

UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

DIVISIÓN DE ESTUDIOS DE POSGRADO

CONTROL GPI DE SEGUIMIENTO DE VOLTAJE DE UN INVERSOR MULTINIVEL MONOFÁSICO: UNA IMPLEMENTACIÓN EN FPGA

TESIS

PARA OBTENER EL GRADO DE:

MAESTRO EN ELECTRÓNICA Y COMPUTACIÓN

PRESENTA

ING. JOSÉ ANTONIO JUÁREZ ABAD

DIRECTOR DE TESIS

DR. JESÚS LINARES FLORES

CO-DIRECTOR DE TESIS

DR. ENRIQUE GUZMÁN RAMÍREZ

Huajuapán de León, Oaxaca a 11 de julio de 2012

Tesis presentada en julio de 2012 ante
los sinodales:

Dr. Marco Antonio Contreras Ordaz.

Dr. Carlos García Rodríguez.

M.C. Esteban Osvaldo Guerrero Ramírez.

Director de tesis:

Dr. Jesús Linares Flores

Co-director de tesis:

Dr. Enrique Guzmán Ramírez

Dedicatorias

*A toda mi familia:
A mis padres, Vicente y Catalina
A mis hermanos, Fernando y Raquel
A quien las casualidades de la vida nos ha puesto
en el mismo tiempo y espacio: Coral
Con mucho cariño
Toño.*

Agradecimientos

Por la dedicación a esta tesis, por su fina atención y
amistad:

Dr. Jesús Linares Flores

Por las imprescindibles aportaciones en la realización de
este trabajo:

Dr. Enrique Guzmán Ramírez

Dr. Marco Antonio Contreras Ordaz

Por su cooperación en la finalización de este trabajo, a
los sinodales:

Dr. Carlos García Rodríguez

M.C. Esteban Osvaldo Guerrero Ramírez

Por su generosa atención:

Carmen Torres Martínez

Resumen

En este trabajo se presenta un control de seguimiento directo de voltaje para el inversor monofásico de cinco niveles, el control se basa en la técnica GPI. La implementación del controlador se hace a través de una tarjeta FPGA SPARTAN-3E1200. Los beneficios prácticos que ofrece el control son: robustez del voltaje de salida ante cambios súbitos de carga del tipo lineal y no lineal, bajo contenido armónico de voltaje y de corriente en la carga ($< 5\%$ *IEEE* – 519), autónomo, reconfigurable y hace uso de un sólo sensor de voltaje para operar. Por otro lado, la implementación de las señales de seguimiento que emplea el controlador, así como también las señales portadoras que utiliza el modulador multinivel basado en la técnica PS-PWM se llevan a cabo dentro del FPGA. La arquitectura segmentada del controlador al igual que las señales de referencia deseadas y portadoras está basada en el estándar *IEEE* – 754 de punto flotante simple de 32 bits. La generación de las señales de seguimiento y de las señales portadoras del modulador PS-PWM se realizó por medio de la técnica de tablas de búsqueda (look-up table) empleando los módulos de BRAM que posee el FPGA; con esto se logró economizar recursos dentro del dispositivo. Además, para la implementación del controlador se utilizan los módulos Ipcores los cuales son proporcionados por el fabricante del FPGA, dichos módulos están optimizados para reducir el consumo de recursos del dispositivo. La descripción del control basado en FPGA se realiza en lenguaje VHDL usando la metodología descendente top-down. Para la medida de la única variable que necesita el controlador se utiliza un sensor de voltaje *ISO* – 124, el cual usa un convertidor analógico digital de 10 bits. Finalmente, se presentan las pruebas de desempeño del sistema en lazo abierto y en lazo cerrado bajo cambios súbitos de carga del tipo lineal y no lineal, se analizan las respuestas en estado estacionario y estado transitorio de las variables de voltaje y de corriente en la carga de salida mediante un instrumento de la calidad de la energía. Adicionalmente, se realizaron pruebas de índice de desempeño respecto a la variación de los parámetros del controlador, los resultados de dichas pruebas se presentan por medio de un conjunto de gráficas en las que se muestra la evolución en el tiempo de la función de la integral del error cuadrático (ISE).

Índice general

Dedicatorias	I
Agradecimientos	III
Resumen	V
1. Introducción	1
1.1. Introducción	1
1.2. Formulación del problema	7
1.3. Justificación	8
1.4. Preguntas de investigación	8
1.5. Hipótesis	9
1.6. Objetivos	10
1.6.1. Objetivo general	10
1.6.2. Objetivos específicos	10
1.7. Metas científicas	10
1.8. Aportaciones	10
1.9. Organización de la tesis	11
I Conceptos Básicos	13
2. Conceptos sobre Sistemas Lineales	15
2.1. Introducción	15
2.2. Conceptos Básicos de Control Lineal	17
2.3. Controlabilidad y observabilidad	19
2.3.1. Controlabilidad	20
2.3.2. Observabilidad	21
2.4. Planitud Diferencial	21
2.5. Parametrización Diferencial	23
2.6. Control Proporcional Integral Generalizado (GPI)	24

3. Dispositivos FPGA	25
3.1. Introducción	25
3.2. Historia de la tecnología FPGA	26
3.3. Estructura genérica de un FPGA	29
3.4. Arquitectura y tecnologías de configuración de los FPGAs	32
3.4.1. FPGA basado en SRAM	33
3.4.2. FPGA basado en Memoria Flash	42

II Metodología empleada

Fase I 45

4. Diseño del controlador por la técnica GPI para el inversor multinivel en cascada	47
4.1. Introducción	47
4.2. Descripción de la metodología	48
4.3. Modelado Matemático	51
4.3.1. Simulación del circuito equivalente	52
4.3.2. Modelo promedio en el espacio de estados	52
4.3.3. Modelo matemático del inversor de tres niveles	53
4.3.4. Modelo matemático del inversor multinivel en cascada con fuentes independientes	57
4.4. Análisis	61
4.4.1. Prueba de controlabilidad al sistema	62
4.4.2. Obtención de la salida plana del sistema	62
4.4.3. Prueba de observabilidad al sistema	63
4.4.4. Obtención del reconstructor de estado	63
4.4.5. Trayectorias de referencia	64
4.4.6. Diseño del Controlador GPI de seguimiento de voltaje	64
4.4.7. Elección de las ganancias del controlador	66
4.5. Simulación	67
4.5.1. Simulación en SIMNON	68
4.5.2. Simulación con MATLAB/Simulink-PSIM	70

III Metodología empleada

Fase II 75

5. Diseño de la arquitectura segmentada en FPGA	77
5.1. Introducción	77

5.2. Partición Modular	78
5.3. Simulación Funcional	79
5.4. Rediseño Digital y Gráfico de Optimización de Flujo de Datos	80
5.4.1. Administrador digital de reloj	82
5.4.2. Módulo del controlador GPI de seguimiento	83
Módulo del generador de onda senoidal y cosenoidal	86
Interfaz de conversión analógico-digital	89
Control del convertidor analógico-digital	90
Encoder de punto flotante	92
5.4.3. Generador PWM para el Modulador Multinivel	94
Generador de Onda triangular	95
Comparadores de magnitud	96
Control de tiempos muertos	96
5.5. Codificación HDL e implementación en FPGA	96
5.5.1. Reportes obtenidos de la implementación del controlador	97
5.6. Validación Experimental	98
5.6.1. Diseño del experimento	102
Pruebas experimentales en estado estacionario	104
Prueba experimental de la respuesta del sistema a un transitorio (carga lineal)	109
Prueba experimental de la respuesta del sistema a un transitorio (carga no lineal)	112
Índice de desempeño(ISE)	112
IV Conclusiones	117
6. Conclusiones y perspectivas	119
Apéndice	123
A. Tendencias en la tecnología FPGA	123
A.1. Unidad de procesamiento	123
A.2. Periféricos analógicos	123
A.3. Arquitecturas de multiprocesamiento	124
A.4. Herramientas de diseño	124

B. Diseño del filtro de salida del convertidor	127
B.1. Introducción	127
B.1.1. Filtro LC	128
B.1.2. Calculo de L y C del filtro de salida	129
B.1.3. Prueba en simulación del filtro de segundo orden con el software PSIM 9	131
B.1.4. Diseño del inductor para AC	132
Paso 1.- Cálculo de la reactancia inductiva	132
Paso 2.- Cálculo de la corriente en la bobina	133
Paso 3.- Cálculo de la potencia aparente	133
Paso 4.- Cálculo del área de ventana	133
Paso 5.- Selección del núcleo a utilizar	133
Paso 6.- Cálculo del número de vueltas	135
Paso 7.- Cálculo del GAP del entrehierro	135
Paso 8.- Cálculo del borde de flujo	136
Paso 9.- Recálculo del número de vueltas considerando el borde del flujo	136
Paso 10.- Cálculo de la densidad de flujo	136
Paso 11.- Calculo del calibre del conductor a emplear	136
Paso 12.- Resultados experimentales obtenidos del filtro diseñado	137
C. Código empleado en SIMNON	141
D. Código descargado en la BRAM del dispositivo FPGA	143
E. Descripción de los diagramas <i>top-down</i> de la implementacion en FPGA	145
E.1. Diagrama RTL	145
Diagrama general (Top-Down) del diseño	147
Descripción de los Módulos del Diagrama General del diseño	147
F. Modulo de comunicación USB PC-FPGA y datos extraídos de la arquitectura propuesta	159
G. Artículos Publicados	163
Bibliografía	165

Indice de tablas

3.1. Familia SPARTAN-6	40
3.2. Características Máximas	42
4.1. Estados de conmutación para el inversor de tres niveles	55
4.2. Estados de conmutación con $(V_{cd1} = V_{cd2})$ (ver Figura 4.5)	61
4.3. Parámetros de diseño del para el convertidor multinivel y controlador GPI	68
5.1. Contenido de la ROM_{vC}	95
5.2. Resumen de recursos usados del FPGA en la implementación del controlador GPI de seguimiento.	99
B.1. Niveles de contenido armónico	129
B.2. Parámetros iniciales de diseño	130
B.3. Parámetros para el diseño del inductor de AC	132
B.4. Parámetros para el diseño del inductor de AC (2)	135

Índice de figuras

1.1. Comparativa del inversor clásico de dos niveles y la topología multinivel en relación a la tecnología de semiconductor que emplean [1].	2
1.2. Aplicaciones de los convertidores multinivel [1].	3
1.3. Clasificación de los métodos de modulación para convertidores multinivel [1].	5
1.4. Figura extraída de [34].	7
2.1. Circuito RLC.	16
3.1. Primer concepto de compuertas programable en campo.	27
3.2. Estructura general de un FPGA actual.	30
3.3. Evolución en el proceso de fabricación de la tecnología de cobre en los FPGAS.	32
3.4. Organización interna de un FPGA de Xilinx.	34
3.5. Estructura básica de un SLICE.	35
3.6. Arreglo de SLICES en un CLB de la familia SPARTAN-3E.	36
3.7. Arreglos básicos de LUTs.	36
3.8. Tipos de interconexiones entre CLBs en la Familia SPARTAN-3.	38
3.9. Diagrama de un bloque dedicado de memoria RAM en la familia SPARTAN-3E.	38
3.10. Tarjeta de desarrollo NEXYS2 con un FPGA de la familia SPARTAN-3E.	39
3.11. Tarjeta de desarrollo con un FPGA de Altera CICLONE IV (National Semiconductor).	40
3.12. Estructura interna del FPGA de la familia CYCLONE IV.	41
3.13. Estructura de un LAB en un FPGA de la familia CYCLONE IV.	41
4.1. Metodología de diseño([91]).	50
4.2. Inversor de tres niveles.	53
4.3. Estados de conducción del inversor puente completo: (A) $v_C = V_{cd}$; (B) $v_C = 0$; (C) $v_C = -V_{cd}$ y (D) $v_C = 0$	54
4.4. Estados de conducción del inversor puente completo considerando un filtro de salida de segundo orden.	56
4.5. Inversor Multinivel de cinco niveles con filtro de salida de segundo orden.	59
4.6. Inversor Multinivel de m -niveles con filtro de salida de segundo orden.	60

4.7. Simulación de Simnon.	69
4.8. Simulación de Simnon.	70
4.9. Modelo del sistema construido en PSIM.	71
4.10. Diagrama conceptual sobre el proceso de cosimulación realizado entre <i>Matlab/Simulink</i> y <i>PSIM</i>	72
4.11. Diagrama general de bloques funcionales construidos en <i>Matlab/Simulink</i>	73
4.12. Resultado de la cosimulación entre entre <i>Matlab/Simulink</i> y <i>PSIM</i> para el inversor multinivel monofásico en cascada de cinco niveles.	74
5.1. Concepto de diseño de los módulos dentro del dispositivo FPGA del controlador GPI de seguimiento para el inversor multinivel en cascada de cinco niveles.	79
5.2. Descripción de los bloques funcionales de la Figura 4.11.	80
5.3. Representación por medio de un diagrama de flujo de datos de las ecuaciones (4.32).	81
5.4. Módulo de administración de reloj.	82
5.5. Representación por medio de un diagrama de flujo de datos de las ecuaciones (4.32) con la adición de los módulos de generación de onda senoidal, administrador de reloj, conversión analógico-digital y modulador PS-PWM.	84
5.6. Representación gráfica de la ecuación (5.3).	85
5.7. Diagrama de bloques del submódulo del generador de onda senoidal y cosenoidal.	87
5.8. Diagrama de tiempos para la generación de las ondas seno y coseno.	88
5.9. Diagrama a bloques de la interfaz de conversión analógico-digital.	90
5.10. Maquina de estados finitos para control de la interfaz del ADC.	91
5.11. Diagrama de tiempos de la adquisición de la interfaz del ADC.	92
5.12. Acondicionamiento de v_C para que pueda ser adquirido por el convertido ADC del microcontrolador dsPIC30F2010.	93
5.13. Ventana del generador de memorias del <i>Xilinx CORE Generator</i>	94
5.14. Diagrama de bloques del generador PS-PWM.	95
5.15. Señales de salida esperadas del Modulador Multinivel PS-PWM.	97
5.16. Proceso de implementación en FPGA.	98
5.17. Reporte de la colocación y ruteo generado por la herramienta EDA <i>Xilinx ISE Design Suite 9.2</i>	100
5.18. Reporte de implementación generado por la herramienta EDA <i>Xilinx ISE Design Suite 9.2</i>	101
5.19. Planta del convertidor Multinivel en Cascada de cinco niveles.	103
5.20. Convertidor Multinivel en Cascada de cinco niveles en lazo abierto (Estado estacionario).	105
5.21. Convertidor Multinivel en Cascada de cinco niveles en lazo cerrado (Estado estacionario).	106

5.22. Salida del convertidor multinivel v_C y de la corriente de carga i_L : (a) Prueba en lazo abierto; (b) Prueba en lazo cerrado.	107
5.23. Análisis de THD en corriente y de voltaje, así como también de Factor de Potencia: (a) THD corriente; (b) THD voltaje; (c) Factor de potencia.	108
5.24. Esquema propuesto para realizar la prueba al sistema ante cambios de carga lineal en lazo abierto y lazo cerrado.	109
5.25. Resultados ante un cambio de carga lineal del sistema en lazo abierto y cerrado: (a) Voltaje de salida del convertidor contra corriente de carga (lazo abierto); (b) Error de voltaje (lazo abierto); (c) Voltaje de salida del convertidor contra corriente de carga (lazo cerrado); (d) Error de voltaje (lazo cerrado).	110
5.26. Resultados de TDH y FP : (a) THD de corriente; (b) THD de voltaje; (c) Factor de potencia.	111
5.27. Esquema propuesto para realizar la prueba al sistema ante una carga no lineal.	112
5.28. Resultados ante un cambio de carga no lineal del sistema en lazo abierto y cerrado: (a) Voltaje de salida del convertidor contra corriente de carga (lazo abierto); (b) Error de voltaje (lazo abierto); (c) Voltaje de salida del convertidor contra corriente de carga (lazo cerrado); (d) Error de voltaje (lazo cerrado).	113
5.29. Resultados de TDH y FP : (a) THD de corriente; (b) THD de voltaje; (c) Factor de potencia.	114
5.30. Función de la Integral del Error Cuadrático (ISE).	115
B.1. Filtro de segundo orden del tipo RLC.	128
B.2. Esquemático construido en PSIM para probar el desempeño del filtro de salida.	131
B.3. Resultado de la simulación en PSIM: (a) Salida del convertidor antes del filtro; (b) Salida del convertidor después del filtro diseñado.	131
B.4. Parámetros de los núcleos de ferrita del fabricante Ferroxcube (Tabla 1).	134
B.5. Parámetros de los núcleos de ferrita del fabricante Ferroxcube (Tabla 2).	134
B.6. Geometría del núcleo elegido para el diseño (Familia ETD).	134
B.7. Calibre AWG del conductor para el bobinado del inductor.	138
B.8. Salida del convertidor multinivel después del filtro de segundo orden, esta gráfica fue obtenida por medio del osciloscopio <i>Tektronix</i>	139
B.9. Análisis de THD_v con el Analizador HIOKI 3197	139
E.1. Diagrama RTL de la implementación del controlador después del proceso de síntesis.	146
E.2. Diagrama general del diseño con un enfoque <i>Top-Down</i> realizado en <i>Xilinx ISE Design Suite 9.2</i>	148
E.3. Diagrama del generador de onda senoidal.	149
E.4. Diagrama del módulo de conversión analógico-digital.	150

E.5. Diagrama del módulo del controlador GPI	151
E.6. Diagrama del generador de las referencias de seguimiento para el controlador GPI	152
E.7. Diagrama de la entrada de control artificial del controlador GPI	153
E.8. Diagrama del reconstructor de estado del controlador GPI	154
E.9. Diagrama del Generador triangular del Modulador SPWM	155
E.10. Diagrama de los comparadores de magnitud del Modulador SPWM	156
E.11. Diagrama del generador de tiempo muerto del Modulador SPWM	157
F.1. Interfaz gráfica de transmisión/recepción de datos vía USB por medio del DLP-USB245M	159
F.2. Error de voltaje obtenido vía USB por medio del DLP-USB245M	160
F.3. Integral del Error de voltaje obtenido vía USB por medio del DLP-USB245M	160
F.4. Salida de control obtenida vía USB por medio del DLP-USB245M	161

Capítulo 1

Introducción

Resumen: En este capítulo se realiza la revisión del estado del arte sobre convertidores multinivel, específicamente sobre inversores multinivel y sus principales topologías, haciendo hincapié en la topología de celdas en cascada; también, en este capítulo se definen las motivaciones que originaron el estudio del inversor multinivel con celdas en cascada para este trabajo.

1.1. Introducción

La dependencia de los combustibles fósiles y el incremento progresivo de los costos por obtener recursos energéticos que satisfagan las necesidades tanto de pequeñas como de grandes ciudades, ha motivado a investigadores a desarrollar nuevas formas de obtener energía de una manera económica, limpia y sobre todo en armonía con el medio ambiente que nos rodea [1]. Durante décadas las fuentes de energía renovables (solar, viento, olas del mar, etc.) han sido objeto de investigación en diversas áreas [2, 3], una de las más importantes es la que se encarga del desarrollo de los convertidores de potencia, cuya función es integrar los sistemas de generación de energía renovable a la red de distribución y transmisión eléctrica, esta integración debe hacerse de una manera segura y eficiente, por tanto es necesario construir convertidores de alta potencia que aseguren la integración de estas etapas y garanticen la calidad de la energía [4].

De acuerdo a la Figura 1.1 el diseño del convertidor de potencia en relación al dispositivo de conmutación ha llevado a seguir caminos bien marcados [1]:

- Aumentar la capacidad en el manejo de la corriente y del voltaje en el convertidor de potencia, esto se consigue mediante la elección de semiconductores de potencia que manejan potencias elevadas a una frecuencia de conmutación entre los 10 a 20 kHz, éste tipo de solución, ya es actualmente una tecnología madura.
- Que el convertidor de potencia trabaje a mayores voltajes y maneje potencias elevadas empleando dispositivos de bajo voltaje. Este tipo de tecnología aún está en proceso de desarrollo y dentro de este grupo se encuentran las plataformas multinivel.

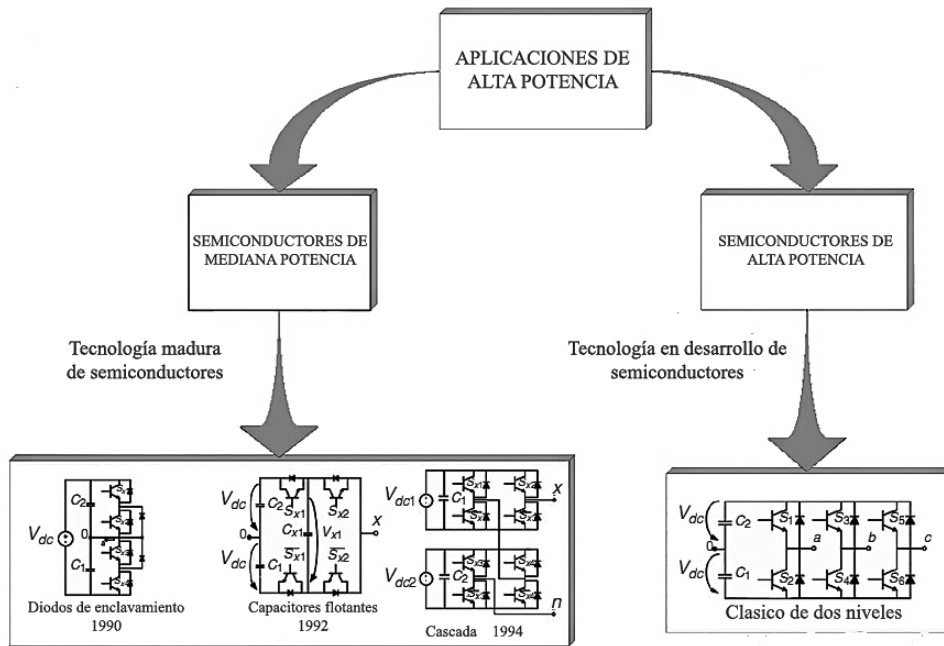


Figura 1.1: Comparativa del inversor clásico de dos niveles y la topología multinivel en relación a la tecnología de semiconductor que emplean [1].

La propuesta de la topología del inversor multinivel en [5], desarrollada en el año de 1979 y cuya idea se consolidó hasta mediados del año de 1990, presentó un convertidor NPC (*Neutral Point Clamped*) cuya estructura es parecida al convertidor clásico de dos niveles, pero con la adición de dos diodos por fase; posteriormente apareció la topología multinivel de diodos de enclavamiento reportado en [6], dicha topología poseía menor contenido armónico comparado con el convertidor NPC y además proponía una nueva técnica de modulación PWM (*Pulse Width Modulation*).

Posteriormente, con la propuesta de la topología multinivel con capacitores de enclavamiento en el año de 1992 presentada en [7] y con la topología multinivel de celdas en cascada del año 1994¹ en [8], debido al rápido crecimiento de las aplicaciones de la electrónica de potencia se amplió la perspectiva de conocimiento relativo a las estructuras multinivel, por lo que a la fecha sus aplicaciones son variadas.

Actualmente se han propuesto nuevas topologías multinivel para emplearse en una gran cantidad de soluciones (ver Figura 1.2), por ejemplo en aplicaciones en donde se emplean a los convertidores del tipo CD-CD [9], para aplicaciones en vehículos eléctricos [10, 11], para aplicaciones en aviones [12], en aplicaciones de fuentes de energía renovables [2, 3], en aplicaciones para la compensación de potencia reactiva y potencia efectiva [13, 14, 15].

Por lo anterior escrito, los convertidores multinivel presentan grandes ventajas con respecto

¹En la literatura estas son las tres estructuras tradicionales que se han investigado ampliamente, las nuevas topologías que se han propuesto son derivadas de éstas.

de balanceo adicional, requerido principalmente en los convertidores multinivel del tipo de capacitores de enclavamiento. Por otro lado, en [18] hacen nuevos desarrollos para reducir el número de semiconductores de conmutación y el número de capacitores en el convertidor, lo cual reduce los costos del convertidor y se logra tener un menor peso y tamaño, además de conseguir una mayor eficiencia. Estos diseños son muy adecuados para aplicaciones en donde se requiere una alta densidad de potencia, como por ejemplo en aplicaciones en vehículos eléctricos híbridos (HEV, *Hybrid Electric Vehicle*) tratados a detalle en [10, 11].

En [1] y [16] se comparan las tres topologías tradicionales de inversores multinivel tomando en cuenta características tales como el número de componentes, complejidad de control, eficiencia, tolerancia a fallos y modularidad; como resultado de dichas comparativas se concluye que la elección de la topología más adecuada dependerá siempre de la aplicación.

La topología en cascada de los inversores multinivel tiene como ventajas la modularidad y la simplicidad (todas las celdas² involucradas son iguales, por lo que su construcción es modular), ventajas que no se observan en las estructuras de capacitor flotante o de diodos de enclavamiento [18]. Por otra parte, la estructura en cascada no tiene problemas de balanceo de voltaje en las fuentes de alimentación (siempre y cuando estén aisladas), además de que éstas no necesitan una re-inicialización especial, desventaja importante en la topología de capacitores de enclavamiento.

Una desventaja de los inversores multinivel en el ámbito industrial es que los transformadores con varios devanados secundarios no son comunes, condición necesaria para obtener fuentes aisladas; por esta situación, la topología antes mencionada se ha ido incorporando al medio industrial con relativa lentitud [1].

Gracias a sus características eléctricas, la topología en cascada se comercializa para aplicaciones de alta potencia ya que su estructura permite conectar las celdas en serie logrando potencias hasta del rango de los 31 MVA [1]; esta estructura también se usa en aplicaciones en donde se requiere una buena calidad de la energía. En [12] por ejemplo se emplea como un filtro activo para evitar la entrada de radiofrecuencias en el sistema de alimentación de los módulos de control de aviones de pasajeros. En [19] y [20] se ha reportado el uso de esta topología en compensación de potencia reactiva. A lo largo de los años se han presentado mejoras en esta estructura multinivel en cascada tanto en convertidores como en inversores, estas mejoras se presentan de manera detallada en [17].

Las técnicas de modulación empleadas en las estructuras multinivel tradicionales (ver Figura 1.3) fueron revisadas en [1, 16, 21, 22] y los autores de estos trabajos reconocen a la técnica de modulación PS-SPWM³ (*Phase Shifted Sinoidal PWM*) como la más adecuada para la topología multinivel en cascada.

La complejidad que involucra el trabajar en lazo cerrado con el inversor multinivel en cascada

²Término aplicado en los inversores multinivel en cascada para describir a un arreglo de cuatro dispositivos de conmutación conectados en puente H.

³Esta técnica es parecida a la empleada en los convertidores de dos niveles, con la diferencia que en ésta se usa más de una portadora.

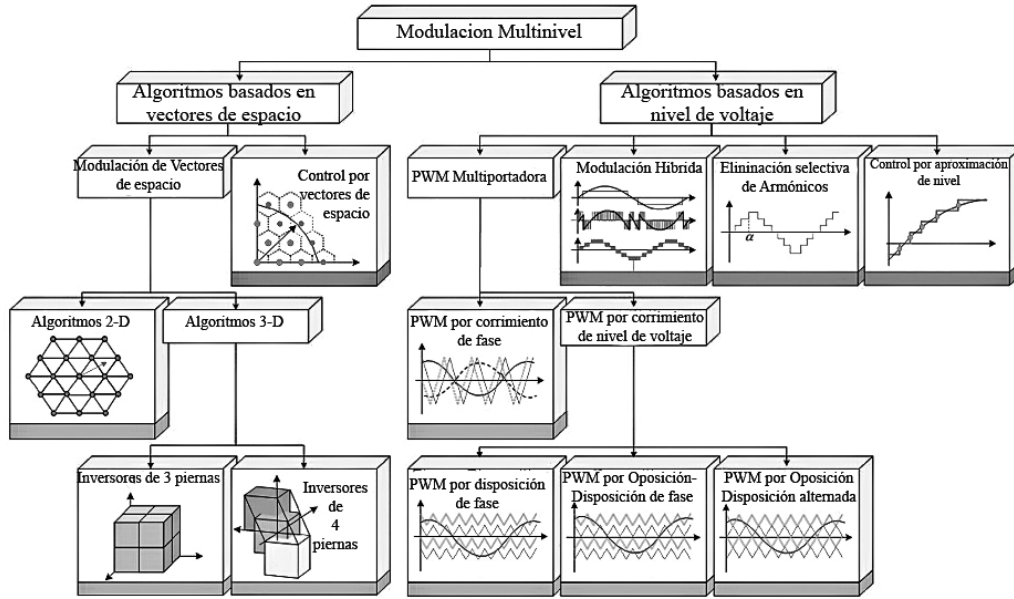


Figura 1.3: Clasificación de los métodos de modulación para convertidores multinivel [1].

depende del número de niveles que se utilicen en la estructura, en general si N_s representa el número de fuentes de poder aisladas, la relación entre el número de niveles y la cantidad de canales PWM necesarios viene dada por las siguientes expresiones:

$$\begin{aligned}
 m &= 2N_s + 1 \\
 l &= 2(m - 1)
 \end{aligned}
 \tag{1.1}$$

donde:

N_s = número de fuentes aisladas.

m = número de niveles en la onda de salida.

l = número de canales PWM para la operación de la estructura multinivel.

Actualmente, los microprocesadores que se adquieren en el mercado y que poseen generadores PWM, no pueden proveer los pulsos PWM para los convertidores multimodulares o convertidores multinivel [23], por lo que en [24, 25, 26], emplean dispositivos FPGAs (*Field Programmable Gate Array*) para hacer sólo el generador multicanal PWM.

Un FPGA es un dispositivo semiconductor que contiene bloques lógicos cuya interconexión y funcionalidad se puede configurar. La lógica reconfigurable de un FPGA, heredada de un Dispositivo Lógico Programable (PLD, *Programmable Logic Device*) [27], puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip (SoC, *System-On-Chip*). Al usar éste tipo de dispositivos se tiene la flexibilidad de modificar el circuito diseñado sin alterar algún

circuito eléctrico físico.

Otras de las ventajas del empleo de los FPGAs en éste tipo de convertidores son: el modelado de arquitecturas paralelas, la simplicidad en la configuración, la reprogramabilidad, el bajo costo (comparado con el precio de los procesadores digitales de señales que son rápidos), por lo que es una buena elección para la implementación de algoritmos de control para prototipos [28, 29, 30].

El modelado de los algoritmos puede realizarse con base en lenguajes descriptores de hardware (HDL, *Hardware Description Languages*) estandarizados por la IEEE, tales como VHDL [31] o Verilog [32]. El uso de un HDL estandarizado representa otra ventaja importante: la portabilidad, es decir el modelado del algoritmo puede ser sintetizado para otras plataformas sin ser modificado o con modificaciones ligeras. Finalmente, la síntesis de un algoritmo para un FPGA, puede ser utilizada en la fabricación de circuitos de aplicación específica (ASIC, *Application-Specific Integrated Circuit*), lo cual abre amplias posibilidades para aplicaciones industriales en cuanto a términos de costo y desempeño [33].

Por otra parte, se han realizado algunos trabajos para el inversor multinivel en cascada con filtro de segundo orden a la salida, por ejemplo en [34] se presenta un inversor con una topología de cinco niveles, en este trabajo se diseña y se implementa un controlador basado en pasividad que regula la corriente en el inductor del filtro y el voltaje en la carga de salida. Para su implementación física este controlador emplea dos dispositivos programables, el primero un FPGA para el modulador multinivel y el segundo un DSP (*Digital Signal Processor*) para su implementación digital, el tipo de solución que presentan los autores no es del todo idónea, ya que ellos emplean dos dispositivos programables para la implementación del controlador, lo cual incrementa el costo de la solución.

El trabajo reportado en [35] presenta una nueva estrategia para la generación del PWM para inversores multinivel que utilizan capacitores como fuentes auxiliares, esta estrategia es implementada en un FPGA pero sólo para regulación de voltaje empleando la técnica de control por Histéresis. Por otro lado, en [19] reportan también un control discontinuo de corriente basado en la técnica de modos deslizantes de un inversor multinivel en cascada para un DSTATCOM⁴ (*Distributed Static Compensador*), en este trabajo presentan el desempeño del controlador de corriente del convertidor a nivel de simulación y a nivel experimental.

En años recientes surgió un tipo de inversor multinivel en cascada llamado inversor multinivel híbrido que atrae la atención de los investigadores, puesto que propone sintetizar una onda de voltaje con bajo contenido armónico utilizando unas pocas celdas en serie, la ventaja de esta topología es que utiliza distintos niveles de voltaje para alimentar a las celdas, por lo que se pueden crear más niveles en el voltaje de salida que minimiza la distorsión armónica total en corriente y en voltaje en la carga de salida, sin aumentar el número de dispositivos de conmutación y de fuentes aisladas. Una desventaja importante de esta propuesta es que las técnicas de modulación empleada no se puede utilizar en convertidores multinivel de estruc-

⁴Compensador de potencia distribuido empleado en líneas de transmisión que tienen bajo factor de potencia o también para brindar estabilidad de voltaje.

tura tradicional, ya que las portadoras poseen distintas frecuencias, para más detalles véanse [36] [37].

1.2. Formulación del problema

De acuerdo a la revisión bibliográfica realizada en este trabajo sobre controladores continuos y discontinuos, en el estado del arte para el inversor multinivel de cinco niveles en cascada con filtro de segundo orden a la salida, se observó que no se ha reportado trabajo alguno sobre el control de seguimiento suave de voltaje basado en la técnica **GPI** para éste tipo de inversores, técnica que mejora el desempeño en cuanto a la eficiencia entrada - salida del inversor multinivel.

Por otra parte, los trabajos reportados sobre controladores continuos implementados en los dispositivos FPGA para éste tipo de convertidores, se encontró solamente un trabajo publicado en un congreso internacional que desarrolla un control pasivo implementado en un DSP que utiliza un modulador multinivel implementado en FPGA 1.4, el cual usa la técnica de modulación PS-SPWM (más detalles ver [34]), en este trabajo hacen regulación de corriente y de voltaje en el inductor del filtro y en el voltaje de la carga, pero no seguimiento. Además, también se realizan pruebas de robustez del controlador ante cambios de carga de tipo lineal y no lineal en la salida del inversor.

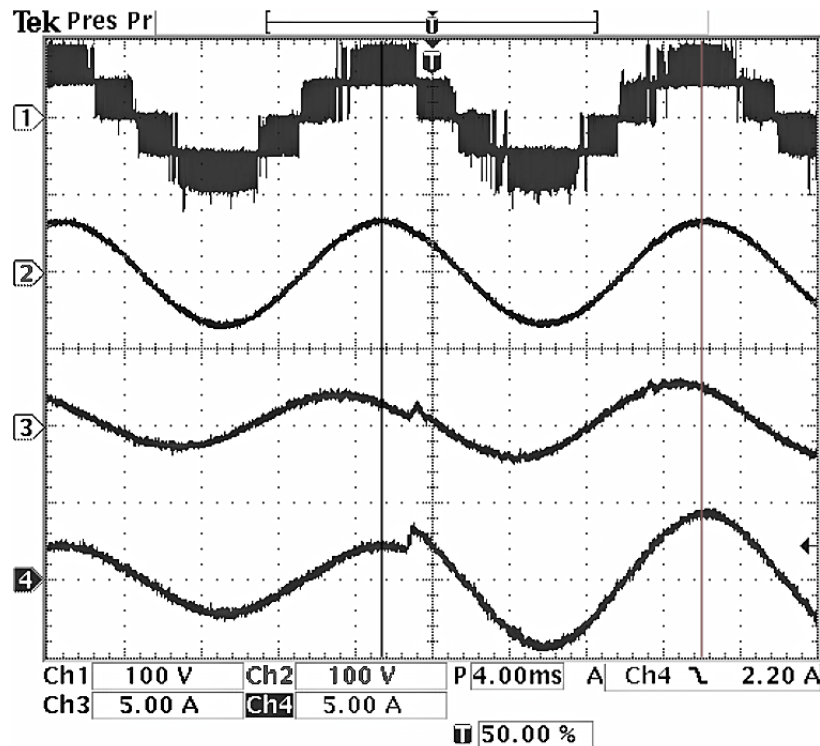


Figura 1.4: Figura extraída de [34].

Por tanto, en el presente trabajo se hace una propuesta de control de seguimiento suave de voltaje basada en la técnica GPI para el inversor multinivel de cinco niveles en cascada con filtro de segundo orden a la salida, este controlador se implementará junto con su modulador en un dispositivo FPGA (*Xilinx Spartan-3E FPGA 1200K*). La arquitectura del algoritmo del controlador-modulador será segmentada en punto flotante de 32 bits con precisión simple apegada al estándar **IEEE-754**, mientras que la técnica de modulación multinivel que se empleará será la **PS-SPWM**.

1.3. Justificación

Teniendo en cuenta que el uso de convertidores de potencia en las redes eléctricas de distribución y transmisión tiene actualmente una amplia aceptación y que la inclusión de las topologías multinivel de estos convertidores garantiza un desempeño muy por arriba de los métodos tradicionales debido a sus características eléctricas tales como: menor contenido armónico en corriente y en voltaje en la carga de salida, modularidad de la estructura multinivel, variedad de topologías; tales características justifican plenamente su aplicación. Además, actualmente con el uso de los dispositivos FPGA para implementar algoritmos de control por retroalimentación para el seguimiento suave de voltaje del tipo sinusoidal, se mejora notablemente el rendimiento del inversor multinivel en cascada en el transporte de potencia entrada-salida, aún cuando se hagan cambios súbitos en su carga de salida del tipo lineal y no lineal.

1.4. Preguntas de investigación

De acuerdo a la revisión bibliográfica realizada en este documento relativo al estado del arte en el que se encuentra el desarrollo teórico-práctico para el control de inversores multinivel a continuación se exponen las siguientes preguntas de investigación:

- **Q1.** ¿Es posible proponer un control de seguimiento suave de voltaje basado en la técnica GPI para un inversor multinivel monofásico de cinco niveles?
- **Q2.** ¿Se puede implementar la lógica necesaria que utiliza la ley control en un FPGA y que esta lógica sea capaz de seguir una señal de voltaje del tipo sinusoidal a 60 Hz con una amplitud deseada?
- **Q3.** ¿Se podrá aplicar la técnica de tablas de búsqueda⁵ para generar las funciones seno

⁵Del término en inglés Look-Up Table (LUT), en informática es una estructura de datos, normalmente un arreglo que se usa para sustituir una rutina de computación con una simple indexación, son muy útiles a la hora de ahorrar tiempo de procesamiento [38, 39].

y coseno necesarias para el control de seguimiento suave de voltaje?

- **Q4.** ¿Es posible implementar en un sólo dispositivo programable en campo (FPGA) la ley de control de seguimiento suave de voltaje junto con un modulador para el inversor multinivel en cascada?
- **Q5.** ¿Será posible que el inversor monofásico multinivel tenga en su carga de salida un factor de potencia cercano a la unidad y que posea un muy bajo contenido armónico en corriente y voltaje para que esté dentro de la norma IEEE 519?

1.5. Hipótesis

- **H1.** Debido a que el modelo promedio del inversor multinivel con filtro a la salida tiene la propiedad de ser plano, se puede diseñar en forma promedio un control vía la técnica GPI, la cual consigue regular el voltaje de salida de la carga del inversor mediante una variable de referencia suave del tipo sinusoidal planificada plenamente mediante la propiedad de planitud diferencial (ver [40, 41]).
- **H2.** Mediante una tarjeta de desarrollo basada en FPGA se puede implementar perfectamente el algoritmo de control de seguimiento suave de voltaje para regular el voltaje de salida de la carga del inversor multinivel. Asimismo, para la conmutación de los dispositivos del inversor se programa el modulador multinivel (PS-SPWM) dentro del mismo dispositivo FPGA.
- **H3.** Utilizando los módulos de memoria embebidos del FPGA se permitirá generar las funciones seno y coseno por medio de tablas de búsqueda, indispensables para realizar seguimiento de voltaje en el control propuesto.
- **H4.** Gracias a la propiedad de ejecución de algoritmos de manera concurrente, únicamente con el FPGA y auxiliado de convertidores analógico a digital es posible implementar la ley de control para el seguimiento suave de voltaje en la carga del inversor y el modulador PS-SPWM; esto con la finalidad de poder regular el voltaje de la carga del inversor multinivel en cascada de forma autónoma.
- **H5.** El voltaje y la corriente en la carga del inversor monofásico multinivel presentan la característica de poseer bajo contenido armónico y por medio del diseño de un filtro de segundo orden correctamente sintonizado se puede obtener un desplazamiento de FP casi cero y un factor de potencia cercano a la unidad.

1.6. Objetivos

1.6.1. Objetivo general

Analizar, diseñar e implementar un control basado en FPGA por la técnica GPI de seguimiento suave de voltaje para un inversor multinivel monofásico en cascada de cinco niveles con filtro a la salida.

1.6.2. Objetivos específicos

- Obtener el modelo matemático en forma promedio del inversor monofásico multinivel en cascada.
- Analizar las propiedades del modelo matemático del inversor monofásico multinivel en cascada para determinar si es posible aplicar la técnica de control GPI.
- Construir la estructura del inversor multinivel en cascada para dos celdas (cinco niveles) con dispositivos de conmutación MOSFET de canal N.
- Construir una arquitectura segmentada en punto flotante de precisión simple (32 bits) acorde al estándar IEEE-754 que será implementada en un FPGA.

1.7. Metas científicas

- Diseño y construcción de una plataforma experimental del sistema de control de un inversor monofásico multinivel topología en cascada de cinco niveles.
- Implementar en un FPGA la estrategia de control por realimentación basada en la técnica GPI para el inversor monofásico multinivel topología en cascada de cinco niveles.
- Publicar los resultados obtenidos en un artículo de revista de arbitraje estricto de la IEEE TIE (IEEE Transactions On Industrial Electronics)

1.8. Aportaciones

Las aportaciones principales al estado del arte de este trabajo de investigación son las siguientes:

- Controlador GPI de seguimiento suave de voltaje basado en FPGA para el inversor multinivel monofásico en cascada con filtro a la salida.
- Plataforma experimental del inversor monofásico de cinco niveles con filtro de segundo orden a la salida y controlador de seguimiento de voltaje implementado de forma autónoma en FPGA.

1.9. Organización de la tesis

Después de explicar los objetivos y características de este trabajo a lo largo de esta sección, el presente documento se encuentra organizado de la siguiente forma:

Después del primer capítulo, este trabajo se divide en dos partes: la primera de ellas sobre los conceptos básicos relacionados con la teoría de los sistemas de control y las tecnologías aplicadas en el presente trabajo de investigación, esta parte contiene a los capítulos 2 y 3, éstos se describen a continuación:

En el capítulo 2 se busca proveer de sustento teórico al diseño del controlador y se realiza la mención de algunos conceptos importantes sobre la teoría de los sistemas de control relacionados con la técnica de control empleada. En el capítulo 3 se realiza una breve revisión de la historia de los dispositivos FPGA y el empleo de los mismos en los sistemas de control, además, se describe brevemente la estructura del dispositivo que se utilizó en la implementación del controlador.

La segunda parte abarca las dos fases de diseño que se realizaron para implementar el controlador en el dispositivo elegido, esta parte incluye a los capítulos 4 y 5, los cuales se describen en seguida:

En el capítulo 4 se describe la metodología que se siguió para implementar el controlador GPI en el FPGA, en éste se contempla desde el modelado matemático hasta la simulación del controlador obtenido. En el capítulo 5 se describe de manera detallada el rediseño digital del controlador (elección del tiempo de muestreo, descripción en lenguaje HDL, la implementación, etc.) hasta llegar a la etapa de pruebas experimentales que se realizaron a la planta junto al controlador.

Finalmente en el capítulo 6 se encuentran las conclusiones y las perspectivas de desarrollo futuro de este trabajo de investigación.

Parte I

Conceptos Básicos

Capítulo 2

Conceptos sobre Sistemas Lineales

Resumen: En este capítulo se realiza la mención de algunos conceptos importantes sobre la teoría de los sistemas de control que son relevantes para la técnica de control empleada en este trabajo; lejos de hacer un análisis profundo de cada concepto, en este capítulo se busca proveer de sustento teórico al diseño del controlador.

2.1. Introducción

Los sistemas de control, de acuerdo a las características dinámicas del sistema, o de su modelo matemático, se pueden clasificar en dos categorías: lineales y no lineales.

Esto es correcto si se define que un sistema de control lineal es aquel que puede ser modelado por ecuaciones diferenciales lineales, y un sistema no lineal es aquel que se puede modelar con ecuaciones diferenciales no lineales. Esta definición, sin embargo, no proporciona más información acerca del problema [46].

Con el fin de profundizar y comprender esta clasificación se puede tomar el circuito RLC de la Figura 2.1, en éste, el voltaje de entrada es $V(t)$, el voltaje del capacitor $V_c(t)$ es la salida y la ecuación diferencial del sistema esta dada por la ecuación (2.1); en éste, R, L y C denotan la resistencia, la inductancia y la capacitancia respectivamente.

$$LC \frac{d^2 V_c(t)}{dt^2} + RC \frac{dV_c(t)}{dt} + V_c(t) = V(t) \quad (2.1)$$

Si se hace la siguiente asignación:

$$\begin{aligned} x_1(t) &= V_c(t) \\ x_2(t) &= \frac{dV_c(t)}{dt} = \frac{dx_1(t)}{dt} \end{aligned} \quad (2.2)$$

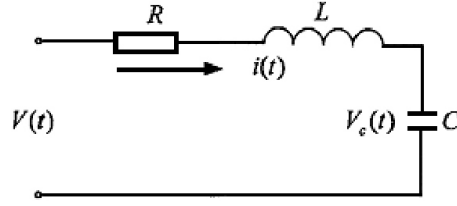


Figura 2.1: Circuito RLC.

entonces la ecuación diferencial (2.1) puede reescribirse en el espacio de estados como:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ x_2(t) &= -\frac{1}{LC}x_1(t) - \frac{R}{L}x_2(t) + \frac{1}{LC}V(t) \end{aligned} \quad (2.3)$$

o en su representación matricial:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{LC} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{LC} \end{bmatrix} V(t) \quad (2.4)$$

La ecuación (2.4) puede ser escrita de manera compacta en la forma:

$$\dot{X}(t) = AX(t) + BV(t) \quad (2.5)$$

La ecuación (2.5) recibe el nombre de forma general de las ecuaciones de estado de un sistema de control lineal. En la ecuación (2.5), si A y B son ambas matrices constantes, el sistema modelado por estas ecuaciones es un *Sistema lineal invariante en el tiempo*, o también llamado un *sistema lineal constante*.

Si ahora se consideran los efectos del envejecimiento de los componentes, se deben tomar en cuenta los efectos de la variación de los parámetros en el circuito a través del tiempo, es decir, R , L y C deben ser considerados como funciones del tiempo, estas pueden representarse como $R(t)$, $L(t)$ y $C(t)$ respectivamente. En este caso las matrices A y B de la ecuación (2.5) no son constantes, por lo que el modelo de la ecuación (2.5) se puede escribir como:

$$X(t) = A(t)X(t) + B(t)V(t) \quad (2.6)$$

La ecuación (2.6) es la forma general de representar las ecuaciones en espacio de estados de los sistemas lineales variantes en el tiempo.

De esta discusión, se puede concluir que de manera general, los sistemas lineales los hay de dos tipos: sistemas *variantes en el tiempo* e *invariantes en el tiempo*.

Continuando con el análisis para los modelos matemáticos de sistemas no lineales, se considera el circuito RLC de la Figura 2.1, pero ahora se supone un tipo especial de capacitor C cuya

capacitancia es el inverso de la función de voltaje, esto es:

$$C = f(V_c) = k_c \frac{1}{V_c} = k_c \frac{1}{x_1} \quad (2.7)$$

Donde k_c es una constante. Sustituyendo la ecuación (2.7) en la (2.3) y considerando $a_1 = -\frac{1}{k_c L}$, $a_2 = -\frac{R}{L}$, $a_3 = \frac{1}{k_c L}$ como constantes, se tiene:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= a_1 x_1^2(t) + a_2 x_2(t) + a_3 x_1(t) V(t) \end{aligned} \quad (2.8)$$

Estas ecuaciones de estado no pueden ser escritas de la forma $\dot{X}(t) = AX + BU$, sólo pueden ser escritas como:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ a_1 x_1^2(t) + a_2 x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ a_3 x_1(t) \end{bmatrix} V(t) \quad (2.9)$$

Donde:

$$\begin{aligned} f(X(t)) &= \begin{bmatrix} x_2(t) \\ a_1 x_1^2(t) + a_2 x_2(t) \end{bmatrix} \\ g(X(t)) &= \begin{bmatrix} 0 \\ a_3 x_1(t) \end{bmatrix} \\ u(t) &= V(t) \end{aligned} \quad (2.10)$$

La ecuación (2.9) tiene la forma:

$$\dot{X}(t) = f(X(t)) + g(X(t))u(t) \quad (2.11)$$

La ecuación (2.11) recibe el nombre de *forma general* de las ecuaciones de estado de un sistema no lineal.

