica como crear una librería de diseño y compilar el archivo *triangular.vhd*, de la manera siguiente:

1. Verificar que el archivo *triangular.vhd* exista en el actual directorio, mediante el comando *ls* en la consola de ModelSim.
2. Crear una librería de diseño para mantener los resultados de simulación. Para crear la librería y el archivo requerido "\_info", introducir los comandos *vlib* y *vmap* como sigue:  
*ModelSim>vlib work*  
*ModelSim>vmap work work*
3. Si la librería de diseño (*work*) existe, ModelSim no sobrescribe la actual librería, pero despliega el siguiente mensaje:  
*# \*\* Warning: (vlib-34) Library already exists at "work".*
4. Compilar el archivo VHDL. Una forma de compilar el archivo es dándole clic derecho al archivo en la parte del área de trabajo del proyecto (*project workspace*) y seleccionar **Compile>Compile All**. Otra forma es especificar el nombre del archivo VHDL con el comando *vcom*, tal como se muestra a continuación:  
*ModelSim>vcom triangular.vhd*  
Si la compilación es exitosa, un mensaje de información aparece en la consola de ModelSim y el compilador liga la librería *work* con los resultados de la compilación, tal como se observa en la figura A.2.

Tabla A.1: Código en VHDL para la generación de una onda triangular

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity triangular is
    Port ( clk : in STD_LOGIC;
          rst : in STD_LOGIC;
          sTriangular: out STD_LOGIC_VECTOR(11 downto 0));
end triangular;
architecture Behavioral of triangular is
begin
    triangular:process(clk)
        variable aux: STD_LOGIC_VECTOR (11 downto 0):="000000000000";
        variable numDatos: integer range 0 to 1024:=0;
        variable edo: integer range 0 to 1024;
        begin
            if falling_edge(clk) then
                if rst='1' then
                    aux:="000000000000";
                    numDatos:=0;
                else --Señal Triangular
                    case numDatos is
                        when 0 to 511 =>
                            sTriangular<=aux;
                            aux:=aux+8;
                            numDatos:=numDatos+1;
                        when 512 =>
                            sTriangular<="111111111111";
                            aux:="111111111111";
                            numDatos:=numDatos+1;
                        when 513 to 1023 =>
                            sTriangular<=aux;
                            aux:=aux-8;
                            numDatos:=numDatos+1;
                        when 1024 =>
                            numDatos:=0;
                            aux:="000000000000";
                        end case;
                    end if;
                end if;
            end process;
        end Behavioral;

```

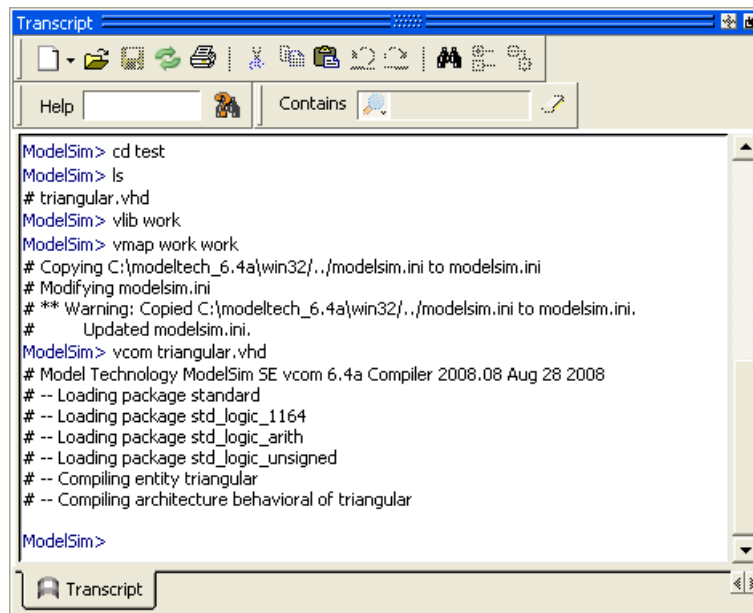


Figura A.2: Consola en ModelSim.

### A.3. Creando el Modelo en Simulink

Una vez compilado el archivo de VHDL se procede a crear el modelo Simulink. Para ello se realizan los siguientes pasos:

1. Ejecutar Matlab. Dentro de Matlab ejecutar el comando *Simulink*. Dentro de Simulink abrir la ventana *Simulink Library Browser*.
2. Arrastrar los siguientes bloques del explorador de librerías en Simulink:
  - Bloque *Step* de la librería de Simulink
  - Bloque de cosimulación HDL de la librería EDA Simulator Link MQ
  - Bloque *Data Type Conversion* de la librería de Simulink
  - Bloque *Scope* de la librería de Simulink
3. Alinear los bloques como se muestra en la figura A.3.



Figura A.3: Alineación de bloques para cosimulación.

Para configurar el bloque *Step* que fungirá como reset del sistema se da doble clic al bloque para entrar al cuadro de diálogo de las propiedades, y se tienen que introducir los siguientes valores:

- Tiempo de paso=60E-9 s
- Valor inicial =5
- Valor final=0
- Tiempo de muestra=10E-9

El cuadro de diálogo debe quedar como se muestra en la figura A.4

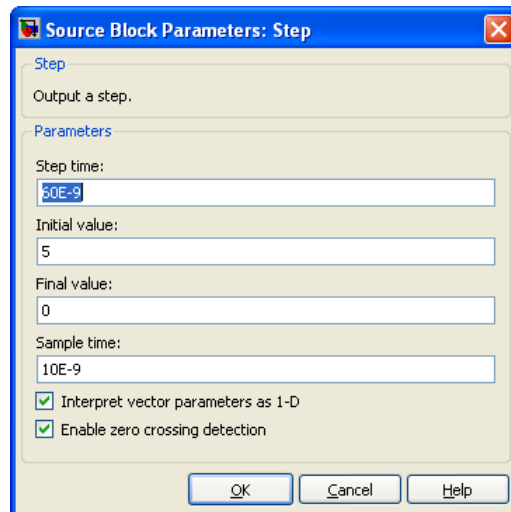


Figura A.4: Parámetros del bloque *Step*.

Debido a que la señal de reset funge como booleano para la entrada en el bloque de cosimulación, es necesario hacer la conversión de tipos de datos, esto se realiza mediante el bloque *Data Type Conversion*, que debe configurarse el parámetro:

- Tipo de dato de salida=Boolean

De tal forma que se tiene que acceder al cuadro de propiedades del bloque y quedar en la forma como se muestra en la figura A.5.

Ahora se configura el bloque de cosimulación HDL que representa el modelo en VHDL de la onda triangular. Empezaremos por los puertos de la manera siguiente:

1. Doble clic sobre el bloque *HDL Cosimulation*. El cuadro de diálogo de las propiedades del bloque aparece. Clic en la pestaña *Ports*.

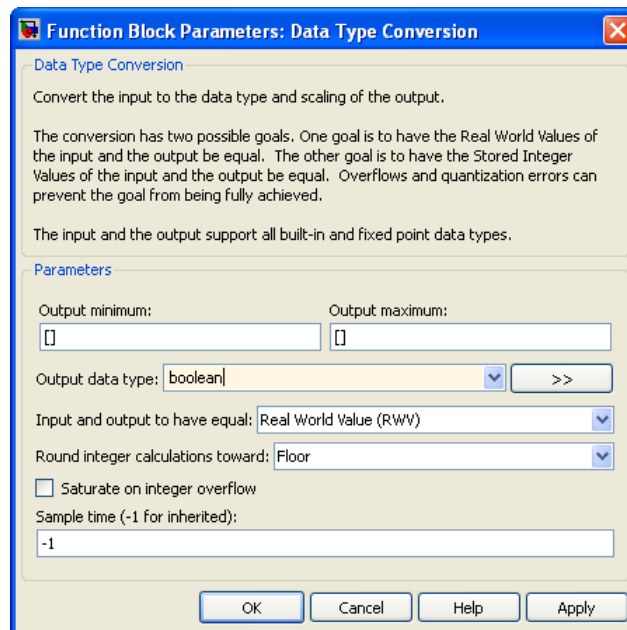


Figura A.5: Propiedades del bloque *Data type conversion*

2. En los cuadros de los puertos, seleccione la señal de muestra `/top/sig1` de la lista de señales y de doble clic sobre ella.
3. Reemplace el nombre de la señal `/top/sig1` por `/triangular/reset`. A continuación de clic en Aplicar, y el nombre de la señal cambia. De clic en Aplicar para guardar los cambios.
4. Seleccione la señal `/top/sig2` y reemplácela por `/triangular/sTriangular`. En el tipo de modo seleccione Output y elija el tiempo de muestra en `10E-9`.
5. Seleccione la señal `/top/sig3` y bórrela mediante el botón *Delete*. Los puertos deben aparecer como la figura A.6.

Ahora se configuran los parámetros de la Conexión, mediante los siguientes pasos:

1. Clic en la pestaña *Connection*.
2. Dejar la opción de modo de conexión como simulación completa.
3. Seleccione el método de socket. Esta opción indica que Simulink y ModelSim se comunicarán vía TCP/IP mediante un puerto socket dedicado.



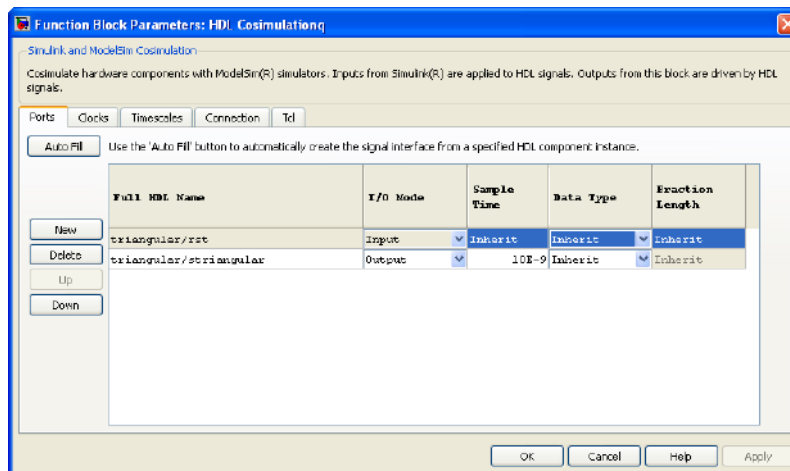


Figura A.6: Parámetros de la pestaña puertos del bloque *HDL Cosimulation*.

4. En el número de puerto o servicio anote el número 4449, o si este puerto se encuentra ocupado en su máquina elija alguno que se encuentre disponible, o bien un nombre de servicio. El panel de control deberá aparecer como lo muestra la figura A.7.

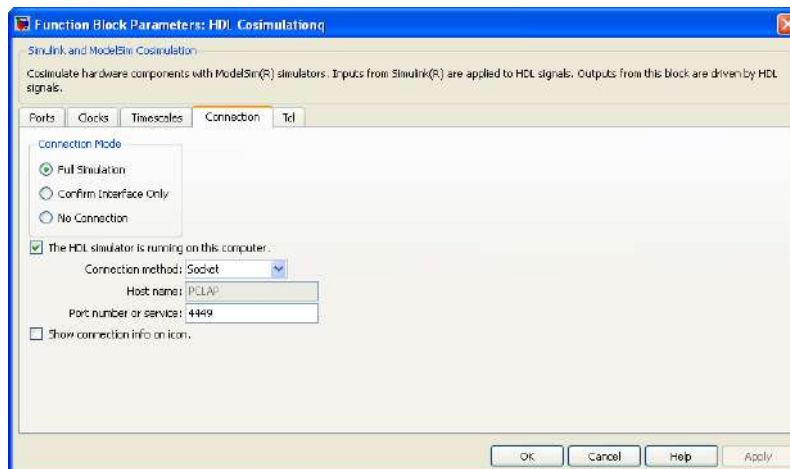


Figura A.7: Parámetros de la pestaña Conexión del bloque *HDL Cosimulation*.

5. Clic en Aplicar.

Lo que sigue es configurar la pestaña de reloj de la siguiente forma:

1. Clic en la pestaña *Clocks*.

2. Clic en el botón *New*. Se agrega una nueva señal vacía de reloj a la lista.
3. Doble clic en la nueva señal para editarla. Asigne la ruta `/triangularr/clk`. Ahora seleccione *falling\_edge*. Configure el periodo en 20ns que corresponde a una frecuencia de 50 MHz.
4. El cuadro de diálogo debe aparecer tal como la figura A.8.

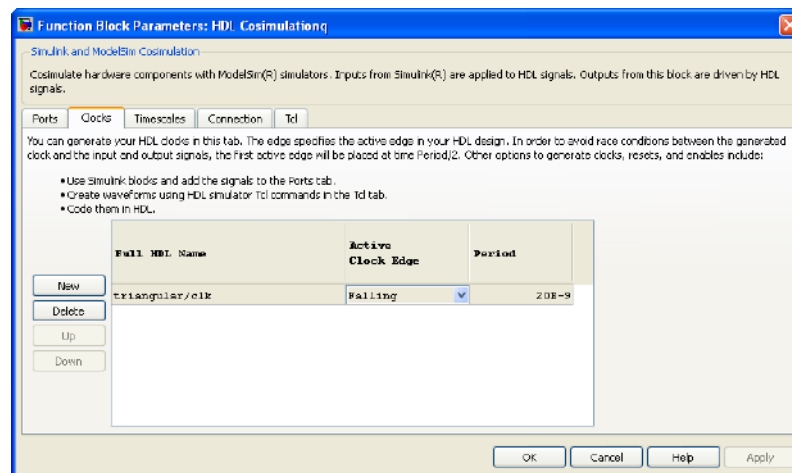


Figura A.8: Parámetros de la pestaña reloj del bloque *HDL Cosimulation*.

5. Clic en Aplicar.

Por último se configura la pestaña de escalas de tiempos, para ello se hace:

1. Clic en la pestaña *Timescales*.
2. Seleccione la configuración que 1 segundo en Simulink corresponda a 1 segundo en el simulador HDL, de tal forma que quede configurado como lo muestra la figura A.9.
3. De clic en OK para cerrar el cuadro de diálogo.

El último paso es conectar los bloques y guardar el modelo, esto mediante los siguientes pasos:

1. Conectar los bloques en la forma como lo muestra la figura A.10.
2. Configurar los parámetros de simulación, de acuerdo a la configuración mostrada en la figura A.11.
3. Guardar el modelo.

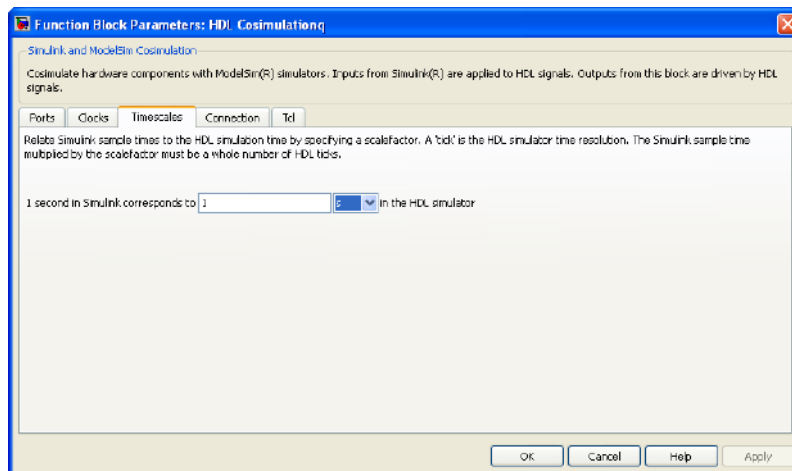


Figura A.9: Parámetros de la pestaña de escalas de tiempo del bloque *HDL Cosimulation*.



Figura A.10: Conexión de los bloques del modelo en Simulink.

## A.4. Configurando ModelSim para trabajar con Simulink

Ahora ya se tiene una representación del modelo de una onda triangular. Para ejecutar ModelSim para trabajar con Simulink se debe introducir el siguiente comando en la ventana de comandos de Matlab:

```
vsim('socketsimulink', 4449)
```

Si se configuró otro número de puerto, se debe de introducir en lugar del número 4449.

## A.5. Cargando instancias de la entidad VHDL para cosimulación con Simulink

En esta sección se explicará como usar el comando *vsimulink* para cargar instancias de VHDL, para la cosimulación con Simulink. Para cargar instancias, se procede con los siguientes pasos:

1. Cambiar el directorio de ModelSim por el directorio donde se encuentra el archivo *triangular.vhd*. Por ejemplo  

```
ModelSim>cd C:\test
```

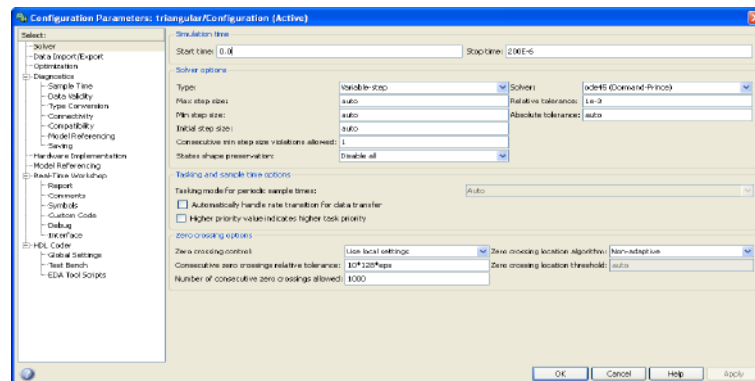


Figura A.11: Configuración de parámetros de simulación.

## 2. Introduzca el siguiente comando

```
ModelSim>vsimulink work.triangular
```

Modelsim inicializa el simulador y se encuentra ahora listo para empezar a simular la entidad triangular en el contexto del modelo de Simulink. El comando de ModelSim despliega mensajes similares a los de la figura A.12.

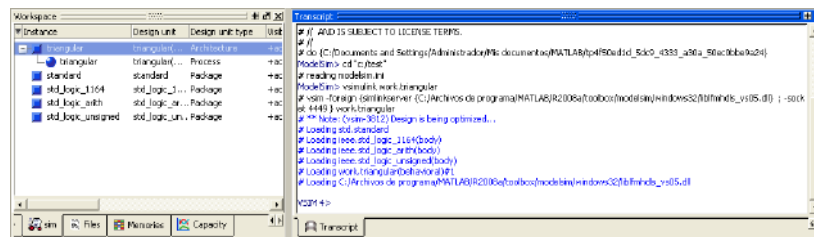


Figura A.12: Mensajes al cargar la entidad del modelo triangular.