Estrictamente hablando, la mayoría de los sistemas dinámicos son no lineales. En el diseño de sistemas de control, el método tradicional es linealizar el sistema original en uno de sus puntos de equilibrio X_e , a partir de esto, el sistema puede ser analizado acorde a los métodos de la teoría de control lineal.

2.2. Conceptos Básicos de Control Lineal

Cada rama de la ciencia posee un conjunto de conceptos básicos, estos elementos básicos componen la teoría y en esencia son parte del marco teórico de la rama en estudio.

En la teoría de control de sistemas lineales, los conceptos básicos incluyen control dinámico de sistemas, entradas y salidas, retroalimentación, variables y vectores de estado, espacio y

ecuaciones de estado, respuesta dinámica y seguimiento de trayectorias, estabilidad, accesibilidad, controlabilidad y observabilidad, índice de desempeño, control óptimo y conceptos básicos de álgebra lineal. Todos estos conceptos básicos no forman parte únicamente de la teoría moderna de control de sistemas, sino además de la teoría de control de sistemas no lineales.

Conceptos tales como transformación de coordenadas no lineales, mapeo no lineal en el espacio de estados y difeomorfismo se comparan con los de sistemas lineales para lograr mayor comprensión. Otro concepto importante es el de sistema de control no lineal afín, dado que es el tipo más importante de sistema de control no lineal, y por sus aplicaciones es el foco principal de investigación. Conceptos como vector de campo, derivada y corchete de Lie son muy importantes en el estudio de los sistemas no lineales de control.

Con el concepto de vector de campo y corchete de Lie es posible discutir el concepto de involutividad, esta es una propiedad muy importante de los vectores de campo y es usada como una condición para realizar la linealización exacta de sistemas no lineales de control, con dicha linealización se puede ilustrar el concepto de grado relativo de un sistema de control, para después discutir la forma normal de linealización de sistemas de control no lineales.

En relación al comportamiento que debe tener un sistema de control lineal, el diseñador debe de considerar las siguientes características [45]:

- **Estabilidad:** La estabilidad debe garantizarse para el modelo nominal, es decir, para el modelo que es usado para el diseño, la estabilidad puede ser local o global. En este punto es importante conocer la región de estabilidad y convergencia del sistema.
- **Exactitud y velocidad de respuesta:** La exactitud puede ser considerada como el movimiento sobre una trayectoria dentro de la región de operación del controlador. Para algunas clases de sistemas, un diseño apropiado del controlador puede garantizar un seguimiento rápido de trayectorias y de manera consistente, independientemente del tipo de trayectoria a seguir.
- **Robustez:** Es la capacidad de respuesta del sistema de control; es la sensibilidad de respuesta ante efectos que no fueron considerados en el diseño tales como disturbios, ruido, dinámicas no modeladas, etc. El sistema de control debe ser capaz de rechazar tales efectos y desarrollar sin problema la tarea de interés.
- **Costo:** En sistema de control está determinado principalmente por el número y tipo de actuadores, sensores y sistemas de procesamiento necesarios para implementarlo. Los actuadores, sensores y la complejidad del controlador (relacionada con los requerimientos de procesamiento) deben de ser elegidos de manera adecuada para cada aplicación en particular.

2.3. Controlabilidad y observabilidad

La teoría de controlabilidad y observabilidad ha sido desarrollada en respuesta a problemas generados por la tecnología científica, especialmente en áreas relacionadas con el control, comunicaciones y computadoras [47].

Parece ser que las primeras medidas conscientes para formalizar este tema como un área de investigación separada del sistema teórico o matemático se llevó a cabo a finales de 1959, por Kalman¹, sin embargo, ha habido muchos resultados dispersos y se podría afirmar que hoy en día, algunos de los principales resultados se han redescubierto más o menos de manera independiente, en todos los países donde se ha alcanzado una etapa avanzada de desarrollo [47].

Los resultados más importantes son los más o menos explícitos y computables para la controlabilidad y observabilidad de ciertas clases específicas de sistemas.

Más allá de estos resultados, existen dos teoremas fundamentales:

Teorema 1 *Un sistema dinámico lineal Σ , real, continuo en el tiempo, de dimensión n y constante tiene la propiedad de que cada conjunto de n valores propios pueden ser producidos por una adecuada realimentación de estado "si y solo si" Σ es completamente controlable [47].*

Teorema 2 (Principio de Dualidad) *Cada problema de controlabilidad en un sistema dinámico lineal que es real, continuo o discreto en el tiempo y de dimensión finita, es equivalente a un problema de controlabilidad en un sistema dual² [47].*

Debido a que los resultados de las preguntas generadas por el teorema 2 y parcialmente debido también a las técnicas matemáticas necesarias para probar el teorema 1 y sus lemas, en los inicios de 1960 estos resultados fueron combinados (Ver teorema 3) y sorpresivamente fue encontrada una relación muy importante entre la controlabilidad y la observabilidad, este resultado está plasmado en el siguiente teorema:

Teorema 3 (Descomposición canónica) *Cada sistema dinámico lineal, real, (continuo o discreto en el tiempo), de dimensión finita y constante puede ser canónicamente descompuesto en cuatro partes, de las cuales sólo una parte es completamente observable y completamente controlable, además de estar involucrada en el comportamiento de la entrada y salida del sistema [47].*

Kalman fue una figura central en el desarrollo de la teoría de sistemas matemáticos sobre la que descansa la mayoría de los métodos de variables de estado. En estos conceptos se afronta la relación que existe entre la entrada y el estado (la controlabilidad), y entre el estado y la salida (la observabilidad), es decir, describen la interacción entre el mundo externo (entradas

¹A Rudolph Kalman se le conoce por su papel en el desarrollo del denominado filtro de Kalman, que fue decisivo en el éxito de los alunizajes del programa Apollo [48]

²Este principio relaciona dos sistemas de tal forma que si el sistema S_1 es de estado completamente controlable (observable) si y sólo si el sistema S_2 es de estado completamente observable (controlable).

y salidas) y las variables internas del sistema (estados).

Se puede decir que la controlabilidad es la propiedad que indica si el comportamiento de un sistema puede ser controlado por medio de sus entradas, mientras que la observabilidad es la propiedad que indica si el comportamiento interno del sistema puede detectarse midiendo sus salidas.

Aunque la mayor parte de los sistemas físicos son controlables y observables, los modelos matemáticos correspondientes tal vez no posean la propiedad de controlabilidad y observabilidad. En este caso es necesario conocer las condiciones bajo las cuales un sistema es controlable y observable.

Las condiciones de controlabilidad y de observabilidad determinan la existencia de una solución completa para el problema del diseño de un sistema de control las cuales se describen a continuación.

2.3.1. Controlabilidad

Consideremos el sistema lineal invariante en el tiempo de n estados y p entradas dado a continuación:

$$\dot{x} = Ax + Bu \quad (2.12)$$

donde $A \in \mathbb{R}^{n \times n}$ y $B \in \mathbb{R}^{n \times p}$.

La controlabilidad relaciona las entradas y los estados del sistema, por lo tanto en la ecuación (2.12) la salida del sistema es irrelevante.

Por tanto se tiene lo siguiente:

Definición 1 *La ecuación de estados (2.12), se dice controlable si para cualquier estado inicial $x(0) = x_0 \in \mathbb{R}^n$ y cualquier estado final $x_1 \in \mathbb{R}^n$, existe una entrada que transfiere el estado x de x_0 a x_1 en un tiempo finito. En caso contrario, la ecuación (2.12), se dice no controlable.*

Para el sistema dado por la ecuación (2.12), se puede determinar si el sistema es controlable examinando la condición algebraica:

$$\text{rango} \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} = n \quad (2.13)$$

donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$ y n representa el orden del sistema.

Para un sistema de única entrada y única salida (**SISO**, *Single Input-Single output*), la matriz de controlabilidad ς se describe en términos de A y B de la siguiente manera:

$$\varsigma = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix} \quad (2.14)$$

donde $\varsigma \in \mathbb{R}^{n \times n}$ recibe el nombre de *matriz de controlabilidad*, y los vectores $B \ AB \ A^2B \ \dots \ A^{n-1}B$ deben ser linealmente independientes si el sistema es controlable.

Por lo tanto si se cumple que el $\det [\zeta] \neq 0$, el sistema es controlable [47], tal como lo muestra la siguiente ecuación:

$$\det [\zeta] = \det \left[\begin{array}{cccccc} B & AB & A^2B & \dots & A^{n-1}B \end{array} \right] \neq 0 \quad (2.15)$$

2.3.2. Observabilidad

El concepto de observabilidad es dual al de controlabilidad, e investiga la posibilidad de estimar el estado del sistema a partir del conocimiento de la salida.

Sea el sistema lineal invariante en el tiempo descrito por el siguiente modelo:

$$\begin{aligned} x(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2.16)$$

Donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, $x(0) = x_0$, $t \in [0, \infty]$ y $y(t) \in \mathbb{R}^m$ es considerado como un vector de salida y $C \in \mathbb{R}^{m \times n}$ como una matriz de salida.

Partiendo del sistema anterior, se tiene lo siguiente:

Definición 2 *El sistema Lineal Invariante en el tiempo se dice observable si, para cualquier tiempo t_1 , el estado inicial $x(0) = x_0$ puede ser determinado mediante el historial de la entrada $u(t)$ y de la salida $y(t)$ dentro de un intervalos $[0, t_1]$.*

En otras palabras el sistema mostrado en (2.16) es completamente observable cuando el determinante de la matriz de observabilidad (ver ecuación (2.17))

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.17)$$

es distinto de cero, es decir:

$$\det [O] = \det \left[\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \right] \neq 0 \quad (2.18)$$

2.4. Planitud Diferencial

Este concepto fue introducido a mediados de los años 90's por Fliess, Lévine [50] y Murray [51], en estos, se introdujo a una clase especial de sistemas de control lineales y no lineales,

descritos por ecuaciones diferenciales ordinarias, que fueron llamados *sistemas diferencialmente planos*, estos forman una estructura de sistemas para los cuales los métodos de control están disponibles una vez que se conoce explícitamente alguna salida plana.

La planitud diferencial es una propiedad estructural de una clase de sistemas lineales y no lineales que permite reconstruir todas las variables de estado y entradas de control de un conjunto de variables específicas (salidas planas) y sus derivadas temporales [52].

Intuitivamente, un sistema se dice que es diferencialmente plano, si se puede encontrar un conjunto de variables llamadas salidas planas de tal forma que todos los estados medidos se puedan determinar a partir de las salidas planas sin integración [50].

En otras palabras, un sistema **SISO** es plano, si existe una *variable endógena*³, llamada salida plana, tal que la entrada u y la salida y pueden ser expresadas como una combinación lineal de la salida plana y un número finito de sus derivadas.

Matemáticamente sería:

Dado el sistema:

$$\begin{aligned}\dot{x} &= f(x, u) \\ z &= h(x)\end{aligned}\tag{2.19}$$

Donde $x \in \mathbb{R}^n$ representa los estados del sistema, $u \in \mathbb{R}$ designa a la entrada y $z \in \mathbb{R}$ es la salida del sistema, por tanto, se dice que el sistema (2.19) es plano si existe una salida:

$$y = \phi(x, \dot{x}, \dots, x^n)\tag{2.20}$$

de tal forma que:

$$x = \Phi(y, \dot{y}, \dots, y^k)\tag{2.21}$$

$$u = \Psi(y, \dot{y}, \dots, y^{k+1})\tag{2.22}$$

$$z = \varphi(y, \dot{y}, \dots, y^k)\tag{2.23}$$

A las ecuaciones (2.21), (2.22), (2.23) en conjunto se les denomina la *parametrización diferencial* del sistema en función de la salida plana dada por la expresión (2.20).

Para determinar la salida plana se tiene la siguiente:

Proposición 1 *La salida plana del sistema lineal, controlable que tiene la siguiente forma en el espacio de estados [53]:*

$$x(t) = Ax(t) + Bu(t)\tag{2.24}$$

Está determinada por un módulo de factor constante y por una combinación de los estados

³Se dice que una variable en un sistema es endógena si esta puede ser expresada como una combinación lineal de la entrada, la salida y un número finito de sus derivadas temporales [53]

obtenidos del último renglón de la matriz de controlabilidad de Kalman, es decir:

$$y = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}^{-1} x \quad (2.25)$$

En el caso de los sistema lineales, la planitud está directamente relacionada a la controlabilidad del sistema (ver [54]), esta relación esta resumida en la siguiente:

Proposición 2 *Un sistema lineal, invariante en el tiempo del tipo **SISO**, dado por la expresión (2.26), es plano si y solo si los polinomios del numerador $n(s)$ y denominador $d(s)$ en la función de transferencia son coprimos⁴.*

En otras palabras, un sistema de la forma (2.26) es plano si y solo si el sistema es controlable.

$$y(s) = \frac{n(s)}{d(s)}u(s) \quad (2.26)$$

Se supone que el sistema de la forma (2.26) tiene un polinomio en el numerador $n(s)$ que puede ser constante, mientras que el polinomio $d(s)$ del denominador es un polinomio de grado ≥ 1 . En resumen, si el sistema es controlable en consecuencia es plano (para mayor referencia ver [53]).

2.5. Parametrización Diferencial

La parametrización diferencial proporciona a la propiedad de planitud diferencial el uso de información sobre las variables de estado, entradas y salidas del sistema en términos de la salida plana, la cual como ya se comentó, relaciona a cada una de las variables del sistema [53].

La aplicación directa de esta parametrización en algunos sistemas se comenta a continuación [53, 55] :

- Una parametrización estática del sistema en equilibrio relaciona los valores de equilibrio de las variables de estado en términos de la salida plana también en equilibrio. Para un seguimiento dinámico de trayectoria en lazo cerrado, la parametrización resulta ser una herramienta muy útil, ya que a través de ella se establecen los valores iniciales, los valores intermedios y los valores finales de los puntos de equilibrio de las diferentes variables de estado.
- Inspección directa de la propiedad de fase mínima⁵ y de fase no mínima de cada una de las variables de estado cuando se toman como salida.

⁴Dos polinomios se dicen coprimos si no tienen factores comunes.

⁵Las funciones de transferencia que no tienen polos ni ceros en el semiplano derecho del plano s son funciones de transferencia de fase mínima; en consecuencia, los sistemas con funciones de transferencia de fase mínima se denominan *sistemas de fase mínima*.

- Obtención de la entrada de control por inspección directa, ya que ésta puede ser expresada en términos de la salida plana y sus derivadas sucesivas, muy útil para la reconstrucción de estados.
- Existen diferentes usos directos de la parametrización diferencial derivados de la propiedad de planitud, pero tal vez una de las aplicaciones más tradicionales y ventajosas de la planitud diferencial se encuentra en la planificación de trayectorias para el control de seguimiento en sistemas diferencialmente planos. La parametrización diferencial proporciona de forma natural una relación explícita entre los estados y la entrada de control para lograr la planificación de la trayectoria de la salida plana; además, permite una evaluación fuera de línea de los estados nominales y de las trayectorias de la entrada de control, sin tener que resolver las ecuaciones diferenciales, estos aspectos son muy importantes en los sistemas de control y en la realización de este trabajo.

2.6. Control Proporcional Integral Generalizado (GPI)

El control proporcional integral generalizado (**GPI**, *Generalized Proporcional Integral*) o control basado en reconstructores integrales, es un desarrollo reciente en el área de control automático. Su principal línea de desarrollo descansa en los sistemas lineales de dimensión finita con algunas extensiones a sistemas diferenciales lineales con retardo y sistemas no lineales [56, 57].

El control GPI esta basado en la reconstrucción estructural del vector de estados a través de entradas y salidas iterativas con compensación integral del error. Con lo que se obtienen las leyes de control basadas en retroalimentación de estados que no necesitan observadores ni cálculos basados en muestreos de la salida [57] . Para mayor referencia ver [53, 58, 60, 117]

Capítulo 3

Dispositivos FPGA

Resumen: Los sistemas de control basados en dispositivos analógicos, a pesar de sus inconvenientes tales como la desviación en sus parámetros y complejidad en su integración permanecen como marco de referencia en términos de rapidez y ancho de banda, por ese motivo, los tiempos de ejecución de los controladores digitales deben ser reducidos, pero se debe mantener la flexibilidad inherente que ofrecen los diseños basados en soluciones digitales; en este capítulo se toma como principal foco a los FPGAs y se realiza una breve revisión sobre el empleo de éste tipo de dispositivos digitales en los sistemas de control.

3.1. Introducción

A la hora de elegir el elemento central de procesamiento de un sistema, muchos diseñadores tienen la impresión que se tiene que escoger ya sea un **FPGA**(FPGAs, *Field Programmable Gate Array*) o un procesador, pero a menudo se pueden utilizar ambas tecnologías sinérgicamente para producir diseños óptimos en términos de rendimiento, escalabilidad y ahorro de materiales.

Desde que Xilinx inventó los FPGAs en 1984, han pasado de ser sencillos chips de lógica de acoplamiento a elementos capaces de reemplazar a los circuitos integrados de aplicación específica (ASICs, *Application-Specific Integrated Circuit*).

Los FPGAs disponibles a mediados de los años 80 se usaban como un dispositivo de lógica complementaria, es decir, ayudaban al procesador a trabajar con otros circuitos discretos dentro del sistema. Sin embargo, el aspecto reprogramable de los FPGAs permitió que los equipos de diseño pudieran resolver problemas fácilmente, o incluso añadir nuevas características de última hora, por lo tanto, en esos años, los FPGAs permitieron a las compañías construir diseños con mayor escala de integración, diseños más pequeños, versátiles y energéticamente más eficientes, lo cual redundaba en bajos costos de producción.

En la etapa temprana de los años 90, los FPGAs empezaron a asumir tareas de control cada vez más complejas. El paralelismo inherente de los FPGAs permitió acoplarse perfectamente

al procesamiento de algoritmos complejos y aplicaciones de alto rendimiento, tareas que solo desempeñaban **DPS**(DSP, *Digital Signal Processor*) de alto rendimiento.

En la última década el FPGA ha tenido un crecimiento grande y acelerado, ha asumido un rol central en aplicaciones de procesamiento de diversos sistemas, ya sea dentro del tratamiento digital de señales, comunicaciones, sistemas de visión, sistemas de control automático, control de motores, etc.

Recientemente, los fabricantes han desarrollado FPGAs que incluyen entre otros: subsistemas de procesamiento ARM, mayor cantidad de periféricos, controladores de memoria, módulos DSP; todo dentro del mismo encapsulado del FPGA.

La tecnología de arreglos de compuertas programables en campo (FPGA) continúa siendo impulsada. Se prevé que el mercado de circuitos integrados de lógica programable (entre ellos los FPGAs) aumente de \$3.5 millones en 2009 a \$9.6 millones para el año 2016 [61], debido en parte a la demanda de dispositivos que brinden flexibilidad y satisfagan las oportunidades que ofrece Internet y los requerimiento de las conexiones de banda ancha.

La adopción de la tecnología FPGA continúa creciendo mientras que las herramientas de alto nivel evolucionan cada día más, para ofrecer a los ingenieros e investigadores con diferentes niveles de experiencia, los beneficios del silicio reprogramable.

3.2. Historia de la tecnología FPGA

Los dispositivos programables, o también llamados arreglos lógicos programables (PLAs, *Programmable Logic Arrays*) fueron desarrollados desde 1970. Sin embargo, por algunos años, su uso fue limitado, principalmente debido a razones tecnológicas.

Al inicio de la década de los años 80, los PLAs empezaron a ser usados en circuitos lógicos, pero tuvieron problemas con el consumo de energía [62]. La idea de fabricar un arreglo de compuertas basado en bloques lógicos personalizables (LBs, *Logic Blocks*) que se pudieran ensamblar arbitrariamente y además estuvieran rodeados en todo su perímetro por bloques de entrada y salida, dio inicio al primer concepto de arreglo de compuertas programable en campo (ver Figura 3.1) o también llamado FPGA, introducido por Xilinx y Freeman, compañía que fue fundada en 1984 y cuyo primer chip consistía en 85,000 transistores [63] (el equivalente aproximadamente a 1000 compuertas) y fue fabricado con un proceso de $2\mu m$.

Durante la década de los años 80 el incremento en precio en la fabricación de los ASICs fue continuo, pero de manera paralela, el avance en la tecnología de manufactura de semiconductores hizo que los FPGAs aparecieran como una alternativa para incrementar el número de aplicaciones donde intervenían los circuitos lógicos digitales, al grado, de permanecer hasta la actualidad como una fase previa a la elaboración de ASICs y también como una herramienta alternativa para realizar prototipos de manera rápida y barata. No obstante, fue hasta la década de los años 90 cuando la tecnología de los FPGAs alcanzó un nivel de madurez de tal

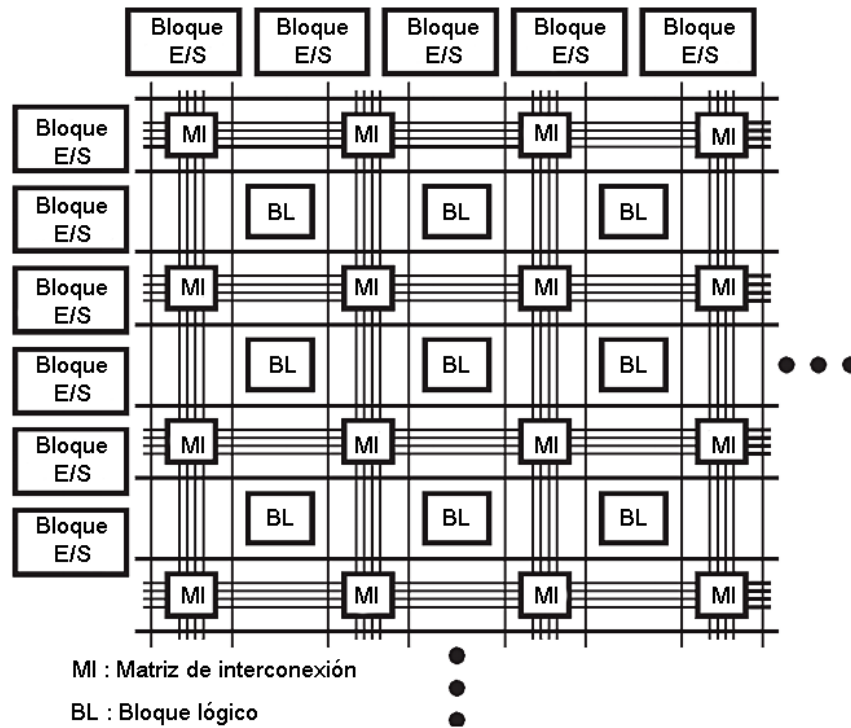


Figura 3.1: Primer concepto de compuertas programable en campo.

forma que se puso en práctica en campos tales como el de las comunicaciones y el procesamiento digital de señales.

Actualmente, los vendedores de lógica programable observan un desplazamiento de los ASICs y de los ASSP ¹ (ASSP, *Application Specific Standard Product*) hacia los FPGAs, abriendo un mercado de oportunidades significativo, esto se debe gracias a factores como: precio de producto, tiempo en salir al mercado, desempeño, rentabilidad, calidad, consumo de potencia, densidad y actualización en campo que ofrecen los dispositivos FPGA [64].

Los circuitos lógicos programables (PLD, *Programmable Logic Device*) poseen hoy en día una reducción en el consumo de potencia, pero la necesidad de alargar la carga de las baterías en los dispositivos portátiles donde se emplean estos PLDs se está incrementando. La demanda en el mercado por la mejora de los equipos de electrónica portátil incluyen la necesidad de poseer un bajo consumo de energía en estado activo y al mismo tiempo, un bajo consumo en estado inactivo, sin embargo, el factor competitivo principal para los PLDs es dar respuesta a las aplicaciones y dispositivos que requieren amplios anchos de banda, tales como procesadores de señales en equipos de comunicaciones y en sistemas de control empleados en la industria. Hoy en día, en lo que respecta a los sistemas de control empleados en la industria, para hacer

¹Es un circuito integrado que implementa una función específica la cual aplica a un mercado más amplio. En contraposición a los ASICs, el cual combina un grupo de funciones y se encuentra diseñado por o para un consumidor, los ASSPs se encuentran disponibles como componentes de consumo masivo. ASSPs se utilizan en todas las industrias, desde las automotrices hasta las de comunicación.

la diferencia en el mercado, los nuevos sistemas de control deben de ser de alto desempeño, muy flexibles y confiables[65, 66, 67].

Al mismo tiempo, una de las motivaciones principales en la construcción de controladores industriales (los cuales regularmente se emplean en electrónica de potencia y control de motores), es disminuir el costo de implementación; en ese sentido, para que el controlador industrial tenga un precio bajo, es necesario reducir el tiempo de desarrollo e implementación del controlador para reducir el tiempo de lanzamiento al mercado y además, por otra parte, también se debe considerar que el consumo de energía del controlador sea reducido [68].

Actualmente la reducción de costes cada día es aún más difícil, dado que estos nuevos sistemas de control industrial se basan en algoritmos de control muy sofisticados, los cuales necesitan mucho más recursos de cómputo y de tiempos de ejecución muy reducidos [69].

Para hacer frente a estos desafíos, los diseñadores pueden basarse en las tecnologías de la electrónica que ya son maduras, estas tecnologías actualmente vienen con herramientas software de desarrollo que son más amigables.

Para implementar eficientemente sistemas de control industriales en tiempo real, el diseñador tiene la posibilidad de elegir entre dos familias principales de tecnología de dispositivos digitales: La primer familia se basa en una plataforma de software puro, los dispositivos asociados a esta familia son microcontroladores y procesadores digitales de señales (DSP); estos componentes integran un microprocesador con varios periféricos que son necesarios para controlar el sistema y comunicarse con el medio industrial.

La diferencia entre microcontroladores [70] y los DSP [71] dada una superficie de silicón, se puede establecer en relación a la unidad central de procesamiento que se emplea, así como también a la capacidad de comunicación y cantidad de periféricos que posee cada plataforma. Generalmente los microcontroladores poseen un conjunto de instrucciones de 16 o 32 bits reducido (RISC, *Reduced Instruction Set Computer*), un núcleo de 16 o 32 bits capaz de procesar el ancho de palabra respectivo y una gran cantidad de periféricos (convertidores analógico-digital, decodificadores de cuadratura, bloques de comunicación: CAN, USART, etc.), mientras que los DSP integran un procesador de alto desempeño y bloques hardware de aceleración de operaciones (MAC ALU, *Multiply and Accumulate Arithmetic Logic Unit*), pero posee pocos periféricos; sin embargo, actualmente las limitaciones entre estas dos plataformas están desapareciendo poco a poco ya que las unidades centrales de proceso de los microcontroladores se hacen cada día más potentes y el número de periféricos en los DSP se está incrementando. Una de las principales ventajas del uso de estas tecnologías es que son económicas y actualmente se consideran maduras, ya que la calidad de las herramientas asociadas para el desarrollo de aplicaciones con estos dispositivos son muy amigables.

La principal limitación de los microcontroladores y DPS es que la ejecución de instrucciones se realiza de manera secuencial, por lo que no se puede aprovechar el procesamiento en paralelo que ofrecen algunos algoritmos de control, y en consecuencia, el desempeño es limitado pues no se puede mejorar la productividad en el procesamiento o elegir mejores anchos de banda.

La otra alternativa disponible de dispositivos digitales para implementación de sistemas de control en la industria son los ya mencionados, arreglos de compuertas programables en campo [72]. Estos componentes consisten en celdas prediseñadas e interconexiones que son totalmente programables por el usuario final, y que pueden ser empleadas para construir arquitecturas adecuadas a la medida para alguna aplicación en específico.

Actualmente existe una gran variedad de diseños de controladores basados en FPGA, pues permiten la implementación eficiente de procesadores de 32 bits o más. Como consecuencia de esto, inicialmente el FPGA fue visto como un microcontrolador programable y los diseñadores combinaron uno o varios procesadores RISC con periféricos y hardware de aceleración dedicados.

Desde esta perspectiva y gracias al incremento de densidad, los FPGAs modernos pueden ser vistos como plataformas digitales SoC² (SoC, *System on Chip*) y actualmente pueden diseñarse arquitecturas específicas para aplicaciones que son exigentes en términos de desempeño y que necesitan altas velocidades de ejecución gracias a la implementación de procesamiento en paralelo que se puede realizar en este tipo de dispositivos [73].

Aunque los microcontroladores y DSP salieron a escena en los años 80 y ahora son tecnología madura, actualmente los FPGAs también poseen herramientas de desarrollo muy amigables. La principal limitación de la tecnología de los FPGAs es el costo, pues comparados con los DSPs y microcontroladores, los FPGAs resultan ser más caros. Otra limitante en el uso de los FPGAs es la dificultad relacionada con la integración de estos con periféricos mixtos (analógicos/digitales), por lo que es necesario adecuar las señales por medio de convertidores analógico-digital (ADC, *Analog to Digital Converter*) o convertidores digital-analógico (DAC, *Digital to analog converter*) según sea el caso para que puedan ser procesadas; actualmente esta limitante está desapareciendo poco a poco, ya que se están introduciendo FPGAs al mercado que integran en su estructura convertidores analógico digital [74].

3.3. Estructura genérica de un FPGA

En la Figura 3.2, se muestra la estructura básica de un **FPGA**, esta consiste en un "mar" de bloques lógicos (LBs, *Logic Blocks*), una red de interconexión y bloques configurables de entrada y salida (*I/O Blocks*), todos estos se describen a continuación:

- **Bloques lógicos configurables:** Los Bloques Lógicos Configurables (CLBs, *Configurable Logic Blocks*) son recursos lógicos que permiten al usuario realizar diferentes funciones; los CLBs están distribuidos en forma matricial en el dispositivo. Estos recursos son de complejidad baja, es decir, las funciones lógicas que se pueden implementar en ellos son sencillas, pero existe un número considerable de ellos, en este caso, se dice

²Se refiere a la integración de todos los componentes de una computadora u otro sistema electrónico en un solo circuito integrado, éste tiene funciones digitales, analógicas o una mezcla de ambas, así como también funciones de radio-frecuencia

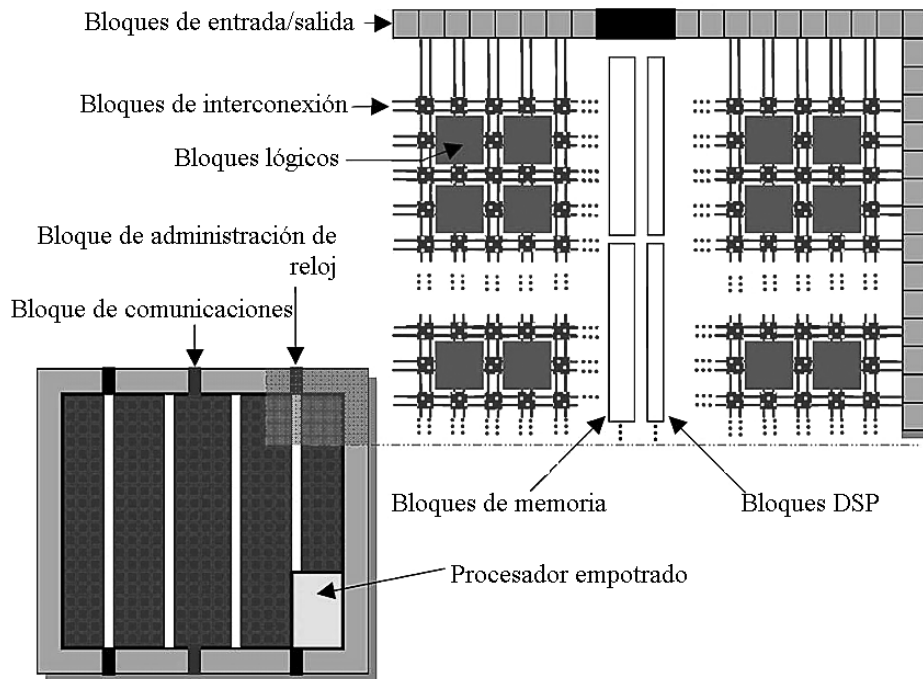


Figura 3.2: Estructura general de un FPGA actual.

que el **FPGA** es de granularidad fina. Se dice que el **FPGA** es de granularidad gruesa cuando los recursos lógicos están formados por memorias de acceso aleatorio llamadas tablas de búsqueda (LUT, *Look-up Tables*), por *flip-flops* para almacenamiento de elementos que dependen de la señal de reloj, de multiplexores que permiten la selección, de reset y de puesta a uno lógico de sus elementos; en un **FPGA** de este tipo el número de CLBs que lo integran es reducido pero tienen la característica de poder implementar funciones de mayor complejidad.

- Bloques configurables de entrada/salida:** La matriz de bloques de CLBs está rodeada por un anillo de bloques de interfaz, denominados bloques configurables de entrada/salida. Estos bloques están dedicados a proporcionar la interconectividad entre el **FPGA** y el exterior, es decir, controlan la entrada y salida 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 dotado de *flip-flops*, latches y buffers de tercer estado. Además, en ocasiones incluye resistores *pull-up*³ y/o *pull-down*⁴) en la salida. La polaridad de la señal de salida es programable.

³En electrónica se denomina *pull-up* a la acción de elevar la tensión de salida de un circuito lógico, a la tensión que, por lo general mediante un divisor de tensión, se pone a la entrada de un amplificador con el fin de desplazar su punto de trabajo

⁴Consiste en una resistencia puesta entre una línea y masa. A diferencia de las resistencias de *Pull-Up* tienen como objetivo dar a esa línea un valor lógico de nivel bajo de masa (0 V) cuando no hay ninguna señal conectada.

- **Interconexiones programables:** Están formados por tres elementos: 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 la matriz de interconexión, este último elemento lógico facilita la comunicación entre los canales de comunicación o también llamados recursos de interconexión.

Existe un tipo de recursos exclusivos de conexión que recibe el nombre de líneas dedicadas a la transmisión de las señales de reloj (CMB, *Clock manager blocks*). Esto es así debido a que las señales de reloj tienen la característica especial de que han de estar conectadas a un gran número de bloques por lo que han de llegar a todos los rincones del FPGA en el menor tiempo posible. Estas líneas de reloj global son diseñadas para obtener tiempos de propagación pequeños y similares. Estas señales de reloj son distribuidas mediante *buffers* de reloj especiales, conocidos como *drivers* de reloj, y están distribuidos alrededor del FPGA. Estos *buffers* se encuentran conectados al reloj principal y lo llevan hacia todas las líneas de reloj global con el fin de que puedan ser utilizadas por cada bloque lógico.

Actualmente debido a que los FPGAs modernos se fabrican con una alta escala de integración, estos poseen en su estructura interna bloques de memoria, bloques DSP, bloques de administración de reloj y bloques de comunicación [62].

Dependiendo de la función a implementar, cada bloque lógico se configura para realizar operaciones combinatorias y/o secuenciales, para las operaciones combinatorias se emplean tablas de búsqueda (LUT, *Look up Tables*), y para las operaciones secuenciales se emplean *Flip-Flops* del tipo D.

Los bloques lógicos también pueden ser habilitados para funcionar como memoria de almacenamiento local (memoria RAM distribuida), registros de corrimiento (SR, *Shift Register*), multiplexores y para realizar operaciones de suma y/o resta.

La estructura interna de cómo están organizados los bloques lógicos definen la arquitectura del FPGA, esta estructura cambia de un fabricante a otro. Existe una gran variedad de FPGAs provistos por varias compañías tales como Xilinx, Altera, Atmel y Lattice. Cada fabricante provee a su FPGA con una arquitectura única.

Para optimizar los recursos de los FPGAs, actualmente se incluyen en su estructura interna bloques DSP (bloques de aritmética), los cuales incluyen multiplicadores, sumadores, restadores y acumuladores, con el mismo propósito, se han agregado bloques de memoria (RAM, ROM, Flash RAM, etc.).

Los bloques de administración de reloj que se incluyen están comúnmente basados en bloques de enganche de fase (PLL, *Phase Locked Loops*) y desempeñan funciones de síntesis la frecuencia, compensa los tiempos de propagación y realiza correcciones de corrimiento de fase en la señal de reloj.

También, los FPGAs modernos incluyen bloques de comunicación que consisten generalmente de registros de transmisión y recepción; en este ámbito se soportan varios protocolos de

comunicación, entre ellos se pueden encontrar: USB, CAN, PCI, SPI, I2C, Ethernet, etc. A lo largo de su desarrollo y para facilitar en los FPGAs la alta escala de integración, el trabajo a altas velocidades y manejar bajos niveles de consumo de energía, los FPGAs han sido sujeto de considerables progresos relacionados con el proceso de fabricación de cobre con tecnología de 40 nm (recientemente Xilinx y Altera han anunciado fabricación de FPGAs con tecnología de cobre de 28 nm [75, 76]), en la Figura 3.3 se muestra la evolución del proceso de fabricación desde 1985. La estructura interna, la forma de operación y la configuración

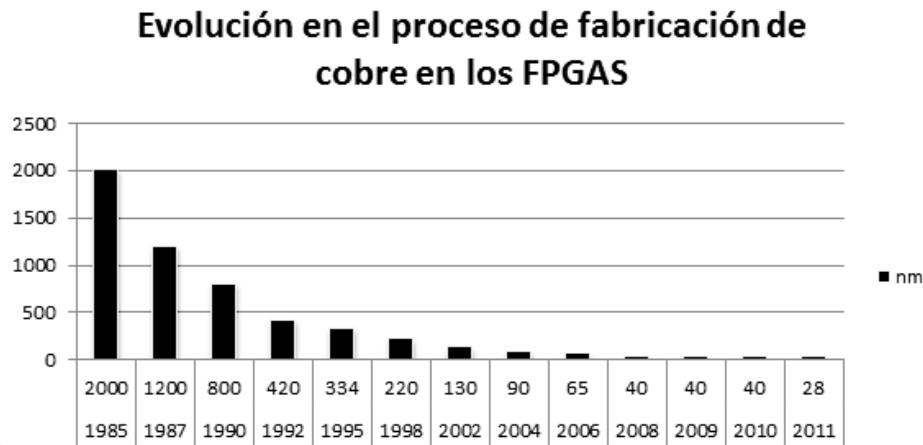


Figura 3.3: Evolución en el proceso de fabricación de la tecnología de cobre en los FPGAS.

de los elementos antes mencionados en los FPGAs son diferentes y dependen principalmente de la familia a la que pertenezca el componente, de la tecnología con la que está hecho y principalmente del vendedor.

3.4. Arquitectura y tecnologías de configuración de los FPGAs

Existen varias formas de clasificar la segmentación y nivel de especialización de los FPGA que hay en el mercado. Se pueden usar los siguientes criterios: tecnología de configuración, tecnología de fabricación y costo. El criterio de clasificación más importante es el de tecnología de configuración.

En la actualidad existen varias tecnologías de configuración de FPGAs, de las cuales se pueden mencionar: SRAM⁵ (*Static Random Access Memory*), EPROM⁶ (*Erasable Programmable Read*

⁵Memoria Estática de Acceso Aleatorio es un tipo de memoria basada en semiconductores que a diferencia de la memoria DRAM, es capaz de mantener los datos, mientras esté alimentada, sin necesidad de circuito de refresco. Sin embargo, sí son memorias volátiles, es decir que pierden la información si se les interrumpe la alimentación eléctrica.

⁶Es un tipo de chip de memoria ROM no volátil

Only Memory), EEPROM⁷ (*Electrically-Erasable Programmable Read-Only Memory*), Fusible⁸, Anti-fusible y tecnología Flash⁹.

El tipo de tecnología define si los dispositivos son reconfigurables o simplemente configurables una sola vez (OTP, One Time Programmable), las tecnologías de configuración más populares son: Anti-fusible, SRAM y Flash, debido a su popularidad, se detallarán dos FPGAs de vendedores diferentes que poseen tecnología SRAM y sólo se describirá a la tecnología Flash.

3.4.1. FPGA basado en SRAM

Esta tecnología se ha convertido en el enfoque dominante para los FPGAs, debido a sus dos ventajas principales: nueva programación y el uso de tecnología de proceso CMOS estándar. Otra ventaja de esta técnica es que las nuevas ideas de diseño pueden ser rápidamente implementadas y probadas. Existen también algunas desventajas, una de ellas es que una memoria SRAM, por ser volátil, debe ser recargada cada vez que se reinicie el sistema. La SRAM está organizada en una capa específica dentro de la estructura del **FPGA**.

Las familias más populares de FPGAs basados en SRAM son las de Xilinx y Altera [77, 78]. Los FPGAs más comercializados de ambos fabricantes y que poseen esta tecnología de programación se encuentran a las familias **VIRTEX** (alto desempeño) y **SPARTAN** (bajo costo) por el lado de Xilinx y por el lado de Altera la familia **STRATIX** (alto desempeño) y **CLONE** (bajo costo).

En el campo las aplicaciones de control industrial, el costo es muy importante, por lo que como ya se mencionó sólo se presenta la descripción general del **FPGA Xilinx SPARTAN-3** y el de Altera **Cyclone-4**.

FPGA Xilinx SPARTAN-3: Los **FPGA SPARTAN-3** de Xilinx están conformadas por un conjunto de Bloques Lógicos Configurables (CLBs, *Configurable Logic Blocks*) organizados matricialmente y rodeados por un perímetro de Bloques Programables de entrada/salida (IOBs, *Input-Output Blocks*). Estos elementos funcionales están interconectados por una matriz de canales de conexión, ésta incluye una red de baja capacitancia para la distribución de señales de reloj de alta frecuencia.

Adicionalmente este dispositivo cuenta con hasta 24 bloques de memoria RAM de 2 Kbytes de doble puerto, cuyo ancho de bus es configurable, además de poseer hasta 12 bloques de multiplicadores dedicados de 18 x 18 bits que entregan productos de 36 bits. Los buses de entrada de estos multiplicadores aceptan datos en complemento a dos (tanto de 18 bits con signo, como 17 bits sin signo). Por cada bloque de RAM hay un multiplicador contiguo conectado, por lo tanto se provee un manejo eficiente de los datos.

⁷EEPROM, es un tipo de memoria ROM que puede ser programado, borrado y reprogramado eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioletas. Son memorias no volátiles.

⁸Tecnología de programación basada en fusibles, solo se pueden programar una vez, lo que las hace poco recomendables para trabajos en laboratorio

⁹La memoria flash es una tecnología de almacenamiento -derivada de la memoria EEPROM- que permite la lectura y escritura de múltiples posiciones de memoria en la misma operación.

Los elementos están organizados como se muestra en la Figura 3.4, en esta estructura existe un anillo de IOBs que rodea un arreglo regular de CLBs. La estructura es atravesada por un arreglo en columna que contiene los bloques de memoria RAM (cada elemento de la familia posee diferente número de estos), y cada bloque está compuesto por varios bloques de 18 Kbits; cada bloque está asociado con un multiplicador dedicado.

Los administradores digitales de reloj (DCM¹⁰, *Digital Clock Manager*) están colocados en los extremos de las columnas de memoria RAM (por lo general, los miembros de la familia SPARTAN-3E poseen cuatro DCMs).

La familia **SPARTAN-3E** posee una red que interconecta los elementos arriba mencionados

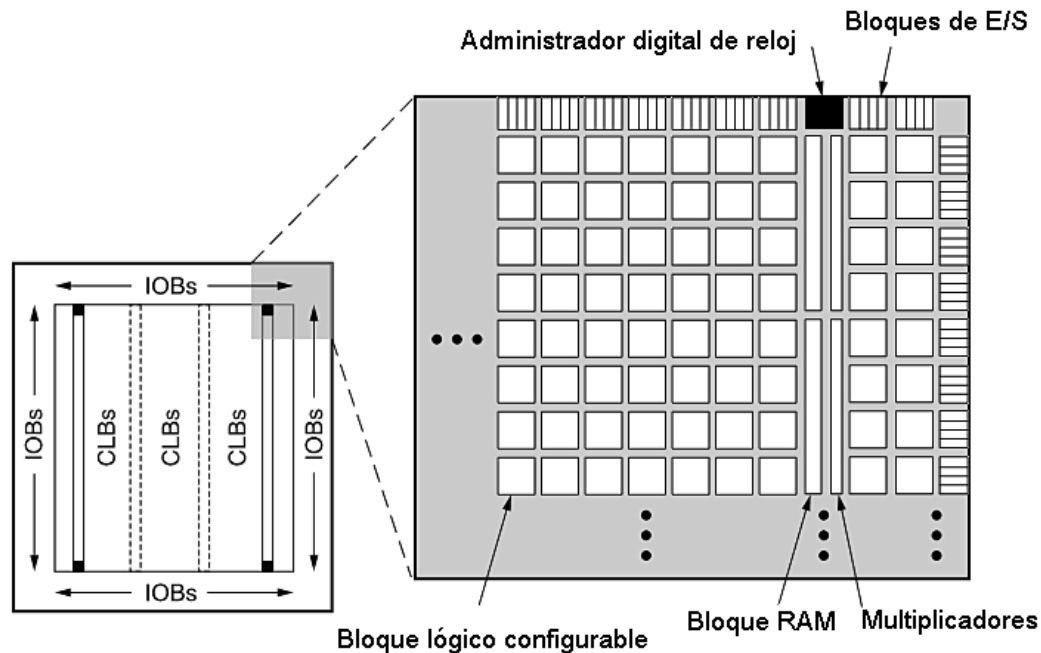


Figura 3.4: Organización interna de un FPGA de Xilinx.

de tal forma, que cada elemento funcional tiene asociada una matriz que permite múltiples conexiones y ruteo.

Los Bloques de Lógica Configurable (CLBs) constituyen el recurso lógico principal para implementar circuitos síncronos o combinacionales, un CLB está formado por cuatro SLICES, en los SLICES se realiza la mayor parte de la funcionalidad del FPGA y suelen estar agrupados de 4 en 4 formando bloques lógicos configurables (CLBs).

Los cuatro SLICES que forman un CLB tienen los siguientes elementos en común: dos generadores de funciones lógicas o tablas de búsqueda (LUT, *Look-up Table*), dos elementos de almacenamiento (registros) y multiplexores programables de función amplia cuyo número depende de la familia de FPGA que se trate, además de lógica de acarreo y compuertas

¹⁰Estos elementos están destinados a proveer una señal de reloj de elevada exactitud, eliminan los cambios de fase en la señal de reloj, así como las desviaciones de esta señal producto de perturbaciones externas, para realizar esta tarea emplean un DLL (*Delay Locked Loop*)

aritméticas, la arquitectura básica simplificada se muestra en la Figura 3.5 y en la Figura 3.6 se muestra el arreglo de SLICE en un SPARTAN-3E para un CLB de esa familia.

Los dos pares de SLICES usan estos elementos para entregar funciones lógicas y funciones

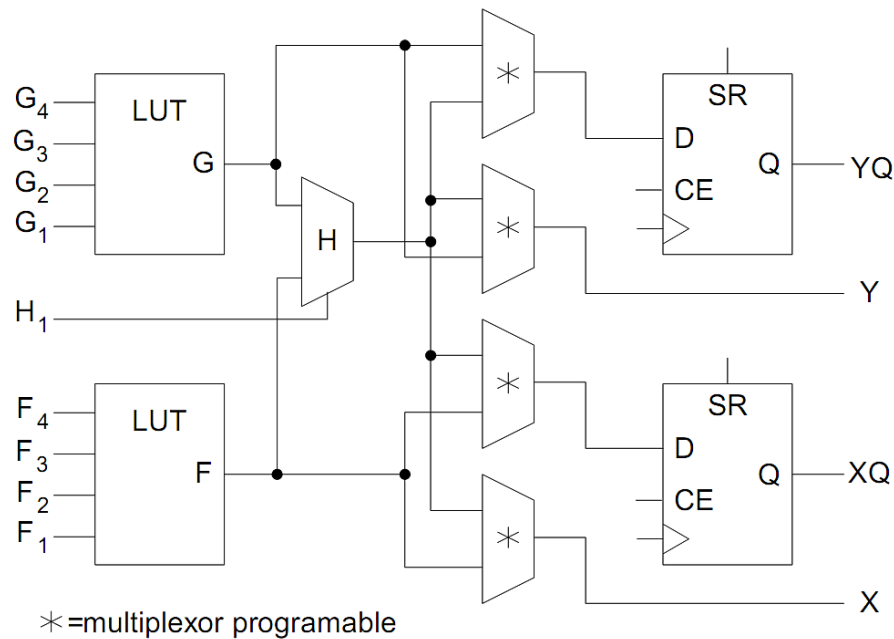


Figura 3.5: Estructura básica de un SLICE.

aritméticas. De la Figura 3.6 se puede observar que el par izquierdo soporta dos funciones adicionales: almacenamiento de datos (como RAM distribuida) y corrimiento de datos con registros de 16 bits.

Los elementos programables más importantes para implementar funciones lógicas en los FPGAs son los generadores reprogramables de funciones lógicas, esta tarea es realizada por las denominadas LUTs o tablas de búsqueda, éstas contienen celdas de memoria SRAM y multiplexores para seleccionar el resultado que se colocará en la salida.

Los generadores de función pueden diseñarse para cualquier número de variables que se desee, esto se logra aumentando el tamaño de la memoria SRAM y la ubicación de selectores que direccionan valores precalculados que son resultado de la evaluación previa de las operaciones lógicas involucradas en el diseño, una forma de representar esto, se observa en la Figura 3.7.

Sin embargo, el número de variables con que pueden diseñarse las LUTs no es una cuestión trivial. Si intentamos realizar una LUT con un número de entradas elevado, el área que ocuparía en el FPGA sería relativamente grande, por lo que el número de SLICES dentro del FPGA se reduciría. Si, por el contrario, se opta por LUTs con pocas entradas, se emplearían pocos SLICES, pero el proceso de ruteo sería complicado y extenso, y se necesitarían muchas conexiones, por lo que el retardo en la red de interconexión programable debido al cableado entre los SLICES sería importante, esta cuestiones resuelta por la herramienta de diseño de

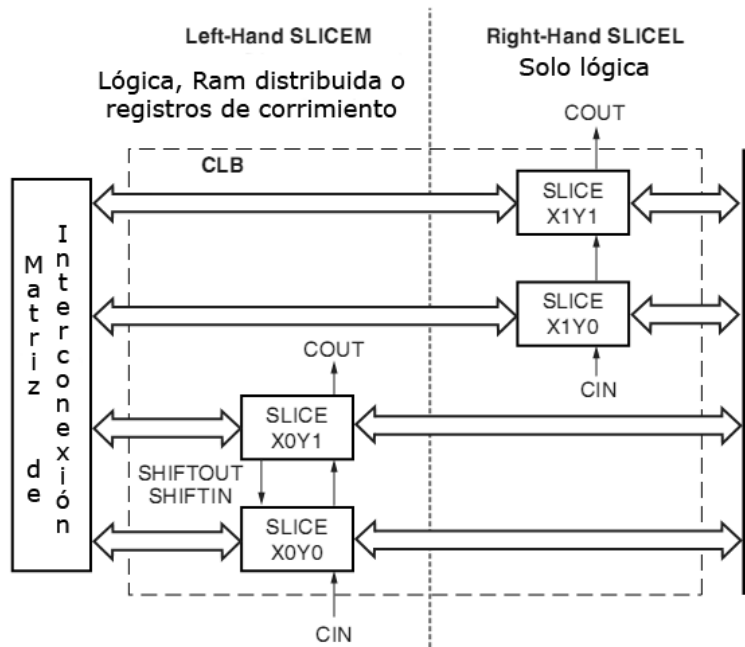


Figura 3.6: Arreglo de SLICES en un CLB de la familia SPARTAN-3E.

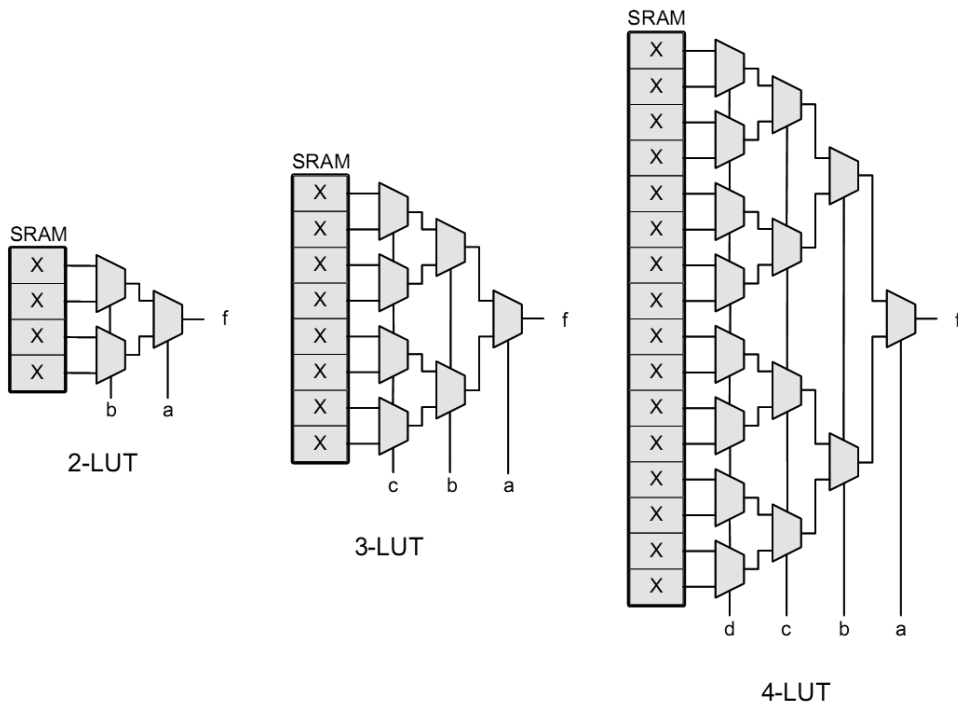


Figura 3.7: Arreglos básicos de LUTs.

alto nivel que se emplea.

Cada CLB del FPGA se encuentra incrustado en la estructura de interconexión, la estructura de interconexión se compone de líneas con conexiones programables, por lo que muchas veces se le llama estructura de conexión programable. En la familia SPARTAN-3E existen cuatro tipos de interconexiones: Líneas largas, líneas hex, líneas dobles y líneas directas.

- **Líneas largas:** Son aquellas que conectan una salida de cada 6 CLBs (ver Figura 3.8(a)), estas líneas poseen baja capacitancia y son capaces de transportar señales de alta frecuencia con mínimos efectos de desfaseamiento, son una buena opción para transportar señales de reloj cuando las ocho líneas dedicadas a este propósito ya fueron empleadas.
- **Líneas HEX:** Conectan una salida de cada tres CLBs, ofrecen mayor conectividad que las líneas largas, pero menos capacidad para transportar señales de alta frecuencia. Ver Figura 3.8(b).
- **Líneas dobles:** Conectan a los CLBs que aún no son conectados por las líneas largas y las líneas Hex, por lo que hacen las conexiones más flexibles. Ver Figura 3.8(c).
- **Líneas directas:** Conecta cada CLB hacia cada uno de sus ocho vecinos. Estas líneas son usadas a menudo para conducir una señal proveniente de un CLB de origen hacia una línea doble, una línea Hex o a una línea larga hasta hacer llegar la señal a su destino. Ver Figura 3.8(d).

Inicialmente los FPGAs disponían de interconexiones heterogéneas de propósito general, aunque en la década pasada se evolucionó hacia una estructura de interconexión jerárquica. Las líneas orientadas al transporte de la señal de reloj están optimizadas para su uso como entradas de reloj a los CLBs, proporcionando un retardo corto.

La familia SPARTAN-3E contiene memoria RAM embebida, ésta se organiza en bloques de 18 Kbits de memoria y es conocida como Bloques de RAM (BRAM); el ancho de bus de datos y el bus de direcciones de cada BRAM es configurable, permitiendo manejar diferentes organizaciones, además la BRAM pueden ser conectada en cascada para crear memorias con palabras más anchas (mayor número de bits) o de mayor profundidad (mayor número de localidades) según según sea el caso.

Los bloques de RAM tienen una estructura de doble puerto (ver Figura 3.9). Estos puertos llamados A y B permiten el acceso independiente al mismo rango de memoria, éste tiene una capacidad máxima de 18432 bits o 16384 bits cuando no se usan las líneas de paridad.

Cada puerto tiene su propio conjunto de líneas de control, de datos y de reloj para las operaciones síncronas de lectura y escritura, estas operaciones tienen lugar de manera totalmente independiente en cada uno de los puertos.

Los FPGAs de la familia SPARTAN-3E poseen una aceptación muy importante en el mercado (un ejemplo de tarjeta de desarrollo con este dispositivo es el mostrado en la Figura 3.10),

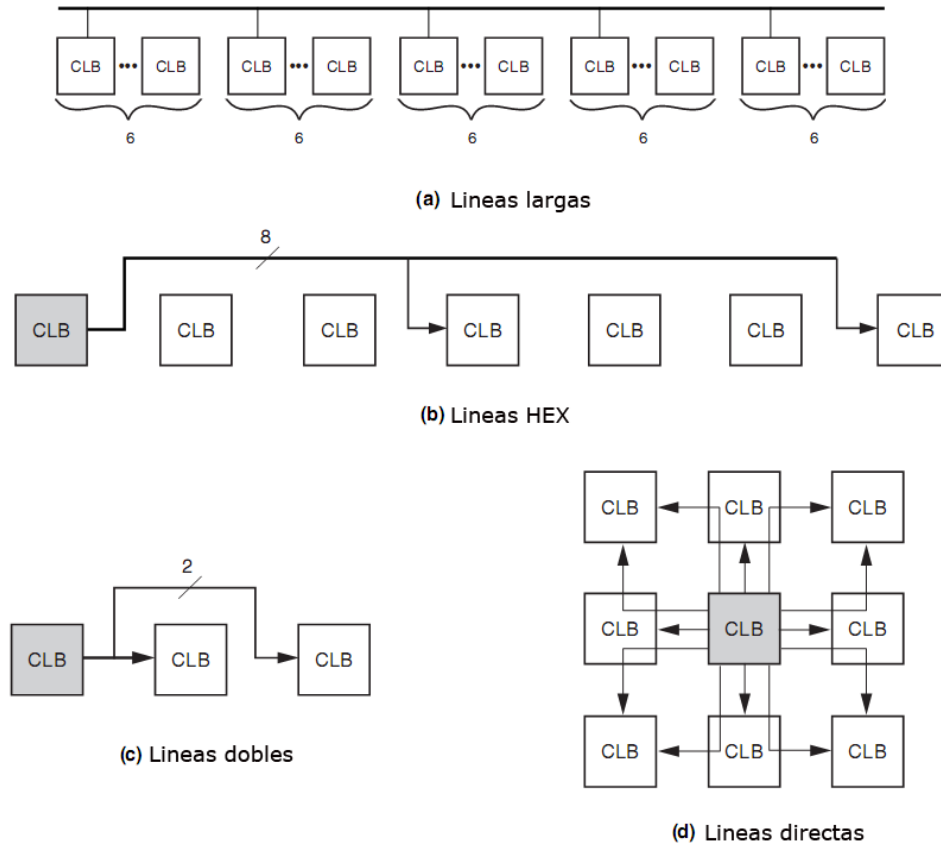


Figura 3.8: Tipos de interconexiones entre CLBs en la Familia SPARTAN-3.

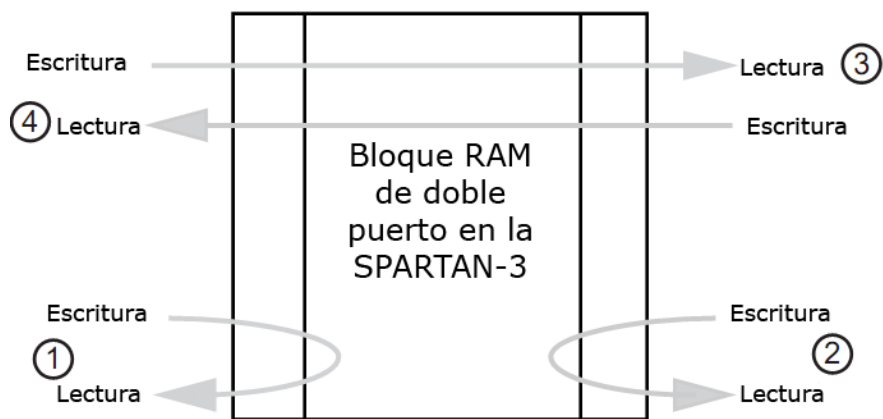


Figura 3.9: Diagrama de un bloque dedicado de memoria RAM en la familia SPARTAN-3E.

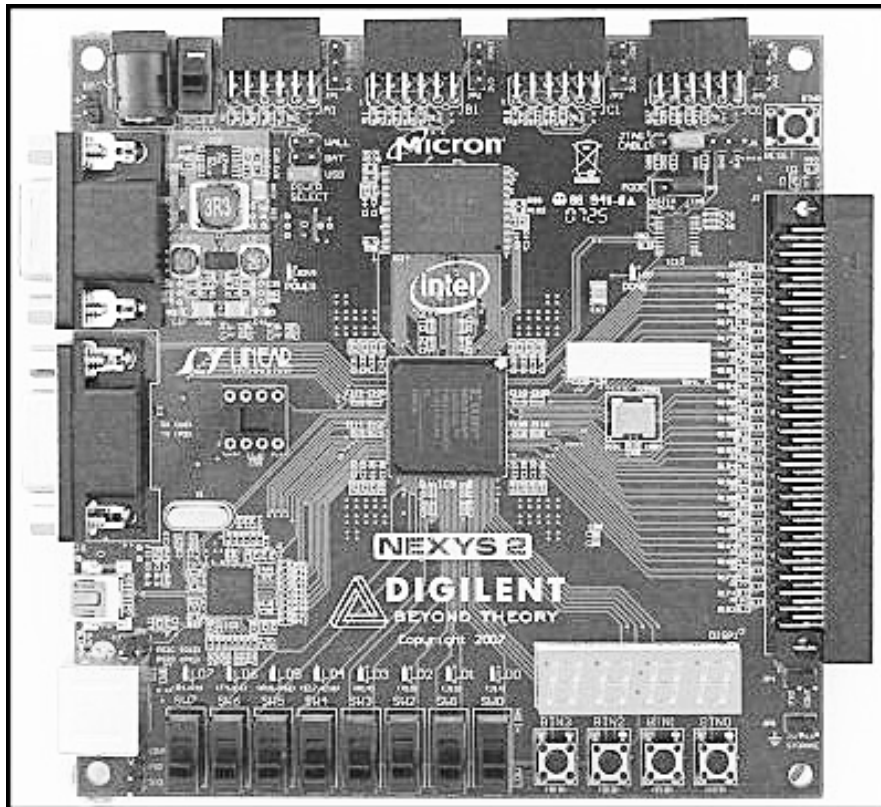


Figura 3.10: Tarjeta de desarrollo NEXYS2 con un **FPGA** de la familia SPARTAN-3E.

existe otro FPGA de Xilinx que es de gama alta cuya configuración se realiza por medio de SRAM y pertenece a la familia **SPARTAN-6**, éste posee mayor número de recursos incrustados, las características máximas de esta familia se pueden observar en la tabla 3.1:

FPGA Altera CYCLONE-4: Una tarjeta de desarrollo con este FPGA se muestra en la

Tabla 3.1: Familia SPARTAN-6

Características máximas de la familia SPARTAN-6	
Numero de CLBS	11519
Entradas y salidas	576
Bloques DSP dedicados	180
Rango de frecuencia de reloj	400-1080 MHz
Tamaño de memoria (RAM, ROM)	6179 Kb
Tecnología de fabricación	45 nm

Figura 3.11 y la estructura interna de los FPGAs de esta familia se presenta en la Figura 3.12. En este tipo de FPGA las celdas lógicas son nombradas como elementos lógicos (LE, *Logic Elements*), los LE pueden trabajar en dos modos: Modo normal y en modo aritmético, el modo normal comúnmente se emplea para implementar funciones lógicas que son combinacionales, el modo aritmético, básicamente se emplea para implementar sumadores, contadores, acumuladores y comparadores, los cuales trabajan en modo secuencial.

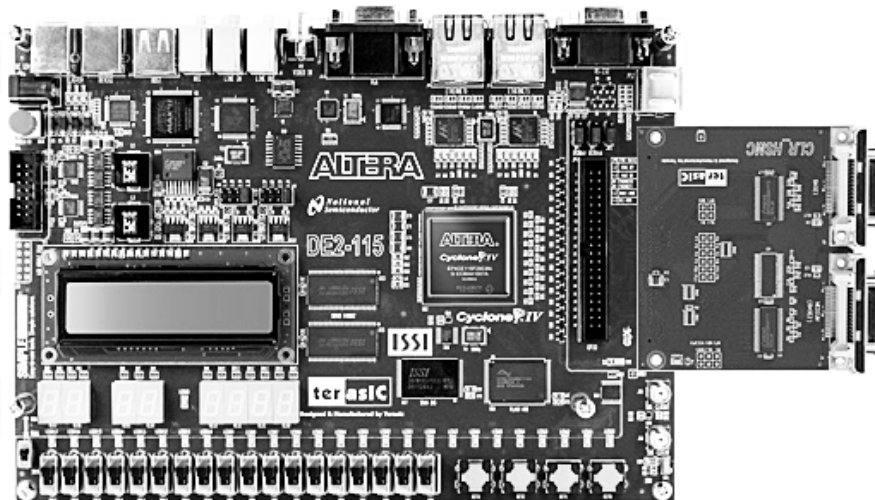


Figura 3.11: Tarjeta de desarrollo con un **FPGA** de Altera CICLONE IV (National Semiconductor).

En términos generales, cada LE de la compañía Altera es equivalente a una celda lógica de Xilinx. Un elemento lógico consiste de LUTs de 4 bits en las que se pueden implementar

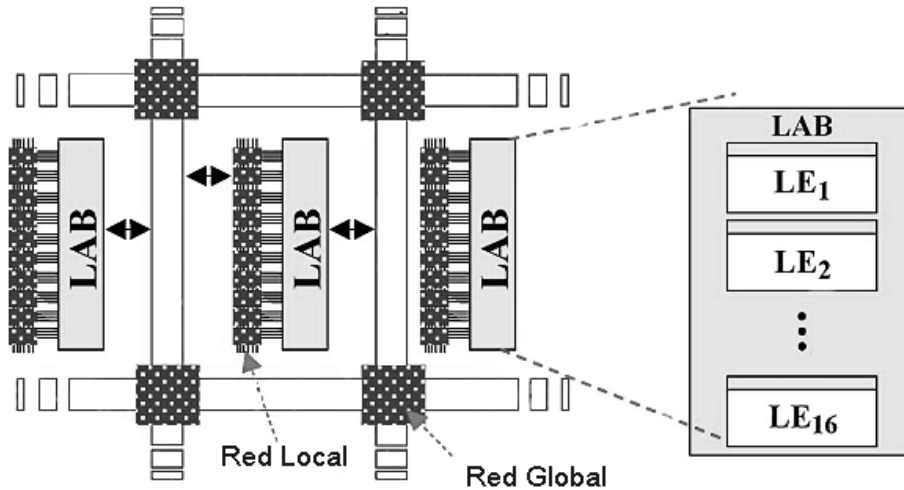


Figura 3.12: Estructura interna del **FPGA** de la familia CYCLONE IV.

operaciones dependiendo del modo en el que se trabaje.

Los elementos lógicos están arreglados en 16 grupos llamados bloques de arreglos lógicos y reciben el nombre de LAB (*LAB, Logic Array Block*). Para formar estos bloques, es necesario que los LE estén comunicados entre sí, dicha comunicación se realiza por medio de una red de interconexión, la cual efectúa la transferencia de señales entre LE del mismo LAB (ver Figura 3.13). Los bloques de interconexión están organizados en dos niveles: red de interconexión local y red de interconexión global [79].

La tabla 3.2 muestra algunas de las características de tres FPGAs basados en SRAM

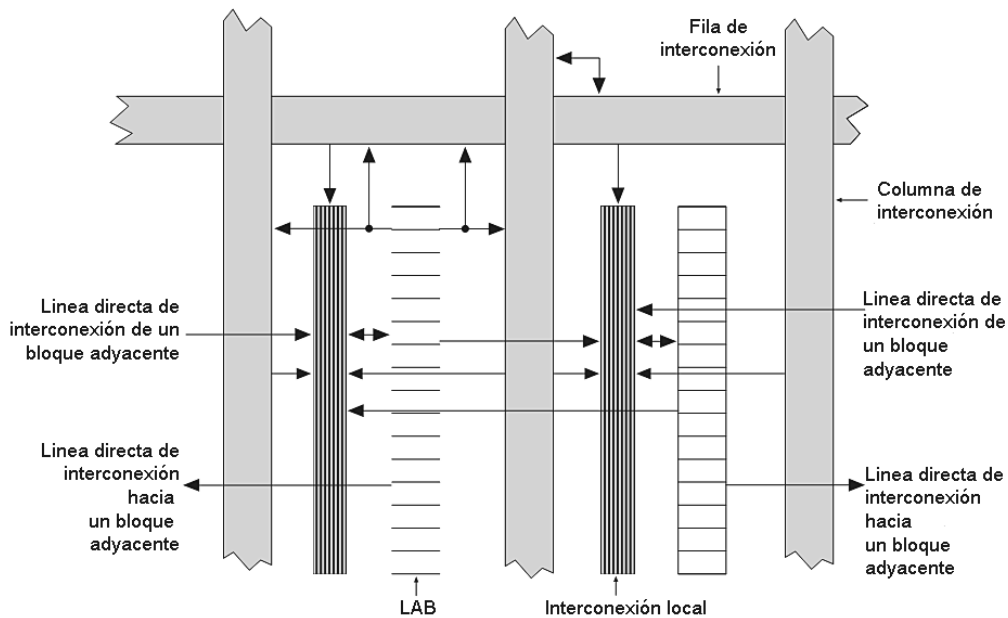


Figura 3.13: Estructura de un LAB en un **FPGA** de la familia CYCLONE IV.

(SPARTAN-3, SPARTAN-6, CICLONE-4), donde se indican los recursos máximos disponibles de cada familia.

Tabla 3.2: Características Máximas

Características máximas de algunos FPGAs basados en SRAM			
Elementos	SPARTAN-3	SPARTAN-6	CYCLONE-4
Número de celdas lógicas	74880	147443	150000
Número de entradas y salidas	633	576	532
Bloques DSP	--	180	360
Rango de frecuencia (MHz)	400 – 1080	400 – 1080	600 – 1300
Tamaño de memoria (RAM, ROM) (Kb)	1872	6179	6480
Número de Mult. dedicados	104	--	--
Número de DCM o PLL	4	12	8
Tecnología de fabricación (nm)	90	45	60

3.4.2. FPGA basado en Memoria Flash

Para contrarrestar las deficiencias de la tecnología SRAM, se usan las tecnologías de programación de puerta flotante, cada conexión contiene dos transistores que comparten una compuerta y así almacenan la información de la programación [74].

Este enfoque es utilizado en celdas de memorias Flash o EEPROM, estas celdas no son volátiles, es decir, no pierden la configuración cuando el dispositivo se apaga, por este motivo, esta tecnología es muy usada en aplicaciones de la rama aeroespacial, ya que garantiza no perder su configuración ante los efectos de la radiación [69].

Históricamente, las celdas de memoria EEPROM no se usaban directamente para cambiar las señales del FPGA, estas células eran de uso general para implementar funciones *AND* cableadas en dispositivos PLD. Actualmente las celdas de memoria Flash, son muy empleadas debido a la eficiencia en área y al bajo consumo de potencia.

La tecnología de programación basada en memoria Flash ofrece varias ventajas, una de las más importantes y que ya fue mencionada es la de ser no-volátil. Esta característica elimina la

necesidad de recursos externos necesarios para almacenar y configurar datos a diferencia de la tecnología de programación basada en SRAM. Además, un dispositivo basado en flash puede funcionar inmediatamente después del encendido sin tener que esperar la carga de datos. La tecnología flash también es más eficiente en cuanto al tamaño, ya que no necesita el mismo número de transistores que la tecnología basada en SRAM.

Si se compara a los FPGAs basados en la tecnología de anti-fusibles y los basados en tecnología flash, ambas son alternativas no volátiles, pero los FPGAs basados en memoria flash son reconfigurables y se pueden programar sin ser removidos de la placa del circuito impreso, características que no posee la tecnología de anti-fusibles.

Actualmente, la tecnología flash permite mediante impulsos eléctricos velocidades de funcionamiento muy superiores frente a la tecnología EEPROM, que sólo permitía actuar sobre una única celda de memoria en cada operación de programación.

Una desventaja de estos dispositivos es que no pueden ser reprogramados infinitamente.

Otra desventaja importante de dispositivos flash es la necesidad de un proceso CMOS no estándar.

Como ejemplos de FPGAs con este tipo de tecnología se puede citar a la familia **Igloo** y **ProASIC** de Actel, por parte, del fabricante Xilinx a la familia **Spartan-3AN**.

Para terminar esta comparativa, los FPGAs basados en SRAM poseen muy altas densidades pero consumen mucha energía, además de que necesitan una memoria externa no volátil para almacenar su configuración; los FPGAs basados en memoria flash no necesitan una memoria para almacenar su configuración, además de que consumen menos energía que la tecnología SRAM, pero su precio aun es elevado y éste aumenta al incrementar la densidad, por lo tanto, la elección de una u otra tecnología lo decidirá la aplicación a desarrollarse.

En el apéndice [A](#) se muestra un breve resumen sobre la tendencia de desarrollo de los recursos de la tecnología FPGA.

Parte II

Metodología empleada

Fase I

Capítulo 4

Diseño del controlador por la técnica GPI para el inversor multinivel en cascada

Resumen: En este capítulo se describe la metodología que fue empleada para implementar el controlador en cuestión en el dispositivo de lógica programable elegido; también se realiza la descripción de las primeras tres fases de diseño que tienen que ver con el modelado del sistema dinámico, el análisis de sus propiedades matemáticas y la simulación del controlador obtenido.

4.1. Introducción

Hoy en día las aplicaciones de la electrónica de potencia son muy sofisticadas, en este contexto, la tecnología digital es de gran interés por que permite la implementación de complejas estrategias de control, con tiempos de ejecución relativamente pequeños [68].

Por otro lado, los controladores analógicos, a pesar de sus inconvenientes tales como la desviación en sus parámetros y la complejidad en su integración permanecen como marco de referencia en términos de rapidez y ancho de banda, por ese motivo, los tiempos de ejecución de los controladores digitales deben ser reducidos, pero manteniendo la flexibilidad inherente que ofrecen los diseños basados en soluciones digitales, esto puede lograrse con la ayuda de plataformas digitales eficientes.

Una posibilidad en las plataformas de diseño digital son los llamados Arreglos de Compuertas Lógicas programables en campo (FPGAs), estos dispositivos toman el beneficio por su alta escala de integración y además porque el diseñador puede ser capaz de construir una arquitectura totalmente dedicada y perfectamente adaptada al algoritmo a implementar.

Como ya se mencionó, usando controladores basados en FPGAs se puede lograr la reducción en el tiempo de ejecución, sin embargo esto es posible siempre y cuando se elija, y se siga rigurosa y eficientemente una metodología para diseñar el controlador. Una metodología con-

siste en un conjunto de pasos y reglas que se siguen en orden durante el proceso de diseño, de tal forma que sea más manejable e intuitivo [68].

En este apartado se describirá la metodología que se utilizó para lograr la implementación final del controlador GPI de seguimiento de voltaje para el inversor multinivel de celdas en cascada de cinco niveles, este proceso se dividió en dos fases, en la primera de estas (la que se desarrolla en este capítulo) se realiza una especificación preliminar del sistema en la cual se incluye el modelado matemático del sistema a controlar, el análisis de las propiedades matemáticas del sistema para verificar si es posible aplicar la técnica de control elegida, en la última parte de esta fase se diseña el algoritmo del controlador del sistema y se realizan algunas simulaciones del desempeño del mismo. Posteriormente, en la segunda fase de esta metodología (próximo capítulo) se trata el desarrollo de la arquitectura basada en FPGA en la que se describen las tareas realizadas para la implementación en el dispositivo en cuestión.

4.2. Descripción de la metodología

En el diseño digital existen plataformas que ya integran funciones de conversión digital-analógica que se pueden combinar con herramientas software para lograr diseños de alto desempeño. Los diseños basados en los dispositivos tales como los DSPs y microcontroladores se considera ahora tecnología madura [86]; algunos DSPs aun con arquitecturas orientadas al control de sistemas eléctricos tienen limitaciones en términos de ancho de banda y en la no concurrencia de las tareas que pueden realizar.

Dependiendo del caso, la arquitectura resultante puede ser toda en Software (Sw) usando los procesadores integrados dentro del FPGA, toda en Hardware (Hw) usando celdas lógicas del FPGA o una combinación de ambas, es decir, una arquitectura mixta software/hardware (Sw/Hw) [68]. Esta personalización puede reducir significativamente el tiempo de ejecución a unos pocos microsegundos, logrando incluso tiempos de respuesta similares a su contra parte analógica.

En ese sentido, varios autores han propuesto y formalizado algunas metodologías de diseño, en todas se encuentran objetivos similares (ver [87, 88, 89, 90]).

La metodología para el desarrollo de este trabajo consta de diez etapas, las cuales se dividen en dos fases:

Fase 1: Esta fase es muy importante, en ella se analiza la viabilidad de la aplicación de la técnica de control elegida, en esta fase se incluyen las tres primeras etapas relacionadas con el modelado, análisis y simulación del inversor monofásico multinivel, estas etapas son independientes del dispositivo a utilizar y se describen ligeramente a continuación:

- [1] **Modelado:** En esta etapa se modela matemáticamente el inversor monofásico de cinco niveles empleando las leyes de Kirchhoff para la obtención de las ecuaciones diferenciales de primer orden en forma continua del sistema.

- **[2] Análisis:** Dando seguimiento a la etapa anterior se analiza el modelo dinámico por medio de algunas herramientas matemáticas tales como el álgebra lineal y de teoría de control lineal (ver Capítulo 2). Esta etapa es muy importante para el desarrollo del proyecto, ya que determina la factibilidad del diseño promedio del controlador basado en la técnica propuesta.
- **[3] Simulación:** En esta fase se simula el modelo matemático obtenido en forma continua en el tiempo, para realizar esta tarea se puede utilizar la pieza de software **SIMNON**. En esta fase se confirma el nivel de aproximación del modelo obtenido del sistema a través de la comparativa a nivel de simulación de las ecuaciones diferenciales que describen el comportamiento dinámico del sistema y el circuito eléctrico realístico del mismo a través de la co-simulación del software **Matlab/Simulink** y el software **PSIM**. Después de esta etapa se obtiene el modelo dinámico del inversor monofásico multinivel de cinco niveles.

De manera paralela, durante esta fase se dimensiona y se construye la plataforma hardware (inversor multinivel en cascada de cinco niveles).

Fase 2: En esta fase la metodología ofrece una solución balanceada entre dos necesidades opuestas:

- Un método que se adapte amigablemente a ingenieros de control que no son expertos en electrónica digital.
- Buen desempeño del controlador.

Los principales pasos de esta fase se muestran en la Figura 4.1 y se describen a continuación (ver [88, 90, 91]):

- **[4] Partición modular:** Este paso es crucial en la metodología y consiste en dividir el algoritmo de control en partes más pequeñas que serán llamados módulos, los cuales deben de ser fáciles de desarrollar y probar. En esta etapa se deben de identificar y extraer acorde a los conceptos de jerarquía y regularidad los módulos que sean independientes y los que puedan ser reusados (moduladores, estimadores, operaciones con vectores, etc.). La jerarquización es usada para dividir un diseño grande o complejo en partes más pequeñas que sean más manejables; la regularidad tiene que ver con maximizar el empleo de módulos que se puedan usar nuevamente.
- **[5] Simulación funcional:** En esta etapa se implementa el modelo funcional del controlador y se valida el funcionamiento con bloques en tiempo continuo empleando la herramienta MATLAB-Simulink.
- **[6] Rediseño digital:** Este paso es muy importante, en éste se incluye la realización del filtro digital, la elección del periodo de muestreo, la elección del formato de coeficientes

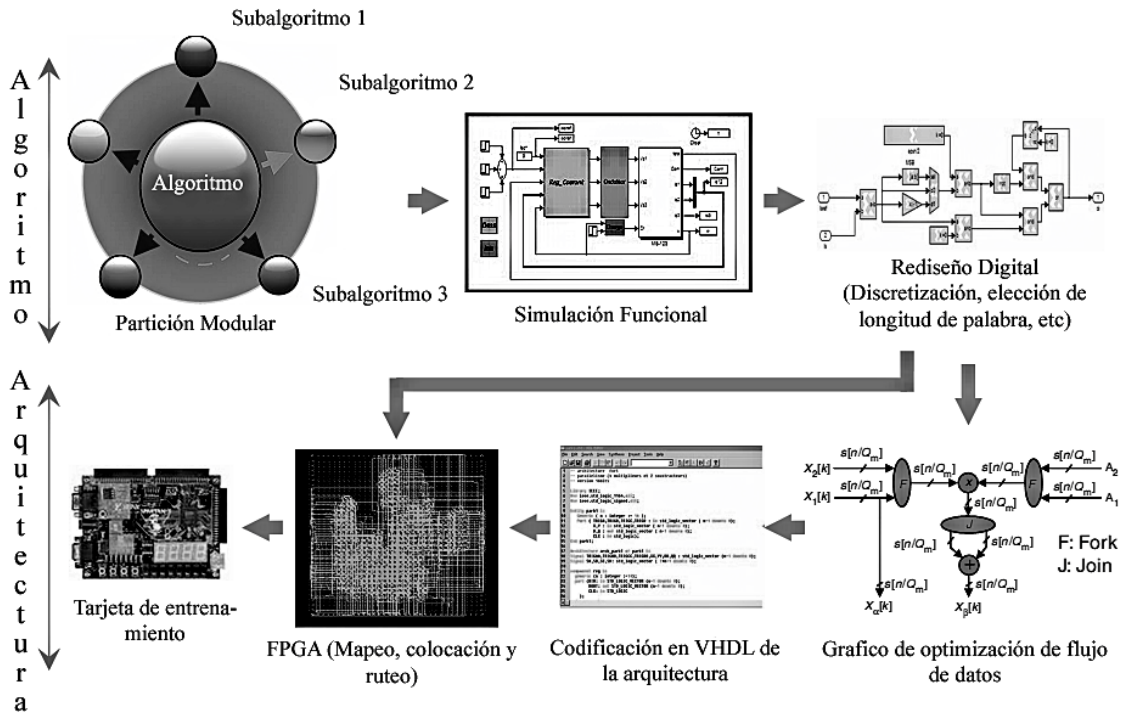


Figura 4.1: Metodología de diseño([91]).

y variables acorde a las necesidades y restricciones del control tales como: longitud de palabra, punto fijo o punto flotante, etcétera.

Las cuatro etapas finales están ligadas al tipo de dispositivo a emplearse, estas son:

- [7] **Gráfico de optimización del flujo de datos:** En esta etapa se modela el flujo de datos del algoritmo para tener el mejor balance en términos de tiempo/área, es decir se debe obtener una representación gráfica de cada módulo del algoritmo. La representación gráfica está compuesta de líneas y nodos, cada línea corresponde a una transferencia de datos y los nodos representan operaciones aritméticas o funciones lógicas o matemáticas.
- [8] **Codificación HDL:** Consiste en la transcripción del gráfico de flujo de datos de manera casi directa a código VHDL, es decir se obtiene de reemplazar cada nodo del gráfico con su correspondiente operador y cada línea con un bus de datos entre operadores. La transferencia de datos puede ser controlada por una unidad de control, que puede ser una máquina de estados finitos que esta sincronizada por la señal de reloj.
- [9] **Implementación en FPGA:** Este es un proceso automático y es realizado por la herramienta EDA que se utilice, consiste en el mapeo de las funciones resultantes, colocación y ruteo.

- [10] **Validación experimental:** Consiste en ajustar el modelo obtenido auxiliándose del procedimiento HIL¹ (*Hardware in the Loop*).

4.3. Modelado Matemático

En general, para implementar sistemas automatizados complejos, flexibles y altamente precisos, se deben construir de manera paralela sistemas de prueba caros y además complejos. Esta complejidad hace que la comprobación del sistema involucrado sea difícil de realizar (o algunas veces imposible), y generalmente obstruye la comprensión global del sistema y la previsión de su comportamiento, cuestiones que son indispensables para minimizar los riesgos de fallo en los sistemas desarrollados.

El modelo matemático de un sistema físico es una representación del sistema por medio de un conjunto de ecuaciones matemáticas, de manera formal « El modelado matemático es una técnica del análisis y diseño de sistemas usando idealizaciones físicas o matemáticas de todas las porciones del sistema. De la integridad del modelo y su apego al sistema físico real, así como del estado de conocimiento del sistema y su ambiente, dependerán las preguntas que puedan ser contestadas con dicho modelo» [92].

Por otra parte, para lograr una aproximación efectiva en los modelos de los sistemas, actualmente está difundido el uso de metodologías y herramientas de simulación.

Las ventajas de la simulación son múltiples: puede reducir el tiempo de desarrollo del sistema, las decisiones pueden comprobarse artificialmente, un mismo modelo puede usarse muchas veces, etc. La simulación posee un empleo más simple que ciertas técnicas analíticas y precisa menos simplificaciones.

La determinación de los modelos y la simulación que describe las relaciones entre la entrada y la salida, son una tarea fundamental en el estudio y proceso del diseño de los convertidores multinivel. Estas relaciones de entrada-salida se vuelven indispensables para el desarrollo de modelos adecuados, los cuales deben permitir obtener toda la información necesaria sobre el convertidor antes de entrar a la etapa de construcción [1].

El modelado de los convertidores multinivel no es una tarea trivial, y su complejidad dependerá del tipo de componentes (lineales o no lineales) que estén involucrados en el sistema.

Históricamente, las técnicas de modelado aplicadas a los convertidores electrónicos de CD han sido aplicadas al estudio de convertidores de CA, logrando con esto diferentes aproximaciones, estas aproximaciones deben elegirse acorde a los objetivos de diseño, por lo tanto regularmente se auxilian de herramientas tales como esquemas de control, desarrollo de controladores, estudios de estado estacionario, estudios de dinámica y transitorios, análisis de estabilidad, etc. [1].

Tomando en cuenta estos comentarios, se pueden desarrollar dos tipos de modelos:

¹Durante este desarrollo, este procedimiento fue empleado para sintonizar el controlador y acoplarlo al sistema físico del convertidor, el procedimiento consiste en un proceso iterativo de sintonía auxiliado de una interfaz USB.

- Simulación del circuito equivalente.
- Modelo promedio en el espacio de estados.

4.3.1. Simulación del circuito equivalente

El modelo del convertidor multinivel puede ser obtenido con la ayuda de algunas herramientas de simulación basadas en SPICE (SPICE², *Simulation Program with Integrated Circuits Emphasis*).

En este caso, el modelo del convertidor multinivel se reduce a la generación de un modelo de circuito eléctrico adecuado que integre plenamente las no linealidades de los dispositivos conmutadores, permitiendo con esto, una completa caracterización de la dinámica del sistema. Si se consideran interruptores ideales, se puede obtener una descripción lineal del convertidor que está basada en los estados de conmutación del convertidor de potencia implicado, es decir el modelo describe los estados discretos posibles del convertidor de potencia. Con este tipo de modelos se puede obtener una buena aproximación y los resultados pueden ser suficientemente buenos en la simulación, siempre y cuando los métodos de integración de las ecuaciones del modelo sean elegidas apropiadamente ya que en tiempo de simulación muchas veces se generan problemas de convergencia [93].

La principal desventaja de este tipo de técnica de modelado es cuando se busca combinar con técnicas avanzadas de control, por lo que muchas veces resulta imposible dicha integración [93], esto se debe a que el modelo resultante usualmente es muy complejo y ofrece problemas durante el análisis en el diseño del controlador [94, 95].

4.3.2. Modelo promedio en el espacio de estados

Los modelos promedio en el espacio de estados pueden ser fácilmente obtenidos por medio de un modelo discreto donde las cantidades que varían con el tiempo se asumen sólo como un valor promedio sobre un periodo de conmutación [1].

Puesto que en los convertidores estas cantidades son variantes en el tiempo, incluso en estado de equilibrio, por lo general es necesario hacer un cambio de coordenadas para convertir cantidades sinusoidales a valores en CD antes del proceso de promediado [96, 97].

Recientemente, una técnica de modelado promediado en el espacio de estados ha sido propuesta, esta técnica se basa en la aproximación sobre un segmento promedio conocido del convertidor (ver [98, 99]), es decir se considera que los dispositivos de conmutación trabajan de manera ideal entre dos o más estados posibles, por lo tanto, el modelo promedio se puede reducir considerablemente empleando sólo el valor promedio (δ_a) de la posición de conmutación. Esta técnica ofrece modelos sencillos para ser utilizados en el diseño de controladores

²Es un estándar internacional cuyo objetivo es simular circuitos electrónicos analógicos y digitales, fue desarrollado por la Universidad de California, Berkeley en 1975 por Donald Pederson

[100], además la simulación de los sistemas modelados de esta manera es más rápida y se resuelve el problema de convergencia gracias a la naturaleza continua de las ecuaciones del modelo obtenido.

Por lo tanto, el uso de esta técnica de modelado supera una de las desventajas tecnológicas en la que este tipo de convertidores se ven involucrados, haciendo la etapa de diseño de los convertidores multinivel una tarea más accesible, en este trabajo se empleará esta técnica de modelado.

4.3.3. Modelo matemático del inversor de tres niveles

Con el fin de simplificar el análisis que concierne a este trabajo, se partirá sólo del modelado para un inversor de tres niveles, y posteriormente se generalizará al caso multinivel con dos celdas.

En el caso del inversor de tres niveles se tiene únicamente un puente H, y está formado por cuatro interruptores totalmente controlados, típicamente transistores MOSFET o IGBTs tal como se muestra en la Figura 4.2. Cada uno de los posibles circuitos que resultan cuando se

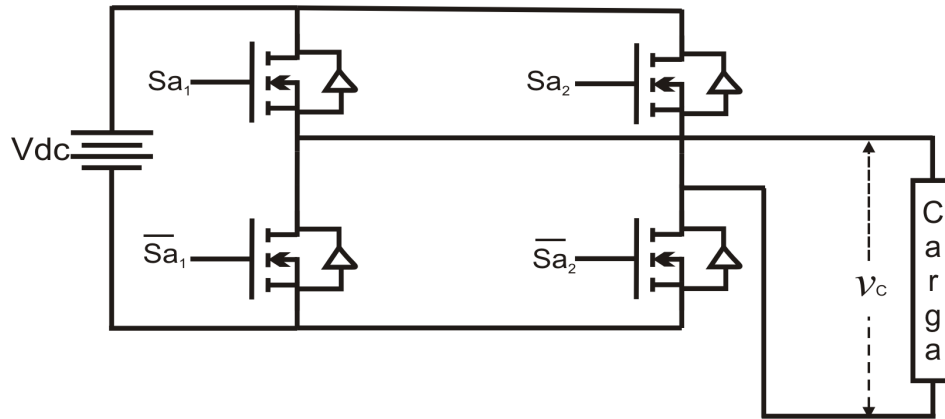


Figura 4.2: Inversor de tres niveles.

conmutan los interruptores $S_{a_{1,2}}$ se muestran en la Figura 4.3. Dado que el inversor está controlado por dos señales binarias, resultan $2^2 = 4$ estados diferentes de conmutación para los dispositivos mismos que se ven resumidos en la tabla 4.1.

Por ejemplo, si se considera el estado de conmutación $(1, 0)$ que se muestra en la Figura 4.3 (A) y en la primera fila de la tabla 4.1, la salida de la rama S_{a_1} es conectada al positivo de la fuente, mientras que la salida de la rama $\overline{S_{a_2}}$ es conectada al negativo de la fuente, generando un voltaje de salida $v_C = V_{cd}$.

Una expresión general para la salida del voltaje es:

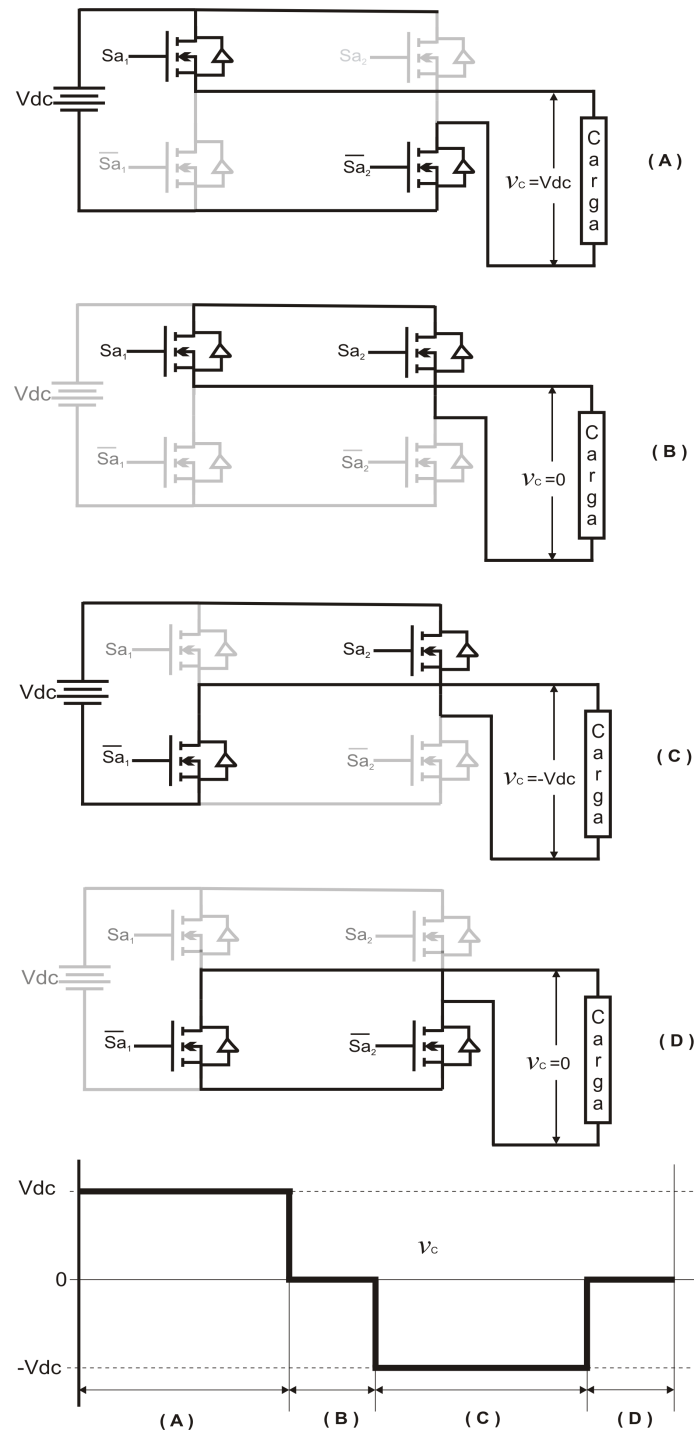


Figura 4.3: Estados de conducción del inversor puente completo: **(A)** $v_C = V_{dc}$; **(B)** $v_C = 0$; **(C)** $v_C = -V_{dc}$ y **(D)** $v_C = 0$.

Tabla 4.1: Estados de conmutación para el inversor de tres niveles

Estados de conmutación (ver Figura 4.3)			
Estado	Sa_1	Sa_2	v_C
(A)	1	0	Vcd
(B)	1	1	0
(C)	0	1	$-Vcd$
(D)	0	0	0

$$v_C = (Sa_1 - Sa_2)Vcd \quad (4.1)$$

donde $Sa_{1,2}$ pertenecen al conjunto discreto:

$$Sa_{1,2} \in \{0, 1\}$$

Reemplazando en la ecuación (4.1) las diferentes combinaciones binarias de las señales de compuerta, es sencillo obtener los diferentes niveles de voltaje listados en la tabla 4.1 y en la Figura 4.3.

De la tabla 4.1 se puede notar que en el caso de las combinaciones (0,0) y (1,1) de las señales de compuerta se tiene un voltaje $v_C = 0$, esta característica es llamada nivel de voltaje de redundancia y puede ser usado para otros propósitos de control ya que éste no afecta el nivel de voltaje generado en la carga. Por lo tanto, existen tres valores diferentes en la salida de voltaje ($Vcd, 0, -Vcd$), razón por la cual se le nombra inversor de tres niveles y en alguna bibliografía ha sido considerado como un inversor multinivel [101].