## A.6. Ejecutando la simulación

Esta sección guía a través del escenario para ejecutar y monitorear la simulación. Para ello se siguen los siguientes pasos:

### 1. Abrir y añadir las señales de la entidad triangular a una ventana de señales mediante el siguiente comando:

```
VSIM n>add wave /triangular/*
```

Aparece una ventana con la señales, como lo muestra la figura A.13.

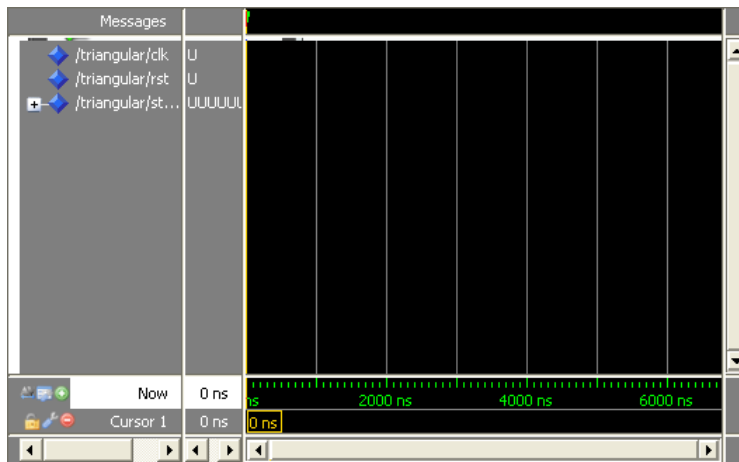


Figura A.13: Señales de la entidad triangular.

2. Ejecute la simulación en Simulink. El valor de la señal *sTriangular* empieza a incrementar y decrementar. Si es necesario puede hacer un acercamiento sobre la señal para poder verla a detalle. Las señales en ModelSim se verán parecido a las mostradas en la figura A.14.

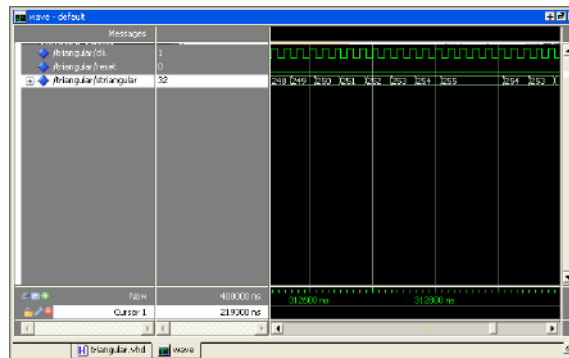


Figura A.14: Señales simuladas en ModelSim.

3. Por último se puede visualizar el resultado en Simulink de la onda triangular, para ello podemos acceder al visor de señales *Scope*, en la cual aparecerá la onda triangular generada, como se muestra en la figura A.15.

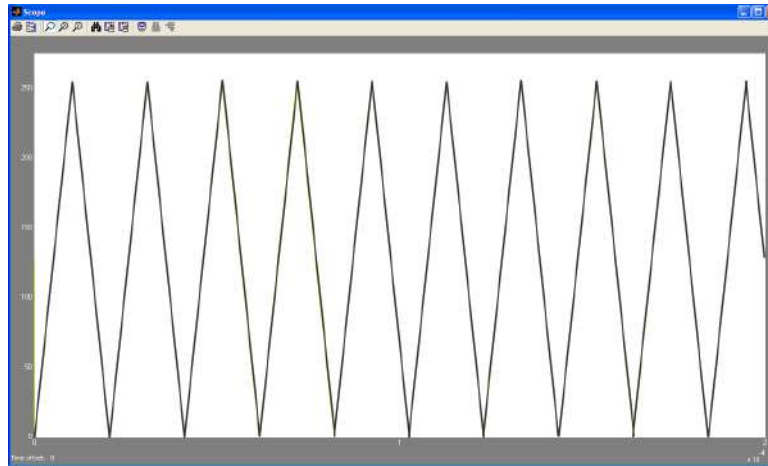


Figura A.15: Señal triangular obtenida en Simulink.

## A.7. Terminando la simulación

En esta sección se dan los pasos necesarios para terminar la simulación, para ello se realizan los siguientes pasos:

1. En ModelSim, se termina la simulación seleccionando el menú *Simulate* y la opción *End Simulation*.
2. Cerrar ModelSim.
3. Cerrar la ventana del modelo en Simulink.

## Apéndice B

# Cosimulación en Matlab/Simulink y Psim

**RESUMEN:** En este apéndice se muestran los pasos para poder realizar la cosimulación entre Matlab Simulink y Psim. Para ello se realiza una cosimulación a lazo abierto del circuito convertidor reductor en Psim, y en Matlab/Simulink se muestra el voltaje medido y se da el valor deseado del voltaje de salida.

### B.1. Introducción

El módulo *SimCoupler* en Psim es una aplicación que viene incluida en el programa que provee una interfaz entre Psim y Matlab/Simulink para cosimulación. Con el módulo SimCoupler se puede simular una parte del sistema en Psim, y el resto en Matlab/Simulink, con esto fácilmente se puede simular un circuito en Psim, y combinar las herramientas que existen en Simulink para realizar diversas cosimulaciones. El bloque *SimCoupler* no viene agregado en Matlab/Simulink, una vez instalado Matlab, se procede a instalar Psim y automáticamente se agrega el bloque *SimCoupler* a Matlab/Simulink.

El módulo de SimCoupler consiste de dos partes, los nodos de entrada/salida en Psim, y el bloque de modelo *SimCoupler* en Simulink, a continuación se muestran los pasos necesarios para realizar una simulación utilizando el módulo *SimCoupler*. Para ello se realiza la simulación a lazo abierto del circuito convertidor reductor. El circuito junto con el PWM serán simulados en Psim, y en Matlab/Simulink solo se observará la señal de voltaje proveniente de la salida del convertidor reductor y además, en el mismo diagrama de Simulink se proveerá la salida deseada, es decir  $u_{av}$ .

## B.2. Creación del circuito en Psim

Para poder cosimular un circuito con Psim y Matlab/Simulink se procede con los siguientes pasos:

1. Crear el circuito en Psim de acuerdo al sistema que se quiera simular; en este caso se realizará la cosimulación de un circuito convertidor reductor tal como se muestra en la figura B.1.

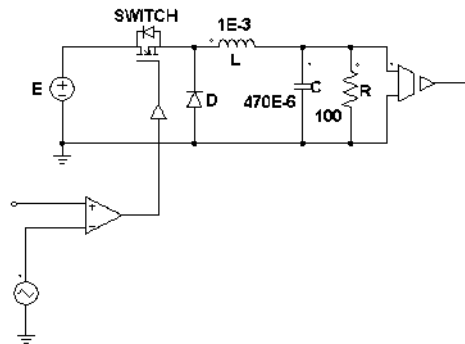


Figura B.1: Circuito del convertidor reductor en Psim, utilizado para simulación.

2. Conectar el nodo *OUT SLINK* al sensor de voltaje en la parte de salida del convertidor reductor y renómbrelo como " $V_o$ ".
3. Conectar un nodo *IN SLINK* a la entrada positiva del comparador en el PWM y renómbrelo como " $U_{av}$ ". El circuito debe quedar tal como se muestra en la figura C.2.

## B.3. Creación del bloque en Matlab/Simulink

La cosimulación de Psim con Matlab/Simulink requiere la creación de un diagrama en Simulink, para ello se realizan los siguientes pasos:

1. Abrir Matlab/Simulink y crear un modelo nuevo. Agregar los bloques *Constant*, *SimCoupler* y *Scope* al nuevo modelo. El bloque *Constant* permitirá configurar la señal  $u_{av}$  que controlará la salida del convertidor reductor. El bloque *SimCoupler* creará la interfaz entre Matlab/Simulink y Psim para comunicar las señales entre ambos programas. Por último el bloque *Scope* permitirá observar la señal de voltaje de salida del convertidor reductor proveniente de Psim.(Figura B.2)
2. Doble clic sobre el bloque *Constant* para acceder a sus propiedades. Configurar el valor constante a 0.5 para obtener una salida del 50 %



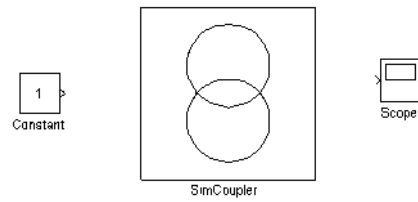


Figura B.2: Bloques utilizados en Simulink para cosimulación.

del valor de entrada en el convertidor reductor, es decir, ya que la entrada es de 24V, se espera tener una salida de 12V.

3. Doble clic sobre el bloque *SimCoupler*. Aparecerá una ventana de diálogo donde debemos de configurar la ruta del archivo en Psim para cosimulación, tal como se muestra en la figura B.3. Para definir la ruta del archivo de Psim, se da clic izquierdo en el botón *Browse...* y en la ventana que aparece ir a la ruta donde se encuentra el archivo y seleccionarlo. Se presiona el botón *Apply* y aparece una nueva ventana mostrando los nodos de entrada y salida del módulo *SimCoupler* que existen en el circuito en Psim (Ver figura B.4). Si todo está configurado correctamente dar clic en el botón *OK* de la nueva ventana, y cerrar la ventana de propiedades del bloque *SimCoupler*. El bloque *SimCoupler* se actualiza con las entradas y salidas existentes en el circuito en Psim (Figura B.5).

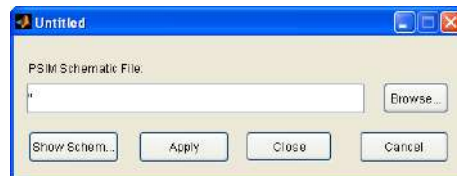


Figura B.3: Cuadro de diálogo para configurar la ruta del archivo en Psim.

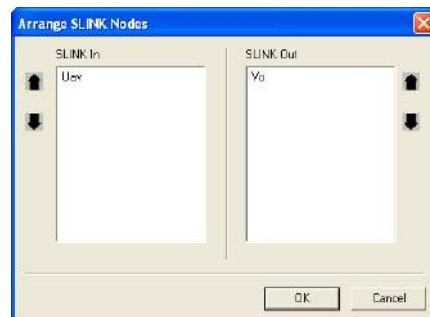


Figura B.4: Nodos de entrada y salida del módulo *SimCoupler*.

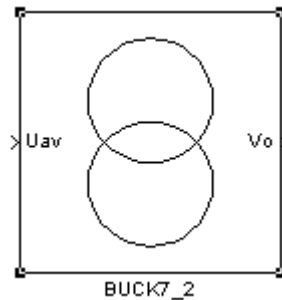


Figura B.5: Bloque *SimCoupler* conectado con Psim.

4. Conectar los bloques en el modelo tal como se muestra en la figura B.6.

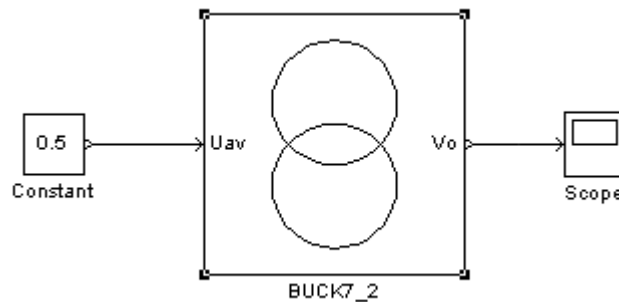


Figura B.6: Conexión de bloques en el modelo.

5. Ir a los parámetros de simulación del modelo (Menu *Simulation-Configuration Parameters*), y configurar el tipo de solución en *Fixed-step*, y el paso de tiempo de  $1 * 10^{-5}$  (figura B.7). El paso de tiempo debe estar sincronizado en Psim, por lo que en el circuito también debe configurarse este mismo paso de tiempo.
6. En dado caso de existir problemas por tiempos de muestreo en el módulo SimCoupler en Simulink, se puede agregar un bloque de ganancia (*Gain*) entre en bloque de la constante y el módulo SimCoupler (Figura B.8). Este bloque de ganancia se configurará con una ganancia de 1 y un tiempo de muestreo de 0.
7. Configurar el tiempo de simulación a 200 ms y correr la simulación. La figura B.9 muestra el voltaje de salida obtenido en Psim y mostrado en Matlab/Simulink con ayuda del bloque *Scope*. Con esto se pueden hacer diferentes aplicaciones sobre control, simulando el circuito en Psim y, la acción de control en Matlab/Simulink.

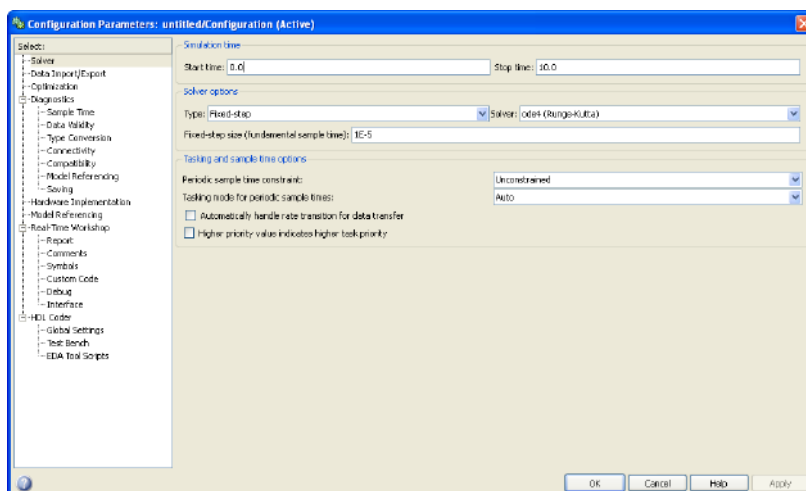


Figura B.7: Parametros de simulación del modelo.

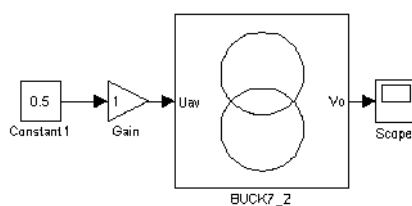


Figura B.8: Bloque de ganancia agregado para evitar problemas con el bloque de *SimCoupler*.

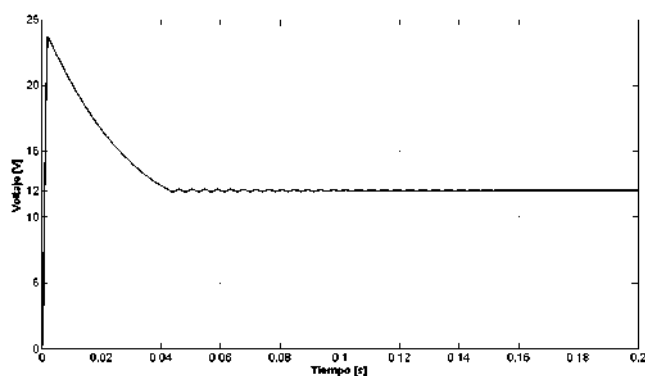


Figura B.9: Resultado de la cosimulación entre Psim y Matlab/Simulink del convertidor reductor.



## Apéndice C

# Diagramas de simulación Simulink

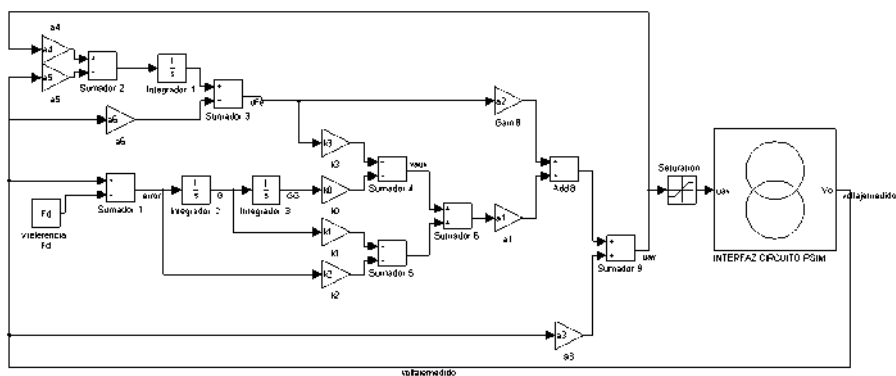


Figura C.1: Diagrama en Simulink para simulación del control GPI con Psim.

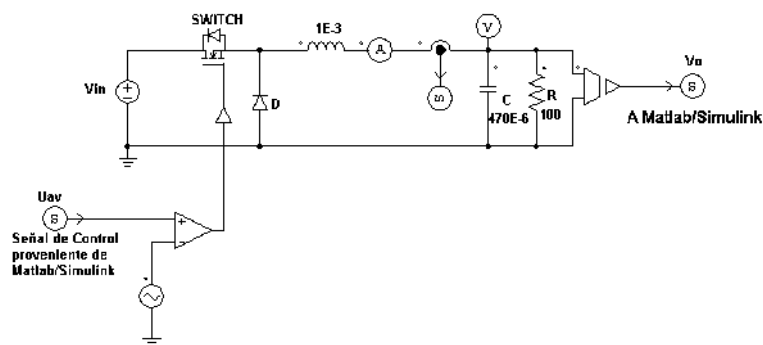


Figura C.2: Circuito del convertidor reductor en Psim, utilizado para cosimulación.



## Apéndice D

### Circuito Impreso

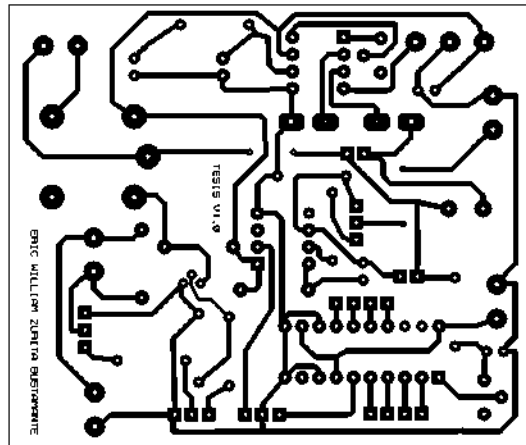


Figura D.1: PCB del convertidor reductor y convertidor analógico digital.