Ahora de manera adicional, en el modelado se considera que a la salida de este circuito se cuenta con un filtro de segundo orden del tipo LC y que la carga del convertidor es de tipo resistiva R (ver Figura 4.4).

Por lo tanto, para obtener las ecuaciones diferenciales que describen a este convertidor se aplicarán de manera directa las leyes de *Kirchhoff* de voltaje y corriente (**LVK** y **LCK** respectivamente) a cada uno de los posibles circuitos que resultan cuando se conmutan los interruptores $Sa_{1,2}$, en la misma forma que en el caso anterior, pero ahora sin tomar en cuenta los estados de conmutación **(B)** y **(D)** ya que en estos no existe contribución de voltaje.

Para el caso **(A)** en el que $sa_1 = 1$ y $sa_2 = 0$

$$L \frac{di_L}{dt} = -v_C + Vcd \quad (4.2)$$

$$C \frac{dv_C}{dt} = i_L - \frac{v_C}{R} \quad (4.3)$$

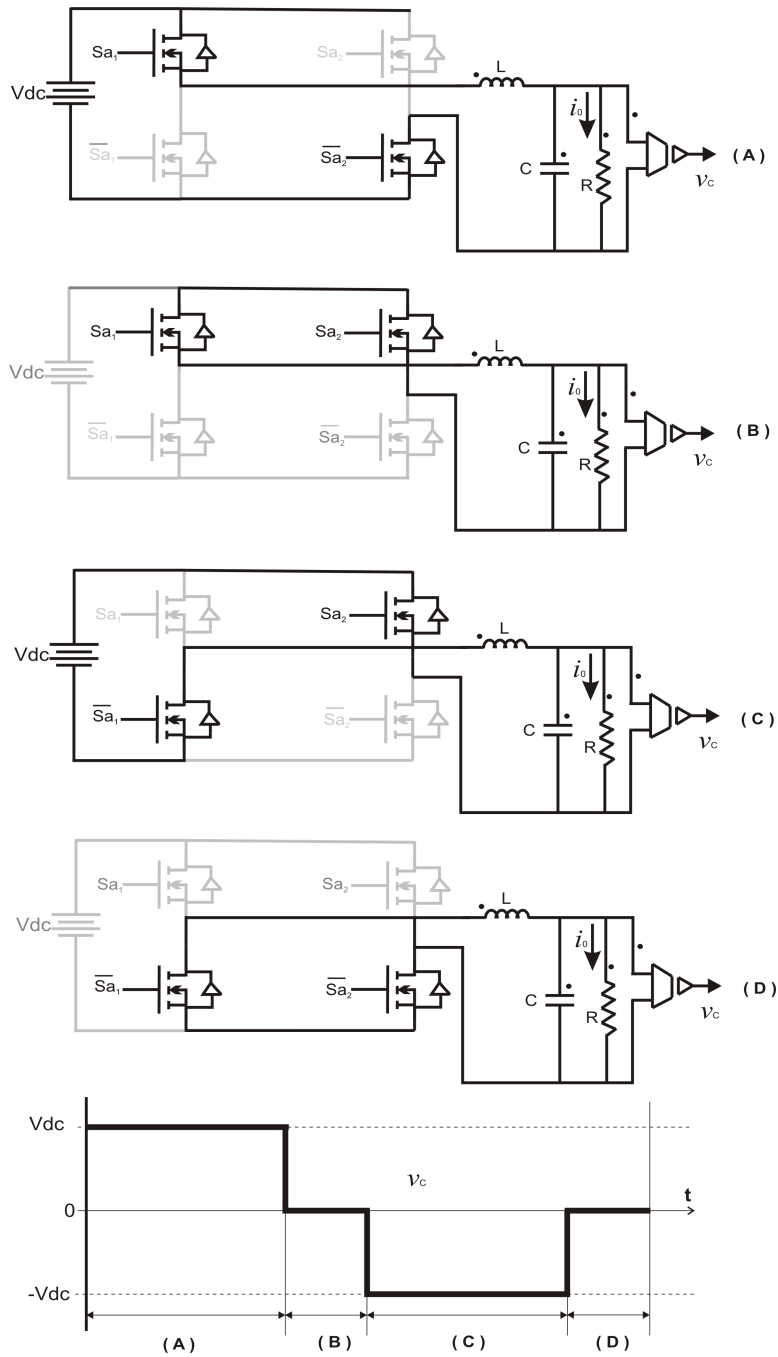


Figura 4.4: Estados de conducción del inversor puente completo considerando un filtro de salida de segundo orden.

Para el caso (C) en el que $sa_1 = 0$ y $sa_2 = 1$

$$L \frac{di_L}{dt} = -v_C - V_{cd} \quad (4.4)$$

$$C \frac{dv_C}{dt} = i_L - \frac{v_C}{R} \quad (4.5)$$

Se puede observar que las ecuaciones (4.3) y (4.5) son iguales, pero las (4.2) y (4.4) cambian en signo, por tanto, el modelo promedio para el convertidor se puede proponer como:

$$\begin{aligned} L \frac{di_L}{dt} &= -v_C + u_{av} V_{cd} \\ C \frac{dv_C}{dt} &= i_L - \frac{v_C}{R} \end{aligned} \quad (4.6)$$

donde u_{av} pertenece al conjunto continuo $u_{av} \in \{-1, 1\}$.

De las Figuras 4.3 y 4.4 se puede observar que cuando un semiconductor no conduce, bloquea el voltaje total V_{cd} de la fuente, por lo tanto, esta topología está reservada para aplicaciones de bajo voltaje, ya que se encuentra limitada principalmente por la tecnología del semiconductor empleado [1]. Una de las ventajas de esta topología es que el puente H puede ser usado como un módulo básico para construir convertidores multinivel, con mas niveles de voltaje y además con voltajes de operación superiores dado que los componentes se colocan en serie, permitiendo ser aplicados en mediano voltaje [1].

4.3.4. Modelo matemático del inversor multinivel en cascada con fuentes independientes

A la fecha y con el fin de aumentar los niveles de potencia que se pueden manejar, así como para mejorar la calidad de la onda de voltaje y corriente de salida, la estructura clásica basada en la conmutación de tres niveles de voltaje ha sido modificada, de tal forma que ahora es posible manejar más de tres niveles de tensión.

En la actualidad, numerosas aplicaciones requieren de altos niveles de potencia, por ejemplo, algunos motores de mediano voltaje, requieren de altos niveles de potencia en el orden de megawatts. Los convertidores de estructura multinivel han sido introducidos como una alternativa en situaciones donde se manejan voltajes medianos³ pero altas potencias; además, los convertidores multinivel también son empleados para interconectar las fuentes de energía renovable (fotovoltaica, viento, celdas de combustible, etc.) a la red eléctrica de una forma más fácil [98].

El concepto de convertidor multinivel fue introducido en 1975 (ver [103]) y el término multinivel inició con el convertidor de tres niveles en [5], posteriormente varios convertidores

³En México se considera de mediano voltaje si se encuentra a más de $1KV$, pero menos de $35kV$ [102]

multinivel se han desarrollado y patentado (ver [104, 105, 106, 107, 108, 109, 110]).

Sin embargo, el concepto elemental de un convertidor multinivel consiste en el manejo de altas potencias por medio de colocar en serie semiconductores de potencia como interruptores y varias fuentes aisladas de CD de bajo voltaje para realizar una conversión y sintetizar una onda escalonada de voltaje.

Capacitores, baterías y fuentes de energía renovables pueden ser usadas como las fuentes múltiples de voltaje. La conmutación de los semiconductores realiza una suma de los voltajes de estas fuentes múltiples para obtener un voltaje alto en la salida; una ventaja importante de este tipo de convertidores tiene que ver con la capacidad de voltaje que deben soportar los semiconductores, ya que sólo depende del valor de la fuente donde será conectado y que regularmente es de bajo voltaje.

Como ya se mencionó en el primer capítulo de este trabajo, una de las topologías que han mostrado una operación más confiable y que ofrece mejores prestaciones en relación a la calidad que se puede lograr en el manejo de la energía, es la que se conoce como de fuentes independientes, la cual tiene una estructura de forma tal que m puentes completos tipo H conectados en cascada, generan una señal de voltaje que toma un número de niveles dado por la expresión (4.7):

$$N = 2m + 1 \tag{4.7}$$

donde:

N = Número de niveles

m = Número de puentes H

Esta señal discontinua es finalmente alimentada a un circuito de filtrado que se encarga de mejorar la calidad de la onda de salida.

Desde otra perspectiva y en adición al problema de diseño de estructuras multinivel, un tema que es primordialmente importante para su buen funcionamiento es el que se refiere al control de estos dispositivos. En este sentido, la presencia de conmutadores en la estructura de los convertidores multinivel ocasiona que se presenten discontinuidades en sus modelos matemáticos, por tanto, el problema de diseño de esquemas de control deja de ser una tarea trivial.

En la Figura 4.5 se muestra un inversor multinivel con fuentes independientes de dos celdas, ya que el número de niveles esta dado por la ecuación (4.7) la cantidad de niveles es de cinco, en esta figura también se observa el filtro de segundo orden a la salida para mejorar la señal de salida en corriente y voltaje.

En este caso los posibles circuitos que resultan cuando se conmutan los interruptores $S_{a1,2,3,4}$ extenderían mucho el análisis dado que el inversor está controlado ahora por cuatro señales binarias, resultan entonces $2^4 = 16$ estados diferentes de conmutación de los dispositivos, para

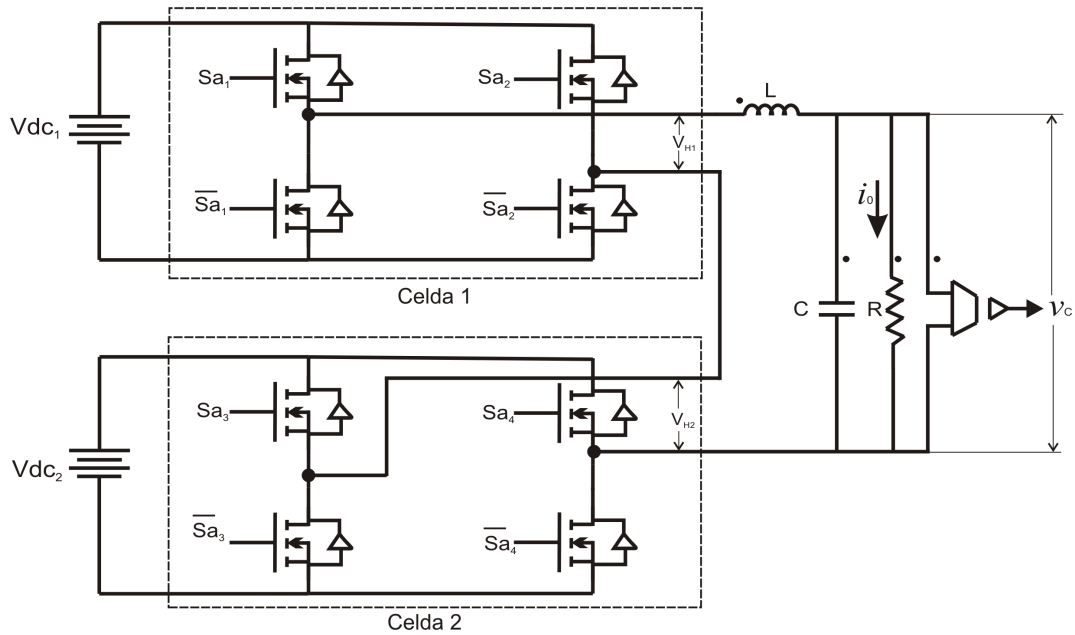


Figura 4.5: Inversor Multinivel de cinco niveles con filtro de salida de segundo orden.

simplificar el análisis estos se ven resumidos en la tabla 4.2, en ésta se asume que todas las fuentes de CD tienen exactamente el mismo valor ($Vcd_1 = Vcd_2$).

De la tabla 4.2 se puede observar que los estados de conmutación para cada puente individual son los mismos que se presentaban en el apartado anterior, por lo tanto, considerando que las etapas se conectan en cascada y dado que las fuentes de CD son iguales, para generalizar en este apartado se puede establecer el modelo promedio dado por la ecuación (4.8) (ver [34, 111]).

$$\begin{aligned} L \frac{di_L}{dt} &= -v_C + u_{av} E \\ C \frac{dv_C}{dt} &= i_L - \frac{v_C}{R} \end{aligned} \quad (4.8)$$

donde $E = Vcd_1 + Vcd_2$, y $u_{av} \in \{-1, 1\}$.

De manera general se puede establecer el modelo promedio para n puentes H (ver Figura 4.6) a partir de la ecuación (4.9):

$$\begin{aligned} L \frac{di_L}{dt} &= -v_C + u_{av} \sum_{i=1}^n Vcd_i \\ C \frac{dv_C}{dt} &= i_L - \frac{v_C}{R} \end{aligned} \quad (4.9)$$

donde $E = Vcd_1 + Vcd_2 + \dots + Vcd_n$, y $u_{av} \in \{-1, 1\}$.

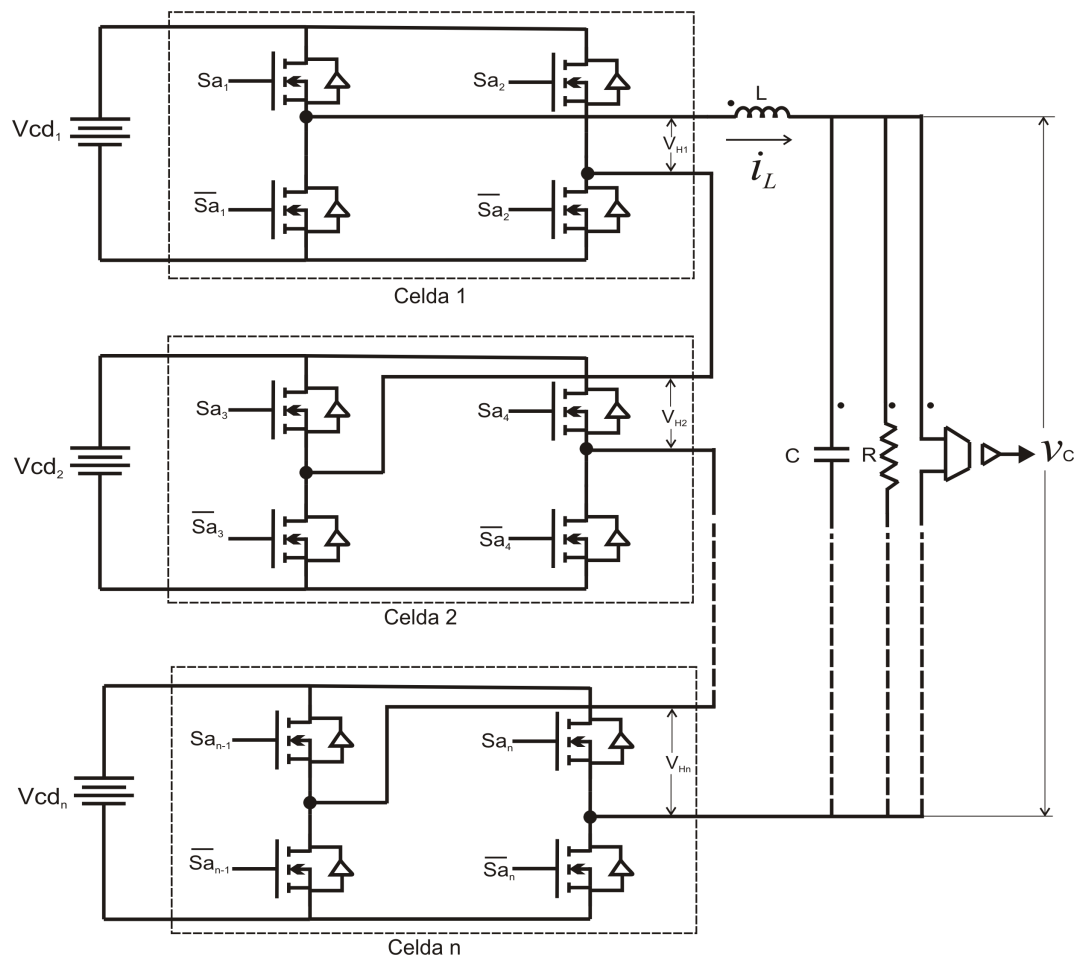


Figura 4.6: Inversor Multinivel de m -niveles con filtro de salida de segundo orden.

Tabla 4.2: Estados de conmutación con ($Vcd_1 = Vcd_2$)(ver Figura 4.5)

Estados de conmutación de un inversor de cinco niveles en cascada						
celda 1		celda 2		Voltaje celda 1	Voltaje celda 2	Voltaje de Salida
Sa_1	Sa_2	Sa_3	Sa_4	Vcd_1	Vcd_1	$v_C = Vcd_1 + Vcd_2$
1	0	1	0	Vcd	Vcd	$2Vcd$
1	0	0	0	Vcd	0	Vcd
1	0	1	1	Vcd	0	Vcd
0	0	1	0	0	Vcd	Vcd
1	1	1	0	0	Vcd	Vcd
0	0	0	0	0	0	0
1	1	0	0	0	0	0
0	0	1	1	0	0	0
1	1	1	1	0	0	0
1	0	0	1	Vcd	$-Vcd$	0
0	1	1	0	$-Vcd$	Vcd	0
0	1	0	0	$-Vcd$	0	$-Vcd$
0	1	1	1	$-Vcd$	0	$-Vcd$
0	0	0	1	0	$-Vcd$	$-Vcd$
1	1	0	1	0	$-Vcd$	$-Vcd$
0	1	0	1	$-Vcd$	$-Vcd$	$-2Vcd$

4.4. Análisis

Partiendo del modelo promedio descrito por las ecuaciones diferenciales en (4.8) que describen el comportamiento dinámico del inversor multinivel en cascada de cinco niveles con fuentes separadas, y suponiendo que el inversor estará sujeto a una perturbación desconocida, el modelo se puede reescribir de la siguiente manera (ver ecuación (4.10)):

$$\begin{aligned}
 L \frac{di_L(t)}{dt} &= -v_C(t) + u_{av}(t)E \\
 C \frac{dv_C(t)}{dt} &= i_L(t) - \frac{v_C(t)}{R} - i_P(t)
 \end{aligned} \tag{4.10}$$

En la ecuación (4.10), $i_L(t)$ es la corriente en el inductor, $v_C(t)$ es el voltaje en el capacitor de salida, $i_P(t)$ es la perturbación de corriente desconocida y $u_{av}(t)$ es la entrada de control promedio que puede tomar valores dentro de un conjunto continuo $u_{av} \in \{-1, 1\}$ (ver [112]) El modelo obtenido es un sistema lineal de segundo orden que tiene la forma típica de la ecuación (4.11):

$$\begin{aligned}
 \dot{x} &= Ax + Bu_{av} + P \\
 y(t) &= c^T x
 \end{aligned} \tag{4.11}$$

Por comparación entre las ecuaciones (4.10) y (4.11) se puede obtener:

$$A = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}; B = \begin{bmatrix} \frac{E}{L} \\ 0 \end{bmatrix}; \quad (4.12)$$

$$P = \begin{bmatrix} 0 \\ -i_p(t) \end{bmatrix}; c^T = \begin{bmatrix} 0 & 1 \end{bmatrix}; \quad (4.13)$$

4.4.1. Prueba de controlabilidad al sistema

En el modelo dado por la ecuación (4.10) si se considera a $i_p(t) = 0$, se puede obtener la matriz de controlabilidad de *Kalman* del sistema de la forma siguiente:

$$C = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix} \quad (4.14)$$

El determinante de la matriz de controlabilidad de la expresión (4.14) está dado por:

$$\frac{E^2}{CL^2} \neq 0$$

Por lo tanto, el modelo del sistema inversor con filtro a la salida es controlable y en consecuencia, es un sistema diferencialmente plano (ver H. Sira-Ramírez *et. al.* [113] y J. Lévine [114])

4.4.2. Obtención de la salida plana del sistema

La salida plana para el sistema se puede obtener basado en la siguiente proposición:

Proposición 3 *La salida plana de un sistema lineal controlable en el espacio de estados y que tiene la forma:*

$$\dot{x} = Ax + Bu \quad (4.15)$$

esta dada por un módulo de factor constante y por la combinación lineal de los estados obtenidos de la última columna de la inversa de la matriz de controlabilidad de Kalman:

$$F = [0, 0, \dots, 1] [B, AB, \dots, A^{n-1}B]^{-1} x \quad (4.16)$$

Acorde a la proposición (3), la salida plana para el sistema está dada por:

$$\begin{aligned} F &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{E}{L} & 0 \\ 0 & \frac{E}{LC} \end{bmatrix}^{-1} \begin{bmatrix} i_L \\ v_C \end{bmatrix} \\ &= \frac{LC}{E} v_C \end{aligned} \quad (4.17)$$

La ecuación (4.17) se puede simplificar tomando a la variable de voltaje como la salida plana de la siguiente forma (ver [113, 114]):

$$F = v_C \quad (4.18)$$

La planitud del sistema implica que todas las variables de estado del sistema, incluyendo la entrada de control, se pueden parametrizar en términos de $F = v_C$ y un número finito de sus derivadas temporales, es decir:

$$v_C = F(t) \quad (4.19)$$

$$i_L = C\dot{F}(t) + \frac{1}{R}F(t) + i_p(t) \quad (4.20)$$

Mientras que la entrada promedio de control está dada por:

$$u_{av} = \frac{LC}{E}\ddot{F}(t) + \frac{L}{ER}\dot{F}(t) + \frac{1}{E}F(t) + \frac{L}{E}\frac{d}{dt}i_p(t) \quad (4.21)$$

4.4.3. Prueba de observabilidad al sistema

Por otra parte, en la ecuación (4.10) se consideró que $i_p(t) = 0$, por tanto, se puede ver que el sistema es observable desde la variable de salida $F = v_C$ dado que la matriz de observabilidad de *Kalman* está definida por:

$$\begin{aligned} O &= \begin{bmatrix} c^T \\ c^T A \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \end{aligned} \quad (4.22)$$

La matriz en la ecuación (4.22) cumple con la propiedad de ser de rango completo, por lo tanto el modelo del sistema es observable desde la salida $F(t) = y(t) = v_C$. Este hecho establece la reconstructibilidad del sistema, ya que todas las variables de estado son parametrizables en términos de la entrada, la salida y un número finito de integrales iteradas de las variables de entrada y salida (ver [115, 116, 117]).

4.4.4. Obtención del reconstructor de estado

Si se integra la ecuación (4.21) por ambos lados y se resuelve para la variable F , se obtiene un estimador integral de la primera derivada de F , esto está dado por:

$$\hat{\dot{F}} = \frac{E}{LC} \int_0^t \left[u_{av}(\tau) - \frac{1}{E}F(\tau) \right] d\tau - \frac{1}{CR}F(t) \quad (4.23)$$

donde,

$$F(t) = y(t) = v_C(t) \quad (4.24)$$

Se puede notar que para un estado inicial diferente de cero, la relación que vincula la derivada de la salida de voltaje del convertidor con el estimador en la ecuación (4.23) está dada por:

$$\dot{F} = \widehat{\dot{F}} + \dot{F}_0 \quad (4.25)$$

Donde \dot{F}_0 denota la razón de cambio inicial de la salida de voltaje. El reconstructor integral dado por (4.23) provee el valor estimado del término derivativo del controlador, sin tener que medir y calcular la derivada de la salida plana (señal medida, $F(t) = y(t) = v_C(t)$).

4.4.5. Trayectorias de referencia

Dado que el controlador formulado en este trabajo se propone sea de seguimiento, aprovechando la propiedad de planitud diferencial expuesta en el apartado anterior, se propone que la trayectoria para el voltaje de salida del convertidor multinivel $F^*(t) = y^*(t) = v_C^*(t)$ sea una señal senoidal del tipo $F^*(t) = A_m \sin(\omega_n t)$, la cual tiene amplitud A_m y frecuencia natural $\omega_n = 2\pi f$, para una frecuencia f de 60 Hz, válida en américa central.

Por lo tanto, la trayectoria de referencia y sus derivadas temporales son:

$$F^*(t) = A_m \sin(\omega_n t) \quad (4.26)$$

$$\dot{F}^*(t) = A_m \omega_n \cos(\omega_n t) \quad (4.27)$$

$$\ddot{F}^*(t) = -A_m \omega_n^2 \sin(\omega_n t) \quad (4.28)$$

4.4.6. Diseño del Controlador GPI de seguimiento de voltaje

Como ya se comprobó en los apartados anteriores, el sistema es controlable, posee salida plana y es observable desde esta salida plana, por lo tanto, ahora se procederá a diseñar el controlador GPI para el inversor multinivel en cascada de cinco niveles con fuentes independientes.

Dado el sistema del convertidor representado por el modelo matemático en ecuaciones de estado:

$$\begin{aligned} \dot{i}_L(t) &= -\frac{1}{L}v_C(t) + \frac{E}{L}u_{av}(t) \\ \dot{v}_C(t) &= \frac{i_L(t)}{C} - \frac{v_C(t)}{CR} - \frac{i_P(t)}{C} \\ y(t) &= v_C(t) \end{aligned} \quad (4.29)$$

se asume lo siguiente:

- a) La única variable que se medirá será el voltaje de salida del convertidor $v_C(t)$.
- b) Para todos los parámetros del sistema ($\{C, L, R, E\}$) se conoce su valor nominal.
- c) La perturbación de corriente es variante con el tiempo, además de ser absolutamente acotada; por otra parte, la perturbación de corriente es completamente desconocida.
- d) La cantidad $\hat{F}(t)$ es una función del tiempo, y es absolutamente acotada para la trayectoria $\dot{F}(t)$ del sistema, y converge alrededor de una vecindad de la trayectoria de referencia $\dot{F}^*(t)$ debido a la retroalimentación de acción suave del controlador de seguimiento.
- e) El voltaje de salida del convertidor $v_C(t)$ y la corriente en la carga i_L deben de estar en fase, buscando obtener un factor de potencia (**PF**) cercano a la unidad.

A partir del modelo dado en las ecuaciones (4.29) se diseña la ley de control para el seguimiento de voltaje de la salida del convertidor multinivel para obtener un voltaje deseado el cual esta denotado por $F^*(t)$, por tanto, se propone una ley de control basada en una retroalimentación de estados de la manera siguiente:

$$\begin{aligned} u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\dot{F} + \frac{1}{E}F \\ v &= \dot{F}^*(t) - k_3 \left[\hat{F}(t) - \dot{F}^*(t) \right] - k_2 [F(t) - F^*(t)] \end{aligned} \quad (4.30)$$

Sí en la ecuación (4.30) se sustituye a la variable de estado \dot{F} por la variable estimada de la derivada de la salida plana \hat{F} dada por la ecuación (4.23), la ecuación (4.30) se verá afectada por desestabilizaciones ocasionadas por la incorporación del estimador de estado \hat{F} , para corregir de manera satisfactoria este efecto ocasionado principalmente por el error intrínseco de la estimación y por la perturbación externa ($i_P(t)$) se empleará una compensación por medio de integrales del error de la salida plana (error de seguimiento), es decir, se establece el error de seguimiento de voltaje por medio de la diferencia entre el valor de la amplitud del voltaje medido a la salida del convertidor y el voltaje de la trayectoria deseada de la siguiente forma:

$$e = F(t) - F^*(t) \quad (4.31)$$

Cuando en el diseño del controlador se incorpora al estimador \hat{F} y además se realiza la compensación por medio de integrales del error, se está formulando un controlador que recibe el nombre de control Proporcional-Integral Generalizado (GPI), en este caso, el control propuesto esta descrito por:

$$\begin{aligned}
u_{av} &= \frac{LC}{E}v + \frac{L}{ER}\widehat{F} + \frac{1}{E}F \\
v &= \ddot{F}^*(t) - k_3 \left[\widehat{F}(t) - \dot{F}^*(t) \right] - k_2 [F(t) - F^*(t)] \\
&\quad - k_1\gamma - k_0\eta \\
\dot{\gamma} &= F(t) - F^*(t) \\
\dot{\eta} &= \gamma
\end{aligned} \tag{4.32}$$

Donde \widehat{F} está descrito por la ecuación (4.23).

La dinámica del error se obtiene mediante la sustitución en la ecuación (4.21) (la cual describe a la entrada de control promedio) en la ecuación del estimador de estado (4.23) y en las ecuaciones del controlador propuesto dadas por la ecuación (4.32), obteniendo la siguiente expresión:

$$\begin{aligned}
\ddot{F}(t) = v &= -\frac{1}{C} \frac{d}{dt} i_p(t) + \ddot{F}^*(t) - k_3 \left[\left(\dot{F} - \dot{F}_0 \right) - \dot{F}^*(t) \right] \\
&\quad - k_2 [F(t) - F^*(t)] - k_1 \int_0^t [F(\tau) - F^*(\tau)] d\tau \\
&\quad - k_0 \int_0^t \int_0^\tau [F(\lambda) - F^*(\lambda)] d\lambda d\tau
\end{aligned} \tag{4.33}$$

La ecuación característica que posee las relaciones integro-diferenciales en términos del error de seguimiento está dado por:

$$\begin{aligned}
&e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e \\
&= -\frac{1}{C} \frac{d^3}{dt^3} i_p(t)
\end{aligned} \tag{4.34}$$

4.4.7. Elección de las ganancias del controlador

De la ecuación (4.34) se puede observar que si la ganancia k_3 se elige igual a $k_3 = \frac{1}{CR}$, esta se reduce a lo siguiente:

$$e^{(4)} + k_3 e^{(3)} + k_2 \ddot{e} + k_1 \dot{e} + k_0 e = -\frac{1}{C} \frac{d^3}{dt^3} i_p(t) \tag{4.35}$$

Por lo tanto, para eliminar los efectos de los valores iniciales en el controlador causado por el reconstructor integral \widehat{F} se propone que los valores de los parámetros de diseño $\{ \underbrace{k_3}_{:=1/CR}, k_2, k_1, k_0 \}$ se elijan con el polinomio característico en lazo cerrado dado por la ecuación (4.36) de tal forma que todas las raíces del mismo se encuentren en la parte izquierda del plano complejo.

$$p(s) = s^4 + k_3s^3 + k_2s^2 + k_1s + k_0 \quad (4.36)$$

Los parámetros del controlador se elegirán por medio del siguiente polinomio característico:

$$p_d(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)^2 = s^4 + \underbrace{4\zeta\omega_n}_{:=k_3} s^3 + \underbrace{(2\omega_n^2 + 4\zeta^2\omega_n)}_{:=k_2} s^2 + \underbrace{4\omega_n^3\zeta}_{:=k_1} + \underbrace{\omega_n^4}_{:=k_0} \quad (4.37)$$

donde los valores de ζ y ω_n son cantidades positivas.

Dado que se asume que $i_P(t)$ es absolutamente acotado entonces existen coeficientes k_i para el diseño del controlador de seguimiento GPI los cuales pueden ser elegidos por medio de una comparación directa término a término entre el polinomio dado en (4.35) y el polinomio característico deseado $p_d(s)$ de cuarto grado dado en la ecuación (4.37).

Se puede observar que del polinomio dado en la ecuación (4.37), $\omega_n = 1/(4\zeta CR)$, entonces las ganancias para el controlador GPI están dadas por:

$$\begin{aligned} k_3 &= 4\zeta\omega_n \\ k_2 &= 2\omega_n^2 + 4\zeta^2\omega_n \\ k_1 &= 4\omega_n^3 \\ k_0 &= \omega_n^4 \end{aligned} \quad (4.38)$$

Es necesario recordar que las raíces del polinomio característico se deben de elegir lo suficientemente lejos en la parte izquierda del plano complejo para lograr disminuir el efecto de la perturbación $i_P(t)$.

Logrando lo anterior, el error de seguimiento del controlador ($e = F(t) - F^*(t)$) y sus derivadas temporales convergerán de manera asintótica alrededor del origen del error de seguimiento dentro de una pequeña región deseada y permisible.

4.5. Simulación

Siguiendo con la fase 1 de la metodología propuesta, y basado en los resultados obtenidos en las secciones 4.3.4 y 4.4 donde se obtiene el modelo promedio del convertidor multinivel y se analizan las propiedades del mismo para proponer el controlador respectivamente, ahora se procederá a realizar la simulación del sistema de ecuaciones que describen el comportamiento del convertidor junto con su controlador; como ya se mencionó, en esta tarea se verifica el nivel de aproximación del modelo propuesto, además de examinar el desempeño del controlador ante perturbaciones exógenas.

Para realizar esta labor se empleará en un primer acercamiento el software **SIMNON**⁴ [118] y después se realizará una cosimulación entre las piezas de software **Matlab/Simulink**⁵ [119] y **PSIM**⁶[120] .

En ambas simulaciones se emplearan los parámetros de diseño mostrados en la tabla 4.3 (verificar los valores del filtro de salida en el Apéndice B):

Tabla 4.3: Parámetros de diseño del para el convertidor multinivel y controlador GPI

Parámetros de diseño para el convertidor multinivel y controlador		
Parámetro	Unidad	Valor
Voltaje de celdas	E	160 Volts
Capacitor de filtrado	C	$10\mu F$
Inductancia de filtrado	L	$3mH$
Resistencia carga	R	75Ω
Frecuencia natural	$\omega_n = 1/(4\zeta CR)$	1178
Coef. amortiguamiento	ζ	0,707
Frecuencia	f	$60Hz$
Ampl. deseada	A_m	145 Volts
Constantes de sintonía (ver ec. (4.38))	k_0	Variables
	k_1	
	k_2	
	k_3	

4.5.1. Simulación en SIMNON

Para poder simularlo en SIMNON es necesario pasar las ecuaciones del convertidor multinivel y del controlador GPI a variables de estado (Ver ecuaciones (4.23) y (4.32)). Como ya se había mencionado, para realizar las simulaciones se emplearan los parámetros concentrados en la tabla 4.3.

El valor deseado de voltaje de acuerdo a la tabla 4.3 es de $A_m = 145$ Volts. El valor de ω_n dado en la sección anterior sirvió de punto de prueba ($\omega_n = 1178$), a partir de ese valor de

⁴**SIMNON**® puede ser utilizado para la simulación de algoritmos de control complejos, para modelos financieros, para analizar la dinámica de robots; todos los sistemas que pueden definirse en términos matemáticos, se pueden simular con este software

⁵**Simulink**® es un entorno para simulación y diseño multidominio basado en modelos para sistemas dinámicos y embebidos. Proporciona un entorno gráfico interactivo y un conjunto de bibliotecas de bloques que le permiten diseñar, simular, implementar y probar una serie de sistemas variables en el tiempo, incluidas las comunicaciones, control, procesamiento de señales, procesamiento de vídeo y procesamiento de imágenes.

⁶**PSIM**® es un software de simulación diseñado específicamente para electrónica de potencia y el control de motores. Provee una simulación rápida y una interfaz de usuario amigable, PSIM proporciona un entorno de simulación de gran alcance para hacer frente a las necesidades de simulación en electrónica de potencia.

manera heurística se incrementó su magnitud, observándose que a partir de $\omega_n = 1300$ la simulación entregó resultados aproximados a los esperados (Ver código en el apéndice C).

En la Figura 4.7 se muestran los resultados para $v_C, i_L, u_{av}, \hat{F}, e$ y $\int edt$; se considera un valor de $\omega_n = 1300, \zeta = 0,707$, y una vez realizada la sustitución en las expresiones (4.38), las constantes del controlador son:

$$\begin{aligned} k_3 &= 3676,4 \\ k_2 &= 3382599,2148 \\ k_1 &= 6,213116000000001E9 \\ k_0 &= 2,8561E12 \end{aligned} \tag{4.39}$$

En la gráfica 4.7(a) se muestra el valor de voltaje de salida del convertidor comparado con el voltaje deseado, en esta gráfica se puede observar que durante el cambio de carga realizado (perturbación de corriente debido a un escalón positivo) el valor de voltaje en la salida del convertidor se conserva constante, pero sufre un ligero sobretiro durante el transitorio, mientras que la corriente en 4.7(b), como es de esperarse se incrementa; en la Figura 4.7(c) se muestra la salida de control del convertidor, en la Figura 4.7(d) se observa el comportamiento del estimador de estado (ver ecuación (4.23)), en las Figura 4.7(e) y (f) se muestra el resultado del error de seguimiento y de la integral de ese error de seguimiento respectivamente.

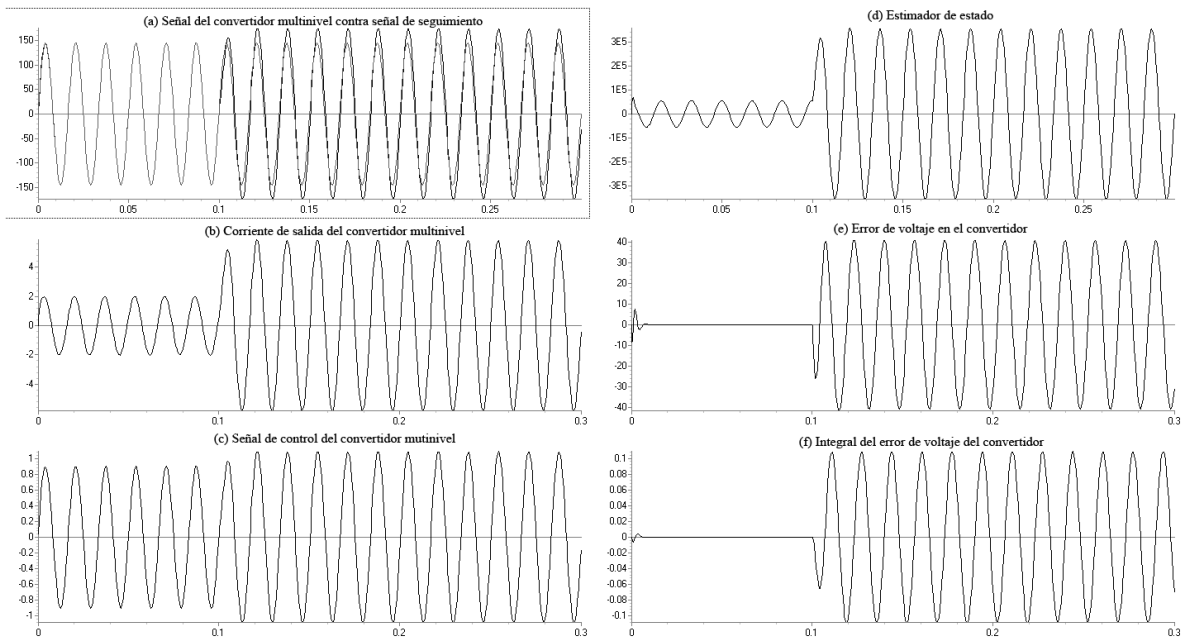


Figura 4.7: Simulación de Simmon.

En la Figura 4.8 se muestran los mismos parámetros que en la Figura 4.7 pero ahora con

valores de $\omega_n = 3000$, $\zeta = 0,707$ (los valores de las constantes se muestran en las expresiones (4.40)), se observa que el sistema se comporta de mejor manera en todos sus parámetros, la señal de seguimiento y el valor del voltaje de salida del convertidor son bastante semejantes y el voltaje de salida no disminuye al realizar un cambio de carga y tampoco se observa sobretiro en la salida de voltaje.

$$k_3 = 8484. \quad (4.40)$$

$$k_2 = 18005998,188$$

$$k_1 = 7,6356E10$$

$$k_0 = 8,0999E13$$

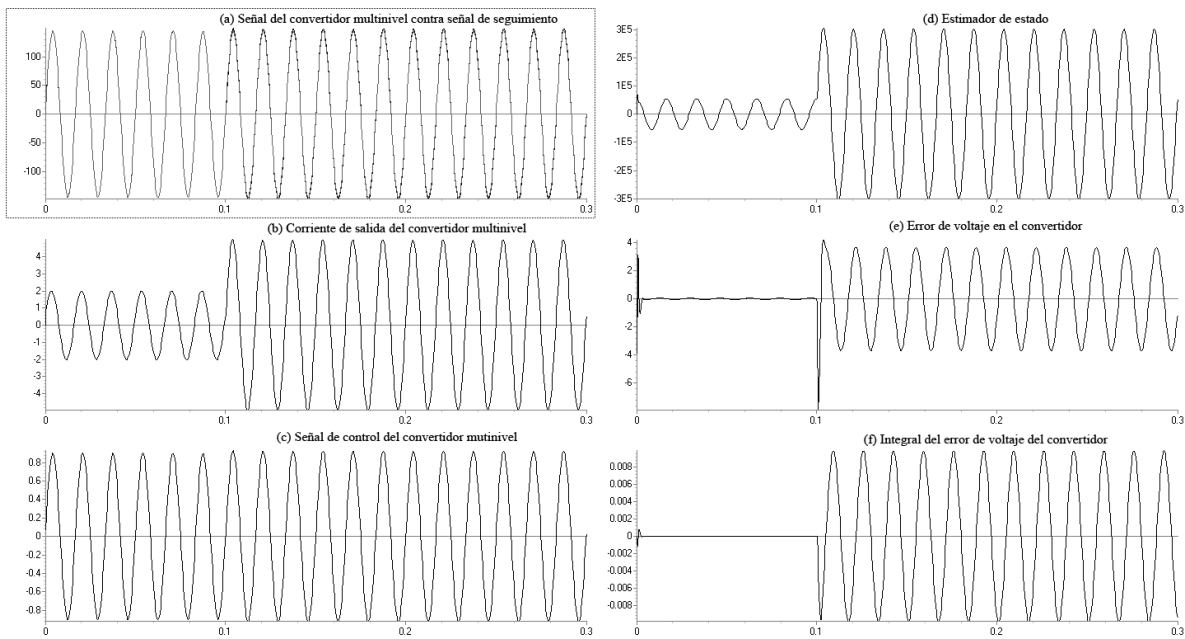


Figura 4.8: Simulación de Simmon.

Con el fin de reforzar los resultados conseguidos, en la siguiente sección se presentara la cosimulación por medio de **Matlab/Simulink-PSIM**.

4.5.2. Simulación con MATLAB/Simulink-PSIM

Para este apartado se hace necesario el manejo de diferentes herramientas de simulación de circuitos. A continuación se enumeran las herramientas empleadas y se justifica su uso brevemente.

- **PSIM:** En este programa se realizó la concepción del esquema eléctrico de forma básica, el cual incluye a las fuentes de tensión independientes de CD, los puentes H de cada celda, etc., a partir de estos, se simula el comportamiento del voltaje y la corriente de salida del convertidor multinivel (Ver Figura 4.9).
- **Matlab/Simulink:** Es un programa de cálculo matemático muy flexible y potente, con posibilidades gráficas para la representación de gráficos. En concreto se ha hecho uso de la biblioteca *Simulink*, que facilita el análisis, el diseño y la simulación de sistemas al incluir rutinas que resuelven los cálculos matemáticos junto con una interfaz sencilla. Se ha utilizado el entorno gráfico que permite dibujar los sistemas como diagramas de bloques. La finalidad de la simulación es mostrar la tensión a la salida del filtro y su comportamiento ante un cambio de carga, esto se realizó en cosimulación con el programa PSIM (ver Figura 4.11)por medio de la herramienta *Simcoupler*⁷[121] , el funcionamiento en general de esta herramienta se ilustra en la Figura 4.10.

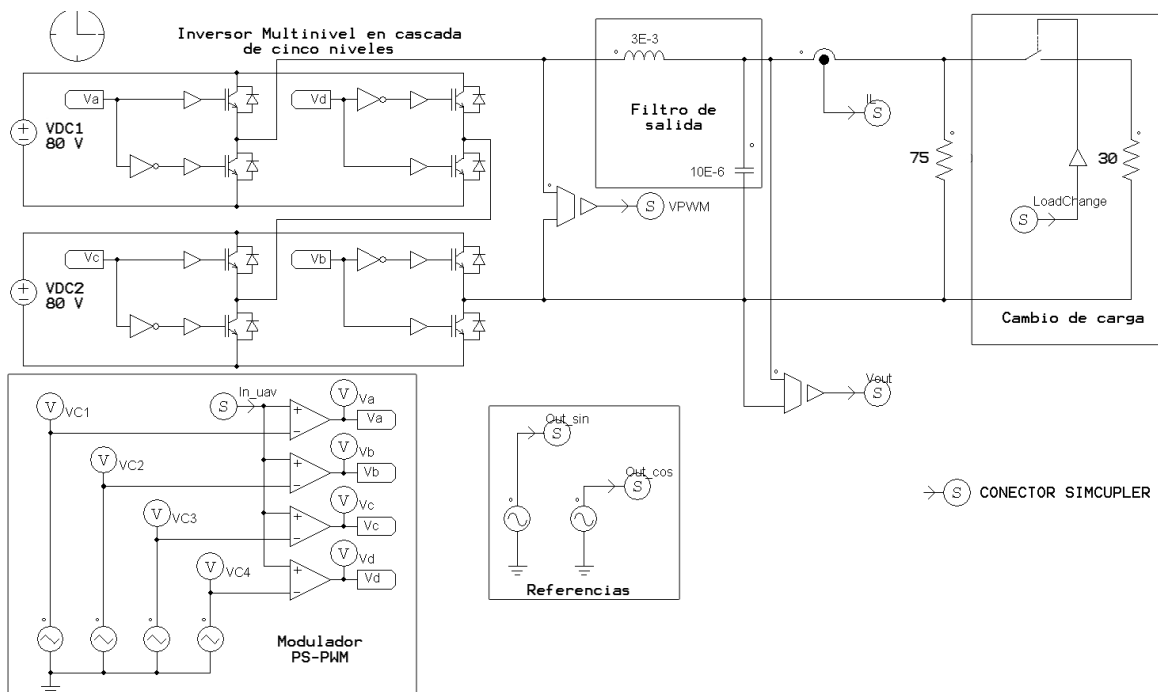


Figura 4.9: Modelo del sistema construido en PSIM.

Para realizar esta simulación se escriben por medio de bloques las ecuaciones (4.23), (4.26), (4.32), (4.31), (4.33), (4.38); el modelo del sistema se describe por medio del circuito eléctrico realístico construido en PSIM.

La descripción de cada uno de los bloques funcionales en Matlab/Simulink se realizará en el

⁷Es un módulo adicional del software PSIM, provee una interfaz entre PSIM y Matlab/Simulink para realizar cosimulaciones, es decir, una parte del sistema a simular es implementado en PSIM y el resto en Matlab/Simulink

Cosimulación con Matlab/Simulink

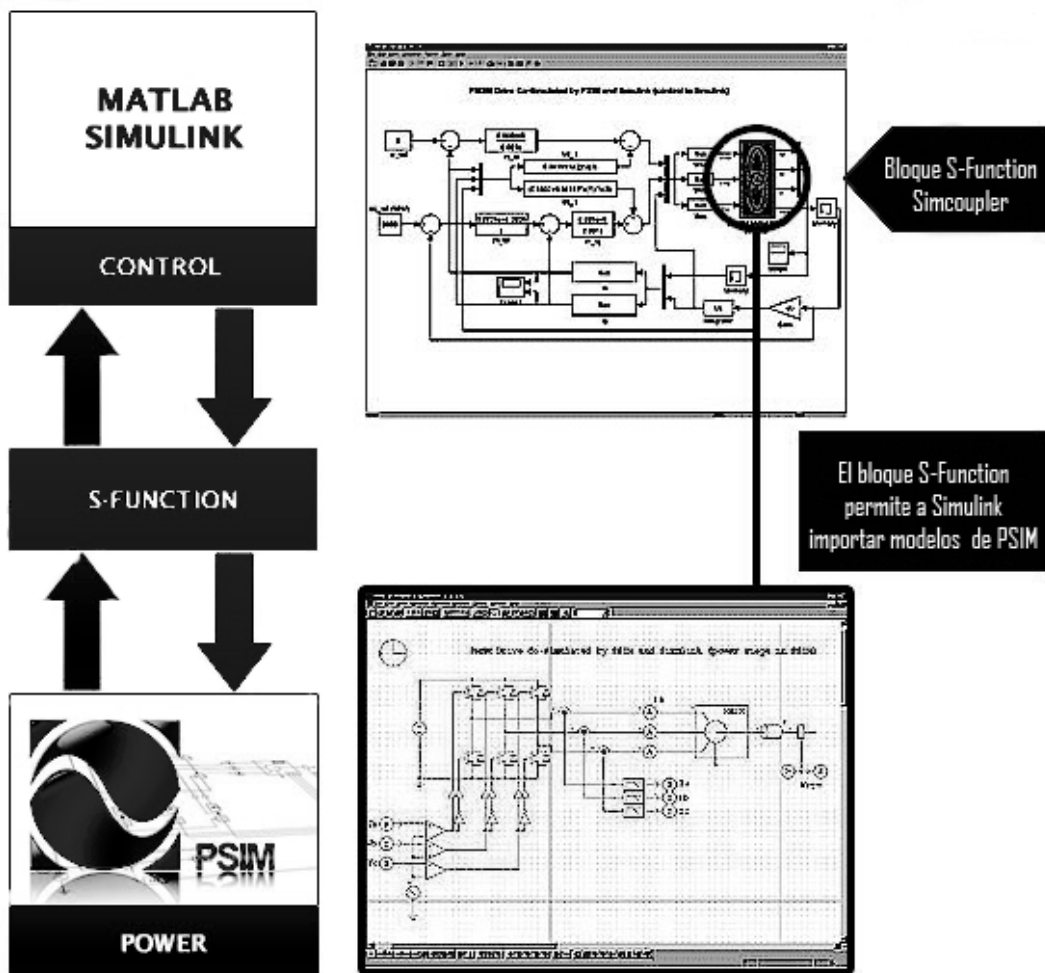


Figura 4.10: Diagrama conceptual sobre el proceso de cosimulación realizado entre *Matlab/Simulink* y *PSIM*.

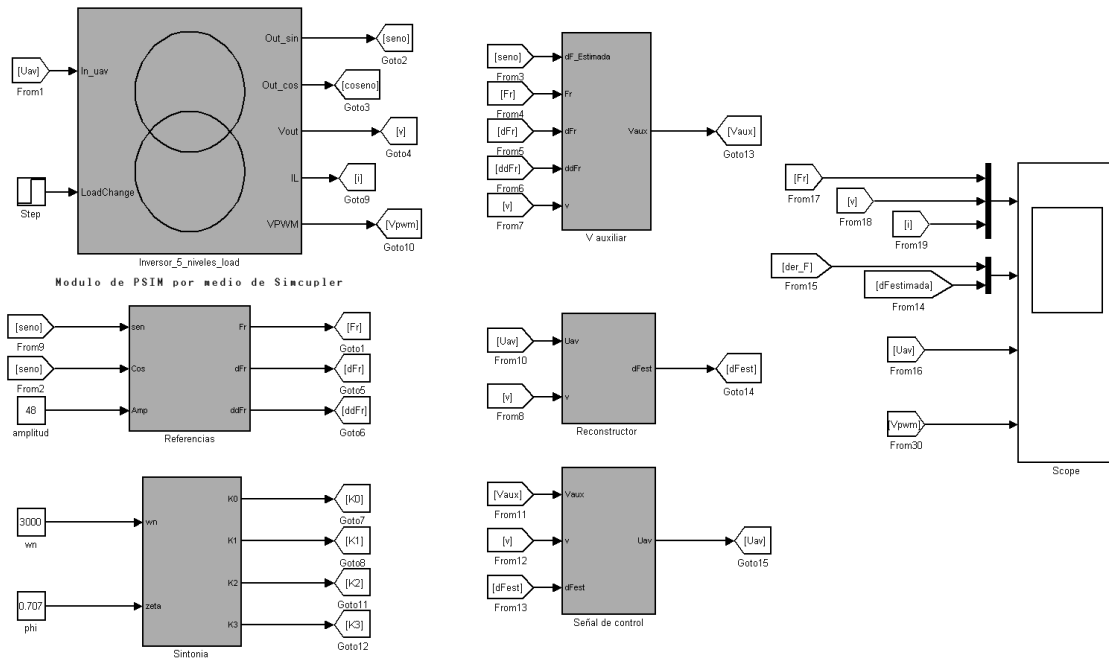


Figura 4.11: Diagrama general de bloques funcionales construidos en *Matlab/Simulink*.

capítulo 5.

En la Figura 4.12 se muestra la respuesta obtenida de la cosimulación Matlab/Simulink-PSIM. Esta simulación se realizó con los valores de $\omega_n = 3000$, $\zeta = 0,707$ y los valores de las constantes que se muestran en las expresiones (4.40), para una amplitud deseada $A_m = 145$ V. Como se puede ver en la Figura 4.12, la amplitud del voltaje de salida se conserva, aunque el sistema sea sujeto a un cambio de carga.

De los resultados obtenidos de las secciones 4.5.1 y 4.5.2 se puede concluir que el modelo obtenido en la sección 4.3.4 es válido, y que al menos en simulación el funcionamiento del controlador tiene un buen desempeño.

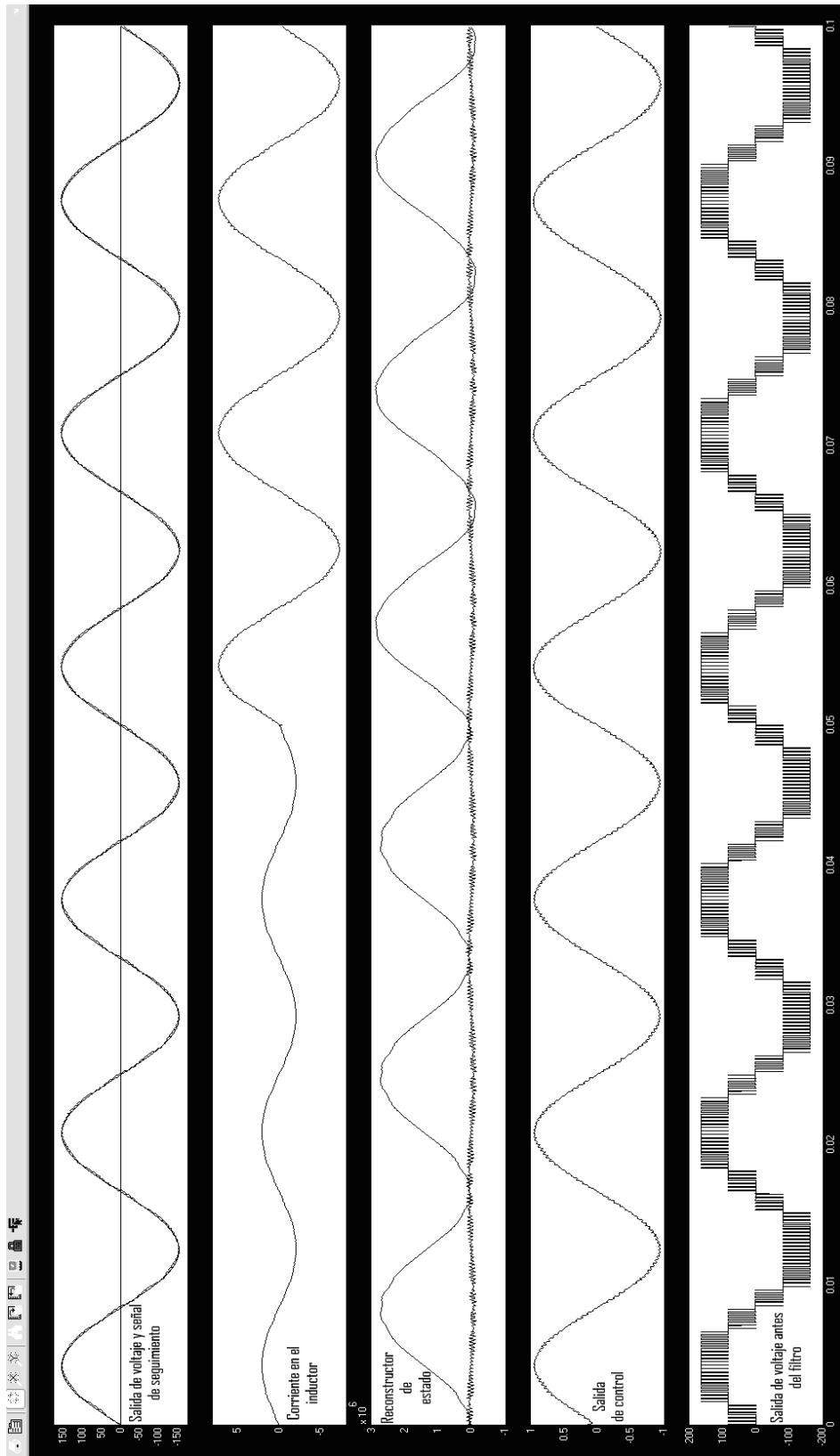


Figura 4.12: Resultado de la cosimulación entre *Matlab/Simulink* y *PSIM* para el inversor multinivel monofásico en cascada de cinco niveles.

Parte III

Metodología empleada

Fase II

Capítulo 5

Diseño de la arquitectura segmentada en FPGA

Resumen: En este capítulo se realiza el rediseño digital del controlador a través de las etapas restantes de la metodología, se realiza la elección del tiempo de muestreo, el rediseño digital, la descripción en el lenguaje **HDL**, la implementación, hasta llegar finalmente a la etapa de pruebas experimentales que se realizaron por último al controlador.

5.1. Introducción

La creación de herramientas EDA ha impactado fuertemente en el diseño y desarrollo de sistemas en el campo de la microelectrónica. La automatización EDA comprende a las herramientas de software y hardware usadas para diseñar, verificar e implementar circuitos electrónicos, sistemas y subsistemas basados en procesadores, DSP o FPGAs.

Para la implementación del controlador propuesto en el capítulo anterior, se ha elegido la herramienta EDA de Xilinx llamada *ISE Design Suite 9.2* y a una tarjeta de desarrollo de la casa Digilent (*Nexys 2*), la cual posee un FPGA que pertenece a la familia *Spartan-3E* de Xilinx.

La herramienta *ISE Design Suite* ofrece un flujo de diseño para FPGAs y CPLDs, permitiendo un acceso instantáneo a las características del lenguaje *HDL*, síntesis y simulación, implementación, colocación en el dispositivo y programación vía interfaz JTAG.

Por otro lado, la tarjeta de desarrollo basada en el FPGA *Spartan-3E*, diseñada por la compañía Digilent, incluye un FPGA *Spartan-3E 1200*, este FPGA es ideal para un amplio rango de aplicaciones de la electrónica de consumo, las cuales incluyen aplicaciones de banda ancha en comunicaciones, redes domésticas, pantallas de proyección, equipos de televisión digital, etc..

Actualmente los FPGAs son muy empleados en aplicaciones de sistemas de control, principalmente por la característica de que este sistema digital ejecuta de manera casi-instantánea

un algoritmo de control, cuestión que es de gran interés para los ingenieros de sistemas de control.

Un FPGA es un buen candidato para realizar sistemas de control gracias a dos características:

1. Una vez que fueron identificadas las operaciones que posee el algoritmo de control estas pueden ser procesadas a través de múltiples componentes en paralelo, siempre y cuando las operaciones posean cierta independencia unas con otras.
2. Los FPGAs permiten usar técnicas de arquitecturas segmentadas (pipeline architectures); en este tipo de estructuras la tarea se divide en etapas de tal forma que se genera un nuevo resultado cada ciclo de reloj.

Las características mencionadas pueden ser usadas para reducir el tiempo al realizar una tarea sin adicionar más hardware.

Continuando con las fases de la metodología que se siguió para realizar este trabajo, en este capítulo se explicará la implementación en hardware del *controlador GPI de seguimiento* correspondiente al rediseño digital del controlador, esta tarea está basada en las especificaciones mostradas en la tabla 4.3 y en las ecuaciones del controlador obtenido en el capítulo 4.

5.2. Partición Modular

El controlador de seguimiento GPI está diseñado basado en una propuesta modular y jerárquica (*Top-Down*) [122], los módulos fueron definidos con diversos niveles de abstracción. Para este diseño se hace una descripción por medio de un esquemático que muestra de manera general los módulos que intervienen en el controlador, los módulos en sí, fueron modelados usando el lenguaje VHDL (VHDL, *Very High Hardware Description Language*), los módulos aritméticos fueron generados usando la herramienta *Xilinx Core Generator*, la cual incluye un generador de componentes pre-optimizados.

El análisis y la simulación expuestos en el capítulo 4 muestran que debido al rango de resultados generados por las operaciones involucradas en el controlador, es necesario usar el formato en punto flotante¹; con esta intención, se eligió al estándar que provee soporte a las operaciones binarias en punto flotante IEEE-754-1985 [123].

De manera general, en la Figura 5.1, se expone una primera partición modular para el controlador GPI de seguimiento basado en FPGA, este proceso genera cuatro módulos:

- Administrador de reloj.
- Ley de control de seguimiento GPI (Controlador GPI).
- Interfaz de adquisición de datos analógico-digital (*Interfaz ADC*).

¹La aplicación en su procesamiento genera un alto rango dinámico de valores numéricos (valores entre $625E^{-12}$ y $16E^8$), por tanto, dado que el formato en punto flotante permite representar valores tan grandes como $(\pm 3,4E^{38})$ y tan pequeños como $(\pm 1,2E^{-38})$ se consideró el adecuado después de compararse con formatos de punto fijo tales como $Q16,16$, $Q0,32$, $Q31,32$ ya que estos no permiten representar dichas cantidades.

- Generador PWM (*Modulador PS-PWM*).

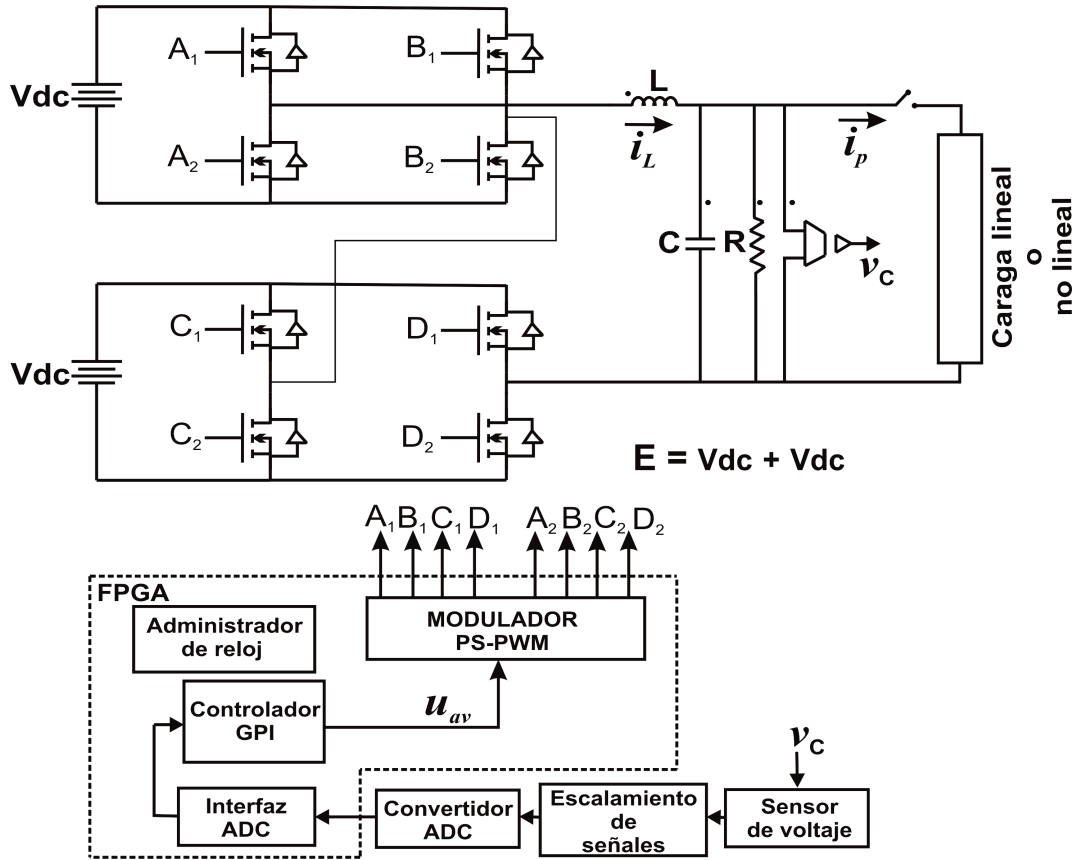


Figura 5.1: Concepto de diseño de los módulos dentro del dispositivo FPGA del controlador GPI de seguimiento para el inversor multinivel en cascada de cinco niveles.

Más adelante se realizará otra partición modular en la fase de implementación del controlador, específicamente sobre el módulo del controlador de seguimiento; esta partición tendrá lugar para incrementar el desempeño del sistema de control; el controlador se modeló como una arquitectura segmentada y será descrita posteriormente.

5.3. Simulación Funcional

En el apartado 4.5.1 y 4.5.2 del capítulo 4 se realizó la simulación del controlador en bloques funcionales, tanto en ecuaciones de estado (**SIMNON**), como por medio de una cosimulación (**Matlab/Simulink**), para complementar esta simulación, se mostrará el contenido de cada bloque funcional de la Figura 4.11.

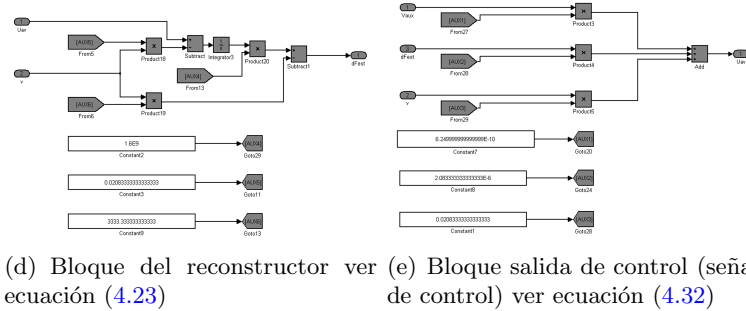
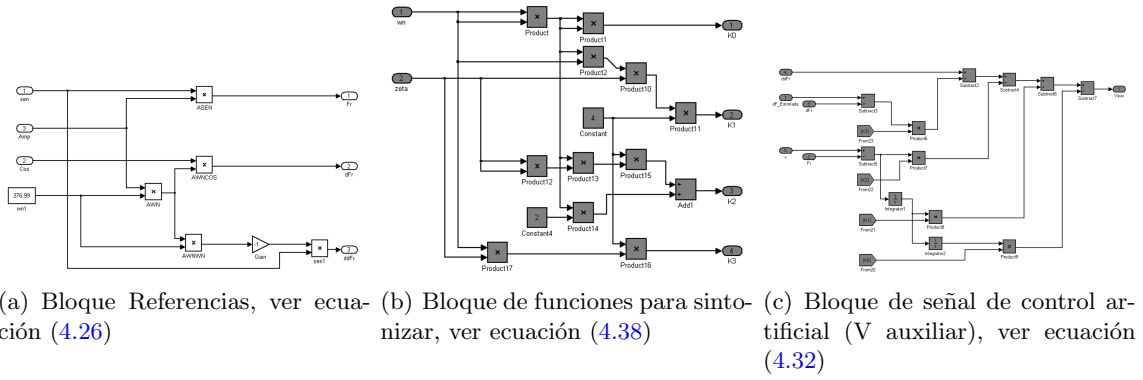


Figura 5.2: Descripción de los bloques funcionales de la Figura 4.11.

5.4. Rediseño Digital y Gráfico de Optimización de Flujo de Datos

En el diseño de un controlador basado en lógica reconfigurable se pueden emplear dos opciones, la primera es diseñar el controlador continuo en el tiempo, y después obtener una aproximación discreta de las ecuaciones continuas que componen el algoritmo del controlador y finalmente, basados en esta aproximación discreta, modelar el controlador, este método es llamado *rediseño digital*; la segunda opción es obtener la función de transferencia discreta del controlador y basado en ésta diseñar el controlador.

En ambos casos debe de ser aplicada una técnica de discretización² para lograr la implementación en el dispositivo de lógica reconfigurable que se eligió.

En este trabajo se seleccionó el *rediseño digital* para realizar la implementación del controlador GPI de seguimiento.

Como ya se comentó, la arquitectura propuesta para el controlador será segmentada (o *pipelining*), las arquitecturas segmentadas³ surgen por la necesidad de aumentar la velocidad de procesamiento; la segmentación puede mejorar de manera dramática el desempeño de un diseño a través de la reestructuración de las rutas largas de procesamiento con varios niveles

²Procedimiento para obtener sistemas en tiempo discreto, que se comporten aproximadamente igual que un sistema en tiempo continuo [125].

³Segmentar una arquitectura significa dividirla en segmentos o etapas, cada segmento (etapa) está definido por un registro que almacena los datos a procesar y otro que almacena los resultados.

más pequeños (segmentos), en otras palabras, el flujo de procesamiento de datos es separado en varios procesos y múltiples señales de reloj.

Este método permite el trabajo a velocidades de reloj más elevadas e incrementa el rendimiento en la salida de datos, pero crea un tiempo de latencia, el cual está directamente relacionado con el número de registros que se empleen en el diseño.

Debido a que un FPGA posee una gran cantidad de registros en su estructura física, usualmente es ventajoso crear estructuras segmentadas en este tipo de dispositivos ya que al hacer uso de los registros internos en términos económicos, no eleva el costo del diseño.

El análisis del tiempo de propagación de las etapas que integran el controlador se realizó a partir de las ecuaciones (4.32), en la Figura 5.3 se representan dichas ecuaciones de manera gráfica por medio de un diagrama de flujo de datos.

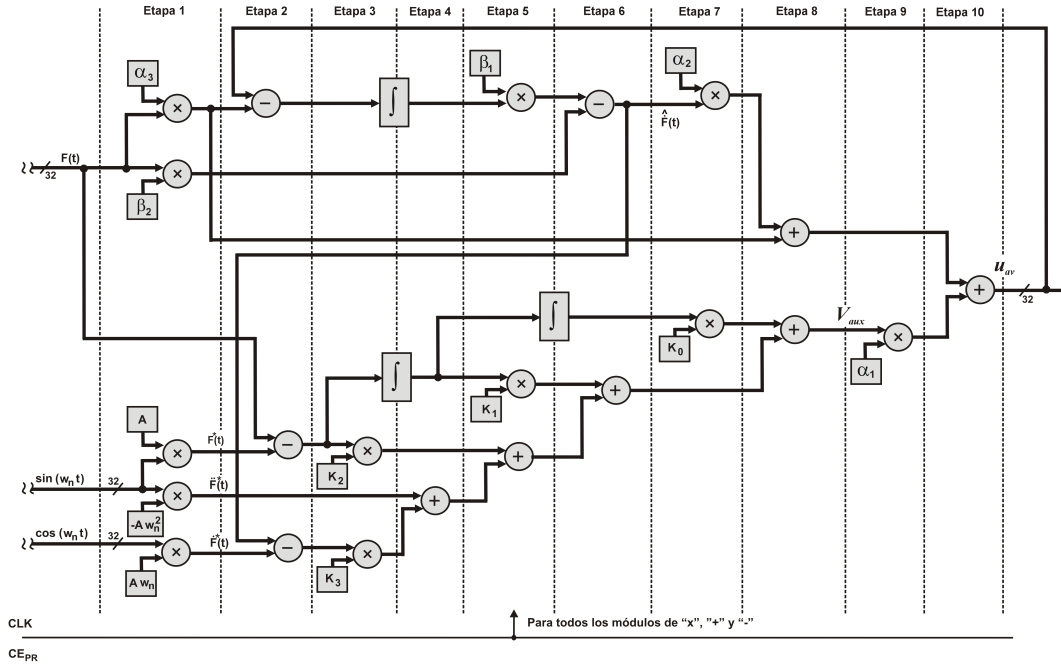


Figura 5.3: Representación por medio de un diagrama de flujo de datos de las ecuaciones (4.32).

La latencia de los multiplicadores, sumadores y restadores generados por la herramienta *Xilinx Core Generator* es de $6T_{CLK}$, $13T_{CLK}$ y $13T_{CLK}$ respectivamente, por lo tanto, el tiempo de propagación t_{CL} generado por los módulos distribuidos de las etapas no segmentadas de

la Figura 5.3 está definido de la siguiente forma:

$$\begin{aligned}
 t_{CL} &= t_{Etapa1} + \dots + t_{Etapa10} & (5.1) \\
 &= 6T_{CLK} + 13T_{CLK} + 6T_{CLK} + 13T_{CLK} + 13T_{CLK} \\
 &\quad + 13T_{CLK} + 6T_{CLK} + 13T_{CLK} + 6T_{CLK} + 13T_{CLK} \\
 &= 102T_{CLK}
 \end{aligned}$$

El análisis del tiempo de propagación en las etapas que integran el controlador, indican que el mayor tiempo de propagación generado es de: $102T_{CLK}$.

Del resultado de la ecuación (5.1) se puede argumentar que los registros de segmentación en las etapas 1 – 10 no son necesarios, ya que el tiempo de propagación dentro de la arquitectura es de $t_{CL} = 2,04\mu s$ (para una señal de reloj de entrada de 50MHz), por lo tanto, basado en la aplicación y en las especificaciones del sistema, el tiempo de muestreo elegido (tiempo entre dato y dato) es de $t_{SP} = 4\mu s$ y el sistema global de reloj (CLK) es fijo de 50MHz⁴.

5.4.1. Administrador digital de reloj

El principal elemento de este módulo es el administrador digital de reloj (DCM, *Digital Clock Manager*). El DCM es un elemento incrustado dentro del dispositivo y que posee la familia de FPGAs *SPARTAN-3E de Xilinx*, este módulo provee de manera flexible un control completo sobre la frecuencia de reloj manteniendo sus características con alto grado de precisión a pesar de las variaciones en operación debido a cambios de voltaje o temperatura.

De la Figura 5.4 la señal CLK es la misma que la señal Clk_{in} , pero el DCM provee la característica de corrección, asegurando una señal de reloj limpia con un 50% de ciclo de trabajo, ya que eliminar el desplazamiento de la señal de reloj es importante para la mayoría de los diseños que funcionan a 50MHz o más.

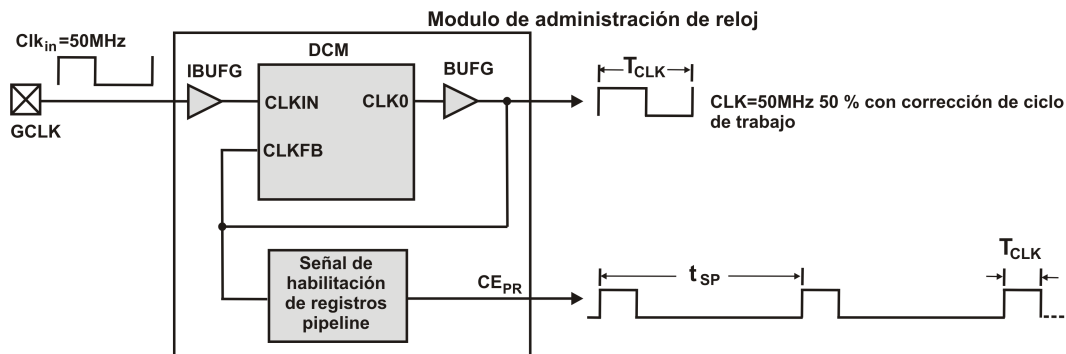


Figura 5.4: Módulo de administración de reloj.

⁴Considerando que los elementos incrustados dentro del FPGA Spartan-3E poseen una frecuencia interna de operación máxima de $230MHz$ ($T_{CLK} = 7,7ns$)[124], es claro notar que el término "tiempo real" es relativo a la aplicación.

Otra característica importante del DCM es que, al estar dentro de la estructura del FPGA, posee una red de distribución de reloj que incluye un circuito digital que no permite el desplazamiento de la señal de reloj y garantiza un tiempo de propagación igual a cero en la distribución de la señal de reloj.

El módulo de administración de reloj, genera el reloj global del sistema para los módulos aritméticos del controlador a partir de la frecuencia principal de $50MHz$ (Clk_{in}) y en adición, también para los registros de pipeline de la señal de habilitación (CE_{PR}) (ver Figura 5.4). La señal de habilitación para los módulos (CE_{PR}) se incluye al módulo DCM, esta señal es generada basada en un contador simple cuya señal de reloj es T_{CLK} . La descripción de este elemento fue modelado a nivel de algoritmo. El ancho de pulso de la señal CE_{PR} es de 1 ciclo de $T_{CLK} = 20ns$; este pulso es generado cada periodo de tiempo $t_{SP} = 4\mu s$ (200 ciclos de reloj T_{CLK}).

5.4.2. Módulo del controlador GPI de seguimiento

Para realizar esta descripción se anexaran dos submódulos (generador de onda senoidal e interfaz de conversión analógico-digital) al diagrama de flujo de datos mostrado en la Figura 5.3, finalmente, el módulo queda en la forma mostrada por la Figura 5.5.

Para la implementación del controlador GPI es necesario obtener la aproximación de las ecuaciones (4.32). Para facilitar esta tarea, estas ecuaciones fueron reescritas como se muestra a continuación:

$$\begin{aligned}
 u_{av}(t) &= \alpha_1 v(t) + \alpha_2 \hat{F}(t) + \alpha_3 F(t) & (5.2) \\
 \hat{F}(t) &= \beta_1 \int_0^t [u_{av}(\tau) - \alpha_3 F(\tau)] d\tau - \beta_2 F(t) \\
 \alpha_1 &= \frac{LC}{E}, \quad \alpha_2 = \frac{L}{RE}, \quad \alpha_3 = \frac{1}{E} \\
 \beta_1 &= \frac{E}{LC}, \quad \beta_2 = \frac{1}{RC}
 \end{aligned}$$

donde:

$$\begin{aligned}
 v(t) &= \ddot{F}^*(t) - k_3 [\hat{F}(t) - \dot{F}^*(t)] - k_2 e(t) \\
 &\quad - k_1 \int_0^t e(\tau) d\tau - k_0 \int_0^t \int_0^\tau e(\lambda) d\lambda d\tau \\
 e(t) &= F(t) - F^*(t) \\
 F^*(t) &= A \sin(\omega_n t), \quad \dot{F}^*(t) = A \omega_n \cos(\omega_n t) \\
 \ddot{F}^*(t) &= -A \omega_n^2 \sin(\omega_n t)
 \end{aligned}$$

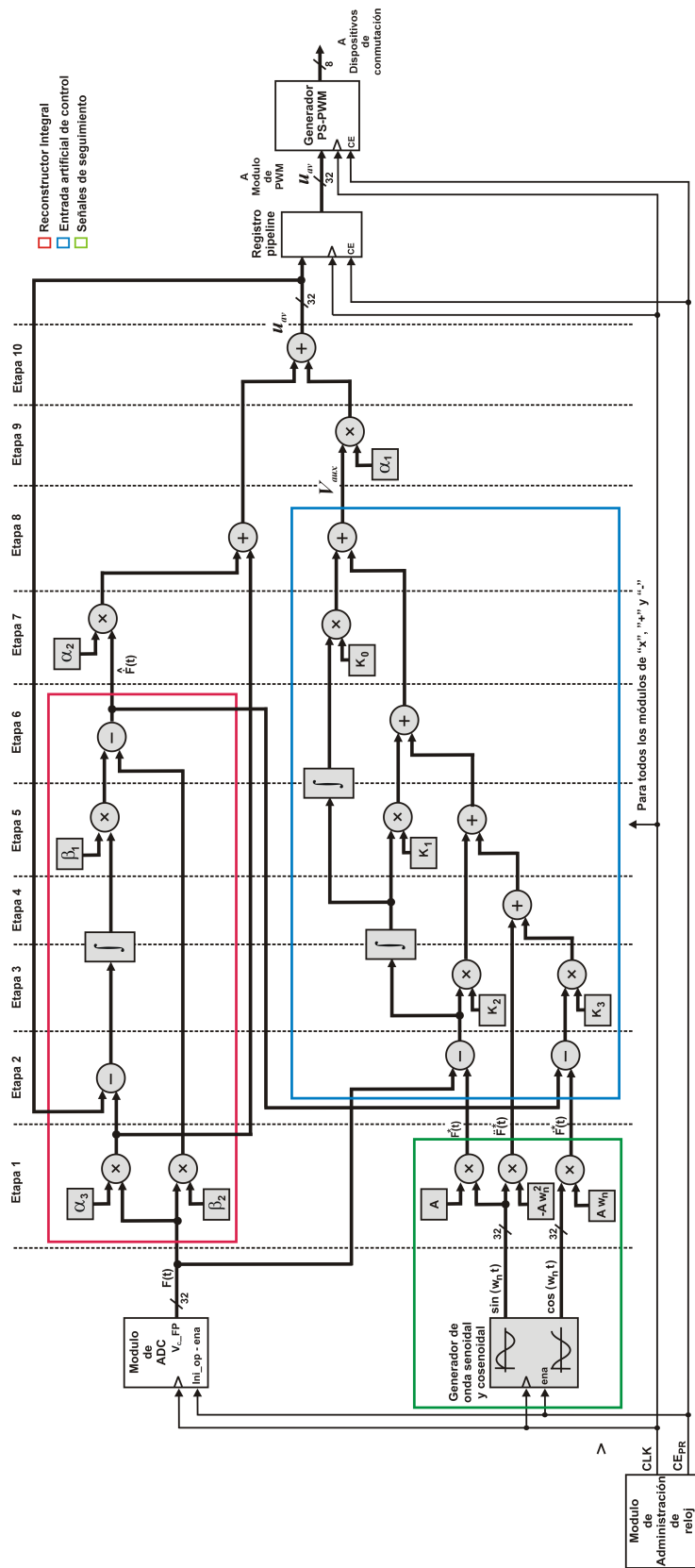


Figura 5.5: Representación por medio de un diagrama de flujo de datos de las ecuaciones (4.32) con la adición de los módulos de generación de onda senoidal, administrador de reloj, conversión analógico-digital y modulador PS-PWM.

Es necesario obtener de la ecuación (5.2) una aproximación discreta de la *integral continua* que aparece en el algoritmo.

El algoritmo de integración que se ha usado para realizar el cálculo de la operación de integración es el *método de integración numérica de Euler* (ver Figura 5.6) [126, 127], este método esta descrito por la siguiente expresión:

$$I(t) = \int_0^t x(\tau) d\tau \approx I[n] = I[n-1] + \Delta t(x[n]) \quad (5.3)$$

donde Δt recibe el nombre de paso de integración⁵.

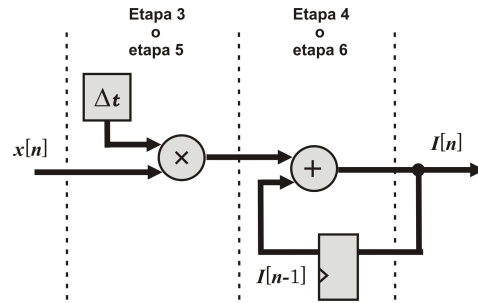


Figura 5.6: Representación gráfica de la ecuación (5.3).

Partiendo de la ecuación (5.3) la integral $I_{est}(t) = \int_0^t (u_{av}(t-1) - \alpha_3 y(t)) dt$ usando el método de Euler esta definida por:

$$I_{est}[n] = I_{est}[n-1] + \Delta t(x[n]) \quad (5.4)$$

donde, $x[n] = u_{av}[n-1] - \alpha_3 y[n]$.

Para el caso de la discretización aproximada de las ecuaciones $I_e(t) = \int_0^t e(t) dt$ y $II_e(t) = \int_0^t \int_0^\tau e(t) dt d\tau$ está dada por las expresiones (5.5) y (5.6) respectivamente.

$$I_e[n] = I_e[n-1] + \Delta t(e[n]) \quad (5.5)$$

$$II_e[n] = II_e[n-1] + \Delta t(I_e[n]) \quad (5.6)$$

donde $e[n] = y[n] - y^*[n]$.

La aproximación discreta para el algoritmo presentado en las ecuaciones (5.2) se obtiene al sustituir las ecuaciones (5.5) y (5.6) está definido como:

$$u_{av}[n+1] = \alpha_1 v[n] + \alpha_2 \hat{F}[n] + \alpha_3 F[n] \quad (5.7)$$

⁵En el algoritmo de la ecuación (5.3) es el intervalo de tiempo de t_n a t_{n+1}

$$\begin{aligned}
v_{aux}[n] &= \ddot{F}^*[n] - k_3(\hat{F}[n] - \dot{F}^*[n]) - k_2e_v[n] \\
&\quad - k_1I_{ev}[n] - k_0II_{ev}[n] \\
e_v[n] &= F[n] - F^*[n] \\
\hat{F}[n] &= \beta_1I_{est}[n] - \beta_3F[n]
\end{aligned}$$

La ley de control será implementada basada en las ecuaciones (5.7).

En la Figura 5.5 el controlador recibe datos de los módulos de la interfaz de conversión analógico-digital y del generador de onda senoidal, ambos módulos tienen registros de segmentación a la entrada. Los datos generados por el controlador son enviados al módulo de generación de *PWM* a través de un registro de segmentación de salida.

La arquitectura propuesta se divide en diez etapas no segmentadas, en la estructura se busca explotar la independencia que existe entre las operaciones que integran las expresiones (5.7), por lo que en cada etapa de la arquitectura propuesta varios elementos trabajan de manera concurrente; la característica de ejecución de procesos de manera concurrente es intrínseca al dispositivo que fue elegido, esta propiedad permite obtener altas velocidades de procesamiento, sin embargo, generalmente lleva a usar mayor cantidad de recursos de implementación.

La arquitectura propuesta consiste de quince multiplicadores, nueve sumadores y cuatro restadores. Como ya se había mencionado, la herramienta *EDA Xilinx ISE Design suite 9.2* incluye una herramienta que genera IP CORES ⁶.

Los módulos de aritmética del controlador son todos de punto flotante de precisión simple (32 bits), apegados al estándar *IEEE – 754* y fueron generados con la herramienta antes mencionada.

Módulo del generador de onda senoidal y cosenoidal

Un aspecto importante de la discretización de las ecuaciones (5.2) es haber obtenido una aproximación discreta de las funciones $\sin\omega_n t$ y $\cos\omega_n t$ usando el formato en punto flotante de precisión simple; ambas funciones poseen una frecuencia de $60Hz$ ($t_{señal} = \frac{1}{60} = 16,6666ms$) y una amplitud pico a pico de ± 1 .

Dentro la arquitectura tratada, esta función la desarrolla el submódulo llamado generador de onda senoidal y cosenoidal.

Basado en el periodo de muestreo $t_{SP} = 4\mu s$, ambas funciones ($\sin\omega_n t$ y $\cos\omega_n t$) fueron aproximadas usando 4096 puntos ($\frac{t_{señal}}{t_{SP}} \approx 4096$).

Una forma eficiente de implementar estas funciones es almacenar sus valores precalculados en memorias ROM (basadas en la BRAM del dispositivo reconfigurable).

Aprovechando la simetría de las funciones en cuestión, sólo un segmento fue calculado y alma-

⁶Un *IPCORE* (*Intellectual Property Core*) es un bloque de lógica o de datos que es usado en el ambiente de los FPGAs o de los ASICs; en esencia es un elemento que se puede reutilizar durante un diseño (el IP CORE debe de ser enteramente portable para cualquier marca de dispositivo lógico programable), pero que previamente fue diseñado; existen tres categorías: *Hard Cores*, *Firm Cores* y *los Soft Cores*.

cenado en las memorias BRAM internas del dispositivo, estas memorias fueron configuradas por medio del programa *IP CORE* de *EDA Xilinx ISE Design* como memorias ROM. Para la función $\sin\omega_n t$, solo el segmento $[0, \pi/2]$ fue calculado y el resto de los segmentos está definido como:

$$\begin{aligned} \sin(\pi/2 + n)_{(\pi/2, \pi]} &= \sin(\pi/2 - n)_{[0, \pi/2]} \\ \sin(x)_{(\pi, 3\pi/2]} &= -\sin(x)_{[0, \pi/2]} \\ \sin(3\pi/2 + n)_{(3\pi/2, 2\pi]} &= -\sin(\pi/2 - n)_{[0, \pi/2]} \end{aligned} \quad (5.8)$$

donde $n \in (0, \pi/2)$.

La Figura 5.7 muestra el diagrama de bloques del submódulo de generación de señales. Este

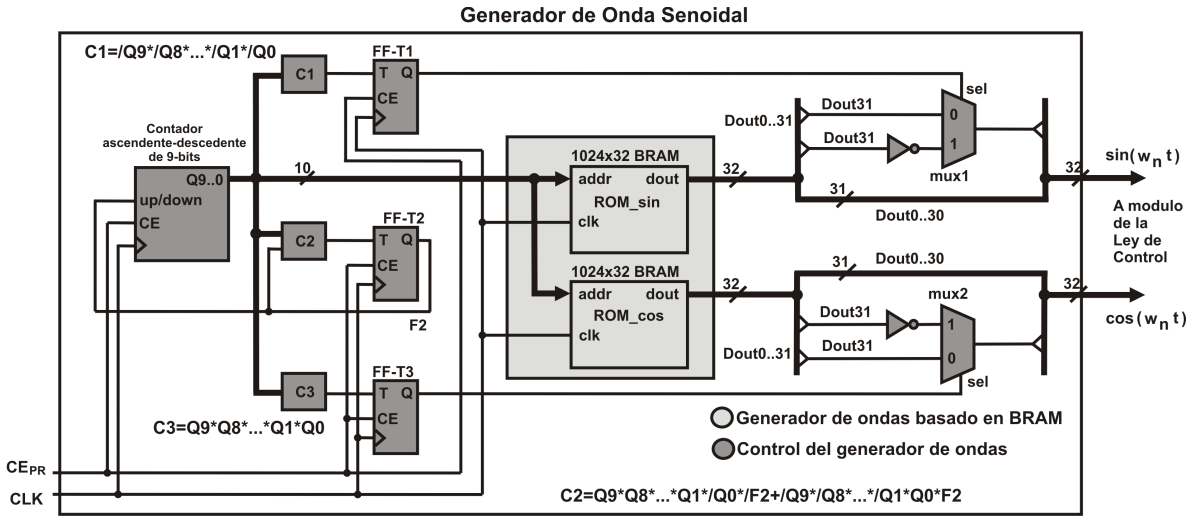


Figura 5.7: Diagrama de bloques del submódulo del generador de onda senoidal y cosenoidal.

submódulo incluye dos memorias ROM (ROM_{sin} y ROM_{cos}), configuradas para almacenar 1024 palabras de 32 bits, cada memoria ocupa dos memorias BRAM del FPGA.

La memoria llamada ROM_{sin} almacena los valores generados por la función $\sin\omega_n t$ la cual fue evaluada cada $4\mu s$ en el intervalo $[0, \pi/2]$. De la Figura 5.7, el contador ascendente - descendente de 9 bits genera la dirección para leer el dato correspondiente de la memoria ROM_{sin} ; $C2$ y $FF-T2$ controlan la secuencia de conteo, $C1$ y $FF-T1$ controlan la generación del segmento negativo de la función $\sin\omega_n t$ (ver también la Figura 5.8).

La generación de la señal senoidal completa en el periodo $[0, \pi/2]$ se da en cuatro tiempos (ver Figura 5.8):

1. De la Figura 5.8 se puede observar que inicialmente el contador inicia con una cuenta ascendente y los *flip-flops* de salida poseen una salida lógica igual a "0", por tanto el segmento $[0, \pi/2]$ de la función $\sin\omega_n t$ es generado cada tiempo de muestreo (t_{SP}); cuando el contador llega al máximo valor permitido (el binario de 1023) pasa al estado

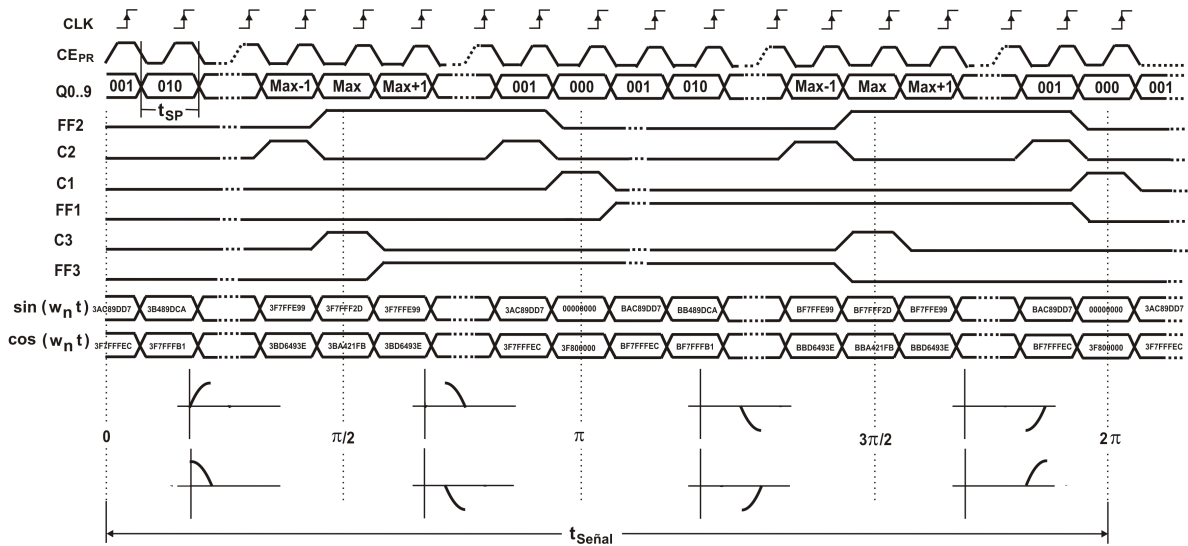


Figura 5.8: Diagrama de tiempos para la generación de las ondas seno y coseno.

siguiente.

2. Cuando se llega al conteo máximo del punto anterior, $C2$ produce una salida lógica igual a "1", lo que produce que $FF - T2$ cambie de estado, y el contador cambia a modo descendente, cuestión que genera el segmento de $(\pi/2, \pi]$ de la función $\sin \omega_n t$.
3. Cuando el contador nuevamente llega al mínimo valor de conteo, $C2$ retorna nuevamente a "0", lo que produce que la señal $FF - T2$ cambie de estado nuevamente a modo ascendente, en esta situación $C1$ produce un cambio en la acción de la señal $FF - T1$ la cual controla el multiplexor ($mux1$) que está encargado de cambiar el signo de la función $\sin \omega_n t$ y por lo tanto se genera el segmento $(\pi, 3\pi/2]$ de la función.
4. Finalmente, cuando el contador llega nuevamente al máximo valor, $C2$ produce un "1" lógico lo cual produce un cambio en la acción de $FF - T2$ y se cambia al modo de conteo descendente y el segmento $(3\pi/2, 2\pi]$ de la función $\sin \omega_n t$ se genera.

La secuencia se repite de manera indefinida.

Aplicando el mismo criterio para la función $\cos \omega_n t$, solo el segmento $[0, \pi/2]$ es precalculado y almacenado en las BRAM (ROM_{\cos}) del dispositivo, el resto de los segmentos son definidos por las siguientes ecuaciones:

$$\begin{aligned}
 \cos(\pi/2 + n)_{(\pi/2, \pi]} &= -\cos(\pi/2 - n)_{[0, \pi/2]} & (5.9) \\
 \cos(x)_{(\pi, 3\pi/2]} &= -\cos(x)_{[0, \pi/2]} \\
 \cos(3\pi/2 + n)_{(3\pi/2, 2\pi]} &= \cos(\pi/2 - n)_{[0, \pi/2]}
 \end{aligned}$$

Interfaz de conversión analógico-digital

La información que se genera por el sensor de voltaje en el convertidor multinivel en cascada de cinco niveles con fuentes independientes es una señal continua en el tiempo (analógica), por lo tanto se debe discretizar para que el FPGA pueda procesarla.

Con este propósito, se empleó el módulo interno de conversión analógico-digital que posee el microcontrolador de la casa *Microchip* dsPIC30F2010 [128].

La compañía Microchip tiene una familia de DSPs denominados dsPIC, que también llama DSC (**DSC**, *Digital Signal Controller*), este dispositivo es un híbrido entre un microcontrolador y un DPS de gama baja.

Microchip divide en dos gamas sus DSC:

- **dsPIC30F**: Gama básica, con las siguientes características:
 - 16 bits, 30 MIPS⁷.
 - 28 a 80 pines.
 - Hasta 144KB de memoria de programación, 8KB de RAM y 4KB de EEPROM.
 - Convertidores *A/D* de hasta 500kbps y de 10 a 12 bits por canal (un dsPIC posee hasta 16 canales de captura *A/D*).

- **dsPIC33F**: Gama alta, con las siguientes características:
 - 16 bits, 40 MIPS.
 - 28 a 100 pines.
 - Hasta 256KB de memoria de programación, 30KB de RAM.
 - Convertidores *A/D* de hasta 2,2Mbps de 12 bits por canal (un dsPIC posee hasta 32 canales de captura).
 - Controlador de *DMA* (8 canales). *A/D*

De manera específica, el dsPIC30F2010 cuenta con las siguientes características:

- 16 bits, 30 MIPS.
- 28 pines.
- Hasta 12KB de memoria de programación, 512B de RAM y 1KB de EEPROM.
- 6 Convertidores de 10 bits por canal (con un tiempo de adquisición de 25μs).