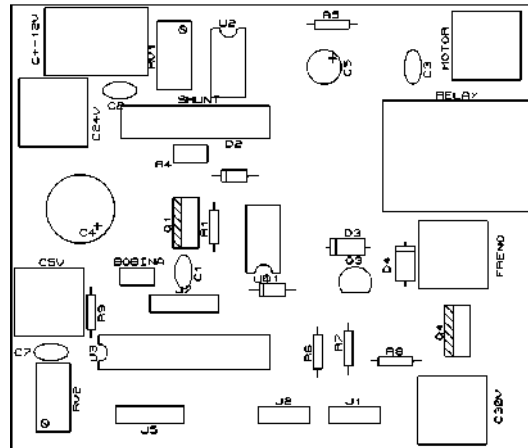


Figura D.2: Guía de montaje de la placa de circuito impreso (*Top silk*).

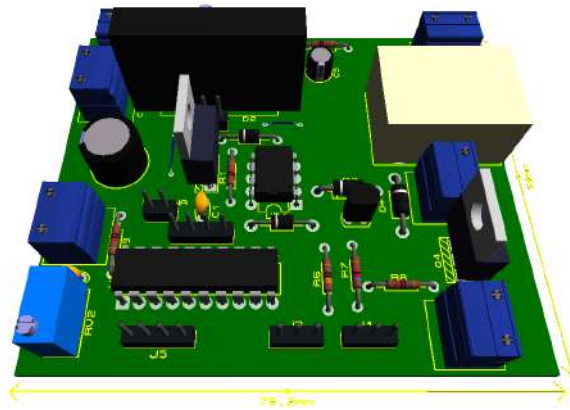


Figura D.3: Vista 3D de la placa de circuito impreso



## Apéndice E

### Artículos publicados

**RESUMEN:** Este apéndice muestra dos publicaciones aceptadas para su publicación que fueron escritas a partir del presente trabajo de investigación. El primer artículo fue aceptado en el Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET 2010, realizado en la ciudad de Cuernava, Morelos. El segundo artículo fue aceptado para su publicación en la revista IEEE Transactions on Industrial Electronics en Febrero del presente año.

---



## Una comparativa en FPGA entre los controladores GPI y PID para la regulación del voltaje de salida en un convertidor CD-CD tipo reductor.

E. W. Zurita Bustamante, J. Linares Flores, E. Guzmán Ramírez

**Resumen:** En este artículo se presenta la comparación de dos controladores promedio implementados en un Arreglo de compuertas programables en campo (FPGA). Los controladores implementados corresponden a un controlador PI Generalizado (GPI) y un controlador Proporcional Integral Derivativo (PID). El controlador GPI presenta una mejor respuesta dinámica que el controlador PID, en cuanto al tiempo de establecimiento para un voltaje de referencia constante y, también presenta una mayor robustez ante el rechazo de perturbaciones para mantener el voltaje de referencia a su salida, cuando al convertidor se le somete a cambios súbitos de carga estática y carga dinámica. Los controladores promedio y sus respectivos moduladores por ancho de pulsos (PWM) son implementados usando una tarjeta NEXYS2 SPARTAN3E1200.

**Abstract:** This paper presents a comparison between two stabilizing average output feedback controllers, implemented on Field Programmable Gate Array (FPGA) facility, for the regulation of the "buck" converter output voltage. A Generalized Proportional Integral (GPI) controller and a Proportional Integral Derivative (PID) controller are implemented using a FPGA and their respective performances duly compared. The GPI controller is found to present a better dynamic response than the PID controller in terms of the settling time while exhibiting a greater degree of robustness regarding disturbance rejection represented by severe changes in static and dynamic loads. The average controllers and their corresponding Pulse Width Modulation (PWM) actuators are implemented using a NEXYS2 Spartan3E1200 FPGA.

**Palabras Clave:** Convertidor reductor, FPGA, PID, GPI.

### Introducción

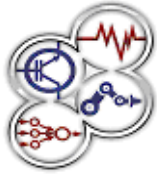
Una infinidad de aplicaciones industriales se hacen a través de los convertidores de potencia del tipo CD-CD, como por ejemplo en aplicaciones de fuentes de alimentación de potencia ininterrumpibles [1], en impulsores de motores CD para tracción eléctrica en los trolebuses [2], en sistemas de iluminación que usan balastos electrónicos [3], etc. Hoy en día los dispositivos de conmutación rápida están disponibles en el mercado con altas velocidades de conmutación y altas capacidades en el manejo de potencia. Ahora es factible diseñar fuentes de alimentación de potencia por arriba del 90% de eficiencia a un bajo costo y tamaño pequeño, además de un peso muy ligero [4-5-6]. Las topologías clásicas de convertidores CD-CD son tres: "Buck", "Boost" y el "Buck-Boost".

El presente trabajo aborda el control de voltaje de salida del convertidor reductor mediante la técnica GPI. Esta técnica de control se ha venido utilizando en los últimos años para el control de convertidores de potencia del tipo CD-CD, debido a que presenta las siguientes características en su desempeño: rápida respuesta dinámica, robustez ante perturbaciones del tipo constante y rampa desconocidas. Además, mediante esta técnica se reduce de forma considerable el uso de sensores en la medición de los estados del sistema, ya que a través de los reconstructores integrales que conforman el controlador se estiman las variables de estado no medidas del sistema [7]-[8]. Así podemos observar en la referencia [9] la utilización de esta técnica GPI para la regulación del voltaje de salida del convertidor reductor basado en "Switched-Capacitor". Este trabajo presenta resultados en simulación de la respuesta transitoria del voltaje de salida del convertidor para una entrada escalón unitario, además muestran la robustez del controlador ante cambios súbitos de carga, es decir, al disminuir el valor de la carga inicial en un 50%. Mientras, en el trabajo reportado por Franco-González, *et al* [7] diseñan un

Eric William Zurita Bustamante, Estudiante de Maestría, División de Estudios de Posgrado, Universidad Tecnológica de la Mixteca, ie031459@ndikandi.utm.mx

Jesús Linares Flores, Instituto de Electrónica y Mecatrónica, Universidad Tecnológica de la Mixteca, jlinares@mixteco.utm.mx

Enrique Guzmán Ramírez, Instituto de Electrónica y Mecatrónica, Universidad Tecnológica de la Mixteca, eguzman@mixteco.utm.mx



controlador multivariable GPI en combinación con un control por modos deslizantes para la regulación de los voltajes de salida de los convertidores del sistema "Boost-Boost" usando solamente mediciones de los voltajes en los capacitores. Además, la robustez del control se prueba conectando súbitamente cargas estáticas no modeladas, en paralelo a la resistencia de carga final. De los resultados experimentales reportados por este artículo muestran perfectamente el desempeño del controlador ante el rechazo de perturbaciones desconocidas por el sistema. La implementación física del controlador multivariable la realizan a través de una computadora y una tarjeta de adquisición de datos. En el trabajo publicado en [10], se presentan resultados a nivel simulación del desempeño del controlador GPI en combinación con un modulador sigma-delta para el control directo de velocidad angular del sistema convertidor "reductor-motor CD". En este trabajo únicamente se monitorea la velocidad angular del sistema, y se consigue arrancar al motor de forma suave. La robustez del controlador del sistema se prueba variando el par de carga del eje del motor, en donde se observa una excelente recuperación de la velocidad de referencia en el eje del motor.

Referente a controladores robustos implementados en FPGA para la regulación del voltaje de salida del convertidor reductor, se tiene el trabajo reportado en [11] el cual presenta la comparativa entre dos controladores promedio del tipo PID. El primero corresponde a un PID clásico, mientras que el segundo controlador corresponde a un controlador PID no lineal. Los resultados experimentales muestran que el controlador PID no-lineal presenta una mejor respuesta dinámica en la variable de salida y un menor tiempo de recuperación de la señal de referencia. Por otro lado Yanxia Gao *et al* [12], presentan el diseño de un control PID clásico y un Modulador digital por ancho de pulso (DPWM) como módulo principal, para un convertidor reductor. Para verificar la eficacia del DPWM, se implementa el DPWM de 11 bits en un FPGA. Los resultados experimentales a una frecuencia fija de 2 MHz muestran la funcionalidad del DPWM, al obtener una rápida respuesta transitoria, y un control robusto ante cambios súbitos en la carga. La implementación la realizan en un FPGA Virtex-II Pro XC2VP30. Finalmente en [13] se muestra la implementación de un estimador de eficiencia en control digital aplicado a un convertidor reductor, basado en una estimación

indirecta de la razón entre la corriente de entrada y salida del convertidor, usando solo un sensor de corriente en el capacitor del convertidor reductor. Los resultados de la implementación muestran que el máximo error obtenido en el estimador es de 5%. En este artículo nos enfocaremos en la comparación de dos controladores promedio implementados en el FPGA, para regular el voltaje a la salida del convertidor CD-CD tipo reductor. El primer controlador implementado es un control GPI, mientras que el segundo controlador corresponde a un control PID. La plataforma experimental en donde se implementaron las leyes de control GPI y PID, presenta las siguientes características:

El modulador por ancho de pulsos se diseña a través de una señal portadora del tipo triangular y un comparador. La principal función de este modulador es la conversión de la señal promedio del controlador a una señal pulsante que activa y desactiva al transistor de potencia del convertidor a una frecuencia fija de 48kHz.

El tiempo de procesamiento del control GPI es de 39.2 $\mu$ s, mientras que el tiempo de procesamiento del control PID es de 20.54 $\mu$ s. Estos tiempos de procesamiento se consiguieron gracias a la ejecución paralela de unidades modeladas dentro de un FPGA. Con el avance de los PLD (*Programmable Logic Device*) como son los CPLD o FPGA se pueden realizar sistemas de control sin la necesidad de un CPU o un DSP [14].

El voltaje de salida se obtiene a través de un convertidor analógico digital (ADC), el cual es el único hardware adicional necesario para poner en funcionamiento al controlador. El ADC utilizado es el ADC0820, el cual es un convertidor de 8 bits.

El resto del documento esta organizado de la siguiente manera: en la siguiente sección se presenta el modelo matemático del convertidor reductor, la tercera sección presenta el controlador promedio GPI para el convertidor reductor. El diseño del control PID es mostrado en la cuarta sección, posteriormente se presenta la arquitectura de los controles implementados. Los resultados experimentales son mostrados en la sexta sección y finalmente se dan las conclusiones del trabajo en la sección siete.



**Modelo Promedio del Convertidor Reductor**

En la figura 1 se muestra el circuito eléctrico del convertidor CD-CD tipo reductor. Usando las leyes de Kirchhoff se obtiene el siguiente modelo promedio del convertidor reductor

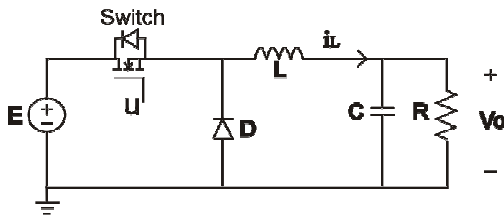


Fig. 1. Circuito eléctrico del convertidor reductor.

$$\begin{aligned} L \frac{di_L}{dt} &= -v_o + Eu \\ C \frac{dv_o}{dt} &= i_L - \left(\frac{1}{R}\right)v_o \end{aligned} \quad (1)$$

Donde  $i_L$  es la corriente de entrada del convertidor,  $v_o$  es el voltaje de salida del convertidor. La entrada de control es representada por la variable  $u$ , la cual denota el voltaje de entrada aplicado a las terminales del switch, la cual toma valores en el conjunto discreto  $\{0,1\}$ . Si se considera un valor promedio en la entrada de control en cada periodo de conmutación del interruptor (Transistor Mosfet), esta entrada de control promedio  $u$  pertenece al conjunto continuo  $[0,1]$ . De (1) se observa que el sistema es de segundo orden de la forma típica:  $\dot{x} = Ax + Bu$ . De ahí, la matriz de controlabilidad Kalman del sistema  $C = [B, AB]$  está dada por:

$$C = \begin{pmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{pmatrix} \quad (2)$$

donde su determinante es

$$\frac{E^2}{L^2 C} \neq 0$$

Por lo tanto el sistema es controlable y a su vez plano, [15]. Por otra parte, la salida plana del sistema se obtiene mediante la siguiente proposición.

**Proposición 2.1**[16] *La salida plana del sistema lineal controlable en la forma de espacio de estado*

$$\dot{x} = Ax + Bu \quad (3)$$

está dada por un módulo de factor constante y por una combinación de los estados obtenidos del último renglón de la matriz inversa de controlabilidad de Kalman

$$C = [B, AB, \dots, A^{n-1}B] \quad (4)$$

es decir

$$F = [0, 0, \dots, 1][B, AB, \dots, A^{n-1}B]^{-1}x \quad (5)$$

De acuerdo a la proposición anterior obtenemos la salida plana del convertidor reductor como sigue:

$$\begin{aligned} F &= [0 \quad 1] \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix}^{-1} \begin{bmatrix} i_L \\ v_o \end{bmatrix} \\ F &= \frac{LC}{E} v_o \end{aligned} \quad (6)$$

Por lo tanto, se puede simplemente tomar como salida plana al voltaje de salida del convertidor, [15]:

$$F = v_o$$

La planitud diferencial del sistema implica que todas las variables de estado del sistema y la variable de entrada de control son parametrizables en términos de  $F=v_o$ , y un número finito de sus derivadas con respecto al tiempo

$$\begin{aligned} v_o &= F \\ i_L &= CF + \frac{1}{R}F \end{aligned} \quad (7)$$



y la entrada de control promedio es,

$$u = \frac{LC}{E} \ddot{F} + \frac{L}{RE} \dot{F} + \frac{1}{E} F \quad (8)$$

Por otra parte, del modelo promedio dado en (1) se puede ver que el sistema es observable, cuando se toma como salida a la variable  $v_o$ , es decir, la matriz de observabilidad de Kalman dada como:

$$O = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \quad (9)$$

cumple con la propiedad de ser de rango completo. Este hecho implica que todas las variables de estado del sistema, que no son la salida plana, pueden estimarse a través de reconstructores integrales que involucran la entrada y la salida.

Por tanto, a partir de integrar la ecuación (8) con respecto al tiempo, se obtiene el estado estimado de  $\hat{F}$ , el cual está dado como:

$$\hat{F} = \left( \frac{E}{LC} \right) \int_0^t \left[ u(\tau) - \frac{1}{E} F(\tau) \right] d\tau - \frac{1}{RC} F \quad (10)$$

### Diseño del Control GPI

A partir de la ecuación de la entrada de control promedio dada en (8), se diseña la ley de control para la estabilización del voltaje de salida del convertidor reductor a un voltaje deseado constante, denotado como  $\bar{F}$ ,

$$u = \frac{LC}{E} v_{aux} + \frac{L}{RE} \dot{F} + \frac{1}{E} F$$

$$v_{aux} = -k_3 \dot{F} - k_2 (F - \bar{F}) \quad (11)$$

De la ecuación de control (11), basada en la técnica de planitud diferencial, se sustituye la variable de estado  $\dot{F}$  por la variable estimada de la derivada de la salida

plana  $\hat{F}$ . Esto implica que esta ley de control dada en (11) pueda verse afectada por valores constantes,

ocasionadas por la incorporación del estimador de  $\hat{F}$  y por perturbaciones externas desconocidas por el sistema. Para corregir adecuadamente este efecto de desestabilización, ocasionados por los errores de estimación y por las perturbaciones externas, se utiliza la compensación mediante integrales del error de la salida plana. El diseño de la ley de control que incorpora el estimador de  $\hat{F}$  y la compensación a través de las integrales del error,  $F = F - \bar{F}$ , recibe el nombre de control Proporcional Integral Generalizado, GPI, el cual es dado como:

$$u = \frac{LC}{E} v_{aux} + \frac{L}{RE} \hat{F} + \frac{1}{E} F$$

$$v_{aux} = -k_3 \hat{F} - k_2 (F - \bar{F})$$

$$-k_1 \int_0^\tau (F - \bar{F}(\tau)) d\tau - k_0 \int_0^\tau \int_0^\lambda (F - \bar{F}(\lambda)) d\lambda \quad (12)$$

La dinámica del error en lazo cerrado se obtiene de igualar la parametrización diferencial de la entrada de control promedio dada en (8) con la del diseño de la ley de control GPI (12).

$$\ddot{F} = -k_3 \hat{F} - k_2 (F - \bar{F})$$

$$-k_1 \int_0^\tau (F - \bar{F}(\tau)) d\tau - k_0 \int_0^\tau \int_0^\lambda (F - \bar{F}(\lambda)) d\lambda \quad (13)$$

Tomando la segunda derivada de la ecuación integro diferencial resultante (13), se tiene la dinámica del error dada por:

$$e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e = 0 \quad (14)$$

El diseño de los coeficientes  $k_3, k_2, k_1$  y  $k_0$  son seleccionados de manera que el polinomio característico correspondiente tenga todas sus raíces ubicadas en la parte izquierda del plano complejo



$$p(s) = s^4 + k_3s^3 + k_2\dot{s} + k_1\dot{s} + k_0s \quad (15)$$

Esta selección de los coeficientes constantes del polinomio,  $\{k_3, k_2, k_1, k_0\}$ , se realiza a través de un polinomio Hurwitz de cuarto orden el cual garantiza la estabilidad exponencial a cero del error de regulación,  $e$ . Por lo tanto, estas ganancias se escogen partiendo de igualar el polinomio (15) con el siguiente polinomio Hurwitz, [17].