Este microcontrolador se ha seleccionado por las siguientes razones:

⁷MIPS es el acrónimo de "millones de instrucciones por segundo", es una forma de medir la potencia de los procesadores, sin embargo, esta medida sólo es útil para comparar procesadores con el mismo juego de instrucciones.

- Únicamente se empleara al dsPIC como convertidor analógico-digital.
- Aprovechar que es reprogramable (la programación se realiza en *lenguaje C*).
- Se empleara sólo un canal de conversión.
- El resultado de la conversión se envía al FPGA de manera paralela (bus de 10 bits).
- Mejor control y manejo de la conversión analógico-digital debido a que es programable.
- El tiempo de conversión que ofrece el dispositivo no resulta ser crítico en la aplicación.

Como ya se menciona el convertidor analógico-digital del dsPIC30F2010 tiene una resolución de 10 bits, un tiempo de conversión de $25\mu s$ (t_{CONV}) y tiene un rango de voltaje de entrada de 0 a 5 volts.

El módulo de la interfaz de conversión analógico-digital está compuesta por dos submódulos:

- **Control del convertidor analógico-digital:** Implementa el protocolo para realizar la adquisición vía bus de 10 bits del dsPIC30F2010 hacia el FPGA.
- **Encoder de punto flotante:** Convierte el valor entero de 10 bits en un valor en formato de punto flotante de precisión simple (32 bits).

El diagrama a bloques del módulo de la interfaz de conversión analógico-digital se muestra en la Figura 5.9. A continuación se describe cada uno de los submódulos que forman parte del

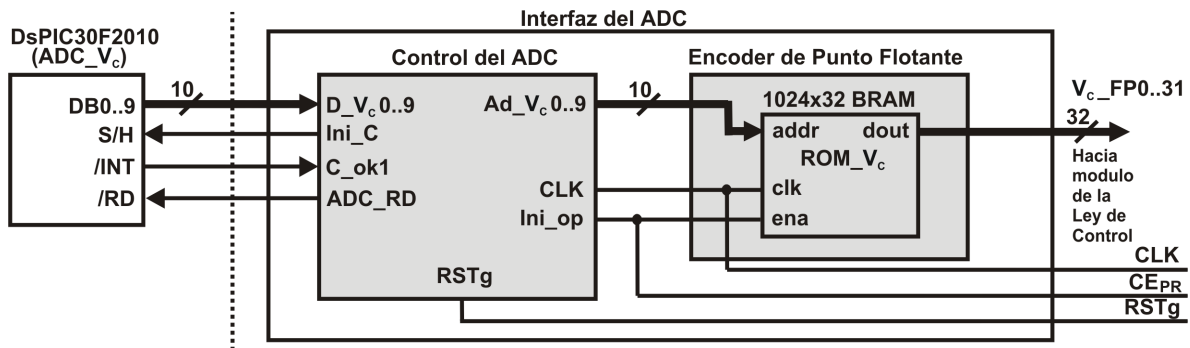


Figura 5.9: Diagrama a bloques de la interfaz de conversión analógico-digital.

módulo de conversión analógico-digital:

Control del convertidor analógico-digital Este submódulo es una sencilla maquina de estados finitos (FSM, *Finite-State Machine*) que implementa el protocolo de comunicación entre el dsPIC30F2010 y el FPGA (ver Figura 5.10).

Cada periodo de tiempo que la señal del habilitador (Ini_C) se pone a "1" lógico, este submódulo inicia al dsPIC30F2010 la adquisición de un nuevo dato.

A continuación se describe su operación auxiliado del diagrama de tiempos mostrado en la Figura 5.10:

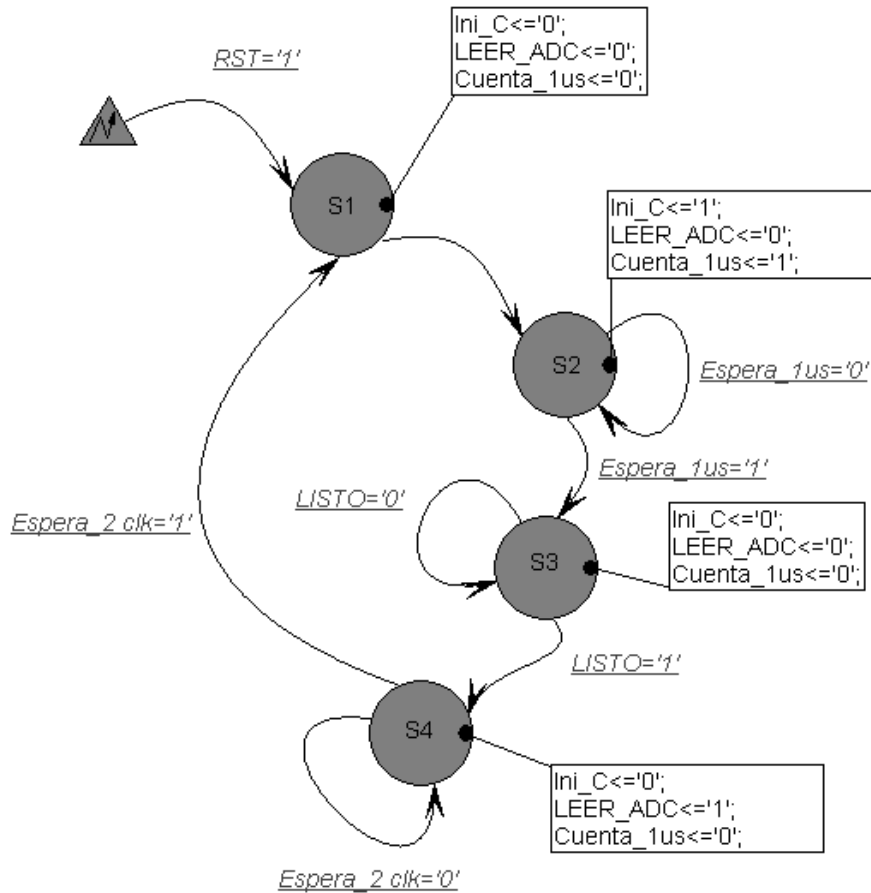


Figura 5.10: Máquina de estados finitos para control de la interfaz del ADC.

- **Estado 1:** Es el estado de reset o de inicio de la máquina de control.
- **Estado 2:** En este estado se indica al dsPIC30F2010 que de inicio con la adquisición del canal analógico e inicie también la conversión, se puede observar en la Figura 5.11 que en este estado, la señal (*Ini_C*) es puesta a nivel lógico "1" durante un mínimo de $1\mu s$, esta temporización es llevada por un contador (*Cuenta_{1us}*) que es habilitado al mismo tiempo que (*Ini_C*), el contador en cuestión coloca un "1" lógico por ($1 T_{CLK}$) una vez que llega la cuenta establecida ($50 T_{CLK}$), debido a este cambio la máquina de control pasa al estado 3.
- **Estado 3:** En este estado, la máquina de control espera el pulso (*Listo*) que es enviado por el dsPIC cuando ya se realizó la conversión y el dato está disponible para ser leído en el bus de $10bits$ (*Adv_C*), pasando ahora al estado 4.
- **Estado 4:** En este estado se respalda por medio de la señal (*Leer_{ADC}*) el valor del bus de 10 bits, el *Dato anterior* se reemplaza por el *Dato nuevo*, la señal (*Leer_{ADC}*) se

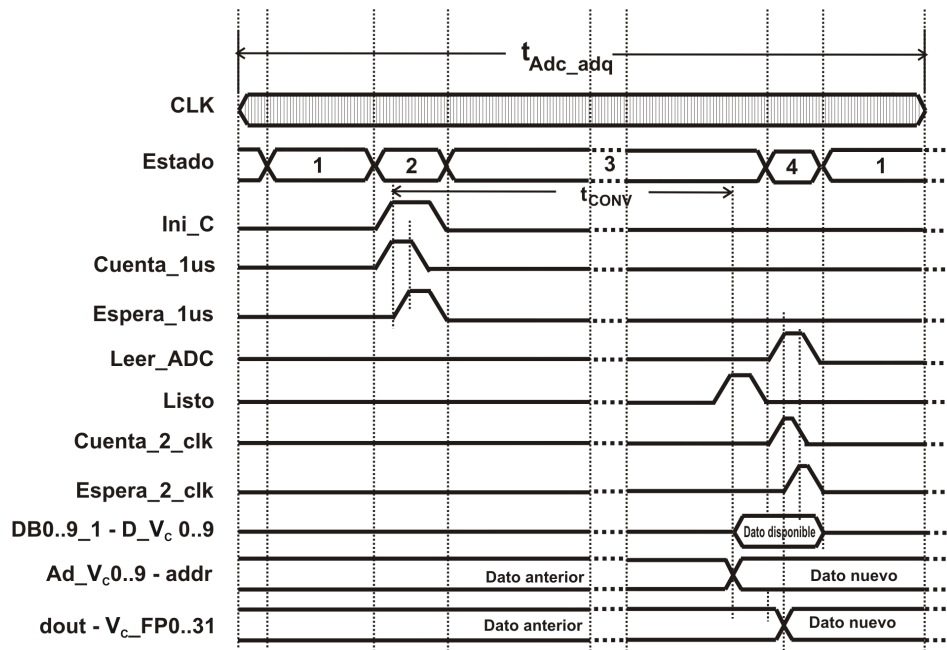


Figura 5.11: Diagrama de tiempos de la adquisición de la interfaz del ADC.

mantiene en alto $2 T_{CLK}$ para garantizar el respaldo del nuevo dato, esta temporización es llevada por un contador ($Cuenta_{2CLK}$) cuyo conteo es activado al mismo tiempo que ($Leer_{ADC}$), una vez llegada a la cuenta ($2 T_{CLK}$), el contador pone un "1" lógico durante ($1 T_{CLK}$), dado que el valor del bus (Ad_{vC}) es la dirección que apunta a los datos en la memoria BRAM del *submódulo Encoder de punto flotante* (ver Figura 5.9); ahora también se tiene un nuevo dato en el bus de 32 bits ($dout_{VcFP}$); después de este estado la maquina regresa al estado de reset (*estado 1*).

Encoder de punto flotante Este submódulo convierte el valor integral leído del bus de datos de 10 bits (Ad_{vC}) a un valor en formato de punto flotante de precisión simple (32 bits). Existen dos factores que influyen en la conversión:

1. **La resolución de del convertidor**, esta viene dada por la siguiente expresión:

$$RES_{ADC} = \frac{V_{ref}}{2^{bits_{ADC}}} = \frac{5Volts}{1024} = 0,0048828125 Volts \quad (5.10)$$

2. **Acondicionamiento de las señales:** El convertidor multinivel es capaz de generar un salida en CA de -160 a 160 Volts, pero el dsPIC30F2010 puede leer sólo un voltaje entre 0 y 5 volts, por lo tanto es necesario contar con una etapa de acondicionamiento de voltaje, que puede estar basada en el tratamiento de la señal por medio de amplificadores operacionales⁸ (ver Figura 5.12)

⁸El término de amplificador operacional (*operational amplifier, OA, o op amp*) fue asignado alrededor de

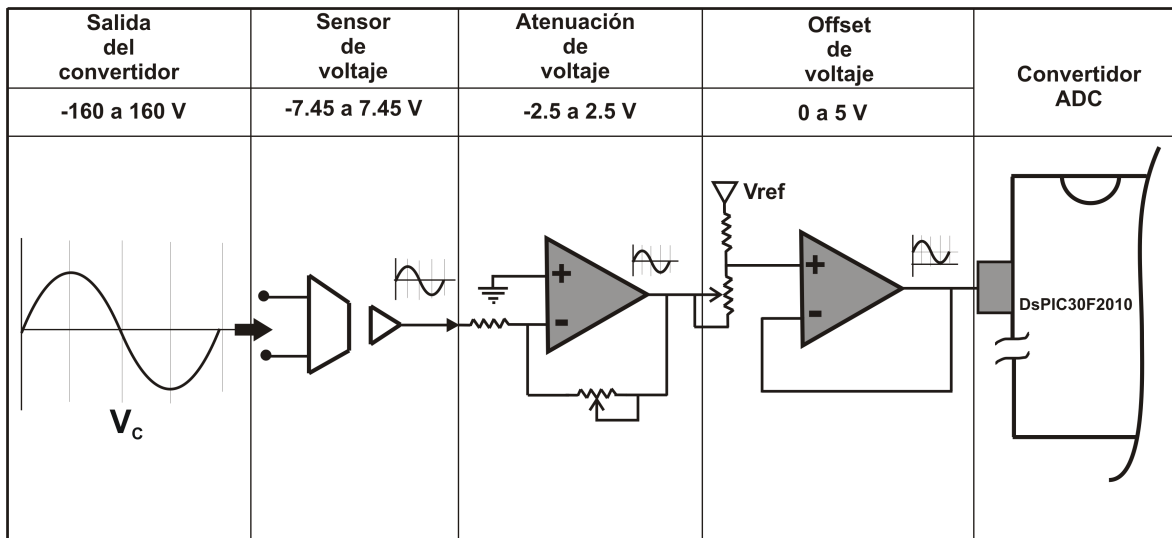


Figura 5.12: Acondicionamiento de v_C para que pueda ser adquirido por el convertidor ADC del microcontrolador dsPIC30F2010.

Partiendo de los datos anteriores, el conjunto de sensor de voltaje-adequación vía *op amp* deberá de entregar una salida entre 0 y 5 volts para poder ser convertidas por el dsPIC30F2010. El acondicionamiento de la salida del convertidor (v_C) incluye un factor de atenuación de 64 V⁹ y un corrimiento de voltaje sobre 0 volts (*offset*) de 2,5 Volts (v_{Coff}) (ver Figura 5.12). Considerando estos factores, la conversión del voltaje medido v_C a punto flotante de precisión simple dentro del FPGA está definido por:

$$\begin{aligned}
 v_{C_FP} &= ((Ad_{v_C})(RES_{ADC}) - v_{Coff})(v_{Catt}) \\
 &= (Ad_{v_C})(0,3125) - 160
 \end{aligned}
 \tag{5.11}$$

Para aplicaciones que requieren memorias grandes en capacidad de almacenamiento, la familia de FPGAs *SPARTAN-3E* de *Xilinx* provee bloques eficientes de memoria dentro de su estructura (BRAM).

La implementación del *Encoder de punto flotante* está basada en bloques internos de BRAM del FPGA, las cuales tienen una capacidad máxima de 18,432 bits y pueden trabajar a una velocidad de salida arriba de 250MHz para todos los anchos de palabra y una latencia de lectura de 2,45ns [124].

La herramienta *Xilinx CORE Generator* también es un avanzado constructor de memorias personalizables que genera memorias optimizadas en área y desempeño usando los recursos de BRAMs en los FPGAs de *Xilinx* (ver Figura 5.13).

El submódulo de *Encoder de punto flotante* consiste en dos BRAMs, organizadas para crear

1940 para designar una clase de amplificadores que permiten realizar una serie de operaciones tales como: suma, resta, multiplicación, integración, diferenciación, etc., importantes dentro de la computación analógica de esa época

⁹Se obtiene al dividir $v_{Catt} = \frac{v_C}{v_{Coff}} = \frac{160}{2,5} = 64$ Volts

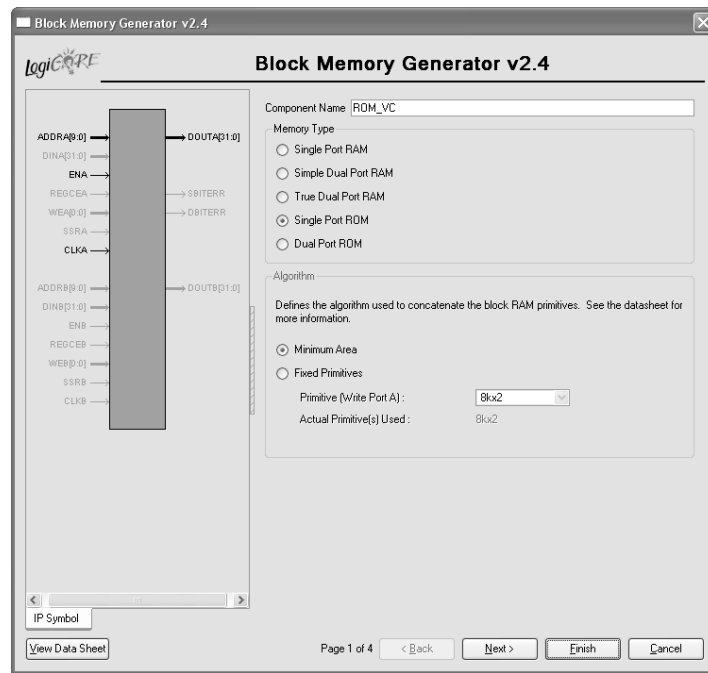


Figura 5.13: Ventana del generador de memorias del *Xilinx CORE Generator*.

una memoria ROM de 1024 palabras de 32 bits, para este diseño, esta memoria recibe el nombre de ROM_{vC} , esta contiene la información necesaria para convertir los valores de v_C al formato de punto flotante de precisión simple.

El dato adquirido de v_C por el dsPIC30F2010 (Ad_{vC}) representa la dirección de la memoria ROM_{vC} , el valor de salida de la memoria ($dout_{vCFP}$) está determinado por (Ad_{vC}).

Los datos almacenados en la memoria ROM_{vC} fueron generados basados en la ecuación (5.11).

En la tabla 5.1 se muestra de manera resumida el contenido de esta memoria, y en el apéndice D se muestra el código del contenido de la memoria empleado por el *Xilinx Core Generator*.

5.4.3. Generador PWM para el Modulador Multinivel

El esquema *PWM* implementado para este diseño es un *PWM* senoidal con corrimiento de fase (PS-PWM, *Phase-Shifted Pulse-Width modulation*) con portadoras múltiples.

Este módulo recibe la señal u_{av} del bloque de la ley de control y genera ocho salidas *PWM*: **A1, A2, B1, B2, C1, C2, D1, D2**, estas señales son usadas para el control de conmutación de los *MOSFETs* de potencia del convertidor multinivel.

Basado en los requerimiento de diseño de la tabla 4.3 referentes al controlador de seguimiento y el filtro de salida, se determino que la frecuencia del *PWM* sera de $F_{PWM} = 1000Hz$.

El generador *PWM* está compuesto por varios bloques funcionales (Ver Figura 5.14), la descripción de cada uno de ellos se da en seguida.

Tabla 5.1: Contenido de la ROM_{v_C}

Contenido de la BRAM de v_C		
Muestra	$v_C(t)$	Hex
0	160	42A40000
1	159.6875	42A3AE00
2	159.375	42A35C00
...
511	0.3125	3E240000
512	0	00000000
513	-0.3125	BE240000
...
1022	-159.375	C2A35C00
1023	-159.6875	C2A3AE00

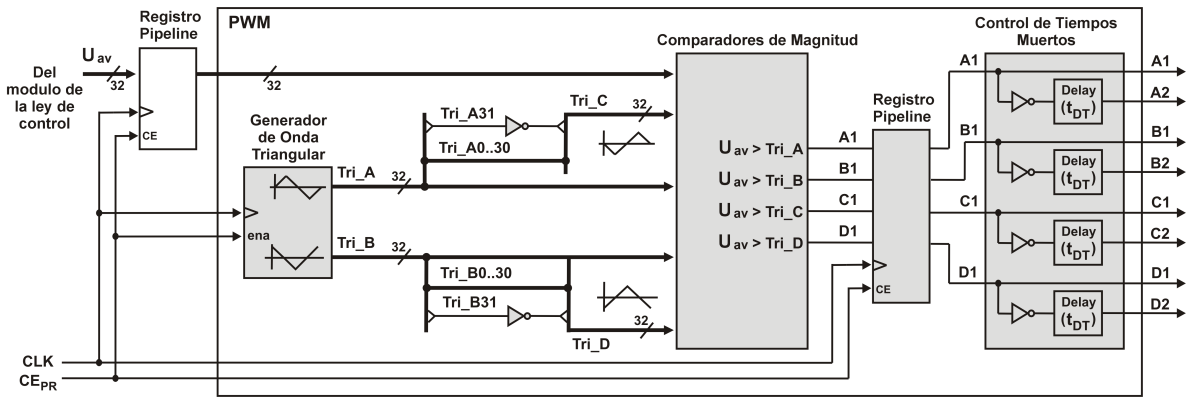


Figura 5.14: Diagrama de bloques del generador PS-PWM.

Generador de Onda triangular

En este bloque se genera una aproximación discreta de dos ondas triangulares con una amplitud pico a pico de ± 1 (definidas como Tri_A y Tri_B), estas ondas triangulares tienen un desfase de 90° , para la discretización de ambas ondas triangulares se usa el formato de punto flotante de precisión simple.

El generador de onda triangular fue diseñado empleando el mismo principio usado en la discretización de las $\sin\omega_n t$ y $\cos\omega_n t$; se sabe que el convertidor posee cinco niveles, por tanto el número de portadoras necesarias es de $n_p = m - 1 = 4$, donde m representa el número de niveles (ver Ecuación 1.2).

Considerando la simetría de las ondas triangulares, sólo el segmento $[0, \pi/2]$ es calculado y almacenado en las BRAMs del FPGA¹⁰, los otros segmentos son generados basados en el segmento ya almacenado.

¹⁰En este caso también la memoria ROM generada tiene un tamaño de 1024 palabras de 32 bits

Si se invierte el bit más significativo de las señales Tri_A y Tri_B , se pueden generar dos nuevas señales triangulares (Tri_C y Tri_D) con las mismas características que las originales (Ver Figura 5.14), pero desfasadas 90° ; esta característica es factible dado que en el formato de punto flotante de precisión simple el bit 32 (bit más significativo) es el encargado de representar el signo.

Comparadores de magnitud

Una vez que ya se tienen las cuatro portadoras (Tri_A , Tri_B , Tri_C y Tri_D) y la señal de control u_{av} , éstas entran al bloque de comparadores de magnitud, este bloque está compuesto de cuatro comparadores de 32 bits, dentro de éste módulo se realiza una comparación de magnitud, entre la señal de control obtenida (u_{av}) y cada una de las triangulares (Tri_A , Tri_B , Tri_C y Tri_D) por medio de las siguientes operaciones: $A1 = u_{av} > Tri_A$, $B1 = u_{av} > Tri_B$, $C1 = u_{av} > Tri_C$, and $D1 = u_{av} > Tri_D$; finalmente se generar cuatro señales de PWM.

Control de tiempos muertos

Este bloque genera otras cuatro señales de *PWM* de la siguiente manera: $A2 = \overline{A1}$, $B2 = \overline{B1}$, $C2 = \overline{C1}$ y $D2 = \overline{D1}$. Además, este bloque añade un retardo entre las señales $A1$, $A2$; $B1$, $B2$; $C1$, $C2$; $D1$, $D2$, este retardo es llamado "tiempo muerto" y tiene la finalidad de evitar un corto circuito entre los dispositivos de conmutación complementarios; la duración del retardo fue calculado tomando en cuenta las características del dispositivo de conmutación empleado (*IRF640*) y esta definido por la siguiente expresión:

$$t_{DT} = t_{MOSFET_{ON}} + t_{MOSFET_{OFF}} = 17ns + 27ns = 44ns \quad (5.12)$$

La Figura 5.15 muestra los detalles de las señales que se esperan generar por el módulo PS-PWM.

5.5. Codificación HDL e implementación en FPGA

La implementación de los módulos y de la arquitectura obtenida en el dispositivo digital (FPGA) agrega algunos pasos adicionales al proceso de diseño que se está siguiendo en este trabajo (ver Figura 5.16).

Los pasos mostrados en la Figura 5.16 son usados comúnmente por las herramientas de desarrollo empleadas en diseños para FPGAs ,por ejemplo: *ISE*, *Quartus*, *Liberio* que son proporcionadas por los fabricantes de FPGAs tales como *Xilinx*, *Altera*, *Actel* (ver [29, 30, 129, 130]).

En este trabajo, la descripción de cada módulo fue realizada en *VHDL*.

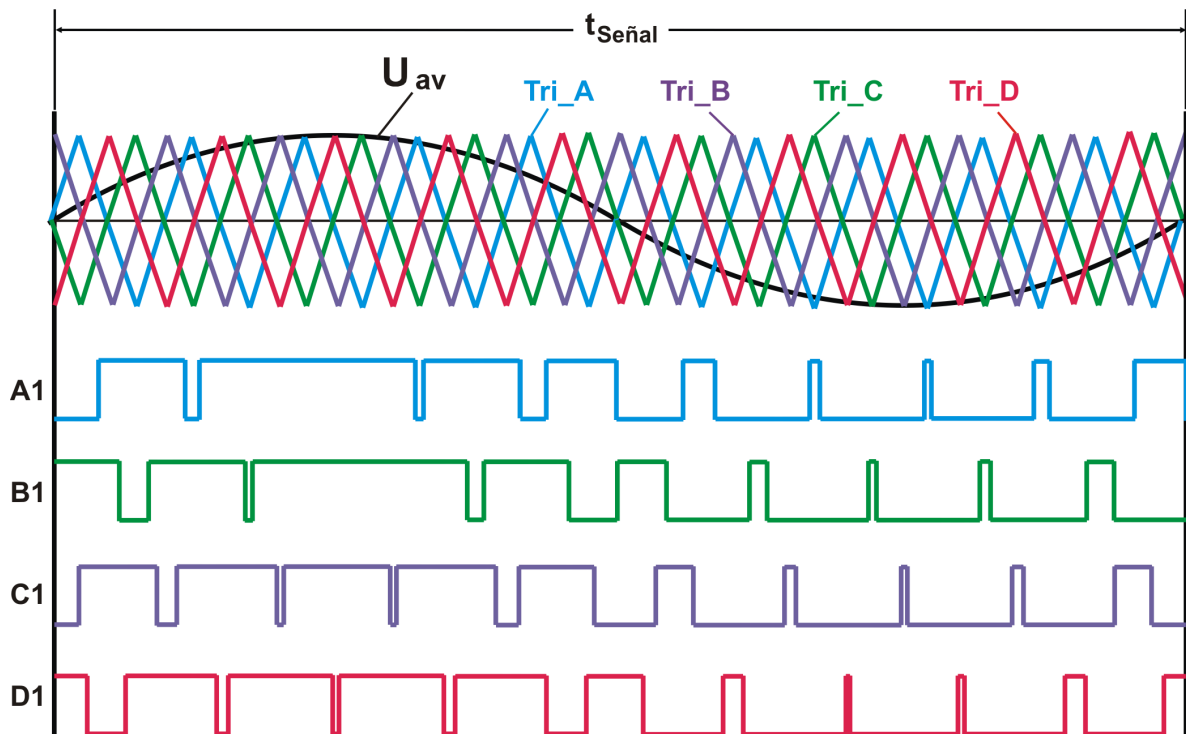


Figura 5.15: Señales de salida esperadas del Modulador Multinivel PS-PWM.

5.5.1. Reportes obtenidos de la implementación del controlador

Debido a la gran cantidad de código generado en cada módulo que forma parte del controlador de seguimiento, sólo se presentará el resumen de los recursos utilizados del FPGA (ver tabla 5.2), el reporte de la etapa de colocación y ruteo (ver Figura 5.17), así como también el reporte final después de la creación del archivo de configuración generado por la herramienta *Xilinx ISE Design Suite 9.2* (ver Figura 5.18).

En el apéndice E se presentan los diagramas conceptuales que fueron construidos desde un enfoque *top-down* con la Herramienta de *Xilinx ISE Design Suite 9.2*.

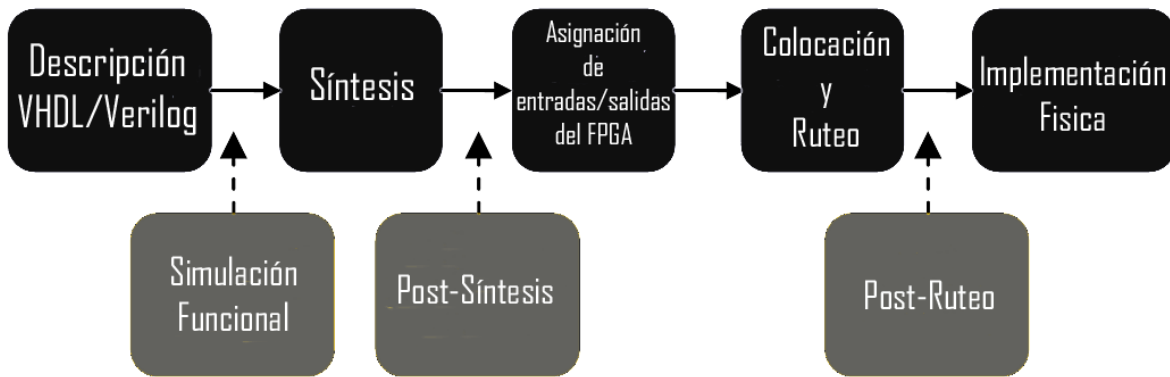


Figura 5.16: Proceso de implementación en FPGA.

5.6. Validación Experimental

En esta etapa se integró el convertidor multinivel en cascada de fuentes independientes de cinco niveles con el diseño del controlador de seguimiento implementado en el FPGA (ver Figura 5.1) y se realizaron tres tipos de pruebas:

1. Comportamiento en estado estacionario.
 - Comportamiento del convertidor en lazo abierto (sin controlador **GPI**).
 - Comportamiento del convertidor en lazo cerrado (con el controlador **GPI**).
2. Respuesta del sistema ante un transitorio provocado por un cambio de carga lineal.
 - Comportamiento del convertidor en lazo abierto.
 - Comportamiento del convertidor en lazo cerrado.
3. Respuesta del sistema ante un transitorio provocado por un cambio de carga no lineal (carga mixta).
 - Comportamiento del convertidor en lazo abierto.
 - Comportamiento del convertidor en lazo cerrado.

Para verificar el comportamiento del sistema se identificaron las siguientes variables:

- *Voltaje de salida* (v_C).
- *corriente de salida* (i_L).
- **THD** (*Distorsión armónica total*) de corriente y de voltaje.
- *factor de potencia* (**PF**).

Tabla 5.2: Resumen de recursos usados del FPGA en la implementación del controlador **GPI** de seguimiento.

Resumen de recursos usados por módulo					
Módulos	<i>Slices</i>	<i>Flip Flops</i>	<i>4 Inputs LUTs</i>	FPGA <i>Emb. elem.</i>	<i>Max. Frec. Mhz</i>
Administración de reloj	13 < 1 %	11 < 1 %	25 < 1 %	1(<i>DCM</i>) 12 %	198.6
Generador de onda seno y coseno	12 < 1 %	13 < 1 %	23 < 1 %	4 <i>BRAMs</i> 14 %	133.6
Generador de PWM	200 2 %	90 < 1 %	372 2 %	4 <i>BRAMs</i> 14 %	194.7
Módulo de interfaz del ADC	29 < 1 %	47 < 1 %	42 < 1 %	4 <i>BRAMs</i> 14 %	246.5
Integrador	29 < 1 %	47 < 1 %	42 < 1 %	4 <i>BRAMs</i> 14 %	246.5
Recursos usados por elemento aritmético independiente (32 bits)					
Sumador	417 4 %	604 3 %	583 3 %	-	178.5
Multiplicador (Emb. FPGA)	186 2 %	294 1 %	182 1 %	4 <i>Mult18x18</i> 14 %	161.8
Multiplicador (lógica)	402 4 %	701 4 %	643 3 %	-	161.9
Comparador de magnitud	47 < 1 %	16 < 1 %	84 < 1 %	-	218.0
Resumen total de recursos empleados					
Controlador GPI	8672 92 %	12639 72 %	15393 88 %	28 <i>Mult18x18</i> 100 %	83.4

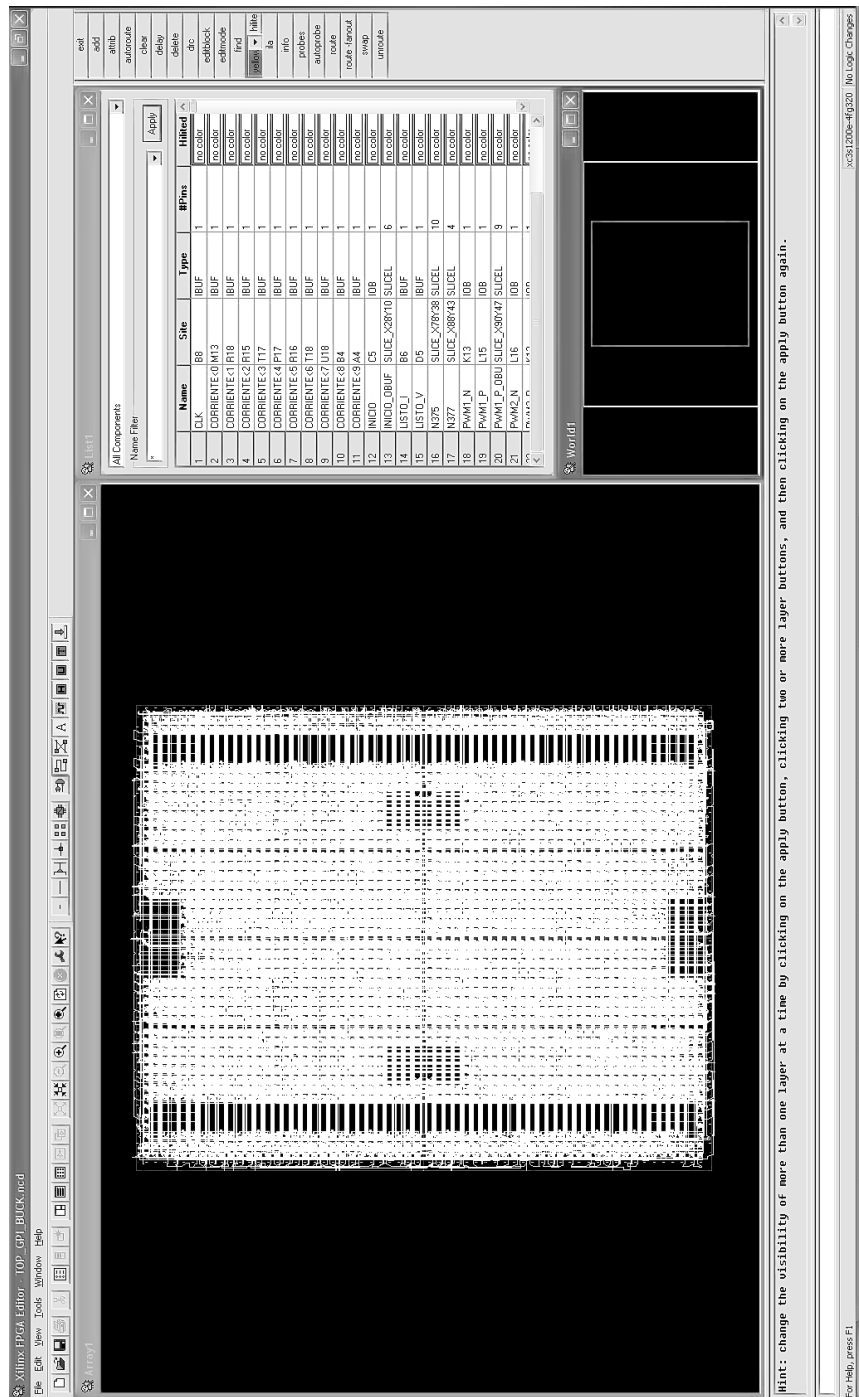


Figura 5.17: Reporte de la colocación y ruteo generado por la herramienta EDA *Xilinx ISE Design Suite 9.2*.

GPBUCK Project Status			
Project File:	GPBUCK.isc	Current State:	Programming File Generated
Module Name:	TOP_GPI_BUCK	o Errors:	No Errors
Target Device:	xc3s1200e-4fg320	o Warnings:	403 Warnings
Product Version:	ISE 9.2i	o Updated:	Jun 26, Ene 17:02:19 2012

GPBUCK Partition Summary			
No partition information was found.			

Device Utilization Summary			
	Used	Available	Utilization
Logic Utilization			
Number of Slice Flip Flops	12,443	17,344	71%
Number of 4 input LUTs	11,439	17,344	65%
Logic Distribution			
Number of occupied Slices	8,670	8,672	99%
Number of Slices containing only related logic	7,711	8,670	88%
Number of Slices containing unrelated logic	959	8,670	11%
Total Number of 4 Input LUTs	113,882	17,344	60%
Number used as logic	11,439		
Number used as a route-thru	1,742		
Number used as Shift registers	701		
Number of bonded IOBs	38	250	15%
IOB Flip Flops	10		
Number of Block RAMs	10	28	35%
Number of GCLKs	1	24	4%
Number of DCMs	1	8	12%
Number of MULT18K18510s	28	28	100%
Total equivalent gate count for design	929,765		
Additional JTAG gate count for IOBs	1,824		

Figura 5.18: Reporte de implementación generado por la herramienta EDA *Xilinx ISE Design Suite 9.2*.

Observando las variables mencionadas se buscó comprobar o rechazar la siguiente hipótesis de investigación (ver sección 1.5):

El voltaje y la corriente en la carga del inversor monofásico multinivel presentan la característica de poseer bajo contenido armónico y por medio del diseño de un filtro de segundo orden correctamente sintonizado se puede obtener un desplazamiento de FP casi cero y un factor de potencia cercano a la unidad, además cumple con la norma IEEE 519 y muestra robustez en la salida de voltaje v_C .

5.6.1. Diseño del experimento

Para verificar la hipótesis se formuló el siguiente diseño experimental.

1. La primer clase de experimentos está orientada a evaluar la **respuesta del sistema en estado estacionario**, en este sentido se le conectará al inversor una carga con resistencia fija de 75Ω (la cual fue modelada con el sistema), la amplitud del voltaje deseado será de $145 V$. En esta etapa se realizarán pruebas de distorsión armónica total y factor de potencia tanto en lazo abierto como en lazo cerrado, los valores obtenidos deberán de estar dentro de los requeridos por la norma **IEEE 519**.
2. En el segundo tipo de experimento se evaluará la **respuesta del sistema a un transitorio** al conectar y desconectar repentinamente una carga lineal. En esta prueba el inversor estará en operación con una carga resistiva de 75Ω , repentinamente se conmutará a un valor de resistencia de 30Ω por medio de un interruptor y después se regresará nuevamente con la carga de 75Ω .
Al igual que en la primera etapa, se efectuarán pruebas de distorsión armónica total y factor de potencia, se espera tener resultados acordes con la norma **IEEE-519**; la amplitud del voltaje deseado será de $145 V$. Esta prueba se realizará en lazo cerrado y lazo abierto, finalmente se hará una comparativa de ambos casos.
3. El experimento final consiste en evaluar la **respuesta del sistema a un transitorio** al conectar y desconectar repentinamente una carga no lineal, ésta consiste en una resistencia de 30Ω , un rectificador en puente completo y un motor de **CA** de $127 V$; se efectuarán pruebas de distorsión armónica total y factor de potencia para una amplitud deseada de voltaje de $145 V$ en lazo abierto y cerrado, para finalizar se compararán ambos resultados.

En la Figura 5.19 se muestra la planta a la que se realizarán las pruebas, éstas fueron realizadas auxiliados de los siguientes equipos y materiales:

- (a) Fuentes de CD de $40 V$ para alimentación de las celdas.

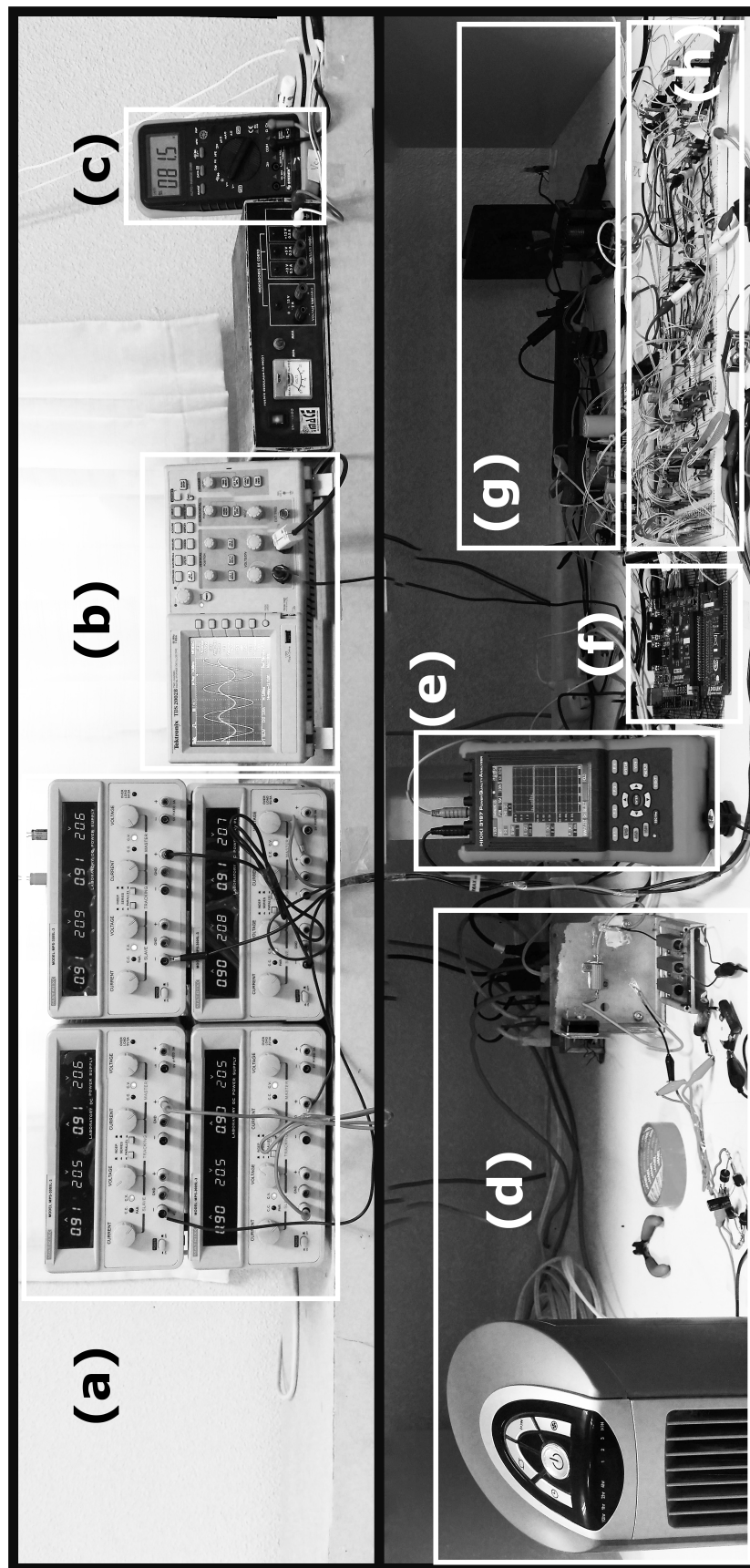


Figura 5.19: Planta del convertidor Multinivel en Cascada de cinco niveles.

- (b) Osciloscopio de dos canales *Tektronix*.
- (c) Multímetro.
- (d) Carga lineal y no lineal (Resistencias de 75 Ω y 30 Ω de 150 y 25 Watts respectivamente, además para la carga no lineal: puente de diodos comercial y Motor de AC).
- (e) Equipo de calidad de la energía *HIOKI 3197*.
- (f) Tarjeta de desarrollo *Nexys 2* con **FPGA** *Spartan-3E 1200*.
- (g) Filtro de salida.
- (h) Inversor multinivel en cascada de 5 niveles.

Pruebas experimentales en estado estacionario

Como ya se describió en la sección 5.6.1, esta prueba se realizó tanto en lazo abierto como en lazo cerrado y tiene la finalidad de verificar el comportamiento correcto en estado estacionario del convertidor, es decir, sin realizar ningún transitorio debido a algún cambio de carga; primero se colocó al convertidor únicamente el modulador PSPWM implementado en el FPGA como se muestra en la Figura 5.20, en esta etapa se realizó el ajuste correspondiente del sensor de voltaje y se verificó el funcionamiento correcto de las celdas. Una vez que se cerró el lazo del sistema fue necesario sintonizar el controlador por medio de las ecuaciones (4.38).

Partiendo de los resultados de simulación mostrados en el capítulo 4 se inició el proceso de sintonía del controlador **GPI** de seguimiento en conjunto con el sistema físico, para realizar esta tarea se utilizó un enfoque **HIL**¹¹ (*Hardware In-Loop*) incrementando de manera positiva el valor de ζ y ω_n hasta lograr el desempeño deseado del controlador; los valores de funcionamiento obtenidos para el sistema son $\omega_n = 3500$ y $\zeta = 0,707$, con estos valores se obtienen las siguientes constantes de sintonía:

$$\begin{aligned}
 k_3 &= 9898 & (5.13) \\
 k_2 &= 48992601 \\
 k_1 &= 1,212505E11 \\
 k_0 &= 1,500625E14
 \end{aligned}$$

Ya calibrado el sensor de voltaje y sintonizado el controlador se procedió a cerrar el lazo de control de acuerdo a como se muestra en la Figura 5.21; para esta prueba, en ambos casos (lazo

¹¹Adicional al procedimiento de sintonía se implementó un controlador de comunicación para transferir las constantes de sintonía y extraer los datos internos de los procesos realizados dentro del FPGA, para implementar la interfaz mencionada esta operación fue realizada por medio de un controlador **USB (DLP-USB245M)** de la compañía *FDTI*, debido a sus características, es ideal para implementar la interfaz entre una *PC* y un sistema autónomo [131] (Ver Apéndice F).

abierto y cerrado) la amplitud deseada fue de (145 V) y se manejó un índice de modulación de 0,90625¹². El resultado de esta prueba para la salida del convertidor (v_C), la corriente de

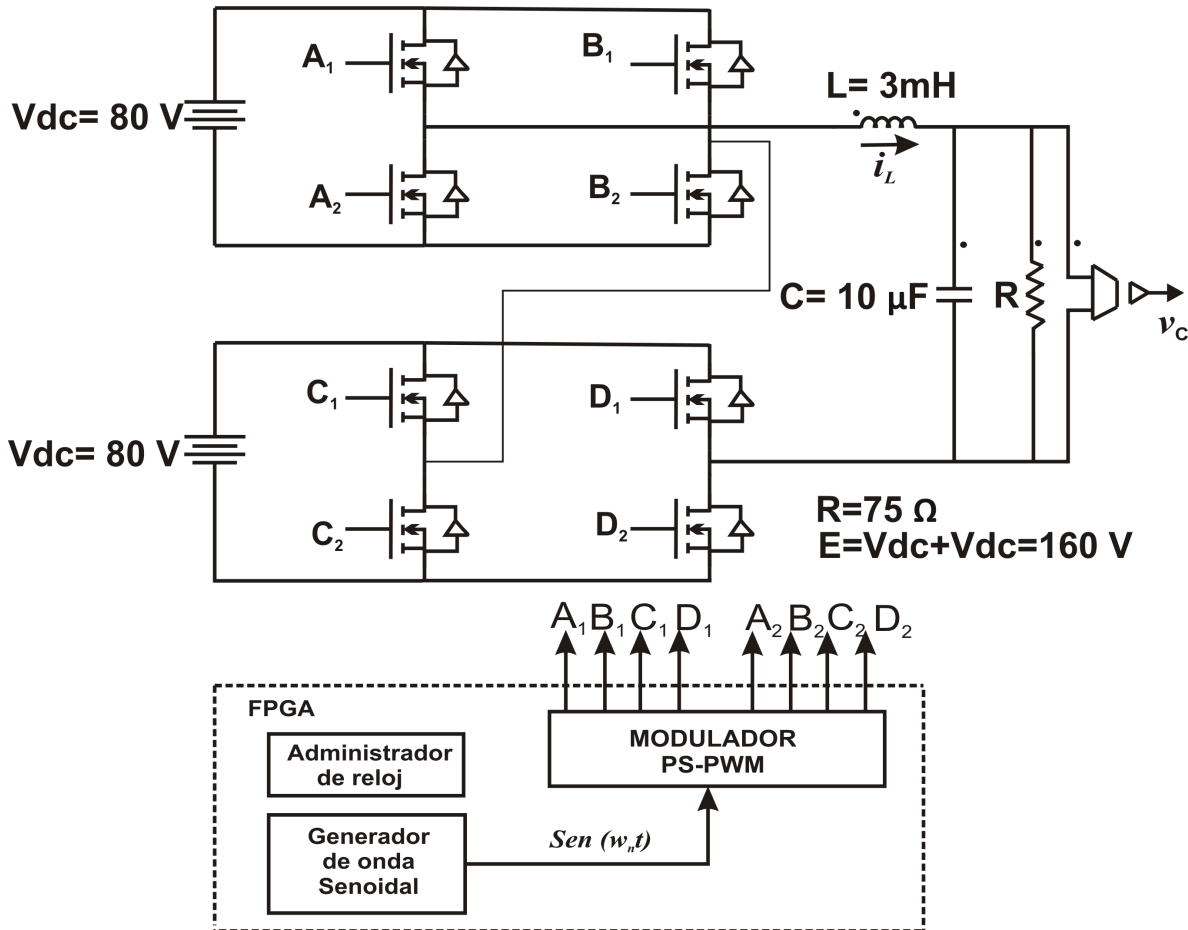


Figura 5.20: Convertidor Multinivel en Cascada de cinco niveles en lazo abierto (Estado estacionario).

carga (i_L), los análisis de THD (voltaje y corriente) y de factor de potencia se muestran en las Figuras 5.22 y 5.23 respectivamente, en éstas se comparan los resultados obtenidos en lazo abierto y en lazo cerrado.