$$(s^2 + 2\zeta\omega_n s + \omega_n^2)^2 = 0 \quad (16)$$

De ahí, se tienen las siguientes designaciones, para cada uno de los coeficientes del polinomio característico

$$\begin{aligned} k_3 &= 4\zeta\omega_n \\ k_2 &= 4\zeta^2\omega_n^2 + 2\omega_n^2 \\ k_1 &= 4\zeta\omega_n^3 \\ k_0 &= \omega_n^4 \end{aligned}$$

Los valores adecuados de  $\zeta$  y  $\omega_n$  con los cuales se obtuvo una mejor respuesta estática y dinámica en cuanto a la regulación del voltaje de salida en el convertidor reductor fueron los siguientes:

$$\zeta = 0.707 \quad \text{y} \quad \omega_n = 2500$$

### Diseño del Control PID

Para comparar las especificaciones de la respuesta transitoria del controlador GPI en la regulación del voltaje de salida del convertidor reductor, se diseña un control Proporcional Integral Derivativo (PID). Para tal fin, la correspondiente función de transferencia del convertidor reductor se obtiene a partir del modelo promedio dado en (1), y es

$$\frac{V_o(s)}{U(s)} = \frac{\frac{E}{LC}}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad (17)$$

Mientras que la función de transferencia del controlador PID, es:

$$F_{PID}(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (18)$$

El diagrama a bloques del sistema controlador-PID se muestra en la figura 2.

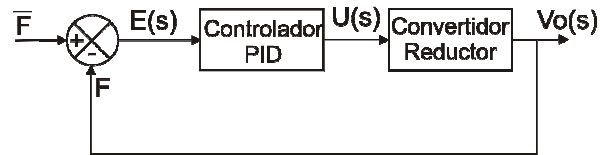


Fig. 2. Control PID en el sistema en lazo cerrado.

La función de transferencia en lazo cerrado se obtiene a partir del diagrama a bloques dado por la figura 2 y este es,

$$H(s) = \frac{(K_p T_d T_i s^2 + K_p T_i s + K_p) \left(\frac{E}{LC}\right)}{s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right)s^2 + \left(\frac{1 + EK_p}{LC}\right)s + \frac{EK_p}{LCT_i}} \quad (19)$$

De la función de transferencia proporcionada por (19), se obtiene el polinomio característico del sistema mostrado en la figura 2.

$$s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right)s^2 + \left(\frac{1 + EK_p}{LC}\right)s + \frac{EK_p}{LCT_i} \quad (20)$$

Los coeficientes  $K_p$ ,  $T_i$ , y  $T_d$  se escogen mediante el polinomio Hurwitz de tercer orden dado por, [17]:

$$(s^2 + 2\zeta\omega_n s + \omega_n^2)(s + \alpha) = 0 \quad (21)$$

Igualando los coeficientes del polinomio característico (20) con el polinomio Hurwitz (21) se tienen los siguientes valores de los parámetros del controlador

$$K_p = \frac{2\zeta\omega_n \alpha LC + \omega_n^2 LC - 1}{E}$$



$$T_i = \frac{EK_p}{LC\alpha\omega_n^2}$$

$$T_d = \frac{LC}{EK_p} \left( \alpha + 2\zeta\omega_n - \frac{1}{RC} \right) \quad (22)$$

Los valores adecuados de  $\zeta$ ,  $\omega_n$ , y  $\alpha$  en la sintonización del controlador PID mediante los cuales se obtuvo una rápida respuesta transitoria en el voltaje de salida del convertidor reductor se determinaron de forma heurística y fueron lo siguientes:

$$\zeta = 0.7071$$

$$\omega_n = 2500$$

$$\alpha = 1000$$

### Plataforma Experimental de los Controladores

Las leyes de control GPI y PID junto con el Modulador de ancho de pulso (PWM) para la regulación del voltaje de salida del convertidor reductor fueron implementadas en una tarjeta NEXYS2 Spartan3E1200. La figura 3 muestra el diagrama a bloques de la implementación de dichos controladores junto con el PWM. Los valores de los parámetros del convertidor reductor son:  $E = 24V$ ,  $R = 100\Omega$ ,  $C = 470\mu F$ ,  $L = 1mH$  y  $f = 48kHz$ . Los valores de los coeficientes constantes utilizados por los controles GPI y PID pueden observarse en la tabla 1 y 2 respectivamente. El único hardware externo interfazado al FPGA para la medición de la variable del voltaje de salida del convertidor es el convertidor analógico digital ADC0820. Las operaciones necesarias para los controladores son realizadas en punto flotante de precisión simple modeladas mediante VHDL, [18].

Tabla 1: Parámetros del Controlador GPI

Parámetro	Valor	Parámetro	Valor
$k_0$	$3.91 \cdot 10^{13}$	$a_1$	$3.92 \cdot 10^{-8}$
$k_1$	$4.42 \cdot 10^{10}$	$a_2$	$8.33 \cdot 10^{-7}$
$k_2$	$2.50 \cdot 10^7$	$a_3$	$8.33 \cdot 10^{-2}$
$k_3$	$7.07 \cdot 10^3$	$a_4$	$2.55 \cdot 10^7$
		$a_5$	$2.13 \cdot 10^6$
		$a_6$	$2.13 \cdot 10^1$

Tabla 2: Parámetros del Controlador PID

Parámetro	Valor
$K_p$	$1.50 \cdot 10^{-1}$
$T_i$	$1.20 \cdot 10^{-3}$
$T_d$	$5.89 \cdot 10^{-4}$

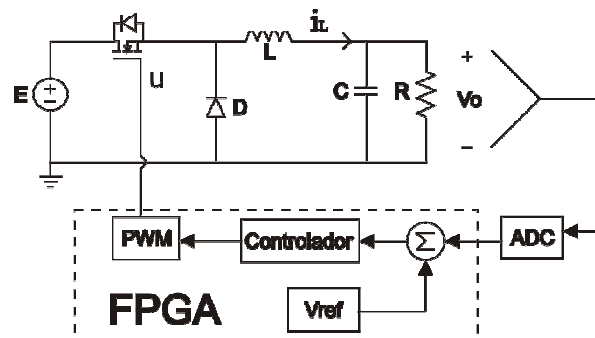


Fig. 3. Diagrama a bloques del control en el FPGA para el convertidor reductor.

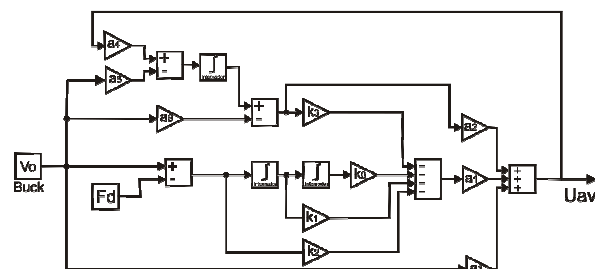


Fig. 4. Control GPI implementado en el FPGA.

### Implementación del control GPI en el FPGA.

Reescribiendo nuevamente las ecuaciones correspondientes al controlador Proporcional Integral Generalizado, se tiene

$$u = a_1 v_{aux} + a_2 \hat{F} + a_3 F$$

$$v_{aux} = -k_3 \hat{F} - k_2 (F - \bar{F}) - k_1 \int_0^\tau (F - \bar{F}(\tau)) d\tau - k_0 \int_0^\tau \int_0^\lambda (F - \bar{F}(\lambda)) d\lambda d\tau$$

$$\hat{F} = \int_0^t [a_4 u(t) - a_5 F(t)] dt - a_6 F \quad (23)$$



donde,

$$a_1 = \frac{LC}{E}, a_2 = \frac{L}{ER}, a_3 = \frac{1}{E}$$

$$a_4 = \frac{E}{LC}, a_5 = \frac{1}{LC}, a_6 = \frac{1}{RC}$$

De (23) se realiza el diagrama a bloques del controlador GPI, el cual se modela en lenguaje VHDL, para su implementación dentro del FPGA. (Ver figura 4).

#### Implementación del control PID en el FPGA.

De igual forma que en el diseño del control GPI, aquí también se realiza un diagrama a bloques del control PID, el cual se modelará en VHDL para posteriormente hacer su implementación. La figura 5, muestra el diagrama a bloques correspondiente al control PID. De (18) se tiene la ecuación de la ley de control PID, la cual se reescribe nuevamente como sigue, para su implementación dentro del FPGA:

$$u = K_p(\bar{F} - F) + K_i \int_0^t (\bar{F} - F(t)) dt + K_d \frac{d}{dt}(\bar{F} - F(t)) \quad (24)$$

donde,

$$K_i = \frac{K_p}{T_i}$$

$$K_d = K_p T_d$$

#### Modulador de ancho de pulsos

El modulador de ancho de pulsos está constituido por una señal portadora y un comparador. La señal portadora es una onda triangular con una frecuencia de 48kHz. Esta señal es comparada con la señal del control proveniente del controlador para generar el PWM (Véase figura 6).

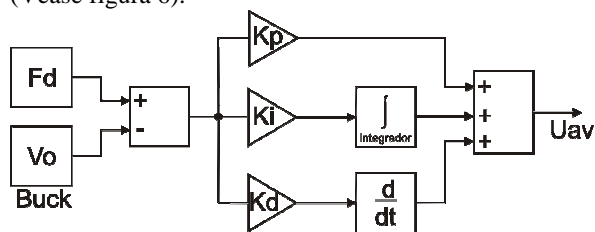


Fig. 5. Control PID implementado dentro del FPGA.

La unidad de la señal portadora y el comparador son unidades independientes modeladas en VHDL. El PWM es de 8 bits.

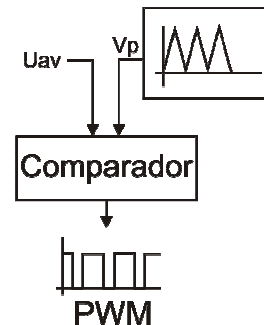


Fig. 6. Modulador de ancho de pulsos.

#### Resultados Experimentales

En esta sección se presentan los resultados experimentales del desempeño del controlador GPI y PID en la regulación del voltaje de salida del convertidor reductor. Los resultados experimentales muestran claramente las especificaciones de la respuesta transitoria de cada uno de los controladores ante una entrada escalón. Estas especificaciones son: tiempo de retardo  $t_d$ , tiempo de subida  $t_r$ , tiempo pico  $t_p$ , sobrepaso máximo  $M_p$  y tiempo de asentamiento  $t_s$ , además se muestra el tiempo de recuperación  $t_{rec}$ , al conectar súbitamente un motor de CD al sistema.

La figura 7.a muestra la respuesta del voltaje de salida del convertidor reductor, para un voltaje de referencia constante de 18 V, usando el controlador GPI. En esta respuesta de voltaje no se observa sobretiro en la señal, mientras su tiempo de establecimiento es de aproximadamente 4.64 ms. En la figura 7.b se muestra la señal de la ley de control GPI.

La figura 8 muestra las dos respuestas de las acciones de control GPI y PID, en la regulación del voltaje de salida del convertidor reductor para un voltaje de referencia constante de 18 V. La señal continua corresponde a la acción de control basada en la técnica GPI, mientras que la señal punteada corresponde a la acción de control PID. El tiempo de establecimiento de la respuesta del voltaje de salida del convertidor mediante la acción de control PID, es 3 veces mayor al tiempo de establecimiento de la acción GPI.



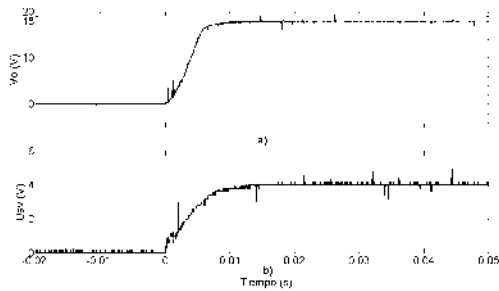


Fig. 7 a) Respuesta transitoria con la técnica de control GPI. b) Señal de control obtenida mediante la técnica GPI.

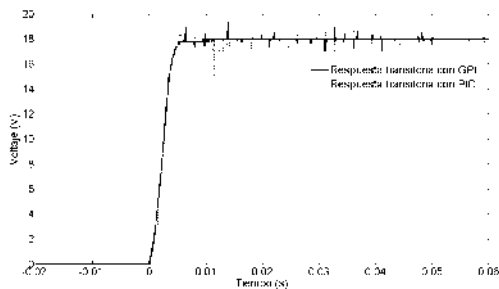


Fig. 8. Respuestas transitorias mediante las técnicas de control GPI y PID.

Para ilustrar mejor la característica de robustez que presenta el controlador GPI en comparación al control PID en la regulación del voltaje de salida del convertidor, se realiza una prueba mediante la conexión súbita de una carga dinámica (motor de CD) a la salida del convertidor reductor. En la señal continua mostrada en la figura 9, se observa claramente la disminución del error en estado estable y una rápida recuperación de la señal de referencia usando la acción de control GPI cuando se conecta súbitamente el motor a la salida del convertidor.

La tabla 3 muestra la comparativa entre estas dos técnicas de control, presentando las principales especificaciones de la respuesta transitoria y tiempo de recuperación al conectar súbitamente cargas dinámicas. Diversas pruebas experimentales se han desarrollado bajo diferentes circunstancias, incluyendo 1) súbitos cambios de resistencia estática en paralelo con la resistencia de carga del convertidor y 2) súbitos cambios de resistencia dinámica en paralelo con la resistencia de carga mediante la conexión de un motor de CD, en conjunto con un freno mecánico. Las señales de las figuras 8 y 9 muestran claramente que la acción

de control GPI presenta una respuesta más rápida que la acción mediante la técnica PID para una señal de referencia constante. Además, el controlador GPI muestra un menor tiempo de recuperación en la regulación de la salida del voltaje que el controlador PID, cuando al sistema se le conecta súbitamente cambios de carga.

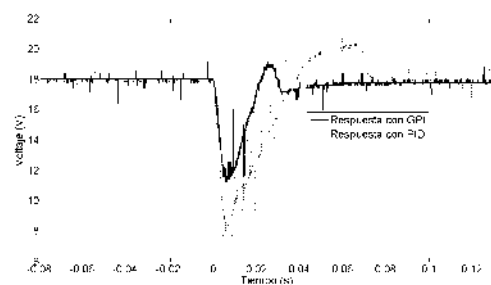


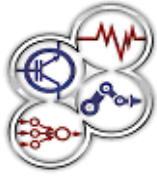
Fig. 9. Robustez de los controladores al conectar súbitamente un motor de CD.

Tabla 3: Comparativa entre los controladores GPI y PID.

Especificaciones de la respuesta transitoria	GPI	PID
Tiempo de retardo $t_d$	2.38 ms	2.52 ms
Tiempo de subida $t_r$	2.96 ms	4.00 ms
Tiempo de pico $t_p$	----	6.24 ms
Máximo sobretiro $M_p$	0 %	2.2 %
Tiempo de asentamiento $t_s$	4.64 ms	13.64 ms
Tiempo de recuperación $t_{rec}$	5.08 ms	7.31 ms

### Conclusiones

En este artículo se ha extendido la técnica del control Proporcional Integral Generalizado para un convertidor CD-CD reductor implementado en un FPGA. El convertidor reductor con control GPI es comparado un control clásico PID. Mediante la comparación se observa que el controlador GPI tiene una mejor respuesta transitoria que el control PID. El tiempo de recuperación para un control PID es mayor que el control GPI al momento de conectar cargas estáticas y dinámicas a la salida del convertidor reductor. Finalmente, los resultados experimentales muestran la efectividad del controlador GPI sobre el controlador PID programados dentro del FPGA para la regulación de voltaje de salida del convertidor reductor.



## Referencias

- [1] S. Okada, T. Nunokawa, T. Takeshita, "Digital Control Scheme of Single-Phase Uninterruptible Power Supply", 31st International Telecommunications Energy Conference, INTELEC 2009, pp. 1-6, 2009.
- [2] J. Sitar, V. Racek, P. Bauer, "DC/DC Converter with Active Filter Supplied from Trolley Net", Power Electronics Specialists Conference, PESC 2007, pp. 383-389, 2007.
- [3] T. B. Marchesan, J. S. da Silveira, M. Cervi, M. A. Dalla Costa, J. M. Alonso, A. Campos, and R. N. do Prado, "Integration Methodology of DC/DC Converters to Supply HPS Lamps: An Experimental Approach", Industry Applications Society Annual Meeting, pp. 1-5, 2008.
- [4] Batarseh, Power Electronic Circuits, John Wiley & Sons Inc, 2004.
- [5] J. Linares Flores, "Modelado, análisis, diseño y simulación del circuito eléctrico convertidor reductor", Ensayo, 2008.
- [6] N. Mohan, T. M. Undeland, W.P. Robbins, Power Electronics Converters Applications, and Design, John Wiley & Sons Inc, 2003.
- [7] Franco González, R. Márquez and H. Sira Ramírez, "On the Generalized Proportional-Integral Sliding mode Control of the "Boost-Boost" Converter", 4th International Conference on Electrical and Electronics Engineering (ICEEE 2007), pp. 209-212, 2007.
- [8] V. Hernandez, and H. Sira Ramírez, "Generalized PI for swinging up and Balancing the inertia Wheel Pendulum", Proceedings of the 2003 American Control Conference, pp. 2809-2814, 2003.
- [9] R. Silva Ortigoza, H. Sira Ramírez, "GPI Control Based on Current Control Scheme for Switched-Capacitor Step-Down DC to DC Converter", 1st International Conference on Electrical and Electronics Engineering, pp.379-384, 2004.
- [10] J. Linares Flores, H. Sira Ramírez, "Sliding Mode-Delta Modulation GPI Control of a DC Motor through a Buck Converter", 2nd Symposium on system, Structure and Control, México, 2004.
- [11] M. He and J. Xu, "Nonlinear PID in Digital Controlled Buck Converters", Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE, pp. 1461-1465, 2007.
- [12] Y. Gao, S. Guo, Y. Xu, S. X. Lin, B. Allard, "FPGA-Based DPWM for Digitally Controlled High-Frequency DC-DC SMPS", 3rd International Conference on Power Electronics Systems and Applications, PESA 2009, pp. 1-7, 2009.
- [13] S. Saggini, W. Stefanutti, P. Mattavelli, and A. Carrera, "Efficiency Estimation in Digitally-Controlled dc-dc Buck Converters based on Single Current Sensing", Power Electronics Specialists Conference, PESC 2008. IEEE, pp. 3581-3586, 2008.
- [14] K. Sugahara, S. Oida, and T. Yokoyama, "High Performance FPGA Controller for Digital Control of Power Electronics Applications", Power Electronics and Motion Control Conference, pp. 1425-1429, 2009.
- [15] M. Fliess, J. Levine, P. Martin and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and applications", International Journal of Control, vol. 61, pp. 1327-1361, 1995.