Los resultados de la Figura 5.23 muestran que la onda de salida del convertidor está dentro los parámetros permisibles por el apartado **IEEE-519** en relación al THD de voltaje y corriente.

¹²Se obtiene de dividir $V_d = 145 V$ entre $E = 160 V$; $m_i = \frac{145}{160} = 0,90625$

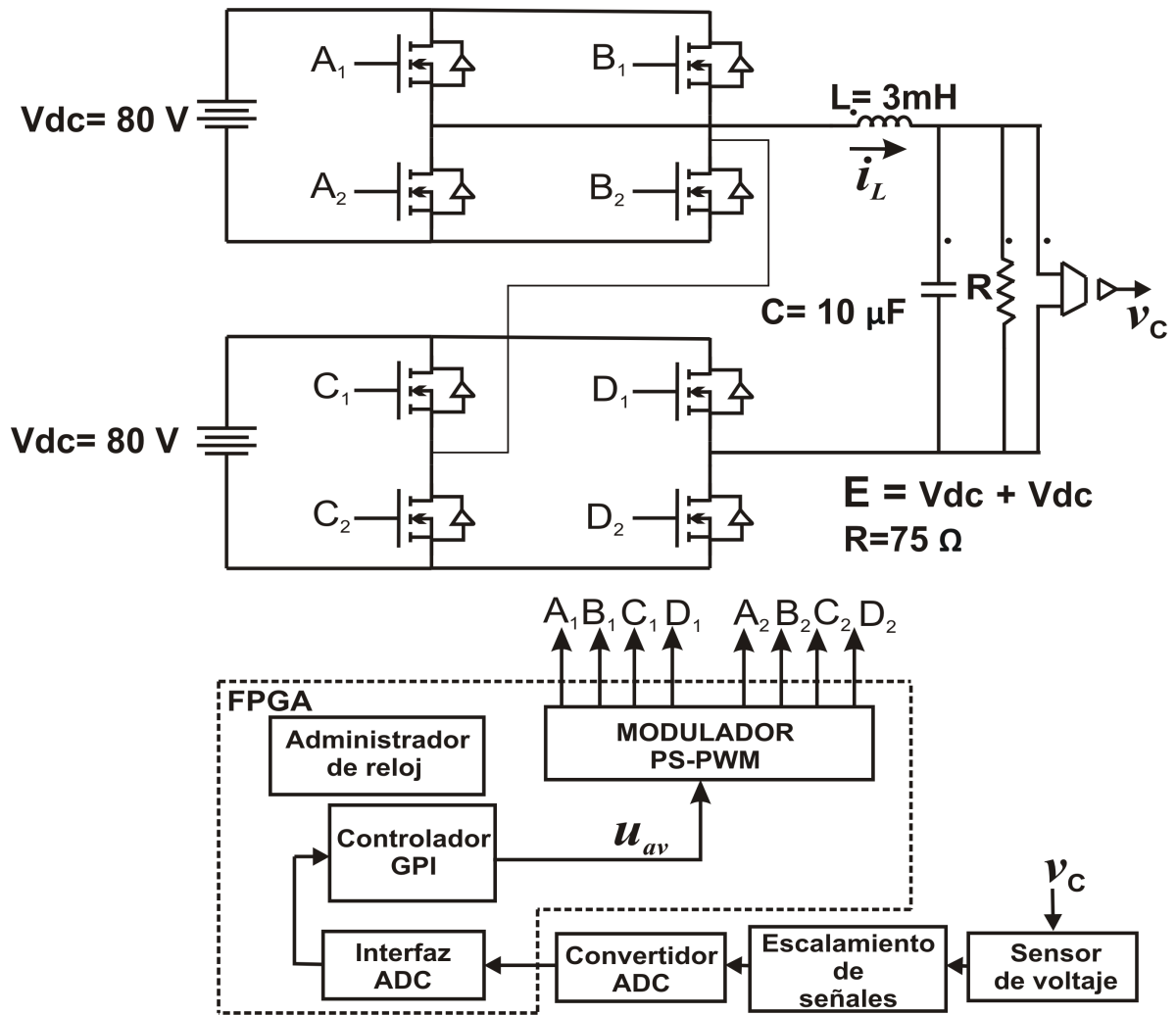


Figura 5.21: Convertidor Multinivel en Cascada de cinco niveles en lazo cerrado (Estado estacionario).

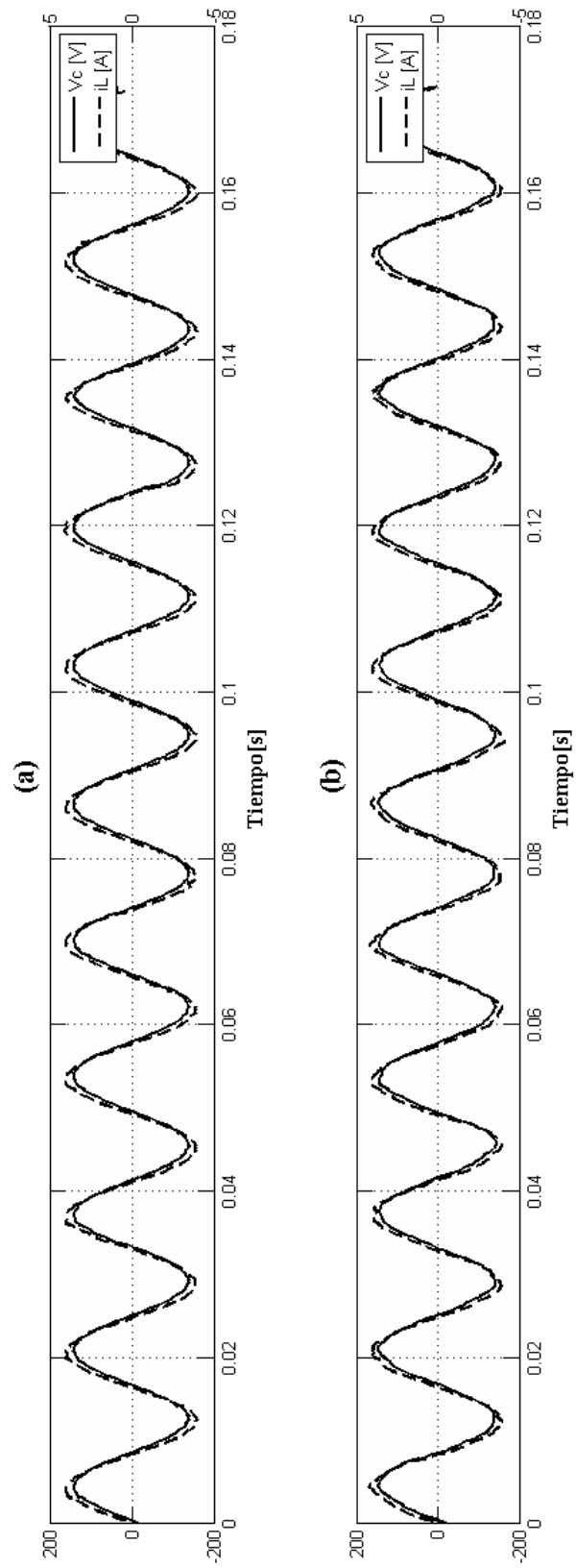


Figura 5.22: Salida del convertidor multinivel v_C y de la corriente de carga i_L : (a) Prueba en lazo abierto; (b) Prueba en lazo cerrado.

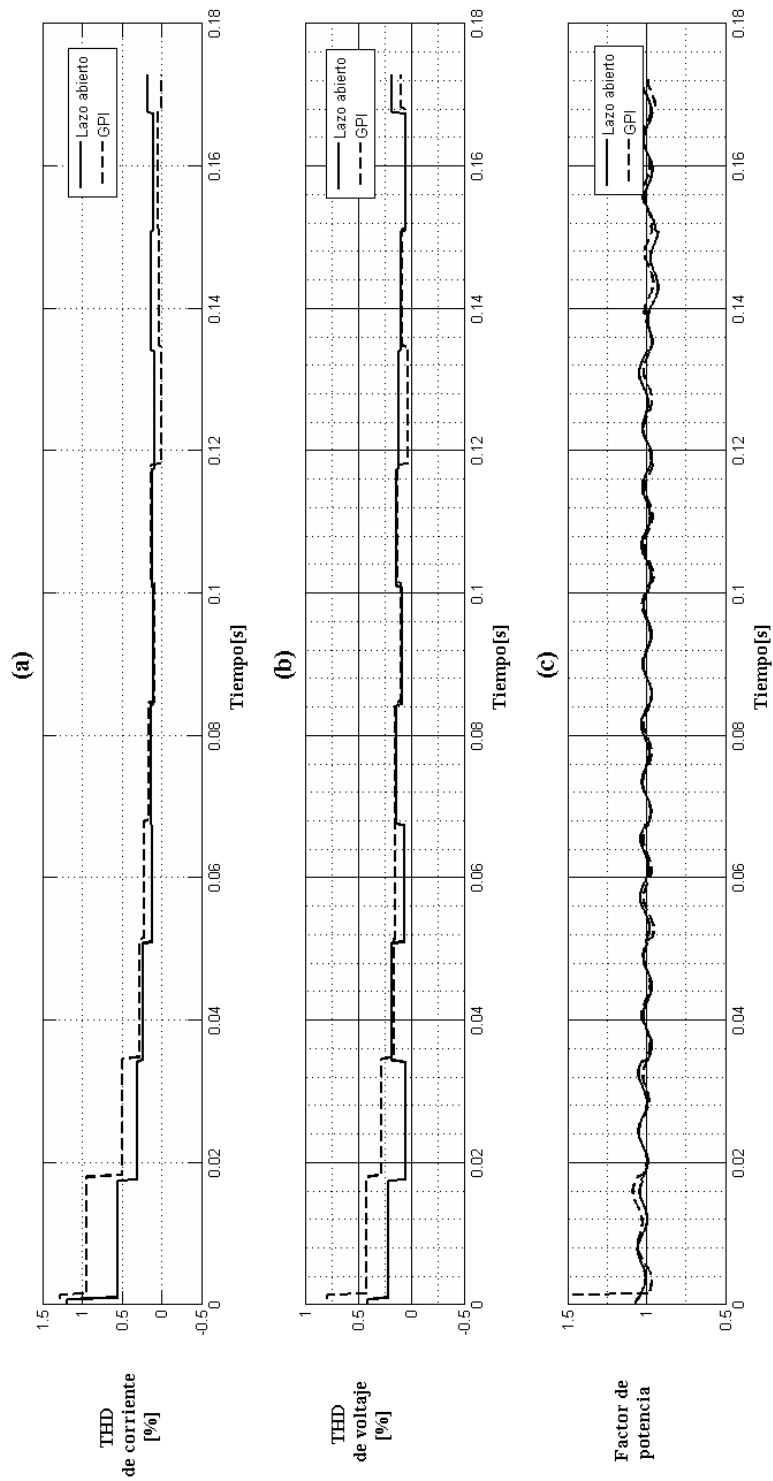


Figura 5.23: Análisis de THD en corriente y de voltaje, así como también de Factor de Potencia: (a) THD corriente; (b) THD voltaje; (c) Factor de potencia.

Prueba experimental de la respuesta del sistema a un transitorio (carga lineal)

Siguiendo con la batería de pruebas correspondientes y auxiliado por el esquema de la Figura 5.24, en la Figura 5.25 (a) y (c) se observa que una vez que el sistema es sujeto a una perturbación repentina (cambiar de posición a **SW1**) tanto en lazo abierto como en lazo cerrado, la corriente del sistema (i_L) se incrementa; por otra parte, si se comparan las gráficas 5.25 (b) y (d) correspondientes al error de voltaje (v_C) en lazo abierto y lazo cerrado respectivamente se puede observar que el error de voltaje en (b) es mayor comparado con (d), por lo tanto, como se esperaba ésta última presenta la característica de robustez deseada. En la Figura 5.26 se muestra una comparativa para el sistema en lazo abierto y cerrado, los resultados son para el THD de corriente y voltaje, además del factor de potencia, mostrando resultados muy parecidos.

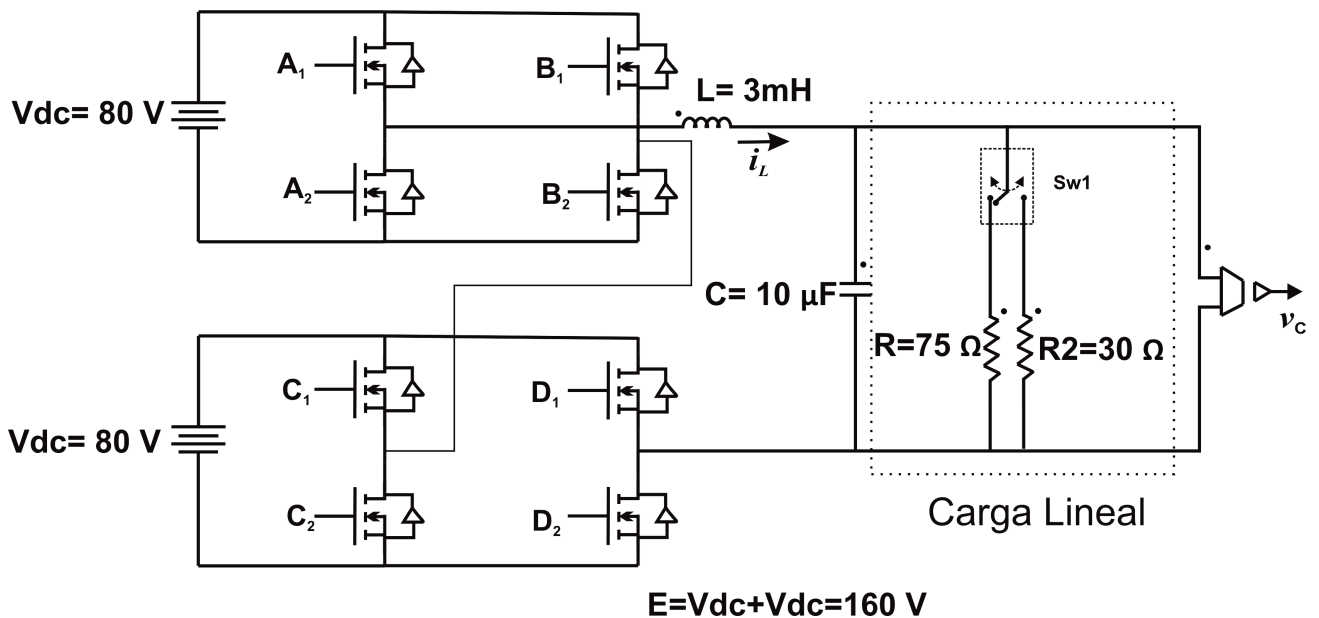


Figura 5.24: Esquema propuesto para realizar la prueba al sistema ante cambios de carga lineal en lazo abierto y lazo cerrado.

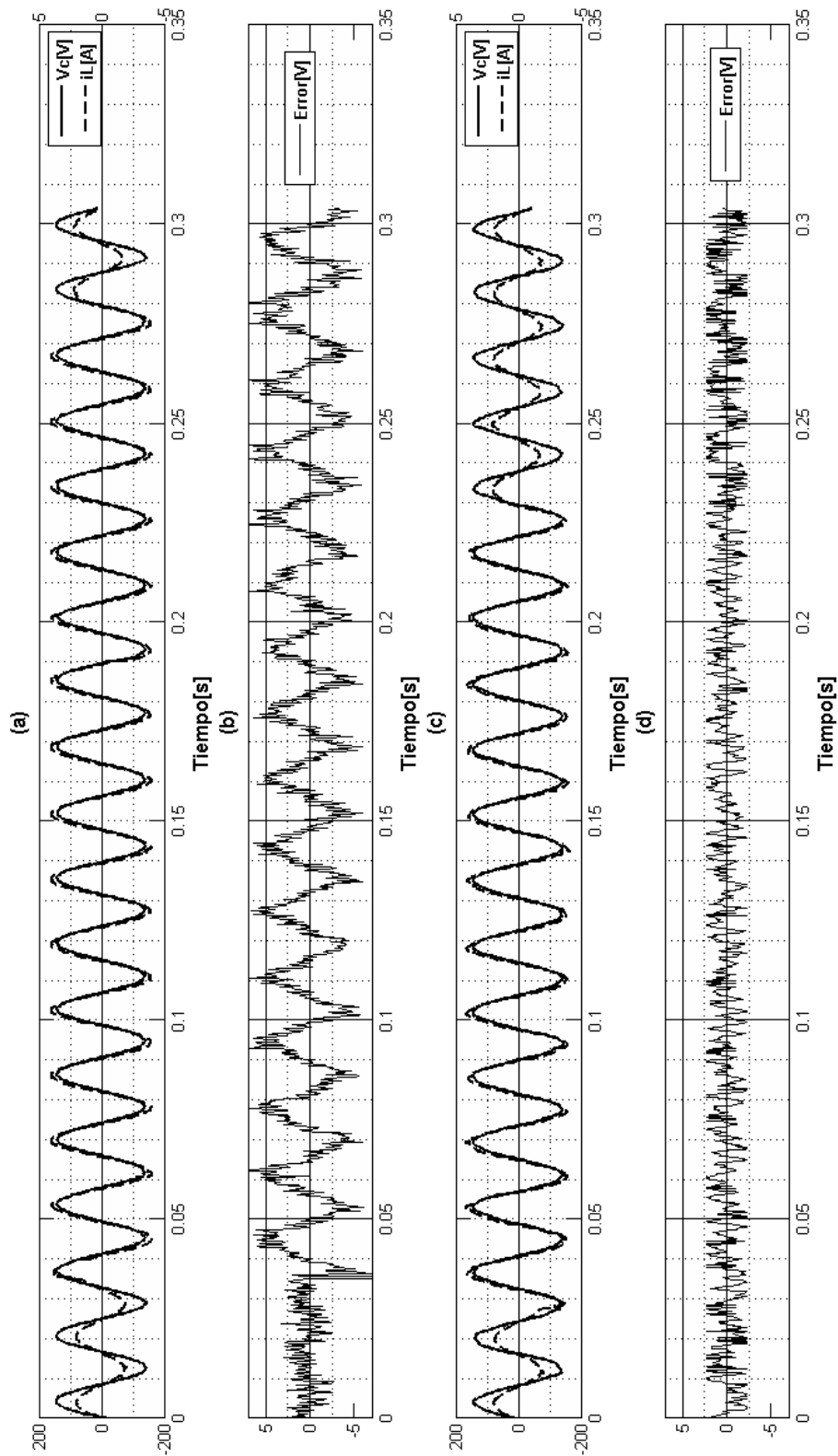


Figura 5.25: Resultados ante un cambio de carga lineal del sistema en lazo abierto y cerrado: **(a)** Voltaje de salida del convertidor contra corriente de carga (lazo abierto); **(b)** Error de voltaje (lazo abierto); **(c)** Voltaje de salida del convertidor contra corriente de carga (lazo cerrado); **(d)** Error de voltaje (lazo cerrado).

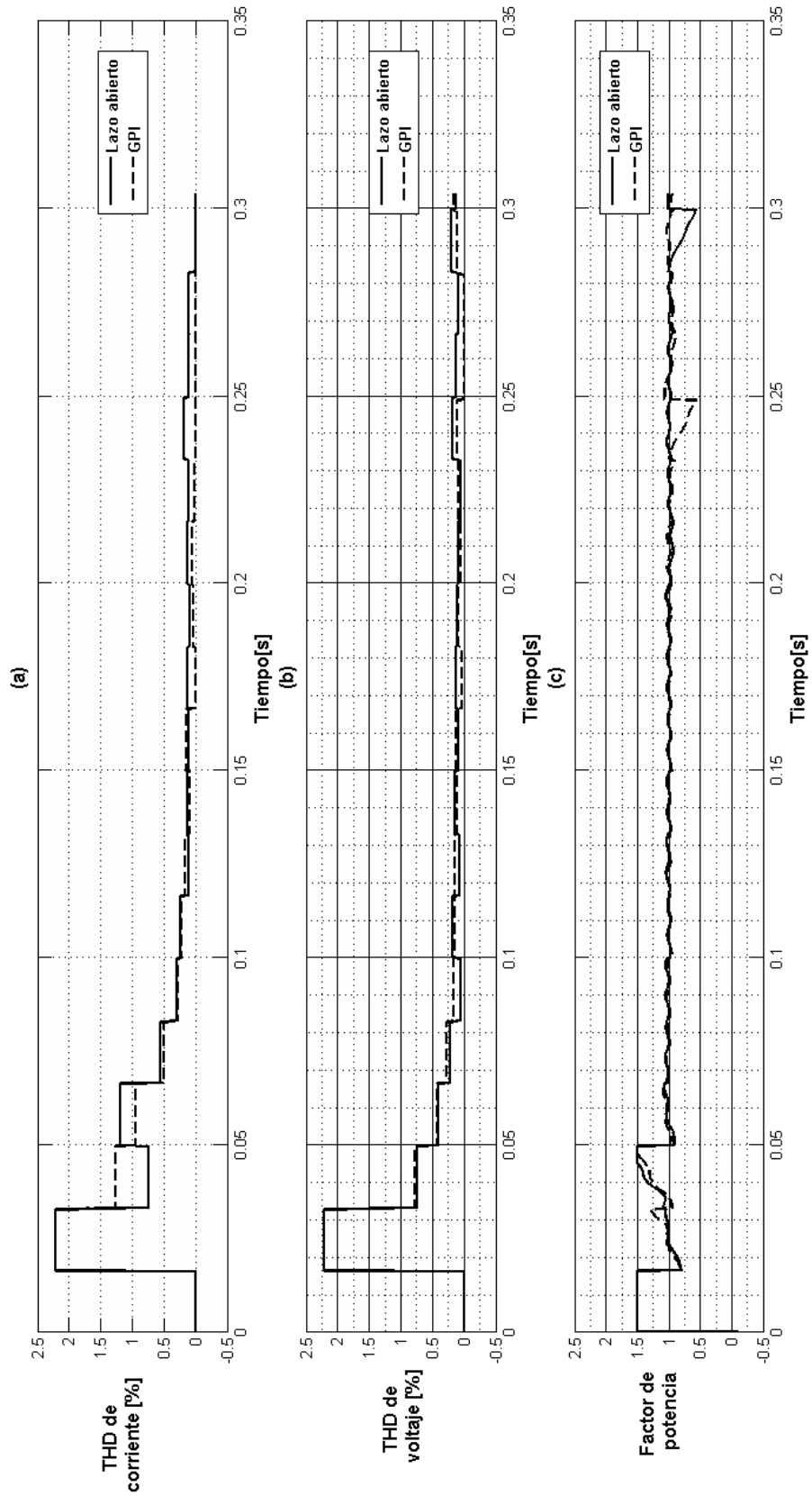


Figura 5.26: Resultados de **TDH** y **FP**: (a) THD de corriente; (b) THD de voltaje; (c) Factor de potencia.

Prueba experimental de la respuesta del sistema a un transitorio (carga no lineal)

En esta prueba se empleó una carga no lineal compuesta por un puente de diodos comercial y un motor de AC de un ventilador ($120 V_{rms}$, $0,38$ Amp, 60 Hz) tal como lo muestra la Figura 5.27; se realizaron las mismas pruebas que en el caso lineal, los resultados se muestran en las Figuras 5.28 y 5.29.

Basado en los resultados obtenidos en las pruebas mostradas por las Figuras 5.25, 5.26,

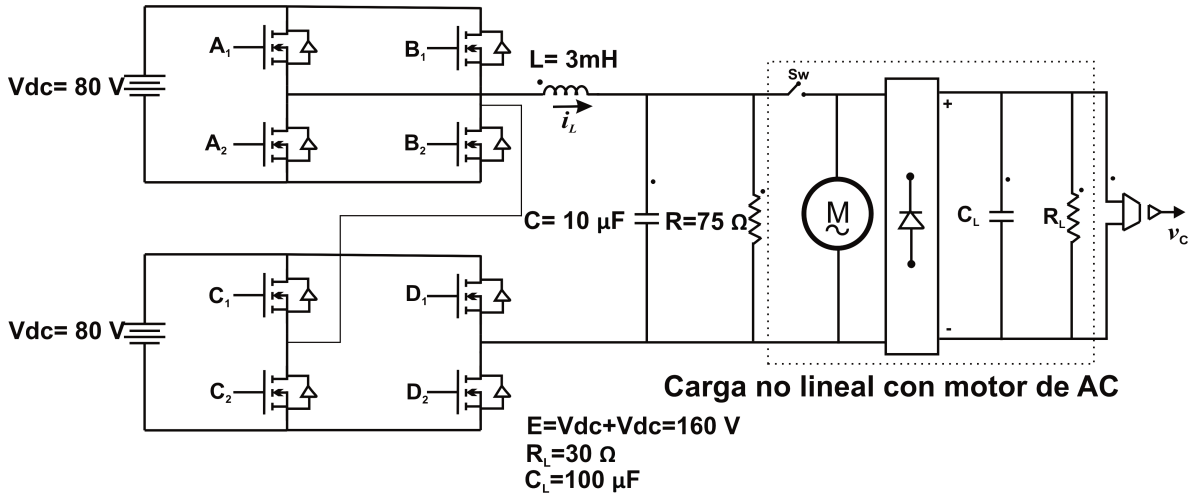


Figura 5.27: Esquema propuesto para realizar la prueba al sistema ante una carga no lineal.

5.28 y 5.29 la hipótesis formulada es verificada y es aceptada como verdadera, ya que en lazo cerrado se observaron y se comprobaron experimentalmente las siguientes características:

- El **THD** en corriente y voltaje posee una distorsión mínima (menor al 5%) permisible por la norma **IEEE-519**, este valor se conserva tanto para cargas lineales como para no lineales.
- El factor de potencia (**FP**) se conserva cercano a la unidad ante cargas del tipo lineal y no lineal (la corriente i_L y el voltaje v_C están en fase).
- El voltaje de salida del convertidor (v_C) se muestra robusto ante perturbaciones (*cambio de carga lineal y no lineal*).

De esta etapa se concluye que el convertidor multinivel muestra una mejora considerable en desempeño una vez que se comparan los resultados en lazo abierto y en lazo cerrado, resultando mejor cuando se integra el controlador **GPI** de seguimiento de voltaje.

Índice de desempeño(ISE)

Las pruebas realizadas en los apartados anteriores fueron perturbaciones exógenas al sistema de control, como ya se observó, el controlador se mostró robusto ante cambios de carga

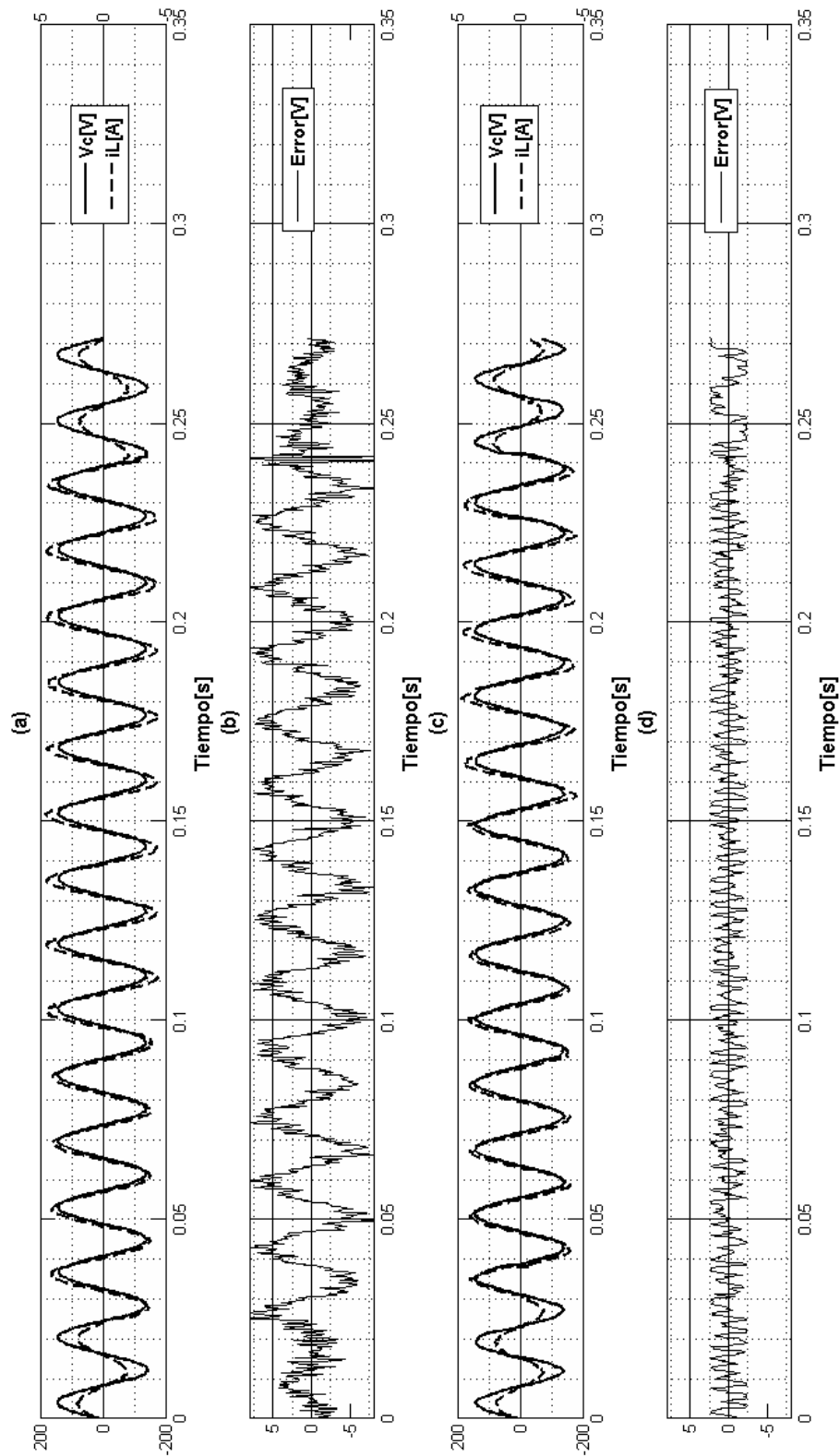


Figura 5.28: Resultados ante un cambio de carga no lineal del sistema en lazo abierto y cerrado: (a) Voltaje de salida del convertidor contra corriente de carga (lazo abierto); (b) Error de voltaje (lazo abierto); (c) Voltaje de salida del convertidor contra corriente de carga (lazo cerrado); (d) Error de voltaje (lazo cerrado).

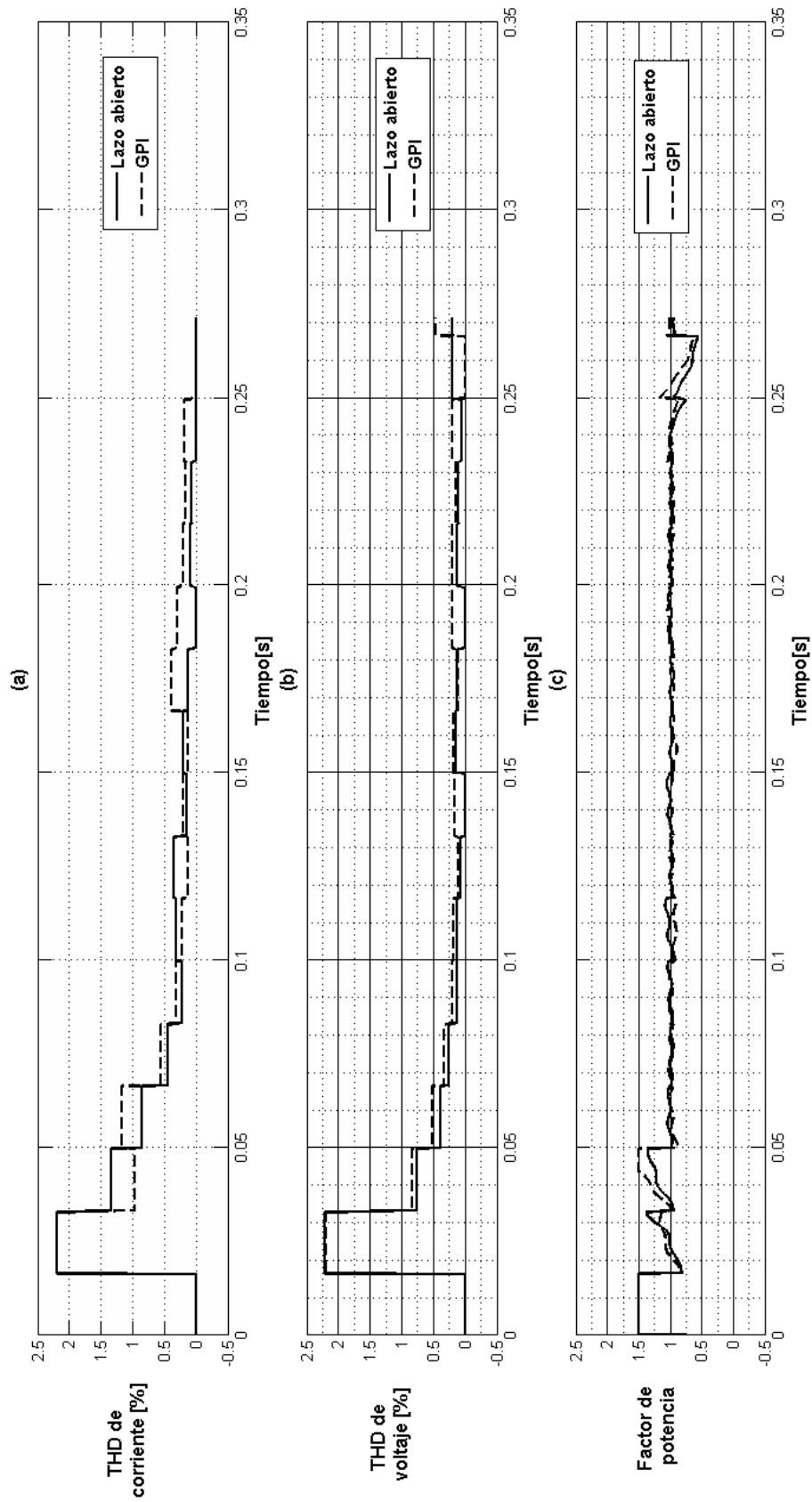


Figura 5.29: Resultados de **TDH** y **FP**:(a) THD de corriente; (b) THD de voltaje; (c) Factor de potencia.

lineal y no lineal, de manera complementaria se planeó una prueba al controlador basada en una perturbación del tipo endógena.

Se observa que en la expresión del controlador (ver ecuación (5.2)), la inversa de la ganancia de control, denotada por α_1 , cancela de manera exacta el parámetro $(V_{dc}/(LC))$. Sin embargo es claro notar que un desconocimiento de este parámetro ¹³ influye en la efectividad del controlador por retroalimentación.

En este caso, si se asume que se conoce el valor preciso de α_1 para el controlador actual, y se le afecta a α_1 por un factor κ (osea $\tilde{\alpha}_1 = \kappa\alpha_1$) en el controlador, en este caso κ representa un porcentaje de desconocimiento de α_1 , con $\kappa = 1$ se obtendría el valor conocido de α_1 , por lo tanto, si se varía $\kappa = 1$ (aumentando o disminuyendo su valor) se puede obtener un conjunto de gráficas en la evolución del tiempo para la función de la Integral del Error Cuadrático (ISE) (Ver Figura 5.30).

Esto está dado por la expresión (5.14) (ver[132])

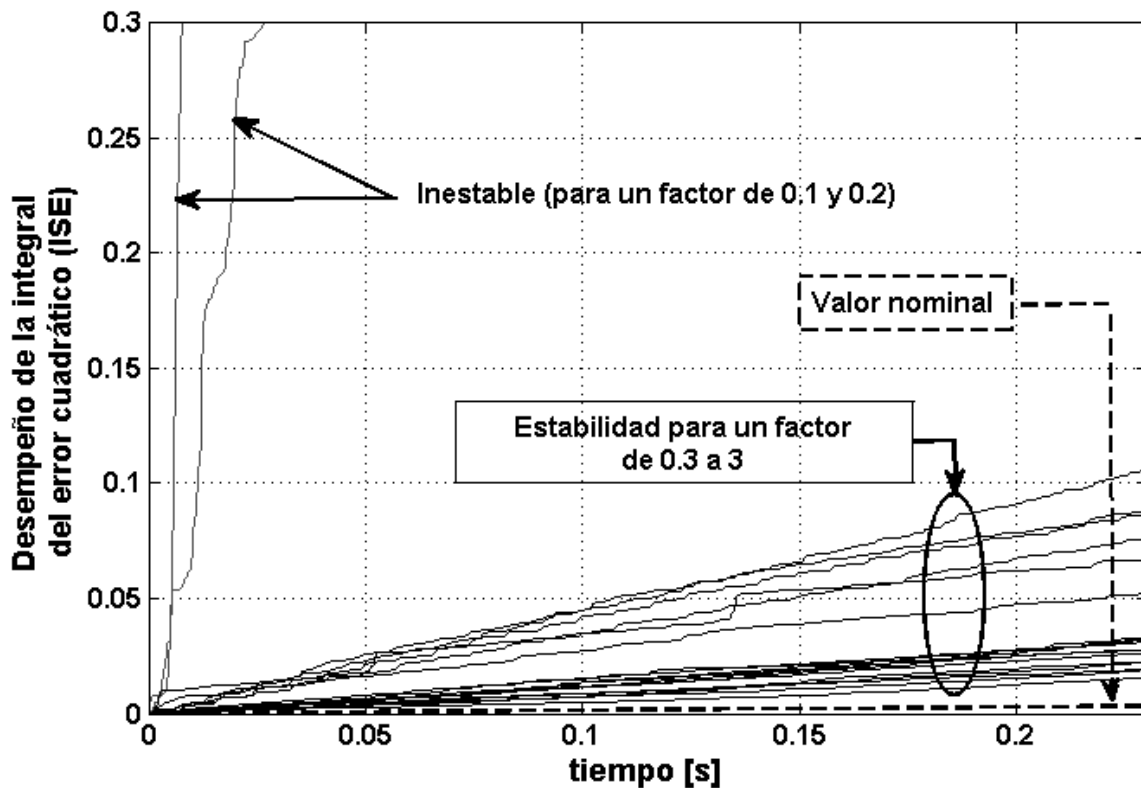


Figura 5.30: Función de la Integral del Error Cuadrático (ISE).

$$ISE_{\kappa}(t) = \int_0^t (F_{\kappa}(\sigma) - F^*(\sigma)) d\sigma \quad (5.14)$$

¹³Esto debido a las variaciones físicas que no fueron modeladas y que son propias de L , C o V_{dc} ; la variación puede ser de alguna o de todas ellas

En esta ecuación $F_\kappa(t)$ es la salida de control obtenida con un valor particular de ganancia κ , donde κ como ya se mencionó se usa para cancelar el término $\tilde{\alpha}_1$.

De la Figura 5.30 el control propuesto se muestra robusto con respecto a una ganancia $\kappa = 0,3$ (equivale a un conocimiento del 70 % de α_1) hasta un valor de $\kappa = 3$ (el cual equivale a un desconocimiento del 300 % de α_1). Una variación rápida en el tiempo revela, por medio del **ISE** un comportamiento inestable para $\kappa = 0,1$ y $\kappa = 0,2$.

De manera práctica, en el caso de este diseño, el valor del error de voltaje nunca es cero, por lo tanto se observa para valores de $\kappa \in [0,3, 3]$ un incremento suave en el valor del **ISE** a lo largo del tiempo; por otra parte para valores de $\kappa > 3,0$ se observa la presencia de ruido en la señal de salida del convertidor.

De los resultados obtenidos se concluye que el controlador GPI presenta un alto grado de robustez ante un amplio rango de variación de parámetros.

Parte IV

Conclusiones

Capítulo 6

Conclusiones y perspectivas

En este trabajo se describió el diseño de un controlador **GPI** de seguimiento para regular la salida de voltaje de un inversor multinivel de fuentes independientes de cinco niveles con filtro de segundo orden (**LC**) a la salida.

El desempeño del controlador fue presentado en el capítulo 5 y posee las siguientes características:

- Las ondas de voltaje y corriente en la salida de la carga presentan un bajo contenido armónico.
- Dado el correcto diseño del filtro de salida, el factor de potencia en la carga de salida es cercano a la unidad, conservándose ante cambios de carga repentinos del tipo lineal o no lineal
- La salida de voltaje del convertidor es robusta ante cambios de carga lineales o no lineales, además tiene una recuperación rápida.

Por otra parte, el diseño implementado en **FPGA** está basado en una metodología actualizada y adecuada para llevar a buen término la implementación de controladores en dispositivos de lógica reprogramable.

El controlador en **FPGA** presentado se diseña e implementa siguiendo una propuesta modular, la cual permite generar código reusable que puede ser parte de un repositorio de código de funciones discretas para ser utilizadas en el diseño de sistemas de control en el futuro.

Gracias a que los valores generados por las operaciones involucradas en el controlador discreto fueron implementadas en módulos de aritmética con punto flotante de precisión simple y basados en el estándar *IEEE-519-1985*, el diseño propuesto permite manejar un amplio rango de exactitud en los resultados; sin embargo, no se puede descartar explorar el manejo de módulos en punto flotante con menor resolución (menor cantidad de bits para operaciones) para buscar reducir el consumo de recursos del dispositivo reprogramable sin sacrificar desempeño del controlador.

El esquema del controlador GPI de seguimiento fue implementado por medio de una arquitectura segmentada considerando la independencia de las operaciones involucradas y de los módulos que integran al mismo, la arquitectura lograda permite reducir el tiempo en completar la tarea de control asignada.

Por otro lado, el tiempo de muestreo definido para la aplicación ($t_{SP} = 4\mu s$), se consideró adecuado para el diseño, dadas las características del mismo. Sin embargo, sería adecuado incrementar el tiempo de muestreo buscando reducir el tamaño de las memorias empleadas (**BRAMs**), sin sacrificar los tiempos de la respuesta, resolución del *PWM* y en general, sin reducir el desempeño del controlador en términos de eficiencia y rapidez.

En términos de uso de recursos, la técnica empleada para generar las ondas triangulares (necesarias para el generador **PSPWM**) y las ondas seno y coseno (necesarias para generar las señales de seguimiento) por medio de una aproximación discreta, resulto ser muy eficiente, esto debido a que se hizo uso de los módulos internos de memoria (**BRAM**) del **FPGA**, sin incrementar hardware adicional, minimizando así el uso de bloques de lógica dentro del dispositivo.

Una característica importante de mencionar es que el controlador **GPI** implementado es de seguimiento, y no de regulación, es decir el controlador es capaz de seguir con un mínimo de error la señal previamente programada.

Apéndices

Apéndice A

Tendencias en la tecnología FPGA

A.1. Unidad de procesamiento

Con el advenimiento de los sistemas SoC (*Soc, System on Chip*), los cuales permiten la combinación de software y hardware han surgido dos tipos de procesadores que se pueden trabajar con los FPGAs, los llamados procesadores sintetizables y los no sintetizables, estos se describen a continuación.

Procesador no sintetizable (*Hard Processor Core*): Está integrado dentro del FPGA y posee tecnología VLSI (*VLSI, Very-large-scale integration*), como regla general, un procesador de este tipo ofrece poca flexibilidad, pero altas velocidades de reloj.

Por ejemplo, la compañía Altera provee de procesadores del tipo *ARM9* dentro de su serie EPXA 10 [78].

El FPGA Virtex-5 de Xilinx integra procesadores *PowerPc440* en su chip [77]. Recientemente Actel ha propuesto su primera familia de FPGA llamada *Fusion* con su procesador *Cortex – M3* [74].

Procesador sintetizable (*Soft Processor Core*): Este tipo emplea celdas lógicas dentro del FPGA para implementar el núcleo del procesador, ejemplos de este tipo de procesador podemos encontrar al *NiosII* de Altera, al procesador *MicroBlaze* de Xilinx y al *Cortex – M1* de Actel. La particularidad que ofrece esta propuesta es la flexibilidad que permite al diseñador al momento de configurar, especificar el número y el tipo de periféricos y el tamaño de memoria; una de las desventajas al momento de usar este tipo de procesadores es que trabajan con velocidades de reloj inferiores a la del sistema donde son incrustados.

A.2. Periféricos analógicos

Las nuevas familias de FPGAs poseen un nuevo nivel de integración y permiten al diseñador implementar tratamiento de señales ya sean analógicas o digitales, un ejemplo de esta mejora lo representa la familia de FPGAs de Actel llamada *Fusion*, este tipo de FPGA

integra un convertidor programable analógico-digital de 12 bits del tipo de aproximaciones sucesivas; esta característica hace que los FPGAs se conviertan en sistemas SoC ideales para aplicaciones en control [129].

A.3. Arquitecturas de multiprocesamiento

Para el caso de aplicaciones digitales complejas y exigentes en cuanto a desempeño y rapidez, actualmente se puede elegir a los dispositivos con arquitecturas múltiples de procesamiento, todas en un solo chip (MPSoCs, *Multiprocessing architectures in a single chip*), los cuales son una alternativa que provee alto nivel de escalabilidad, particularmente en términos de potencia y desempeño, comparado con núcleos de procesamiento monolíticos [81].

Uno de los problemas importantes con este tipo de dispositivos es que el diseño de la infraestructura de comunicaciones generalmente es serial, por lo tanto, la escalabilidad está restringida por la serialización del acceso al canal de comunicación y por las múltiples peticiones de acceso. Una propuesta que busca resolver este tipo de problemas es el concepto de NoCs (NoCs, *Network on Chip*), el cual provee reusabilidad, flexibilidad y un sistema de comunicaciones global muy eficiente [82].

En este contexto el uso del lenguaje llamado *SystemC*¹ resulta interesante, ya que éste permite la simulación completa del sistema multiprocesador y también la estrategia de comunicación a implementarse [83].

Para estas aplicaciones, los sistemas operativos en tiempo real o también llamados RTOS (RTOS, *Real Time Operating Systems*) son esenciales, ya que estos ofrecen un conjunto enriquecido de servicios que proveen soporte básico. Algunos ejemplos de RTOS usados en este tipo de tareas son: **Linux embebidos** y el **MicroC/OS-II** [84].

A.4. Herramientas de diseño

De manera paralela a la evolución del FPGA, las herramientas de diseño también han madurado.

Hoy en día, los fabricantes de FPGAs proveen un conjunto de herramientas muy completo que permiten iniciar el proceso de diseño partiendo de la descripción de hardware (ya sea usando el lenguaje VHDL o Verilog) y llevan al diseñador hasta la generación del archivo de configuración, el cual será descargado en el FPGA que se usará en el diseño [77, 78, 85].

Generalmente, las herramientas de diseño incluyen alguna para el diseño de hardware, alguna herramienta para el proceso de verificación (la cual incluye editor VHDL/Verilog, sintetizador, colocación y ruteo y finalmente la implementación), librerías del propio vendedor, y además,

¹SystemC es frecuentemente descrito como un lenguaje de descripción de hardware tal como VHDL y Verilog, pero es más adecuado describirlo como un lenguaje de descripción de sistemas, puesto que es realmente útil cuando se usa para modelar sistemas a nivel de comportamiento.

alguna herramienta de simulación y depuración de errores.