- [16] H. Sira Ramírez, S. K. Agrawal, Differentially Flat Systems, Marcel Dekker, Chap 2, 2004.
- [17] J. Linares Flores, A. Antonio Garcia, A. Orantes Molina, "Smooth Starter for a DC Machine through a DC-to-DC buck converter", 2008.
- [18] M. Overton, Numerical Computing with IEEE Floating Point Arithmetic, SIAM, 2001.

## Autores



Eric William Zurita Bustamante nació el 20 de Agosto de 1984 en Santos Reyes Nopala, Oaxaca, México. Hizo estudios de lic. en Electrónica en la Universidad Tecnológica de la Mixteca. Actualmente es alumno de la Maestría en Electrónica en la misma Universidad, y es becario por parte del CONACYT.



Jesús Linares Flores, nació el 17 de Junio de 1970 en la ciudad de Atlixco, Puebla, México, hizo sus estudios de Lic. en Electrónica en la BUAP, posteriormente realizó sus estudios de Maestría en Ciencias en la UDLA-PUE con la especialidad en Electrónica de Potencia, en el 2006 recibió el grado Doctor en Ciencias otorgado por el Cinvestav-IPN en el Departamento de Ingeniería Eléctrica, Sección Mecatrónica, desde el año 2006 pertenece al SNI. Actualmente, es Director del Instituto de Electrónica y Mecatrónica de la Universidad Tecnológica de la Mixteca.



Enrique Guzmán Ramírez obtuvo el grado de Doctor en Ciencias de la Computación en el 2008 en el Centro de Investigación en Computación del IPN. Actualmente funge como profesor-investigador de tiempo completo en el Instituto de Electrónica y Mecatrónica de la Universidad Tecnológica de la Mixteca.

# A comparison between the GPI and the PID controllers for the stabilization of a DC-DC “buck” converter: A Field Programmable Gate Array implementation

E. Zurita-Bustamante<sup>†</sup>, J. Linares-Flores<sup>‡</sup>, E. Guzmán-Ramírez<sup>‡</sup>, and H. Sira-Ramírez<sup>††</sup>

**Abstract**—This paper presents a comparison between two stabilizing average output feedback controllers, implemented on Field Programmable Gate Array (FPGA) facility. A Generalized Proportional Integral (GPI) controller and a Proportional Integral Derivative (PID) controller are implemented using a FPGA and their respective performances duly compared. The GPI controller is found to present a better dynamic response than the PID controller in terms of the settling time while exhibiting a greater degree of robustness regarding disturbance rejection represented by severe changes in static and dynamic loads. The average controllers and their corresponding Pulse Width Modulation (PWM) actuators are implemented using a Spartan 3E1600 FPGA.

**Index Terms**—DC-DC power conversion, Buck Converter, Field programmable gate arrays, GPI and PID Control systems.

## I. INTRODUCTION

**M**ANY industrial applications of voltage regulation are accomplished via DC-DC power converters. For example, Uninterruptible Power Supplies (UPS) [1], are used in DC motor drivers for electric traction on trolleys [2]. Lighting systems, using electronic ballasts, also benefit from the use of such devices [3]. Today, switching devices are currently available exhibiting high switching speeds and high power-handling capabilities. It is, nowadays, possible to design low cost, light weight, small size, switched-mode power supplies with efficiencies beyond 90% [31], [5], [6], [7]. The classic power converter topologies in use are, generally speaking, the “buck”, boost and the buck-boost.

In this paper, a Generalized Proportional Integral (GPI) control scheme is presented to regulate the output voltage of a “buck” converter. This control is introduced by Fliess *et al.*

Manuscript received August 06, 2010; Accepted for publication February 2, 2011.

Copyright ©2011 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

<sup>†</sup> E. Zurita-Bustamante is with the Division de Estudios de Postgrado, Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca, Mexico, e-mail: zube84@hotmail.com

<sup>‡</sup> J. Linares-Flores and E. Guzmán-Ramírez are with the Institute of Electronics and Mechatronics, Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca, Mexico, e-mails: jlinares@mixteco.utm.mx, eguzman@mixteco.utm.mx.

<sup>††</sup> H. Sira-Ramírez is with the Sección de Mecatrónica, Centro de Investigación y Estudios Avanzados del IPN (CINVESTAV-IPN), 07300 México D.F., Mexico e-mail: hsira@cinvestav.mx.

in [8] as a means to circumvent classical observers and base output feedback loops in terms of structural state estimators and iterated tracking error integral compensation. The GPI technique has been used in recent years to control DC-DC power converters, because of the following characteristics: fast dynamic response and enhanced robustness with respect to unknown constant, and ramp, disturbances. In addition, using this technique significantly reduces the use of sensors measuring the states of the controlled system. GPI controllers are based on integral state reconstructors processing the available inputs and outputs [9], [10] (see also: [11], [12], [13], [14]).

On the other hand, [15] a GPI control scheme is proposed for DC-DC power converters based on the use of an indirect current control scheme (CCS) for several converter topologies which are non-minimum phase from the available output. In particular, this allows regulation towards a desired output voltage for the switched-capacitor (SC) step-down DC-DC converter. The proposed GPI controller scheme is found to be robust with respect to sudden constant load variations and it only requires the measurement of the output voltage of the inverter. However, the main features of the proposed GPI controller are evaluated only through computer simulations. In the work by Franco-Gonzalez *et al.* [11] multi-variable DC-to-DC converter is presented of the *Boost-Boost* type constituted by two cascaded *Boost* converters in continuous conduction mode. Each converter feeds an independent resistive load. A sliding mode feedback controller, based on the Generalized Proportional Integral (GPI) approach, is developed for the regulation task. The feedback control scheme uses only output capacitor voltage measurements, as well as input signals represented by the switch positions. In addition, robustness of the feedback scheme is tested by non-modeled sudden load resistance variations in the last resistive load. The experimental implementations of the controller were made by a computer with a data-acquisition card.

In the work published in [16], results were presented, at a simulation level, of the performance of the GPI controller for a smooth *starter* of a DC motor based on a switch controlled DC-to-DC power converter of the “buck” type. The switched input actuator was proposed as a sigma-delta modulator (see [9]). The scheme proposes a direct regulation of the motor shaft speed using the flatness of the combined system.

Concerning robust controllers, implemented on FPGA, for

the regulation of the output voltage of the “buck” converter, the work reported in [17] presents a comparison between two PID average controllers. In the work reported in [18] a modular design of embedded feedback controllers is proposed using field-programmable gate array. They proposed a Distributed arithmetic scheme, which is a bit-serial computation algorithm that performs multiplication using an LUT-based scheme, to control a temperature system. In [19] a digital implementation of an observation strategy of the flying capacitor voltages dedicated to stacked multicell converters is performed. They design a sliding-mode observer devoted to the flying capacitor voltages, its digitizing and implemented into the FPGA. Naouar et. al [20] show the benefits of using FPGAs on industrial control systems, specifically control techniques applied to ac machine drives.

Yanxia Gao *et al.* [21], presented the design of a classic PID control and a Digital Pulse Width Modulator (DPWM) as the main module, for a “buck” converter. To verify the effectiveness of DPWM, an 11 bits DPWM was implemented on an FPGA. Experimental results using a fixed sampling frequency of 2 MHz demonstrate the ease of implementation of the DPWM. The implementation was performed on an FPGA Virtex-II Pro XC2VP30. The article by [22] shows the implementation of an efficiency estimator in a digital control scheme applied to a “buck” converter. The approach is based on an indirect estimation of the ratio between the output and input current using a single current sensing. Experimental results on a synchronous “buck” converter, with the efficiency estimator, was also implemented in an FPGA showing that the efficiency may be estimated with errors of the order of 5%. In the work reported in [23] a “buck” converter is controlled through a High-Bandwidth Multisampled Digitally using Ripple compensation. The multisampling techniques reduce the PWM phase lag and ultimately breaking the bandwidth limitation. The control proposed is a feedback technique, and needs no preliminary knowledge of converter parameters. The experimental results on a 1.2V-10A at a frequency of 500 kHz has been implemented on a Xilinx Spartan 3 FPGA.

Recently, Lahoucine Ikhajine *et al.* present a fully integrated FPGA-based solution for motor control [37]. After making a brief description of the proposed control system, the authors have focused firstly on the description of the FPGA-based resolver processing unit (RPU) for extracting the rotor position and speed. Then, a description of the FPGA-based motor controller has been achieved. The FPGA-based RPU offers a good estimation accuracy of the position and speed. However, some studies have to be made in order to improve the treatment quality and particularly the synchronous demodulation. Furthermore, resource optimization has to be carried out in order to reduce the consumed FPGA hardware resources. Finally, in [38] a FPGA-based intelligent-complementary sliding-mode control (ICSMC) is proposed to control the mover of a permanent magnet linear synchronous motor (PMLSM) servo-drive system to track periodic-reference trajectories. In the ICSMC development, a radial-basis function-network (RBFN) estimator with accurate approximation capability was modeled using VHDL language, furthermore, the adaptive-learning algorithms for the online

training of the RBFN were derived using the Lyapunov theorem to guarantee the closed-loop stability.

In this article, we will focus on the comparison of two average output feedback controllers, implemented in a FPGA, to stabilize the output voltage of a “buck” power converter around a desired constant output reference voltage. We implement first a GPI controller and obtain the performance features related to settling time and recovery with respect to sudden, static and dynamic, load changes. Next, we implement a second controller, on the same converter, corresponding to a classical PID controller and proceed to evaluate the same features examined for the GPI control scheme. The average control inputs are used as a duty ratio generator in a PWM control actuator. The experimental setup, used for the comparisons, has the following features:

- The PWM actuator is implemented through a triangular carrier signal and a comparator. The main function of this modulator is the average signal conversion to a pulsing signal that activates and deactivates the converter power transistor, at a switching frequency of 48kHz.
- The processing time control for the GPI is  $39.2\mu\text{s}$ , while the processing time for the PID is  $20.54\mu\text{s}$ . These processing times were achieved thanks to the parallel execution of units modeled within a FPGA [24]-[25]. Progress of PLD(Programmable Logic Device) like FPGA or CPLD enable the realization of a digital control system for power electronics thus avoiding the use of microprocessor (CPU or DSP) [26].
- The output voltage is obtained through an Analog to Digital Converter (ADC), which is the only additional hardware needed to operate to the controllers. The used ADC is the ADC0820, which is an 8 bits converter.

The remaining sections of this paper are organized as follows. In next Section, the mathematical model of the “buck” converter is presented. The Sections III and IV describe the GPI controller and PID controller designs respectively. Section V presents the design requirements of the Controllers. The proposed architectures of the implemented controls are found in Section VI. The results of the implementation of the GPI controller and the PID controller in the FPGA, are provided and discussed in Section VII. Finally, Section VIII contains the conclusions of this paper.

## II. THE “BUCK” CONVERTER MODEL

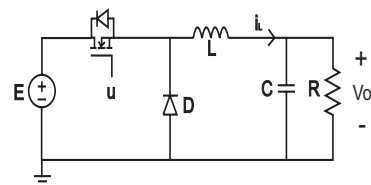


Fig. 1. The electrical circuit of the “buck” converter.

Consider the ‘buck’ converter circuit, shown in Figure 1. The system is described by the following set of differential

equations

$$\begin{aligned} L \frac{di_L}{dt} &= -v_o + Eu \\ C \frac{dv_o}{dt} &= i_L - \left(\frac{1}{R}\right)v_o \\ y &= v_o \end{aligned} \quad (1)$$

where  $i_L$  represents the inductor current and  $v_o$  is the output capacitor voltage. The control input  $u$ , representing the switch position function, takes values in the discrete set  $\{0, 1\}$ . The system parameters are constituted by:  $L$  and  $C$  which are, respectively, the input circuit inductance and the capacitance of the output filter, while  $R$  is the load resistance. The external voltage source exhibits the constant value  $E$ . The *average state model* of the “buck” converter circuit, extensively used in the literature [5], [16], [27], may be directly obtained from the original switched model, (1), by simply identifying the switch position function,  $u$ , with the *average control*, denoted by  $u_{av}$ . Such an average control input is frequently identified with the *duty ratio* function in a Pulse Width Modulation implementation. The control input  $u_{av}$  is restricted to take values in the closed interval  $[0, 1]$ . From (1), the “buck” converter system is clearly a second order linear system of the typical form:  $\dot{x} = Ax + bu$  and  $y = c^T x$ .

$$\begin{aligned} A &= \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \\ b &= \begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix} \\ c^T &= [0 \quad 1] \end{aligned} \quad (2)$$

Hence, the Kalman controllability matrix of the system  $\mathcal{C} = [b, Ab]$ , is given by:

$$\mathcal{C} = \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix} \quad (3)$$

The determinant of the controllability matrix is

$$\frac{E^2}{L^2 C} \neq 0$$

The system is controllable and, hence, differentially *flat* (See Fliess *et al.* [28]). This flat output for the “buck” converter system is obtained from the following proposition.

**Proposition 2.1** [27] *The flat output of a linear controllable system in state space form:*

$$\dot{x} = Ax + bu \quad (4)$$

is given, modulo a constant factor, by the linear combination of the states obtained from the last row of the inverse of the Kalman controllability matrix

$$F = [0, 0, \dots, 1][b, Ab, \dots, A^{n-1}b]^{-1}x \quad (5)$$

According to the previous proposition, the flat output of the “buck” converter is given by

$$\begin{aligned} F &= [0 \quad 1] \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix}^{-1} \begin{bmatrix} i_L \\ v_o \end{bmatrix} \\ F &= \frac{LC}{E} v_o \end{aligned} \quad (6)$$

Therefore, we can simply take as a flat output the output voltage variable [28]:

$$F = v_o$$

The flatness of the system implies that all state variables of the system, including the control input variable, are parameterizable in terms of  $F = v_o$  and a finite number of its time derivatives. Indeed,

$$\begin{aligned} v_o &= F \\ i_L &= C\dot{F} + \frac{1}{R}F \end{aligned} \quad (7)$$

and the average control input is obtained as

$$u_{av} = \frac{LC}{E} \left( \ddot{F} + \frac{1}{RC}\dot{F} + \frac{1}{LC}F \right) \quad (8)$$

Moreover, from the average model, given in (1), we can see that the system is also observable from the output variable  $v_o$ , i.e., the Kalman Observability Matrix, given by,

$$\mathcal{O} = \begin{bmatrix} c^T \\ c^T A \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{RC} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \quad (9)$$

complies with the property of being full-rank. Therefore, the system model is observable for the output  $y = F = v_o$ . This fact establishes the *reconstructibility* of the system, i.e., all the system state variables are parameterizable in terms of the input, the output, and a finite number of iterated integrals of the input and of the output variables (see [8]). By integrating both sides of (8) and solving for the variable  $\hat{F}$ , we have an integral estimator of the first time derivative of  $F$ ,

$$\begin{aligned} \hat{F} &= \left(\frac{E}{LC}\right) \int_0^t [u_{av}(\tau) - \frac{1}{E}F(\tau)]d\tau - \frac{1}{RC}F \\ F &= v_o \end{aligned} \quad (10)$$

Note that, for non-zero initial states, the relations linking the actual values of the converter output voltage derivative to the structural estimate in (10) is given by

$$\dot{F} = \hat{F} + \dot{F}_0 \quad (11)$$

where  $\dot{F}_0$  denotes the unknown initial rate of change of the output voltage.

### III. GPI CONTROLLER DESIGN

From (8), we propose the following feedback control law for the stabilization of the “buck” converter output voltage, around a desired constant reference value  $\bar{F}$ ,

$$\begin{aligned} u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\dot{F} + \frac{1}{E}F \\ v &= -k_3\dot{F} - k_2(F - \bar{F}) \end{aligned} \quad (12)$$

For the GPI feedback controller, we replace the unmeasured state variable,  $\dot{F}$ , by its structural estimated variable,  $\hat{F}$ , given on (10). However, this implies that the closed loop system is affected by the constant estimation error present in  $\hat{F}$ , as acknowledged in (11). To suitably correct for the destabilizing effect of the structural estimation errors and the

effect of possible external perturbations, GPI control uses iterated integral error compensation as follows:

$$\begin{aligned} u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\hat{F} + \frac{1}{E}F \\ v &= -k_3(\hat{F}) - k_2(F - \bar{F}) - k_1\gamma - k_0\eta \\ \dot{\gamma} &= F - \bar{F} \\ \dot{\eta} &= \gamma \end{aligned} \quad (13)$$

Let  $e = F - \bar{F}$  denote the stabilization error. The stabilization error dynamics is obtained by substituting (11), and the controller equations (13) into the differential parameterization of the average control input given in (8). We obtain the GPI controller as (13).

$$\begin{aligned} \ddot{F} &= -k_3(\dot{F} - \dot{F}_0) - k_2(F - \bar{F}) - k_1 \int_0^t (F(\tau) - \bar{F})d\tau \\ &\quad - k_0 \int_0^t \int_0^\tau (F(\lambda) - \bar{F})d\lambda d\tau \end{aligned} \quad (14)$$

The characteristic equation of the integro-differential relation (14), in terms of the stabilization error, is given by:

$$e^{(4)} + k_3e^{(3)} + k_2\ddot{e} + k_1\dot{e} + k_0e = 0$$

The values of the design parameters  $\{k_3, k_2, k_1, k_0\}$  are chosen so that the closed loop characteristic polynomial,

$$p(s) = s^4 + k_3s^3 + k_2s^2 + k_1s + k_0$$

has all its roots in the left half of the complex plane.

The controller parameters were chosen so as to achieve the following desired closed loop characteristic polynomial,

$$p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)^2 \quad (15)$$

Taking into account that  $\zeta$  and  $\omega_n$  are positive quantities. Hence, the gains of the GPI controller are given by:

$$\begin{aligned} k_3 &= 4\zeta\omega_n, \quad k_2 = 4\zeta^2\omega_n^2 + 2\omega_n^2, \\ k_1 &= 4\zeta\omega_n^3, \quad k_0 = \omega_n^4; \end{aligned}$$

#### IV. PID CONTROLLER DESIGN

In order to compare the performance of the transient response of the GPI controller, a classical Proportional Integral Derivative (PID) controller was also designed and implemented. The corresponding transfer function of the converter, obtained from the average model given in (1), is

$$\frac{V_o(s)}{U_{av}(s)} = \frac{\frac{E}{LC}}{s^2 + \frac{1}{RC}s + \frac{1}{LC}} \quad (16)$$

While the transfer function of the PID controller, is:

$$F_{PID}(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \quad (17)$$

The block diagram of the PID controlled system is shown in Figure 2.

The closed loop transfer function is readily found to be

$$H(s) = \frac{(K_p T_d T_i s^2 + K_p T_i s + K_p) \left(\frac{E}{LC}\right)}{s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right)s^2 + \frac{(1+EK_p)}{LC}s + \frac{EK_p}{LC T_i}}$$

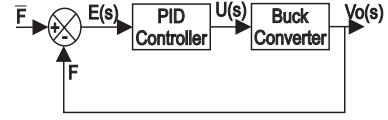


Fig. 2. PID control in closed loop.

The closed loop characteristic polynomial of the PID controlled system is then given by

$$s^3 + \left(\frac{1}{RC} + \frac{EK_p T_d}{LC}\right)s^2 + \frac{(1+EK_p)}{LC}s + \frac{EK_p}{LC T_i} = 0 \quad (18)$$

The coefficients  $K_p$ ,  $T_i$  and  $T_d$  are chosen so that (18) becomes a third order Hurwitz polynomial of the form:

$$p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)(s + \alpha) \quad (19)$$

Taking into account that  $\zeta$ ,  $\omega_n$  and  $\alpha$  are positive quantities. Equating the characteristic polynomial coefficients (18) with those of the desired Hurwitz polynomial (19), we obtain the following values of the parameters for the PID controller,

$$\begin{aligned} K_p &= \frac{2\zeta\omega_n \alpha LC + \omega_n^2 LC - 1}{E} \\ T_i &= \frac{EK_p}{LC \alpha \omega_n^2} \\ T_d &= \frac{LC}{EK_p} \left(\alpha + 2\zeta\omega_n - \frac{1}{RC}\right) \end{aligned} \quad (20)$$

#### V. DESIGN REQUIREMENTS OF THE CONTROLLERS

Figure 3 shows the open-loop response of the buck-converter system with the following specifications:  $L = 1mH$ ,  $C = 100\mu F$ ,  $R = 100\Omega$ ,  $E = 24V$ ,  $f = 48.828 KHz$ ,  $\Delta v_o/v_0 = 0.013\%$ ,  $\Delta i_L = 0.092$ , and  $\bar{u}_{av} = D = 0.75V$  (duty cycle or duty cycle) [31]. The output voltage response is a steady-state error of 5.56% and has a settling time of 15ms. On the other hand, we get that the diagram bode of the transfer function given by (16) with the same specifications, has a gain margin  $Gm = \text{Inf dB}$  (at  $\text{Inf rad/sec}$ ) and a phase margin  $Pm = 0.377 \text{ deg}$  (at  $1.58 \times 10^4 \text{ rad/sec}$ ). Given that the buck converter system has infinite gain margin, it can withstand greater changes in system parameters before becoming unstable in closed loop [30]. Since the system has this characteristic, we will design our controllers in closed loop with the following requirements: Overshoot less than 4.32%, Settling time less than 5 milliseconds, Steady-state error less than 1% and Maximum sampling time of  $40\mu s$ .

In this case, we aim to have an overshoot less than 4.32%, a settling time less than 5ms, and a steady-state error less than 1% for a desired output voltage of 18V. We will tune in our controllers in closed loop through (15) and (19) with a damping coefficient of  $\zeta = 0.707$ ,  $\alpha = 1000$ , and natural frequency of  $\omega_n = 2500$ . Hence,

- a) The GPI controller gains obtained by the design requirements were:

$$\begin{aligned} k_3 &= 7070, \quad k_2 = 2.5 \times 10^7, \\ k_1 &= 4.42 \times 10^{10}, \quad k_0 = 3.90 \times 10^{13}; \end{aligned}$$

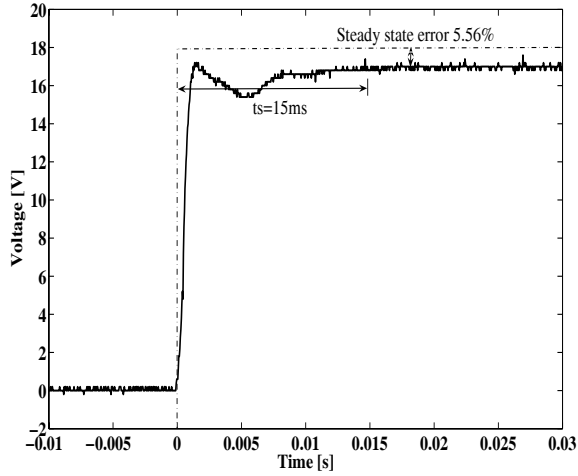


Fig. 3. Output voltage transient responses of the “buck” converter in open loop.

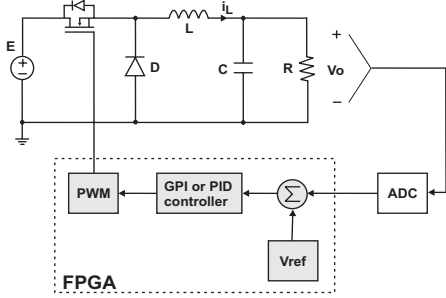


Fig. 4. Block diagram of the FPGA-based control system using the GPI or PID controller.

- b) The PID controller gains obtained by the design requirements were:

$$k_p = 0.15, T_i = 1.2 \times 10^{-3}, T_d = 5.9 \times 10^{-4};$$

## VI. THE CONTROL ALGORITHM ARCHITECTURE.

The GPI and PID control laws and the Pulse Width Modulator (PWM) actuator for the regulations of output voltage of the buck converter were implemented in a Spartan 3E board. Figure 4 illustrates the block diagram of the FPGA-based control system based on GPI or PID controller. The parameters values for the “buck” converter and specifications were given in the previous section. The values of the design coefficients used by the GPI and PID controllers were obtained in base to design requirements given in the previous section. The only external hardware connected to the FPGA for measuring the “buck” converter output voltage was the analog digital converter ADC0820.

### A. GPI Implementation into the FPGA.

We rewrite the equations for the GPI controller as,

$$\begin{aligned} u_{av} &= a_1 v + a_2 \hat{F} + a_3 F & (21) \\ v &= -k_3(\hat{F}) - k_2(F - \bar{F}) - k_1 \int_0^t (F(\tau) - \bar{F}(\tau)) d\tau \\ &\quad - k_0 \int_0^t \int_0^\tau (F(\lambda) - \bar{F}(\lambda)) d\lambda d\tau \\ \hat{F} &= \int_0^t (a_4 u - a_5 F) dt - a_6 F \end{aligned}$$

where,

$$\begin{aligned} a_1 &= \frac{LC}{E}, a_2 = \frac{L}{ER}, a_3 = \frac{1}{E}, \\ a_4 &= \frac{E}{LC}, a_5 = \frac{1}{LC}, a_6 = \frac{1}{RC} \end{aligned}$$

Now, to implement the GPI controller, it is necessary to obtain a discrete approximation of the equation (21). Furthermore, it is necessary to be considered that the most important condition is that the operation frequency of the controller must be a high value and this feature depends directly on the application specifications.

An important aspect in the discretization of (21) is the obtaining of a discrete approximation of the continuous integral. In the solution of this problem, we have used the Adams-Bashforth method of the second order [32], [33]. This method is given by:

$$y[n+1] = y[n] + \frac{1}{2} \Delta t (3\dot{y}[n] - \dot{y}[n-1]) \quad (22)$$

where  $\Delta t$  is the interval of time between two consecutive moments of the solution (usually called *step of integration*), also this method needs from the knowledge of the current derivative,  $\dot{y}[n]$ , and the derivative evaluated in a previous moment  $\dot{y}[n-1]$ .

Then, if the integral is defined by  $IF(t) = \int_0^t (a_4 u(t) - a_5 F(t)) dt$ , using the Adams-Bashforth method, is defined as

$$IF[n+1] = IF[n] + \frac{1}{2} \Delta t (3\dot{IF}[n] - \dot{IF}[n-1]) \quad (23)$$

where,  $\dot{IF}[n] = a_4 u[n] - a_5 F[n]$  and  $\dot{IF}[n-1] = a_4 u[n-1] - a_5 F[n-1]$ .

On the other hand, if the integral is defined by  $I(t) = \int_0^t (F(\tau) - \bar{F}(\tau)) d\tau$ , its discrete approximation is defined as

$$I[n+1] = I[n] + \frac{1}{2} \Delta t (3\dot{I}[n] - \dot{I}[n-1]) \quad (24)$$

where,  $\dot{I}[n] = F[n] - \bar{F}[n]$  and  $\dot{I}[n-1] = F[n-1] - \bar{F}[n-1]$ . Now, if  $II(t) = \int_0^t \int_0^\tau (F(\lambda) - \bar{F}(\lambda)) d\lambda d\tau$ , its discrete approximation is defined as

$$II[n+1] = II[n] + \frac{1}{2} \Delta t (3\dot{II}[n] - \dot{II}[n-1]) \quad (25)$$

where,

$$\dot{II}[n] = I[n] + \frac{\Delta t (3(F[n] - \bar{F}[n]) - F[n-1] - \bar{F}[n-1])}{2}$$

and

$$\dot{I}I[n] = I[n-1] + \frac{\Delta t(3(F[n-1] - \bar{F}[n-1]) - F[n-2] - \bar{F}[n-2])}{2}$$

Hence, the discrete representation of (21) is defined as

$$\begin{aligned} u_{av}[n+1] &= a_1 v[n] + a_2 \hat{F}[n] + a_3 F[n] \\ v[n] &= -k_3 \hat{F}[n] - k_2 (F[n] - \bar{F}[n]) - k_1 I[n] \\ &\quad - k_0 II[n] \\ \hat{F}[n] &= IF[n] - a_6 F[n] \end{aligned} \quad (26)$$

Based on (26) the GPI controller implementation on reconfigurable logic was realized. For this purpose, the Xilinx ISE Design Suite 12.2 tool and the Spartan 3E board was used; the Spartan 3E board, designed by Digilent Company, includes a Spartan-3E1600 FPGA. Furthermore, to acquire information of the buck converter an Analog-Digital Converter, ADC0820, was chosen. The ADC0820 is a 8 bits resolution converter, it offers a  $2 \mu s$  conversion time and it has an 0 to 5 Volts analog input voltage range.

The GPI controller design is based on Top-Down methodology and the schematic description was chosen as top level; the GPI controller modules was modeled using the hardware description language VHDL (algorithm level modeling), generated using the Xilinx CORE Generator tool or using pre-optimized elements that the device in use includes.

The architecture proposed for the discrete GPI controller is shown in Figure 5. Exploiting the independence existing in diverse modules, in each stage of the architecture several modules can be working concurrently, this allows obtaining high processing speeds, however it uses many resources.

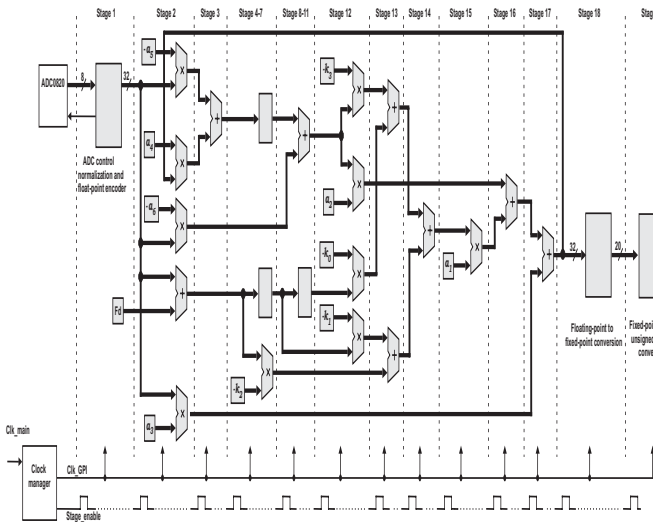


Fig. 5. Architecture proposed for the discrete GPI controller implemented into the Spartan-3E1600 FPGA.

It is clear that the processing in real-time is relative to the application, that is to say, based on the application, the interval of time between the processing of 2 data must be chosen. So that our application operates in real-time, the maximum time of processing in the GPI controller specifications, is fixed at  $40 \mu s$ .

The GPI controller works with a frequency of 50 MHz (Clk\_GPI). The implemented architecture is composed of 19 stages and each of them needs 100 cycles to fulfill its function ( $2 \mu s$ ), this indicates that the processing time of one data is  $38 \mu s$  (time between 2 consecutive data delivered by the controller to the next module, the PWM).

The *clock manager module* generates the frequencies necessary for the GPI controller (Clk\_GPI) and for the PWM (Clk\_PWM) from the principal frequency of 50 MHz (Clk\_main), see Figure 6. The principal element of this module is the Digital Clock Manager (DCM). The DCM is embedded on the Spartan3E FPGA's families and it provides flexible complete control over clock frequency, maintaining its characteristics with a high degree of precision despite normal variations in operating temperature and voltage.

The Clk\_GPI signal is the same as the Clk\_main signal, but the DCM provides a correction clock feature, ensuring a clean Clk\_GPI output clock with a 50% duty cycle.

The Clk\_PWM signal is derived from the input clock (Clk\_main) by a Digital Frequency Synthesizer (DFS) included in the DCM. The frequency of the Clk\_PWM is 25 MHz and has a 50% duty cycle correction too.

The "GPI stage enable" function also is included in *clock manager module* (see Figure 6). The "GPI stage enable" is a simple finite-state machine that generates the enable signals of the proposed architecture stages, Stage\_en(1) ... Stage\_en(19). The description style adopted for this module was algorithm level modeling.

The pulse width of all stage enable signals is  $t_{SE}$  (1 Clk\_main cycle, 20 ns); the time between two active stage enable signals is  $t_S$  (100 Clk\_main cycles,  $2 \mu s$ );  $t_{FL}$  is the time from Stage\_en(1) is active to Stage\_en(19) is active, 1800 Clk\_main cycles ( $36 \mu s$ );  $t_{CE}$  is the enable cycle time (1900 Clk\_main cycles,  $38 \mu s$ ), this is the time that the GPI controller needs to deliver new information to the PWM.

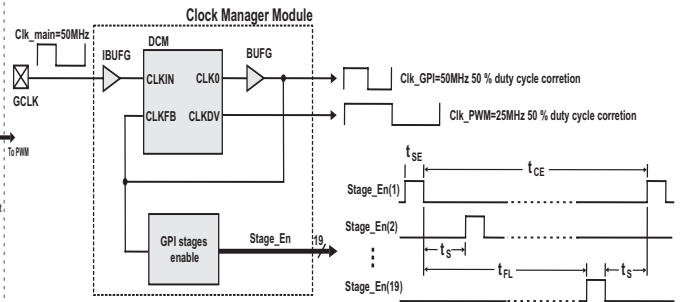


Fig. 6. Clock manager module.

The *ADC control, normalization and float-point encoder module* was described based on algorithm level modeling. This module carries out three functions. First, it controls the necessary signals to acquire information from the ADC0820. Second, it normalizes the acquired information; the voltage range that the buck converter delivers to GPI controller is 0 to 24 Volts and the voltage that the ADC0820 can process is in a range from 0 to 5 Volts; due to this, a non-inverting amplifier with a gain of 0.25 is used to connect the buck converter with the ADC; therefore, it is necessary to compensate this



attenuation on the information obtained from the ADC, this process is called normalization. Third, due to the range of results that the operations of the discrete GPI controller generate, it is necessary to use a floating point format; for this intention, the IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985 [34] was chosen; the last function of this module is to codify the information normalized in single-precision floating-point format.

The Xilinx ISE Design Suite 12.2 includes the CORE Generator tool, which allows generating pre-optimized elements for Xilinx's FPGA. Using this tool the addition and multiplication operations in single-precision floating-point, standard Std-754, can be generated. In our discrete GPI controller architecture, the *float point Adder and Multiplier modules* have been generated using the CORE Generator.

Finally, the result of the GPI controller must be delivered to a 8-bits PWM, for this reason the last two stages of the controller convert the results from single-precision fixed-point to 8 bit unsigned binary. The *Float-point to Fixed-point conversion* and *Fixed-point to 8 bit unsigned binary conversion* modules are doing this work. Both modules are VHDL description based on algorithm level modeling.

### B. PWM module

A single up-down counter unit and one comparator unit are used to create the PWM signal required to drive the buck converter, see Figure 7.

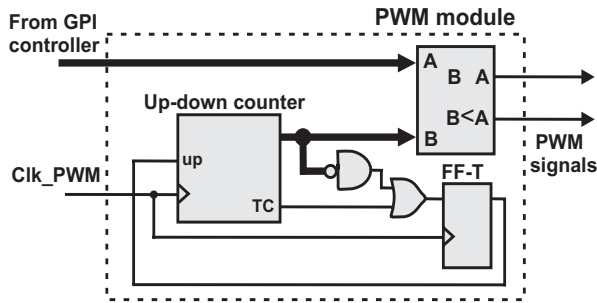


Fig. 7. PWM module.

The maximum count value of the counter, in conjunction with the speed of the clock used to drive the counter, determines the PWM period frequency:

$$\begin{aligned} \text{PWM frequency} &= \frac{\text{Clk\_PWM}}{2(\text{maximum count}+1)} \\ &= \frac{25 \text{ MHz}}{2 \times 256} = 48.828 \text{ khz} \end{aligned}$$

The counter counts from 0 to its maximum value, then from its maximum value to 0. The period of the PWM is measured from zero point to zero point in the counters cycle.

The compare unit compares the count value from the counter unit with the GPI controller output value. The compare unit has two PWM outputs, one is asserted when the count value is  $\geq$  the GPI controller output value, and the other is asserted when the count value is  $<$  the GPI controller output value.

The implementation result of individual modules and the complete architecture for discrete GPI controller are reported in Table I.

TABLE I  
DISCRETE GPI CONTROLLER IMPLEMENTATION RESULT.

Mod.	Slices	Flip-Flops	4-input's -LUT's	Pre-opt.-elem.	Max. Freq. (MHz)
1	53 ( $< 1\%$ )	50 ( $< 1\%$ )	100 ( $< 1\%$ )		150.9
2	404 ( $< 4\%$ )	581 ( $< 3\%$ )	559 ( $< 3\%$ )		190.4
3	403 ( $< 4\%$ )	702 ( $< 4\%$ )	638 ( $< 3\%$ )		161.9
4	223 ( $< 2\%$ )	86 ( $< 1\%$ )	426 ( $< 2\%$ )		140.706
5	23 ( $< 1\%$ )	40 ( $< 1\%$ )	41 ( $< 1\%$ )		168.5
6	60 ( $< 1\%$ )	31 ( $< 1\%$ )	104 ( $< 1\%$ )		138.2
7	12977 (87%)	20490 (69%)	19775 (67%)	1 BRAM (2%) 1 DCM (12%)	59.8

- 1.-ADC control, normalization and float-point encoder.
- 2.-Float point -add.
- 3.-Float point -mult.
- 4.-Float point to Fixed point conver.
- 5.-Fixed point to 8 bit un-signed binary cover.
- 6.-PWM.
- 7.-GPI Controller.

### C. Discrete PID controller implemented on the FPGA.

In a manner similar to that used for implementation the discrete GPI controller, the discrete PID controller was synthesized on a FPGA.

The equations for the continuous PID controller are defined as

$$\begin{aligned} u_{av} &= K_p(\bar{F}(t) - F(t)) + K_i \int_0^t (\bar{F}(t) - F(t))dt \\ &+ K_d \frac{d(\bar{F}(t) - F(t))}{dt} \end{aligned} \quad (27)$$

where,

$$\begin{aligned} K_i &= \frac{K_p}{T_i} \\ K_d &= K_p T_d \end{aligned}$$

The discretization of the equation (27) needs a discrete approximation of the continuous integral and a discrete approximation of the continuous derivative. Applying the equation (22), the discrete approximation of the continuous integral is obtained.

On the other hand, the discrete approximation of the continuous derivative is obtained based on finite differences method of the first order [35], using the backward difference. This method is given by the equation:

$$\left( \frac{\partial y}{\partial t} \right)_n \approx \frac{y[n] - y[n-1]}{\Delta t} \quad (28)$$

Then, if the derivative is defined by  $D(t) = \frac{\partial(\bar{F}(t)-F(t))}{\partial t}$ , using the finite differences method, its discrete approximation is defined as

$$D' [n] = \frac{D [n] - D [n - 1]}{\Delta t} \quad (29)$$

where,

$$D [n] = \bar{F} [n] - F [n]$$

and

$$D [n - 1] = \bar{F} [n - 1] - F [n - 1]$$

Hence, the discrete representation of the equation (27) is defined as

$$u_{av} [n + 1] = K_p (\bar{F} [n] - F [n]) + K_i I [n] + K_d D' [n] \quad (30)$$

Based on equation (30) the PID controller was implemented on reconfigurable logic. This controller was implemented using the same tools and devices that GPI controller.

The PID controller design is based on Top-Down methodology and the schematic description was chosen as top level. The Figure 8 shows the architecture proposed for the discrete PID controller. For obtaining high processing speeds, in the modeling of this controller the independence existing in some modules of the architecture is exploiting, therefore several modules can be working concurrently.

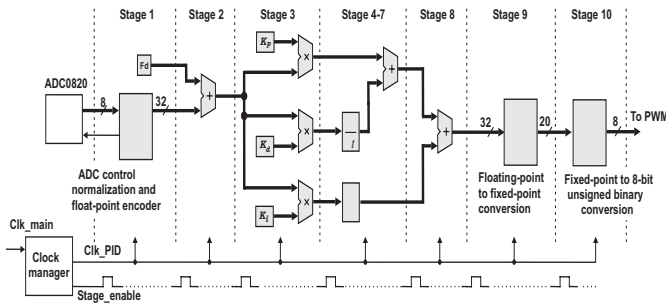


Fig. 8. Architecture proposed for the discrete PID controller implemented into the Spartan-3E1600 FPGA.

The Figure 8 shows that all the discrete GPI controller modules were used in the architecture of the discrete PID controller.

The PID controller work with a frequency of 50 MHz (Clik\_PID). The implemented architecture is composed of 10 stages and each of them needs 100 cycles to fulfill its function (2  $\mu$ s), this indicates that the processing time of one data is 20  $\mu$ s (time between 2 consecutive data delivered by the controller to the next module, the PWM)

TABLE II  
DISCRETE PID CONTROLLER IMPLEMENTATION RESULT.

Mod.	Slices	Flip-Flops	4-input's -LUT's	Pre-opt.-elem.	Max. Freq. (MHz)
1	5668 (38%)	8722 (29%)	8737 (29%)	1 BRAM (2%) 1 DCM (12%)	60.37

### 1.-PID Controller.

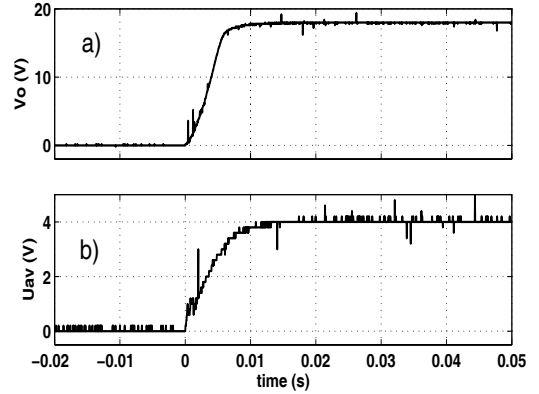


Fig. 9. Output voltage transient response of the “buck” converter with the GPI control action.

The implementation result of the complete architecture for discrete PID controller are reported in Table II.

## VII. EXPERIMENTAL RESULTS

This section presents the experimental results of the performance of the GPI controller and the PID controller in regulating the “buck” converter output voltage. The experimental results depict the specifications of the transient response for each one of the controllers, using a step reference signal. These specifications are: delay time  $t_d$ , rise time  $t_r$ , time of peak  $t_p$ , percentage of overshoot,  $M_p$ , and settling time,  $t_s$ . In addition, we considered the recovery time  $t_{rec}$ , when an armature controlled DC motor load is suddenly connected to the converter output voltage terminals.

Figure 9a) shows the GPI controlled response of the “buck” converter output voltage for a constant reference voltage of 18 V. In this voltage response, an overshoot is not present in the recorded signal, while the settling time is approximately 4.64 ms. Figure 9b) shows the GPI average control law input signal. Figure 10 shows the performances of the GPI and PID control laws, in the stabilization task for the “buck” converter output voltage. As before, we used a constant reference of 18 V. The continuous line corresponds to the GPI controlled response, while the dotted line corresponds to the PID controlled converter. The settling time of the response of the “buck” converter output voltage through the PID controller, is 3 times greater than that obtained with the GPI controller. The output voltage transient response of the buck converter with the GPI control scheme, exhibits a different dynamic performance that of the PID control scheme, due to the structure of the controller, since this controller incorporates a double integrator of the error regulation. As a consequence, the GPI controller tuning was done through a fourth order Hurwitz polynomial, while the PID controller tuning was done with a third order Hurwitz polynomial.

To illustrate the robustness of the GPI controller, in comparison to that of the PID control, we made a test by suddenly connecting a dynamic load (represented by an armature controlled DC motor) at the output of the “buck” converter. The

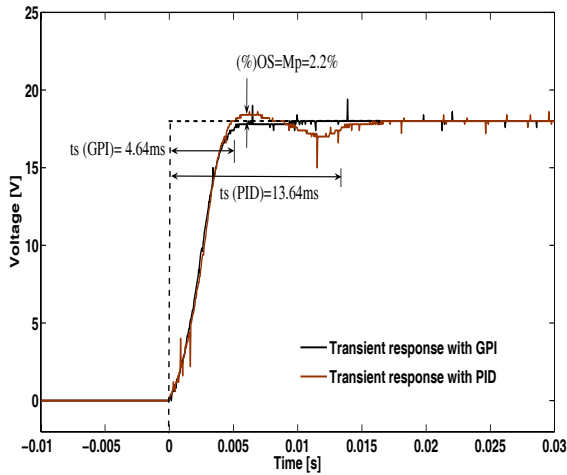


Fig. 10. Output voltage transient responses of the “buck” converter with the GPI and the PID control schemes.

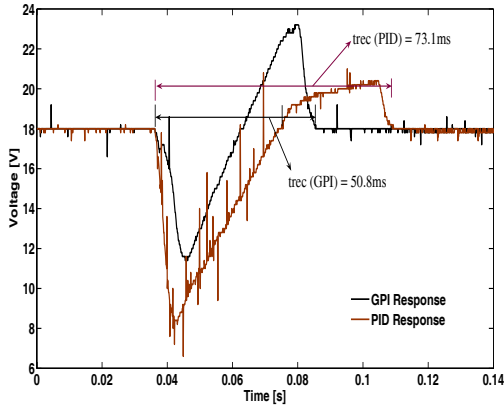


Fig. 11. Output voltage response of the “buck” converter with sudden connection a DC motor.

continuous line, shown in Figure 11, depicts the behavior of the temporarily perturbed converter’s output voltage and the fast recovery of the output voltage to the desired reference signal when the converter is controlled via a GPI controller option. The GPI controller also results in reduced noise features affecting the regulated output voltage response (see Figure 11). The reduced noise in the response is due to that the GPI controller incorporates the estimated value of the first time derivative of  $F$ , while that the PID controller uses the backward difference scheme to calculate the value of the first time derivative of  $F$ . In addition, the output voltage with GPI controller and the average voltage of the controller,  $u_{av}$ , can be shown in Figure 12a) and 12b) respectively, that Figure shows that the maximum voltage,  $u_{av}$ , is 5 V, and the GPI control responds immediately to the external perturbations.

Figure 13 shows the performance index between the GPI controller and PID controller when the buck converter is connected a sudden dynamic load (DC Motor) parallel with the load resistance. This performance index is the integral of

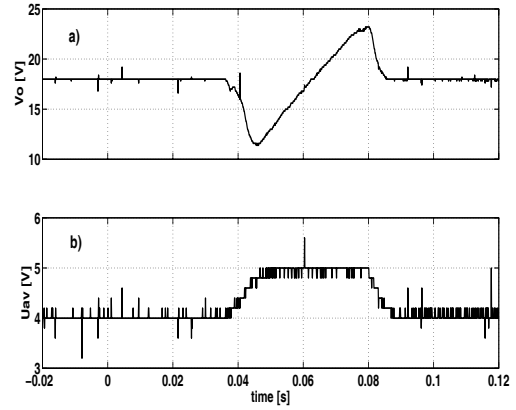


Fig. 12. Output voltage transient responses of the “buck” converter with the GPI scheme and average voltage of the GPI control.

the square of the error, ISE, which is defined as [36]:

$$ISE = \int_0^T e^2(t) dt \quad (31)$$

The upper limit  $T$  is a finite time chosen somewhat arbitrarily so that the integral approaches a steady-state value. It is usually convenient to choose  $T$  as the settling time  $t_s$ . The step response of 18 Volts for the GPI and PID feedback controllers buck converter system are shown in Figure 13(a), and the errors in Figure 13(b). The errors squared are shown in Figure 13(c), and the integral of the errors squared in Figure 13(d). This criterion will discriminate between excessively overdamped and excessively underdamped systems. The minimum value of the integral occurs for a compromised value of the damping. The performance index of Equation (31) is easily adapted for practical measurements because a squaring circuit is readily obtained [36]. Hence, using a quantitative method, we verified the performance of the buck converter system and noted that the integral of the square of the errors for a damping coefficient of 0.707, the error index of the GPI controller, was smaller than PID controller (see Figure 13(d)).

Table III exhibits the comparison between the performances of the two synthesized controllers. The main specifications of the transient response, along with the sudden dynamic load insertion recovery time, are depicted in this Table. The bandwidth of the GPI controller is greater than the PID controller (see Table III), these frequencies are calculated in the closed-loop through the damping ratio and settling time [30]. The damping coefficient value is 0.707, while the values of settling time are: 4.64 ms for the GPI and 13.64 ms for the PID.

Extensive tests were performed under different load conditions, including: 1) Sudden static load changes by activating a resistive load in parallel with the output load resistance of the “buck” converter. 2) Sudden dynamic load insertions in parallel with the load resistance and 3) The use of a magnetic brake on the motor with sudden insertions of the brake. The graphs of the figures 10 and 11, show clearly how the GPI control action presents a faster response than that achievable with the PID controller for a constant reference signal. Also,

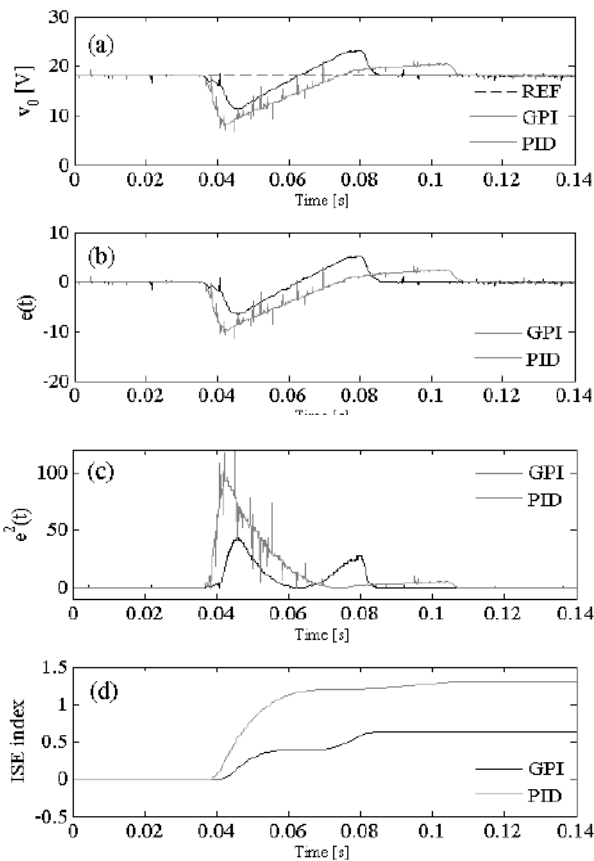


Fig. 13. The calculation of the ISE of the GPI controller and PID controller.

TABLE III  
COMPARISON BETWEEN GPI AND PID CONTROLLERS.

Specifications of the transient response	Controllers	
	GPI	PID
<b>Delay time</b> $t_d$	2.38 ms	2.52 ms
<b>Rise time</b> $t_r$	3.96 ms	4 ms
<b>Time of peak</b> $t_p$	–	6.24 ms
<b>Percentage of overshoot</b> $M_p$	0%	2.2%
<b>Settling time</b> $t_s$	4.64 ms	13.64 ms
<b>Recovery time</b> $t_{rec}$	50.8 ms	73.1 ms
<b>Bandwidth</b> $\omega_{BW}$	1219.5 Hz	414.85 Hz

the GPI controller show a faster recovery in the output voltage regulation than the PID controller when to the system is subject to sudden, unforeseen, changes in the load.

### VIII. CONCLUSIONS

In this paper, we have applied the Generalized Proportional Integral control scheme, synthesized via a Field Programmable Gate Array implementation, for the output voltage regulation in a DC/DC power converter of the “buck” type. The performance of the GPI control action was compared, under several load conditions, with that obtained from a classical PID control action, also synthesized via a FPGA. We conclude that the GPI controller has a better transient response than that achieved with a PID control action. When we connect a

static and a dynamic load to the “buck” converter output, we observed that the GPI control results in a significantly faster response, than that obtained with the PID control, regarding the output voltage recovery time to the desired reference. Finally, the experimental results show the effectiveness of the FPGA realization of both the GPI controller and the PID controller, in this case, programmed into the FPGA, with relatively the same ease of implementation.

Since the ISE index shows that the GPI control obtains a better performance from the buck converter than the PID control, this methodology can be used to design switched-mode power supplies with efficiency greater than 90.

### IX. FUTURE WORKS

As a continuation of this research, the following future works are proposed:

1. The new controllers implementations on FPGA, like passivity based-control, anti-windup, RSR polynomial, back-stepping, etc.
2. In order to obtain high frequency operation, implementing a pipeline structure of the proposed architectures.

### REFERENCES

- [1] S. Okada, T. Nunokawa, and T. Takeshita, “Digital Control Scheme of Single-Phase Uninterruptible Power Supply,” Proc. 31st International Telecommunications Energy Conference, INTELEC 2009, pp. 1-6, 2009.
- [2] J. Sitar, V. Racek, and P. Bauer, “DC/DC Converter with Active Filter Supplied from Trolley Net,” Proc. Power Electronics Specialists Conference, PESC 2007, pp. 383-389, 2007.
- [3] T. B. Marchesan, J. S. da Silveira, M. Cervi, M. A. Dalla Costa, J. M. Alonso, A. Campos, and R. N. do Prado, “Integration Methodology of DC/DC Converters to Supply HPS Lamps: An Experimental Approach,” Proc. Industry Applications Society Annual Meeting, pp. 1-5, 2008.
- [4] I. Batarseh, *Power Electronic Circuits*. New Jersey: John Wiley & Sons Inc, 2004.
- [5] J. Linares Flores and H. Sira-Ramírez, “DC motor velocity control through a DC-to-DC power converter,” Proc. 43rd IEEE Conf. Dec. Control, pp. 5297-5302, 2004.
- [6] J. Linares Flores, A. Antonio Garcia, A. Orantes Molina, Smooth Starter for a DC Machine through a DC-to-DC buck converter, *Revista de Ingeniería Investigación y Tecnología*. ISSN 1405-7743 FI-UNAM, Vol. XII, No. 2, to appear in 2011.
- [7] N. Mohan, T. M. Undeland, W.P. Robbins, *Power Electronics, Converters Applications, and Design*. 3rd ed. New York: John Wiley & Sons Inc, 2003.
- [8] M. Fliess, R. Marquez, E. Delaleau, and H. Sira-Ramírez, “Correcteurs Proportionnels-Intégraux Généralisés,” ESAIM: Control, Optimisation and Calculus of Variations, Vol.7, pp. 23-41, 2002.
- [9] H. Sira-Ramírez, “Sliding Modes,  $\Delta$ -modulators, and Generalized Proportional Integral Control of Linear Systems” *Asian Journal of Control*, Vol. 5, No. 4, pp. 467-475, December 2003.
- [10] H. Sira-Ramírez, “On the Generalized PI Sliding Mode Control of DC-to-DC Power Converters: A Tutorial” *International Journal of Control*, Special Issue in Honor of Professor V.I. Utkin’s 60th birthday. Vol. 76, Nos. 9/10, pp. 1018-1033, June/July 2003.
- [11] A. Franco González, R. Márquez and H. Sira Ramírez, “On the Generalized-Proportional-Integral Sliding mode Control of the Boost-Boost Converter,” Proc. 4th International Conference on Electrical and Electronics Engineering (ICEEE 2007), pp. 209-212, 2007.
- [12] V. Hernández and H. Sira Ramírez, “Generalized PI for swinging up and Balancing the inertia Wheel Pendulum,” Proc. American Control Conference, pp. 2809-2814, 2003.
- [13] J. Becedas, I. Payo and V. Feliu, “Generalised proportional integral torque control for single-link flexible manipulators”, Proc. IET Control Theory & Applications, vol. 4, pp. 773 - 783, May 2010.

- [14] J. Becedas, J. R. Trapero, V. Feliu and H. Sira-Ramírez. "Adaptive Controller for Single-Link Flexible Manipulators Based on Algebraic Identification and Generalized Proportional Integral Control", Proc. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 39, pp. 735 - 751, June 2009.
- [15] H. Sira-Ramírez and R. Silva-Ortigoza, "Control Design Techniques in Power Electronics Devices", Springer-Verlag, Power Systems Series, London, 2006. ISBN: 1-84628-458-9.
- [16] J. Linares Flores, H. Sira Ramírez, "Sliding Mode-Delta Modulation GPI Control of a DC Motor through a Buck Converter," Proc. 2nd Symposium on System, Structure and Control, México, 2004.
- [17] M. He and J. Xu, "Nonlinear PID in Digital Controlled Buck Converters," Proc. Applied Power Electronics Conference, APEC 2007, IEEE Twenty Second Annual, pp. 1461-1465, 2007.
- [18] Y. F. Chan, M. Moallem, and W. Wang, "Design and implementation of modular FPGA-based PID controllers", IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1898-1906, August 2007.
- [19] G. Gateau, A. M. Lienhardt, and T. Meynard, "Digital sliding-mode observer implementation using FPGA," IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1865-1875, August 2007.
- [20] M.-W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhdja, and N. Patin, "FPGA-based current controllers for AC machine drives - A review," IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1824-1842, Aug. 2007.
- [21] Y. Gao, S. Guo, Y. Xu, S. X. Lin, and B. Allard, "FPGA-Based DPWM for Digitally Controlled High-Frequency DC-DC SMPS," Proc. 3rd International Conference on Power Electronics Systems and Applications, PESA 2009, pp. 1-7, 2009.
- [22] S. Saggini, W. Stefanutti, P. Mattavelli, and A. Carrera, "Efficiency Estimation in Digitally-Controlled dc-dc Buck Converters based on Single Current Sensing," Proc. IEEE Power Electronics Specialists Conference, PESC 2008, pp. 3581-3586, 2008.
- [23] L. Corradini, P. Mattavelli, E. Tedeschi and D. Trevisan, "High-Bandwidth Multisampled Digitally Controlled DC-DC Converters Using Ripple Compensation", Proc. IEEE Transactions on Industrial Electronics, vol. 55, pp. 1501 - 1508, April 2008.
- [24] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems-A review," IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1824-1842, August 2007.
- [25] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design tools, and application domains of FPGAs," IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1810-1823, August 2007.
- [26] K. Sugahara, S. Oida, and T. Yokoyama, "High Performance FPGA Controller for Digital Control of Power Electronics Applications," Proc. Power Electronics and Motion Control Conference, pp. 1425-1429, 2009.
- [27] H. Sira Ramírez and S. K. Agrawal, *Differentially Flat Systems*. New York: Marcel Dekker, 2004.
- [28] M. Fliess, J. Lévine, P. Martin and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and applications," *Int. Journal of Control*, vol. 61, pp. 1327-1361, 1995.
- [29] M. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, 2001.
- [30] W. C. Messner and D. W. Tilbury, *Control tutorial for matlab and simulink: A web-Based Approach*, Addison-Wesley, 1999.
- [31] I. Batarseh, *Power Electronic Circuits*. John Wiley & Sons, Inc., 2004.
- [32] U. Ascher, L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM: Society for Industrial and Applied Mathematics. 1998.
- [33] E. Kreyszig, *Advanced Engineering Mathematics*. John Wiley & Sons. 2006.
- [34] IEEE Computer Society. IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985. 1985.
- [35] R. Burden, and F. Douglas, *Numerical Analysis*. Brooks/Cuelo. 2000.
- [36] R. C. Dorf and R. H. Bishop, *Modern control systems*. -12th ed. New Jersey: Prentice Hall, 2010.
- [37] L. Idkhajine, E. Monmasson, M. Wissem Naouar, A. Prata, and K. Bouallaga, "Fully Integrated FPGA-Based Controller for Synchronous Motor Drive", IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pp. 4006-4017, October 2009.
- [38] F.-J. Lin, J.-C. Hwang, P.-H. Chou, and Y.-C. Hung, "FPGA-Based Intelligent-Complementary Sliding-Mode Control for PMLSM Servo-Drive System", IEEE Transactions on Power Electronics, vol. 25, no. 10, pp. 2573-2587, October 2010.



**Eric William Zurita-Bustamante** received the bachelor's degree in Electronic Engineering in 2007 from Universidad Tecnológica de la Mixteca. He is currently working toward the MSc. degree at Universidad Tecnológica de la Mixteca. His researches interests include development and implement hardware architectures on reconfigurable logic for Power Electronics, Automatic Control and Image Processing Applications.



**Jesús Linares-Flores** received the bachelor's degree in Electronics Engineering from Universidad Autónoma de Puebla in 1994, the MSc. degree from Universidad de las Américas-Puebla in 1999, and the Ph.D. degree from Centro de Investigación y de Estudios Avanzados del I.P.N. in 2006. Since 2007, he is representative of the academic corp with name Automatization and Control of the mechatronics systems UTMIX-CA-24-PROMEP. He is Director of the Institute Electronics and Mechatronics at Universidad Tecnológica de la Mixteca. He is author of 20 technical papers in credited journals and international conferences. He is interested in the theoretical and practical aspects of feedback regulation of linear and nonlinear dynamic systems with special emphasis in Passivity-Based Control techniques and its applications in Power Electronics.



**Enrique Guzmán-Ramírez** obtained the Electronic Engineers degree from Escuela Superior de Ingeniería Mecánica y Eléctrica - IPN in 1992. He later obtained the MSc degree in Computer Engineering, Digital Systems specialty, in 2003, and the PhD degree in Computers Science, in 2008, all from the Centro de Investigación en Computación - IPN. At present, he is a Research Professor at the Universidad Tecnológica de la Mixteca. His researches interests include development and implement hardware architectures on reconfigurable logic for Control, Artificial Neural Networks and Image Processing applications.



**Hebertt Sira-Ramírez** obtained the Electrical Engineers degree from the Universidad de Los Andes in Mérida (Venezuela) in 1970. He later obtained the MSc in EE and the Electrical Engineer degree, in 1974, and the PhD degree, also in EE, in 1977, all from the Massachusetts Institute of Technology (Cambridge, USA). Dr Sira-Ramrez is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), a Distinguished Lecturer from the same Institute and a Member of the IEEE International Committee. He is a coauthor of several books on Automatic Control and the author of over 400 technical papers in credited journals and international conferences. Dr. Sira-Ramrez is interested in the theoretical and practical aspects of feedback regulation of nonlinear dynamic systems with special emphasis in Variable Structure feedback control techniques and its applications in Power Electronics.



# Bibliografía

- [1] N. Mohan, T. M. Undeland, y W. P. Robbins. *Power Electronics Converters, Application and Design*. John Wiley & Sons, INC., segunda edición, 2003.
- [2] J. Linares-Flores, A. Antonio-García, y A. Orantes-Molina. Arranque suave para un motor de cd a través de un convertidor reductor cd-cd. *Revista de Ingeniería Investigación y Tecnología.*, XII(2), 2010.
- [3] J. Benavent, A. Abellán, y E.Figueroa. *Electrónica de Potencia*. Alfaomega publicaciones, 2000.
- [4] J. A. Gualda, S. Martínez, y P. M. Martínez. *Electrónica industrial: técnicas de potencia*. Marcombo, 1992.
- [5] I. Batarseh. *Power Electronic Circuits*. John Wiley & Sons Inc, 2004.
- [6] M. Rashid. *Electrónica de Potencia*. Prentice Hall, 1995.
- [7] Jaw-Kuen Shiau, Chun-Jen Cheng, y Ching-En Tseng. Stability analysis of a non-inverting synchronous buck-boost power converter for a solar power management system. En *IEEE International Conference on Sustainable Energy Technologies. ICSET 2008*, páginas 263 –268, 2008.
- [8] T. Friedli, S. D. Round, y J. W. Kolar. Modeling the space elevator - a project oriented approach for teaching experimental power electronics. En *European Conference on Power Electronics and Applications*, páginas 1 –10, 2007.
- [9] D. G. Lamar, J. Sebastian, M. Arias, A. Rodriguez, y M. M. Hernando. A low-cost battery charger with high power factor correction based on standard peak-current mode integrated controllers. En *European Conference on Power Electronics and Applications, EPE '09*, páginas 1 –10, 2009.
- [10] Hong-Chan Chang, Cheng-Chuan Chen, Li-Chien Huang, y Chien-Chin Lin. Implementation of a power supply with the characteristics of solar panels under partially shaded. En *IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, páginas 1472 –1476, 2010.

- 
- [11] Hongmei Li y Xiao Ye. Sliding-mode pid control of dc-dc converter. En *5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, páginas 730 –734, 2010.
- [12] H. Sira-Ramírez, J. Cortés-Romero, y A. Luviano-Juarez. A gpi sliding mode control approach for linear controllable switched systems. En *11th International Workshop on Variable Structure Systems (VSS)*, páginas 289 –294, 2010.
- [13] J. Linares-Flores, J. Reger, y H. Sira-Ramírez. Speed-sensorless tracking control of a dc-motor via a double buck-converter. En *45th IEEE Conference on Decision and Control*, páginas 6229 –6234, 2006.
- [14] A. Kwasinski y P. T. Krein. Passivity-based control of buck converters with constant-power loads. En *Power Electronics Specialists Conference. PESC 2007*, páginas 259 –265, 2007.
- [15] H. Sira-Ramírez. Sliding modes,  $\delta$ -modulators, and generalized proportional integral control of linear systems. En *Asian Journal of Control*, Vol. 5, páginas 467 –475, 2003.
- [16] H. Sira-Ramírez. On the generalized pi sliding mode control of dc-to-dc power converters: A tutorial. En *International Journal of Control*, Vol. 76, páginas 1018–1033, 2003.
- [17] A. Franco-González, R. Márquez, y H. Sira-Ramírez. On the generalized-proportional-integral sliding mode control of the "boost-boost" converter. En *4th International Conference on Electrical and Electronics Engineering, ICEEE 2007*, páginas 209 –212, 2007.
- [18] V. M. Hernández y H. Sira-Ramírez. Generalized pi control for swinging up and balancing the inertia wheel pendulum. En *Proceedings of the 2003 American Control Conference*, Vol. 4, páginas 2809 – 2814, 2003.
- [19] J. Becedas, I. Payo, y V. Feliu. Generalised proportional integral torque control for single-link flexible manipulators. *IET Control Theory Applications*, 4(5):773 –783, 2010.
- [20] J. Becedas, J.R. Trapero, V. Feliu, y H. Sira-Ramírez. Adaptive controller for single-link flexible manipulators based on algebraic identification and generalized proportional integral control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(3):735 –751, 2009.
- [21] H. Sira-Ramírez y R. Silva-Ortigoza. *Control Design Techniques in Power Electronics Devices*. Springer-Verlag, 2006.



- [22] J. Linares-Flores y H. Sira-Ramírez. Dc motor velocity control through a dc-to-dc power converter. En *43rd IEEE Conference on Decision and Control, CDC*, Vol. 5, páginas 5297 – 5302, 2004.
- [23] He Mingzhi y Xu Jianping. Nonlinear pid in digital controlled buck converters. En *Applied Power Electronics Conference, APEC 2007*, páginas 1461 –1465, 2007.
- [24] Y. F. Chan, M. Moallem, y W. Wang. Design and implementation of modular fpga-based pid controllers. *IEEE Transactions on Industrial Electronics*, 54(4):1898 –1906, 2007.
- [25] A. M. Lienhardt, G. Gateau, y T. A Meynard. Digital sliding-mode observer implementation using fpga. *IEEE Transactions on Industrial Electronics*, 54(4):1865 –1875, 2007.
- [26] M. W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhodja, y N. Patin. Fpga-based current controllers for ac machine drives;a review. *IEEE Transactions on Industrial Electronics*, 54(4):1907 –1925, 2007.
- [27] Yanxia Gao, Shuibao Guo, Yanping Xu, Shi Xuefang Lin, y B. Allard. Fpga-based dpwm for digitally controlled high-frequency dc-dc smps. En *3rd International Conference on Power Electronics Systems and Applications, PESA 2009*, páginas 1 –7, 2009.
- [28] S. Saggini, W. Stefanutti, P. Mattavelli, y A. Carrera. Efficiency estimation in digitally-controlled dc-dc buck converters based on single current sensing. En *Power Electronics Specialists Conference, PESC 2008*, páginas 3581 –3586, 2008.
- [29] National Instruments. Do i need a real-time system? <http://zone.ni.com/devzone/cda/tut/p/id/10342>, último acceso (Agosto de 2011).
- [30] E. Figuerés, J. Benavent, y G. Garcerá. *Simulación de Circuitos Electrónicos de Potencia*. Alfaomega, 2001.
- [31] R. Ortega, A. Loría, P. J. Nicklasson, y H. Sira-Ramírez. *Passivity-based Control of Euler-Lagrange Systems*. Springer, 1998.
- [32] M. Fliess, J. Lévine, Ph. Martin, y P. Rouchon. Sur les systèmes non linéaires différentiellement plats. *C.R. Acad. Sci. Paris*, 1992.
- [33] Ph. Martin, R. M. Murray, y P. Rouchon. Flat systems, equivalence and trajectory generation. Reporte técnico, California Institute of Technology, 2003.

- 
- [34] C. García-Rodríguez. Estimación de estados por métodos algebraicos. Tesis de Maestría, Centro de Investigación y de Estudios Avanzados del I.P.N., 2005.
- [35] H. Sira-Ramírez y S. Agrawal. *Differentially Flat Systems*. Marcel Dekker Inc, 2004.
- [36] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.
- [37] J. Lèvine. *Analysis and Control of Nonlinear Systems: A Flatness-based approach*. Springer, 2009.
- [38] M. Fliess, R. Márquez, E. Delaleau, y H. Sira-Ramírez. Correcteurs proportionnels-intégraux généralisés. En *ESAIM: Control, Optimisation and Calculus of Variations*, Vol. 7, páginas 23–41, 2002.
- [39] M. L. Overton. *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM, 2001.
- [40] J. L. Hennessy y D. A. Patterson. *Arquitectura de Computadores, un enfoque cuantitativo*. McGraw-Hill, 1993.
- [41] Andrew S. Tanenbaum. *Organización de computadoras : un enfoque estructurado*. Prentice Hall, 2000.
- [42] IEEE. Standard for binary floating-point arithmetic, std 754-1985 edition, 1985.
- [43] J. Deschamps, G. Antoine, y G. Sutter. *Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems*. John Wiley & Sons, INC., 2006.
- [44] J. Ruz-Ortíz. *VHDL: de la tecnología a la arquitectura de computadores*. Editorial Síntesis, 1997.
- [45] F. Pardo y J. Boluda. *VHDL Lenguaje para síntesis y modelado de circuitos*. RA-MA, 1999.
- [46] M. Mendoza-Manzano. Procesamiento y análisis digital de imágenes mediante dispositivos lógicos programables. Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, 2009.
- [47] D. Pellerin y D. Taylor. *VHDL Made Easy!* Prentice Hall, 1997.
- [48] D. Maxinez y J. Alcalá-Jara. *VHDL El arte de programar sistemas digitales*. Compañía Editorial Continental, 2002.
- [49] S. Olloz, E. Villar, Y. Torroja, y L. Teres. *VHDL: Lenguaje Estándar de Diseño Electrónico*. McGraw-Hill, 1998.

- 
- [50] I. Kuon, R. Tessier, y J. Rose. *FPGA Architecture: Survey and Challenges*. Now Publisher, 2008.
- [51] International Rectifier. Ir2117(s)/ir2118(s) & (pbf), 2007. Data Sheet No. PD60146 Rev O.
- [52] J. A. Cortés-Rodríguez. Diseño y construcción de un sistema de alimentación para un arreglo rgb de tres led de potencia. Tesis de Licenciatura, Universidad Tecnológica de la Mixteca, 2009.
- [53] International Rectifier. Ir2114sspb/ir2214sspb, 2009. Data Sheet No. PD60213 revL.
- [54] STMicroelectronics. Irf640/fp. 1999. Datasheet.
- [55] Design Automation Standards Committee of the IEEE Computer Society. Ieee standard for vhdl register transfer level (rtl) synthesis, 2004. IEEE Std 1076.6-2004.
- [56] Design Automation Standards Committee of the IEEE Computer Society. Ieee standard for verilog register transfer level synthesis, 2002. IEEE Std 1364.1-2002.
- [57] National Semiconductor. Adc0820 8-bit high speed mp compatible a/d converter with track/hold function, 1995. Datasheet.
- [58] T. Akai. *Métodos numéricos: Aplicados a la Ingeniería*. Limusa Wiley, 2007.
- [59] Inc The MathWorks. *Release Notes for Matlab R2008a*.
- [60] Powersim Inc. *PSIM User's Guide*.
- [61] A. O'Dwyer. *Handbook of PI and PID controller tuning rules*. Imperial College Press, tercera edición, 2009.
- [62] B. Kuo. *Sistemas de Control Automático*. Prentice-Hall, séptima edición, 1996.
- [63] K. Ogata. *Ingeniería de Control Moderna*. Prentice-Hall, tercera edición, 1998.