Algunos ejemplos de estas herramientas son el **ISE** (*Integrated Software Environment*) de Xilinx; **Quartus** de Altera y **Libero** de Actel. Las herramientas ya mencionadas poseen la característica de poder asociarse con otras herramientas de simulación y depuración, por ejemplo **Modelsim** en el caso de simulación y **ChipScope** para depuración, ambos de Xilinx. Por otra parte, los fabricantes de FPGAs para buscar adaptarse a la tendencia en el avance de los SoC, proporcionan herramientas de desarrollo software, módulos IP y herramientas de personalización para sus procesadores embebidos.

Por ejemplo, Xilinx proporciona una plataforma de desarrollo integrado (EDK, *Embedded Development Kit*) [77]; Altera ofrece una herramienta de diseño embebido (EDS, *Embedded Design Suite*)[78] y Actel provee de su plataforma a la que llama **Softconsole** [74].

Apéndice B

Diseño del filtro de salida del convertidor

B.1. Introducción

Las técnicas de modulación PWM senoidales poseen las mejores características para usarse en convertidores para aplicaciones de filtros activos. Como ya se mencionó en este trabajo, se utilizó el método PWM de corrimiento de fase (PSPWM), en éste, el rizo de tensión en la salida está en proporción directa al número de portadoras utilizadas y por tanto, al número de niveles del inversor; en esta técnica se menciona la frecuencia del rizo de tensión originada por las portadoras, principalmente por que este parámetro influye directamente en el tamaño del filtro en la salida del inversor.

Se realizó la elección de esta técnica de modulación ya que posee la característica de desplazar el contenido armónico generado de manera implícita por la técnica a frecuencias más elevadas que las otras técnicas de modulación, logrando con esto reducir el tamaño del filtro de salida y permitiendo una reproducción más fiel de la señal de salida por parte del inversor multinivel. Existen varios criterios para diseñar el filtro de salida de un inversor, para elegir el adecuado regularmente se debe de considerar el tipo de respuesta que presenta el filtro de tal forma que la tensión de salida proporcionada sea de la mayor calidad posible (menor THD en la onda de corriente y de voltaje).

Los requerimientos más importantes que se deben de tomar para el diseño del filtro de salida son:

- Atenuar al máximo los armónicos que poseen mayor amplitud.
- Obtener la amplitud de la onda con la frecuencia fundamental con la menor atenuación posible.
- Reducir en la medida de lo posible el tamaño del filtro.
- Lograr que el porcentaje de distorsión armónica en corriente y voltaje se encuentre lo

más bajo posible, o en su defecto que se encuentre dentro de lo establecido por la norma IEEE-519 (THD <5%).

De lo anterior, se puede señalar que el desempeño del inversor en relación a la calidad de la energía que puede proveer, depende en gran medida del filtro de salida, por lo tanto se debe de tener especial cuidado en su diseño.

B.1.1. Filtro LC

La función de transferencia para el filtro LC mostrado en la Figura B.1 está definida por la Ecuación B.1.

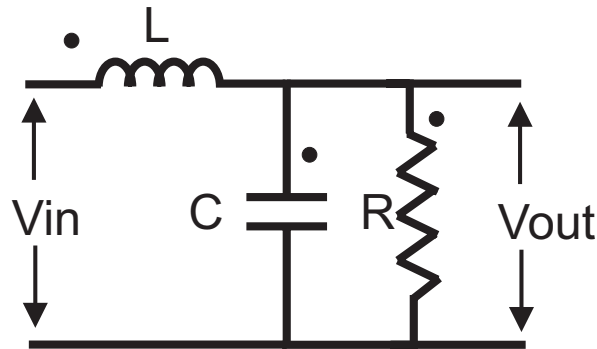


Figura B.1: Filtro de segundo orden del tipo RLC.

$$H(s) = \frac{\omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0} \quad (\text{B.1})$$

Donde:

$$\omega_0 = \sqrt{\frac{1}{LC}} \quad (\text{B.2})$$

$$Q = R\sqrt{\frac{C}{L}} \quad (\text{B.3})$$

de las ecuaciones B.2 y B.3 se tiene que:

ω_0 es la frecuencia de resonancia.

Q es el factor de calidad del filtro.

Como en todo filtro, el valor de la frecuencia de resonancia está definido por los valores de L y C, mientras que el factor de calidad del filtro depende del valor de R, por lo tanto, la respuesta en frecuencia es independiente de la carga. Para el diseño del filtro de salida, por lo general se cuenta con tres aproximaciones [133]:

- Filtros Butterworth.
- Filtros Chebyshev.

- Filtros Bessel.

Cada uno de los filtros mencionados presenta su propio desempeño y cuentas con sus propias características; el filtro *Butterworth* tiene una respuesta en frecuencia que es más plana comparada con la de sus contrapartes, es decir no presenta un pico de ganancia alrededor de la frecuencia de corte, por lo tanto presenta una respuesta aproximada a un filtro ideal; por otra parte el filtro *Chebyshev* introduce oscilaciones en su banda de ganancia, por lo que para aplicaciones que no deben poseer transitorios bruscos no es aceptable; por último, el filtro *Bessel* presenta un desempeño pobre en la banda de atenuación e introduce un desfase significativo entre la tensión de entrada y la tensión de salida.

Para diseñar el filtro de salida del convertidor se eligió al filtro *Butterworth* especialmente por el tipo de respuesta que presenta.

Para calcular los valores de la inductancia y la capacitancia se pueden emplear las expresiones B.2 y B.3.

B.1.2. Cálculo de L y C del filtro de salida

Para el cálculo de los componentes de inductancia y capacitancia que formaran parte del filtro de salida se deben conocer la siguiente información:

- La frecuencia de rizo de la tensión de salida (F_{rizo}).
- La frecuencia de corte (F_{corte}).
- El valor de la resistencia con la que se modeló el sistema.

La frecuencia de corte (F_{corte}) es la frecuencia máxima permitida para lograr la atenuación deseada del contenido armónico que está presente en la salida del convertidor multinivel, el valor de la frecuencia de corte F_{corte} se eligió así considerando el cálculo del filtro para una atenuación a partir una década por debajo de la frecuencia de rizo, por lo tanto, si se eleva la frecuencia de las señales portadoras del modulador PSPWM se podrá usar el mismo filtro.

En la Tabla B.1 se pueden observar los límites para THD en baja tensión. Se planea que el

Tabla B.1: Niveles de contenido armónico

Niveles de contenido armónico máximo para diferentes aplicaciones.	
Clase de sistema	Distorsión Armónica Total(THD)
Aplicaciones criticas	3 %
Sistemas Generales	5 %
Sistemas especializados	10 %

valor a obtener en relación al THD deberá de ser menor del 5 %, por lo que el diseño entraría

dentro de la categoría de sistemas generales.

Para iniciar el cálculo se puede emplear el valor de la frecuencia de corte (F_{corte}) y obtener la frecuencia angular (ω_0) a partir de la Ecuación B.4, por lo tanto:

$$\omega_{cf} = 2 * \pi * F_{corte} \quad (B.4)$$

$$= 2 * \pi * (9,6 \text{ kHz})$$

$$= 1800\pi = 5654,8667 \text{ rad/seg} \quad (B.5)$$

A partir de este resultado, empleando la ecuación B.2 se tiene que:

$$LC = \frac{1}{(\omega_{cf})^2} = 3,127197026 \times 10^{-8} \quad (B.6)$$

De la Ecuación B.6, se puede proponer un valor de capacitancia que pueda lograrse por medio de un arreglo de capacitores de valor comercial, este valor **NO** debe ser el más pequeño que se pueda construir, dado que de ser considerado así, los valores de inductancia resultantes pueden ser relativamente grandes, trayendo consigo dificultades para construir la inductancia, pudiendo originar un problema técnico de diseño, no realizable de manera práctica.

Para este caso, el valor de la capacitancia se propone de $C = 10\mu\text{F}$, por lo tanto, al sustituir en la expresión B.6 y resolver para L , arroja un resultado dado por la ecuación B.7:

$$\begin{aligned} L &= \frac{3,127197026 \times 10^{-8}}{10 \times 10^{-6}} \\ &= 3,127197026 \times 10^{-3} \\ &\approx 3 \times 10^{-3} \text{ H} = 3\text{mH} \end{aligned} \quad (B.7)$$

Finalmente, los datos de diseño del filtro quedan concentrados en la Tabla B.2

Tabla B.2: Parámetros iniciales de diseño

Concentrado de parámetros para inicial el diseño del filtro.			
Parámetro	Nomenclatura	valor	Unidad
Frecuencia de portadoras	F_c	2400	Hz
Número de portadoras	N_c	4	–
Frecuencia de rizo	F_{rizo}	9600	Hz
Resistencia	R	75	Ω
Inductor	L	3	mH
Capacitancia	C	10	μF

B.1.3. Prueba en simulación del filtro de segundo orden con el software PSIM 9

Con la finalidad de demostrar el desempeño del filtro obtenido, se probó el comportamiento del mismo con la ayuda del software **PSIM**, el esquemático del inversor se muestra en la Figura B.2 y en la Figura B.3 se muestra la comparativa entre la salida del inversor antes del filtro (a) y después del filtro (b), en el recuadro de la izquierda de la misma figura se nota que el valor de THD de la señal de voltaje en la salida (V_{pwm}) es hasta de un 26,9%; después del filtro se reduce de manera considerable esta parámetro hasta un $THDv = 0,206\%$, por lo tanto, se concluye que el cálculo arroja resultados dentro de los parámetros esperados.

Partiendo de los resultados anteriores ahora se diseñara el inductor considerando un valor de

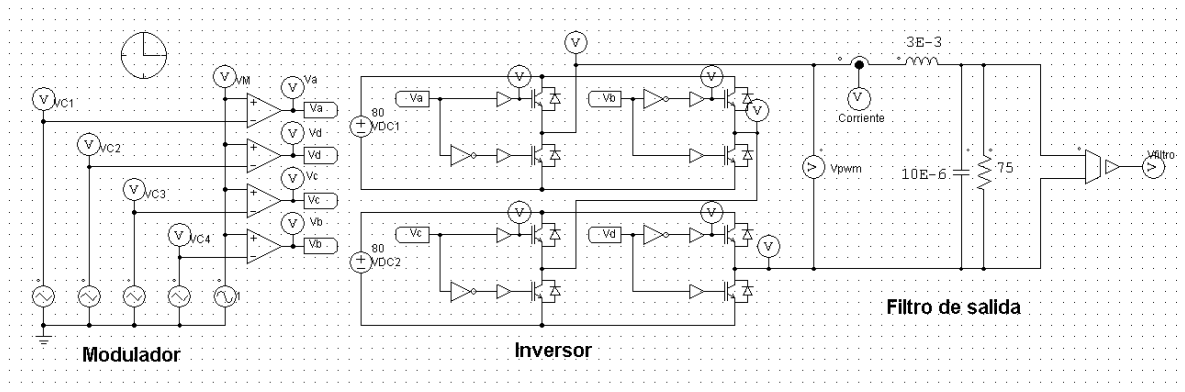


Figura B.2: Esquemático construido en PSIM para probar el desempeño del filtro de salida.

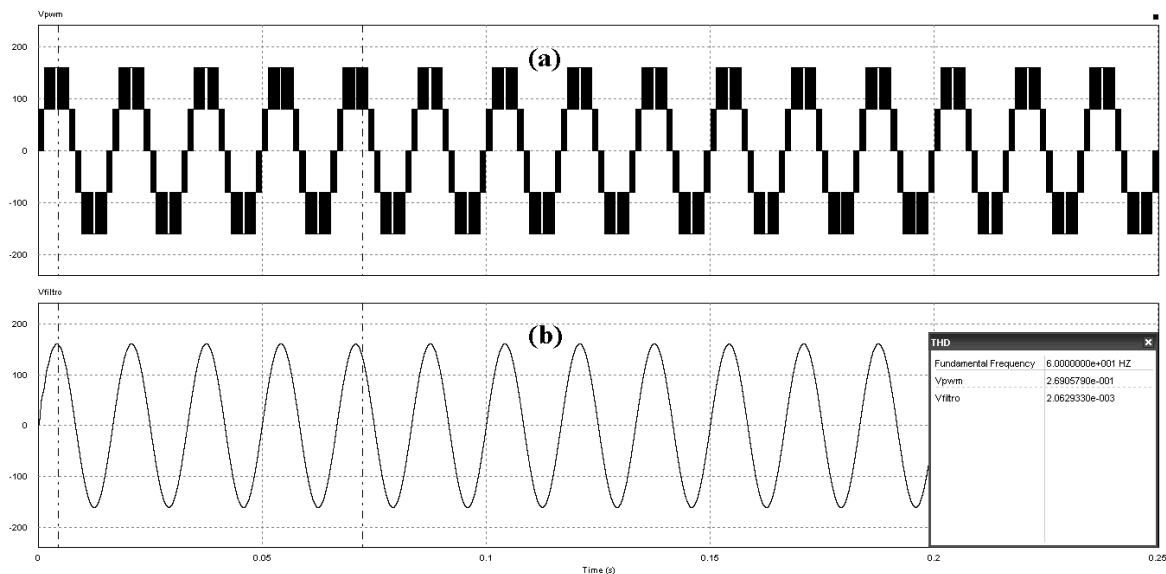


Figura B.3: Resultado de la simulación en PSIM: (a) Salida del convertidor antes del filtro; (b) Salida del convertidor después del filtro diseñado.

$L = 3mH$, para esto, se seguirá el procedimiento mostrado en [134] para el diseño de bobinas para AC (Cap. 10).

B.1.4. Diseño del inductor para AC

El diseño de un inductor para CA es muy similar a la de un transformador. Si en la bobina no existen componentes de CD, los cálculos del diseño son relativamente sencillos.

En algunas aplicaciones, el valor de la inductancia es especificada, y en otros casos, se especifica la corriente, por lo tanto, si la corriente es especificada, entonces el valor de la inductancia será calculada o recíprocamente.

Para el caso de este diseño, el valor de la inductancia necesaria para cumplir los requerimientos del diseño del filtro es de 3mH, por lo que en este caso se calculará el valor de la corriente en el inductor a partir de los datos concentrados en la tabla B.3:

Tabla B.3: Parámetros para el diseño del inductor de AC

Valores de diseño para la construcción física del inductor.			
Parámetro	Nomenclatura	valor	Unidad
Inductancia	$L_{deseada}$	3	mH
Voltaje	V_{ac}	160	V
Frecuencia de diseño	f	2400	Hz
Densidad de corriente	J	300	Amp/cm^2
Densidad de Flujo	B_{ac}	0,5, propio de la ferrita	Teslas
Factor de utilización de ventana	K_u	0.2	%
Factor de forma	K_f	4.44	--

Paso 1.- Cálculo de la reactancia inductiva

Para iniciar con el diseño del inductor se realiza el cálculo de la reactancia inductiva de acuerdo con la Ecuación B.8 de la manera siguiente:

$$\begin{aligned}
 L &= \frac{X_L}{2\pi f} \therefore & (B.8) \\
 X_L &= 2\pi f L \\
 &= 2\pi(2400)(3 \times 10^{-3}) \\
 &= 45,2389 \Omega
 \end{aligned}$$

Paso 2.- Cálculo de la corriente en la bobina

$$\begin{aligned}X_L &= \frac{V_L}{I_L} \therefore & (B.9) \\I_L &= \frac{V_L}{X_L} \\&= \frac{160 \text{ V}}{45,2389 \Omega} \\&= 3,536 \text{ Amp.}\end{aligned}$$

Paso 3.- Cálculo de la potencia aparente

$$\begin{aligned}P_t &= VA = V_L I_L & (B.10) \\&= (160 \text{ V})(3,536 \text{ A}) \\&= 560 \text{ Watts}\end{aligned}$$

Paso 4.- Cálculo del área de ventana

Para este cálculo, los valores de K_f , K_u , B_{ac} , f y J fueron extraídos de la Tabla B.3, por tanto

$$\begin{aligned}A_p &= \frac{VA(10^4)}{K_f K_u B_{ac} f J} & (B.11) \\&= \frac{560 \text{ Watts}(10^4)}{(0,2)(4,44)(0,5 \text{ T})(2400 \text{ Hz})(300 \text{ amp/cm}^2)} \\&= 17,518 \text{ cm}^4\end{aligned}$$

Paso 5.- Selección del núcleo a utilizar

Tomando como base el valor del área de ventana obtenido en el paso anterior, comparando dicho valor en la tabla mostrada en la Figura B.4, verificando en la columna correspondiente a A_p , se toma el valor inmediato superior, para este caso corresponde al núcleo modelo **ETD-59**, se obtienen los datos de esta tabla y se complementan con los de la tabla de la Figura B.5; estos datos son necesarios para continuar con el cálculo, en la Figura B.6 se muestra la geometría del núcleo elegido y en la Tabla B.4 se muestran los datos obtenidos de las tablas mencionadas y se concentran los datos que son necesarios para el diseño.

ETD, Ferrite Cores (Ferroxcube)											
Part No.	W_{tCu} grams	W_{tFe} grams	MLT cm	MPL cm	W_a	A_c cm ²	W_a cm ²	A_p cm ⁴	K_g cm ⁵	A_t cm ²	*AL mh/1K
					A_c						
ETD-29	32.1	28.0	6.4	7.20	1.865	0.761	1.419	1.0800	0.0517	42.5	1000
ETD-34	43.4	40.0	7.1	7.87	1.757	0.974	1.711	1.6665	0.0911	53.4	1182
ETD-39	69.3	60.0	8.3	9.22	1.871	1.252	2.343	2.9330	0.1766	69.9	1318
ETD-44	93.2	94.0	9.4	10.30	1.599	1.742	2.785	4.8520	0.3595	87.9	1682
ETD-49	126.2	124.0	10.3	11.40	1.627	2.110	3.434	7.2453	0.5917	107.9	1909
ETD-54	186.9	180.0	11.7	12.70	1.609	2.800	4.505	12.6129	1.2104	133.7	2273
ETD-59	237.7	260.0	12.9	13.90	1.410	3.677	5.186	19.0698	2.1271	163.1	2727

*This AL value has been normalized for a permeability of 1K. For a close approximation of AL for other values of permeability, multiply this AL value by the new permeability in kilo-perm. If the new permeability is 2500, then use 2.5.

Figura B.4: Parámetros de los núcleos de ferrita del fabricante Ferroxcube (Tabla 1).

ETD, Ferrite Cores (Ferroxcube)													
Part No.	A cm	B cm	C cm	D cm	E cm	G cm	Part No.	A cm	B cm	C cm	D cm	E cm	G cm
ETD-29	3.060	2.270	3.160	0.980	0.980	2.200	ETD-49	4.980	3.610	4.940	1.670	1.670	3.540
ETD-34	3.500	2.560	3.460	1.110	1.110	2.360	ETD-54	5.450	4.120	5.520	1.890	1.890	4.040
ETD-39	4.000	2.930	3.960	1.280	1.280	2.840	ETD-59	5.980	4.470	6.200	2.165	2.165	4.500
ETD-44	4.500	3.250	4.460	1.520	1.520	3.220							

Figura B.5: Parámetros de los núcleos de ferrita del fabricante Ferroxcube (Tabla 2).

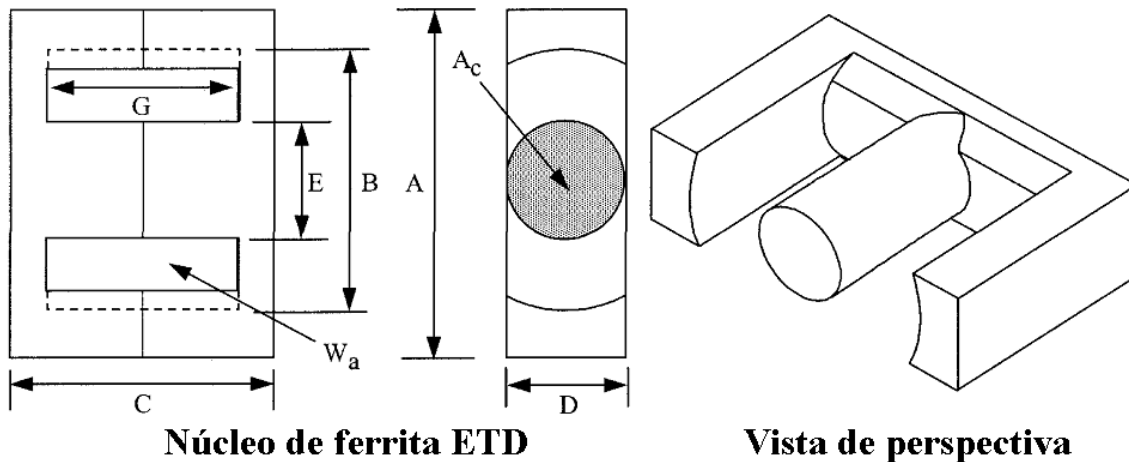


Figura B.6: Geometría del núcleo elegido para el diseño (Familia ETD).

Tabla B.4: Parámetros para el diseño del inductor de AC (2)

Parámetros del núcleo ETD-59 para el diseño y construcción física del inductor.			
Parámetro	Nomenclatura	Valor	Unidad
Modelo del núcleo		ETD-59	
Longitud de ruta magnética	MPL	13.9	cm
Peso del núcleo	W_{tfe}	260	gramos
Longitud promedio de vuelta	MLT	12.9	cm
Área del material	A_c	3.677	cm^2
Área de ventana	W_a	5,186	cm^2
Producto de áreas	A_p	19.0698	cm^4
Geometría del núcleo	K_g	2.1271	cm^5
Área de la superficie	A_t	163.1	cm^2
Longitud de bobinado	G	4.5	cm^2
Lengua	E	2.165	cm

Paso 6.- Cálculo del número de vueltas

$$\begin{aligned}
 N_L &= \frac{V_L(10^4)}{K_f B_{ac} f A_C} & (B.12) \\
 &= \frac{160 V(10^4)}{(4,44)(0,5 T)(2400 Hz)(3,677 cm^2)} \\
 &= 81,66 Vueltas \\
 &\approx 82 Vueltas
 \end{aligned}$$

Paso 7.- Cálculo del GAP del entrehierro

$$\begin{aligned}
 l_g &= \frac{0,4\pi N_L^2 A_C(10^{-8})}{L} & (B.13) \\
 &= \frac{(0,4)(\pi)(82 Vueltas)(3,677 cm^2)(10^{-8})}{3 \times 10^{-3} H} \\
 &= 0,1027 cm
 \end{aligned}$$

Paso 8.- Cálculo del borde de flujo

$$\begin{aligned} F &= \left(1 + \frac{l_g}{\sqrt[2]{A_C}} \ln \frac{2(G)}{l_g}\right) \\ &= \left(1 + \frac{0,1027 \text{ cm}}{\sqrt[2]{3,677 \text{ cm}^2}} \ln \frac{(2)(4,5 \text{ cm})}{0,1027 \text{ cm}}\right) \\ &= 1,2396 \end{aligned} \quad (\text{B.14})$$

Paso 9.- Recálculo del número de vueltas considerando el borde del flujo

$$\begin{aligned} N_{L(\text{new})} &= \sqrt[2]{\frac{l_g L}{0,4\pi A_C F (10^{-8})}} \\ &= \sqrt[2]{\frac{(0,1027 \text{ cm})(3 \times 10^{-3} \text{ H})}{0,4\pi (3,677 \text{ cm}^2)(1,2396)(10^{-8})}} \\ &= 73,35 \text{ Vueltas} \end{aligned} \quad (\text{B.15})$$

Paso 10.- Cálculo de la densidad de flujo

El valor propuesto en la Tabla B.4 para la densidad de flujo B_{ac} es de $B_{ac} = 0,5 \text{ T}$, por lo que el cálculo de este valor a partir de los datos ya obtenidos no debe ser superior para evitar problemas de saturación en el núcleo:

$$\begin{aligned} B_{ac} &= \frac{V_L(10^4)}{K_f N_{L(\text{new})} A_C f} \\ &= \frac{(160 \text{ Volts})(10^4)}{(4,44)(73,35 \text{ Vueltas})(3,677 \text{ cm}^2)(2400 \text{ Hz})} \\ &= 0,5567 \text{ Tesla} \end{aligned} \quad (\text{B.16})$$

A partir del resultado obtenido se concluye que el valor de la densidad de flujo no supera por mucho al propuesto al inicio del cálculo, por lo tanto, no debe de existir saturación del núcleo elegido.

Paso 11.- Cálculo del calibre del conductor a emplear

En esta sección se determinará el calibre del conductor a usar en el arrollamiento de la bobina, este parámetro está definido por dos cantidades: la corriente deseada y el valor de la

densidad de corriente propuesta, matemáticamente se determina a partir de la ecuación B.17:

$$\begin{aligned} A_{wL(B)} &= \frac{I_L}{J}(10^3) && \text{(B.17)} \\ &= \frac{3,5 \text{ Amp}}{300 \text{ Amp/cm}^2} \\ &= 11,66666 \text{ cm}^2 \end{aligned}$$

Consultando la imagen B.7 donde se muestra la tabla sobre el calibre del conductor, el valor del área base se consigue al buscar un valor inmediato superior al obtenido, se busca la correspondencia en la columna numero 2 de la tabla en cuestión, en este caso, el conductor a emplear debe de ser igual o superior al calibre #16 AWG.

Con los datos obtenidos, se procede a construir la bobina.

Paso 12.- Resultados experimentales obtenidos del filtro diseñado

Una vez construida la bobina se procedió a realizar la prueba experimental de su funcionamiento en el filtro diseñado, esta prueba se enfocó en el desempeño de THDv y fue corroborado por medio de un Analizador de la calidad de la energía **HIOKI 3197**, los resultados se muestran por medio de las Figuras B.8 y B.9.

Wire Table												
AWG	Bare Area		Resistance $\mu\Omega/\text{cm}$ 20°C	Heavy Synthetics								
	$\text{cm}^2(10^{-3})$	cir-mil		Area		Diameter		Turns-Per		Turns-Per		Weight gm/cm
			$\text{cm}^2(10^{-3})$	cir-mil	cm	Inch	cm	Inch	cm^2	Inch^2		
1	2	3	4	5	6	7	8	9	10	11	12	13
10	52.6100	10384.00	32.7	55.9000	11046.00	0.2670	0.105	3.9	10	11	69	0.46800
11	41.6800	8226.00	41.4	44.5000	8798.00	0.2380	0.094	4.4	11	13	90	0.37500
12	33.0800	6529.00	52.1	35.6400	7022.00	0.2130	0.084	4.9	12	17	108	0.29770
13	26.2600	5184.00	65.6	28.3600	5610.00	0.1900	0.075	5.5	13	21	136	0.23670
14	20.8200	4109.00	82.8	22.9500	4556.00	0.1710	0.068	6.0	15	26	169	0.18790
15	16.5100	3260.00	104.3	18.3700	3624.00	0.1530	0.060	6.8	17	33	211	0.14920
16	13.0700	2581.00	131.8	14.7300	2905.00	0.1370	0.054	7.3	19	41	263	0.11840
17	10.3900	2052.00	165.8	11.6800	2323.00	0.1220	0.048	8.2	21	51	331	0.09430
18	8.2280	1624.00	209.5	9.3260	1857.00	0.1090	0.043	9.1	23	64	415	0.07474
19	6.5310	1289.00	263.9	7.5390	1490.00	0.0980	0.039	10.2	26	80	515	0.05940
20	5.1880	1024.00	332.3	6.0650	1197.00	0.0879	0.035	11.4	29	99	638	0.04726
21	4.1160	812.30	418.9	4.8370	954.80	0.0785	0.031	12.8	32	124	800	0.03757
22	3.2430	640.10	531.4	3.8570	761.70	0.0701	0.028	14.3	36	156	1003	0.02965
23	2.5880	510.80	666.0	3.1350	620.00	0.0632	0.025	15.8	40	191	1234	0.02372
24	2.0470	404.00	842.1	2.5140	497.30	0.0566	0.022	17.6	45	239	1539	0.01884
25	1.6230	320.40	1062.0	2.0020	396.00	0.0505	0.020	19.8	50	300	1933	0.01498
26	1.2800	252.80	1345.0	1.6030	316.80	0.0452	0.018	22.1	56	374	2414	0.01185
27	1.0210	201.60	1687.0	1.3130	259.20	0.0409	0.016	24.4	62	457	2947	0.00945
28	0.8046	158.80	2142.0	1.0515	207.30	0.0366	0.014	27.3	69	571	3680	0.00747
29	0.6470	127.70	2664.0	0.8548	169.00	0.0330	0.013	30.3	77	702	4527	0.00602
30	0.5067	100.00	3402.0	0.6785	134.50	0.0294	0.012	33.9	86	884	5703	0.00472
31	0.4013	79.21	4294.0	0.5596	110.20	0.0267	0.011	37.5	95	1072	6914	0.00372
32	0.3242	64.00	5315.0	0.4559	90.25	0.0241	0.010	41.5	105	1316	8488	0.00305
33	0.2554	50.41	6748.0	0.3662	72.25	0.0216	0.009	46.3	118	1638	10565	0.00241
34	0.2011	39.69	8572.0	0.2863	56.25	0.0191	0.008	52.5	133	2095	13512	0.00189
35	0.1589	31.36	10849.0	0.2268	44.89	0.0170	0.007	58.8	149	2645	17060	0.00150
36	0.1266	25.00	13608.0	0.1813	36.00	0.0152	0.006	62.5	167	3309	21343	0.00119
37	0.1026	20.25	16801.0	0.1538	30.25	0.0140	0.006	71.6	182	3901	25161	0.00098
38	0.0811	16.00	21266.0	0.1207	24.01	0.0124	0.005	80.4	204	4971	32062	0.00077
39	0.0621	12.25	27775.0	0.0932	18.49	0.0109	0.004	91.6	233	6437	41518	0.00059
40	0.0487	9.61	35400.0	0.0723	14.44	0.0096	0.004	103.6	263	8298	53522	0.00046
41	0.0397	7.84	43405.0	0.0584	11.56	0.0086	0.003	115.7	294	10273	66260	0.00038
42	0.0317	6.25	54429.0	0.0456	9.00	0.0076	0.003	131.2	333	13163	84901	0.00030
43	0.0245	4.84	70308.0	0.0368	7.29	0.0069	0.003	145.8	370	16291	105076	0.00023
44	0.0202	4.00	85072.0	0.0316	6.25	0.0064	0.003	157.4	400	18957	122272	0.00020

Figura B.7: Calibre AWG del conductor para el bobinado del inductor.

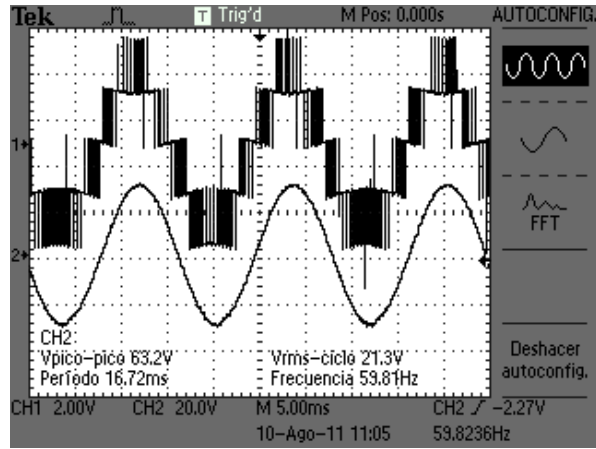


Figura B.8: Salida del convertidor multinivel después del filtro de segundo orden, esta gráfica fue obtenida por medio del osciloscopio *Tektronix*.

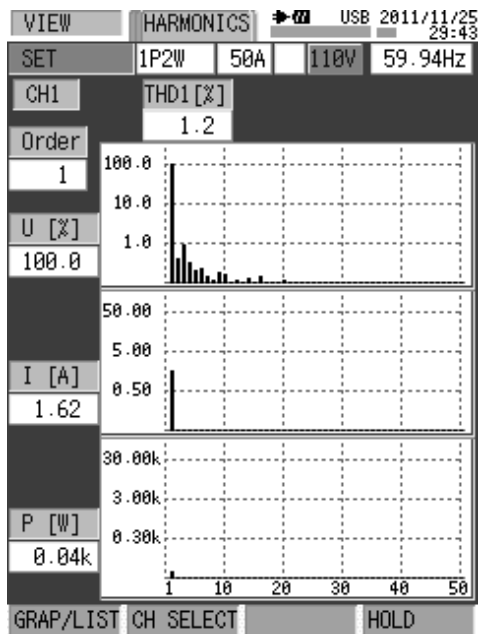


Figura B.9: Análisis de **THD_v** con el Analizador **HIOKI 3197**.

Apéndice C

Código empleado en SIMNON

Lista C.1: Descripción del modelo matemático del convertidor multinivel y controlador GPI en SIMNON

```

1 CONTINUOUS SYSTEM GPIBUCK
2 " Version:      1.0
3 " Abstract:
4 " Description:
5 " Revision:    1.0
6 " Author:      José Antonio Juárez Abad
7 " Created:     08/11/2011
8 "-----
9 " States, derivates and time:
10 STATE   vC  iL  lamda  etha  u
11 DER     dvC diL dlamda detha du
12 TIME   t
13 "-----
14 "CAMBIO DE CARGA PROGRAMADO
15 "-----
16 Ra= if (t>0.1) then 15 else 30
17 "-----
18 "MODELO DINAMICO DEL CONVERTIDOR
19 "-----
20 diL=(-vC+E*Uav)/L
21 dvC=(iL-(vC/Ra))/C
22 dlamda=ev
23 detha=lamda
24 "-----
25 "FUNCIONES DE REFERENCIA
26 "-----
27 Fr=A*SIN(wn1*t)
28 dFr=A*wn1*COS(wn1*t)
29 ddFr=-A*wn1*wn1*sin(wn1*t)
30 wn1=2*3.141592*frec
31 "-----
32 "CONTROLADOR GPI
33 "-----
34 F=vC
35 ev=F-Fr
36 "-----
37 Uav=aux1*Vaux+aux2*dFest+aux3*F
38 aux1=(L*C)/E
39 aux2=L/(E*R)
40 aux3=1/E
41 edF=dFest-dFr
42 "-----
43 "CONTROL ARTIFICIAL
44 "-----
45 Vaux=ddFr-k3*(edF)-k2*(ev)-k1*(lamda)-k0*(etha)
46 "-----
47 "ESTIMADOR DE ESTADO
48 "-----
49 du=Uav-aux5*F
50 dFest=aux4*u-aux6*F
51 aux4=E/(L*C)
52 aux5=1/E
53 aux6=1/(R*C)
54 "-----
55 " CONSTANTES PARA SINTONIA
56 "-----
57 k0=wn*wn*wn*wn
58 k1=4*phi*wn*wn*wn
59 k2=4*phi*phi*wn+2*wn*wn
60 k3=4*phi*wn
61 "-----
62 " PARAMETROS DE DISEÑO
63 "-----
64 L:3E-3
65 C:10E-6
66 R:75
67 E:160
68 phi:0.707
69 wn:1300
70 frec:60
71 A:145
72 END

```


Apéndice D

Código descargado en la BRAM del dispositivo FPGA

Lista D.1: Código del contenido de la memoria empleado por el *Xilinx CORE Generator* (*.coe)

```
1 ;*****
2 ;*****      Single Port Block Memory .COE file      *****
3 ;*****
4 ; Sample memory initialization file for Single Port Block Memory,
5 ; v3.0 or later.
6 ; Autor: José Antonio Juárez Abad
7 ; This .COE file specifies initialization values for a block
8 ; memory of depth=1024, and width=32. In this case, values are
9 ; specified in hexadecimal format.
10 memory_initialization_radix=16;
11 memory_initialization_vector=
12 42a40000,
13 42a3ae00,
14 42a35c00,
15 42a30a00,
16 42a2b800,
17 42a26600,
18 42a21400,
19 42a1c200,
20 42a17000,
21 42a11e00,
22 42a0cc00,
23 42a07a00,
24 42a02800,
25 ...
26 0
27 ...
28 c2a0cc00,
29 c2a11e00,
30 c2a17000,
31 c2a1c200,
32 c2a21400,
33 c2a26600,
34 c2a2b800,
35 c2a30a00,
36 c2a35c00,
37 c2a3ae00;
38 ;*****
```

Apéndice E

Descripción de los diagramas *top-down* de la implementación en FPGA

E.1. Diagrama RTL

Como resultado del proceso de síntesis se presenta el diagrama **RTL** (**RTL**, *Register Transfer Level*) mostrado por la figura [E.1](#) que fue obtenido de la herramienta EDA ISE Design Suite 9.2.

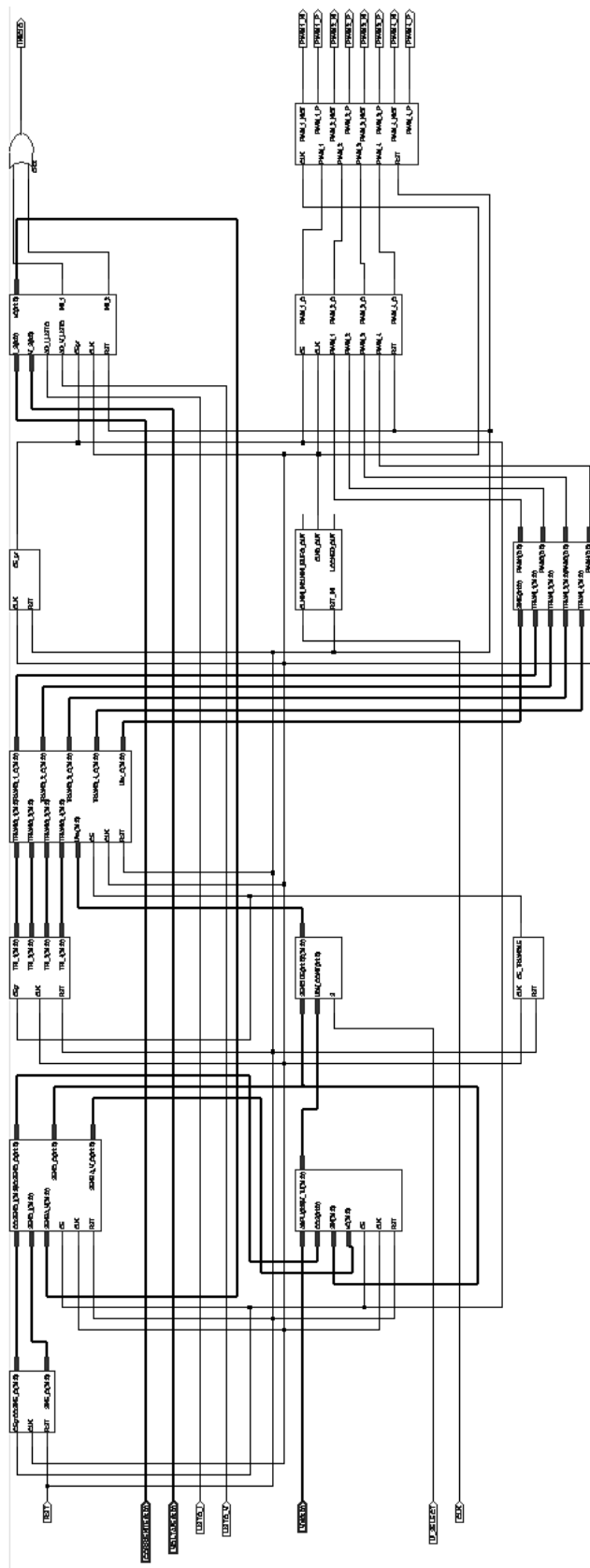


Figura E.1: Diagrama RTL de la implementación del controlador después del proceso de síntesis.

Diagrama general (Top-Down) del diseño

Descripción de los Módulos del Diagrama General del diseño

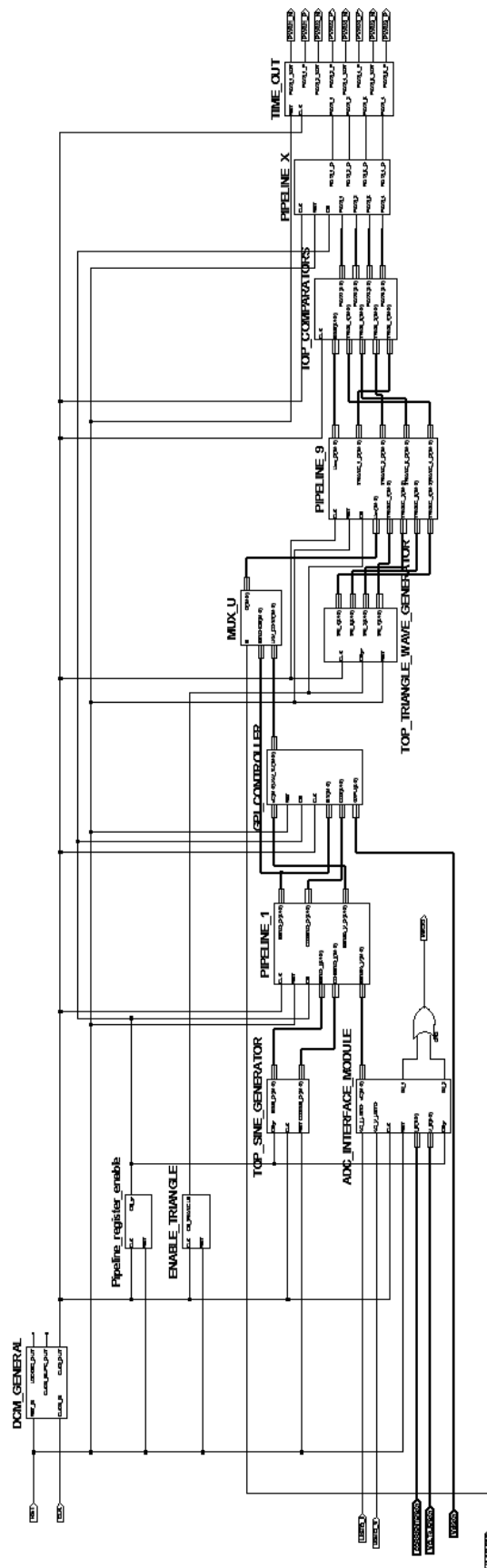


Figura E.2: Diagrama general del diseño con un enfoque *Top-Down* realizado en *Xilinx ISE Design Suite 9.2*.

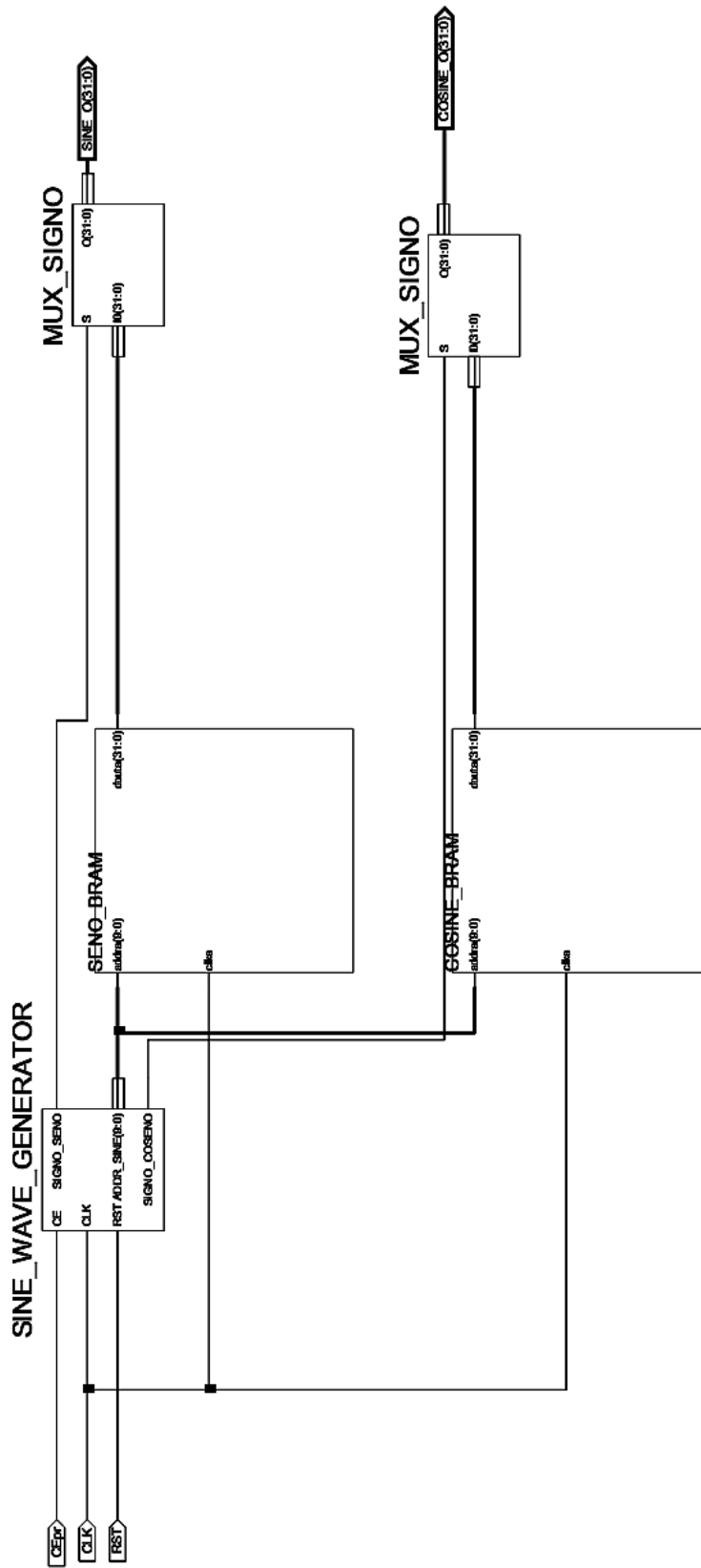


Figura E.3: Diagrama del generador de onda senoidal.

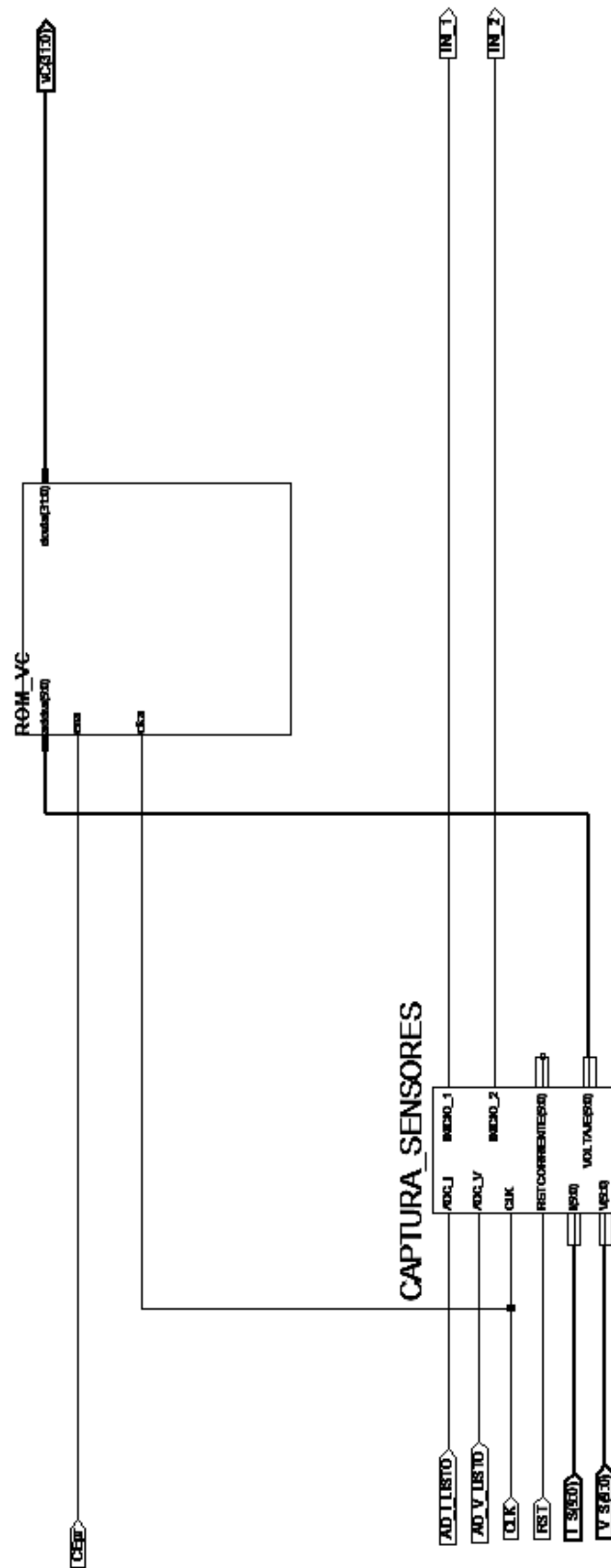


Figura E.4: Diagrama del módulo de conversión analógico-digital.

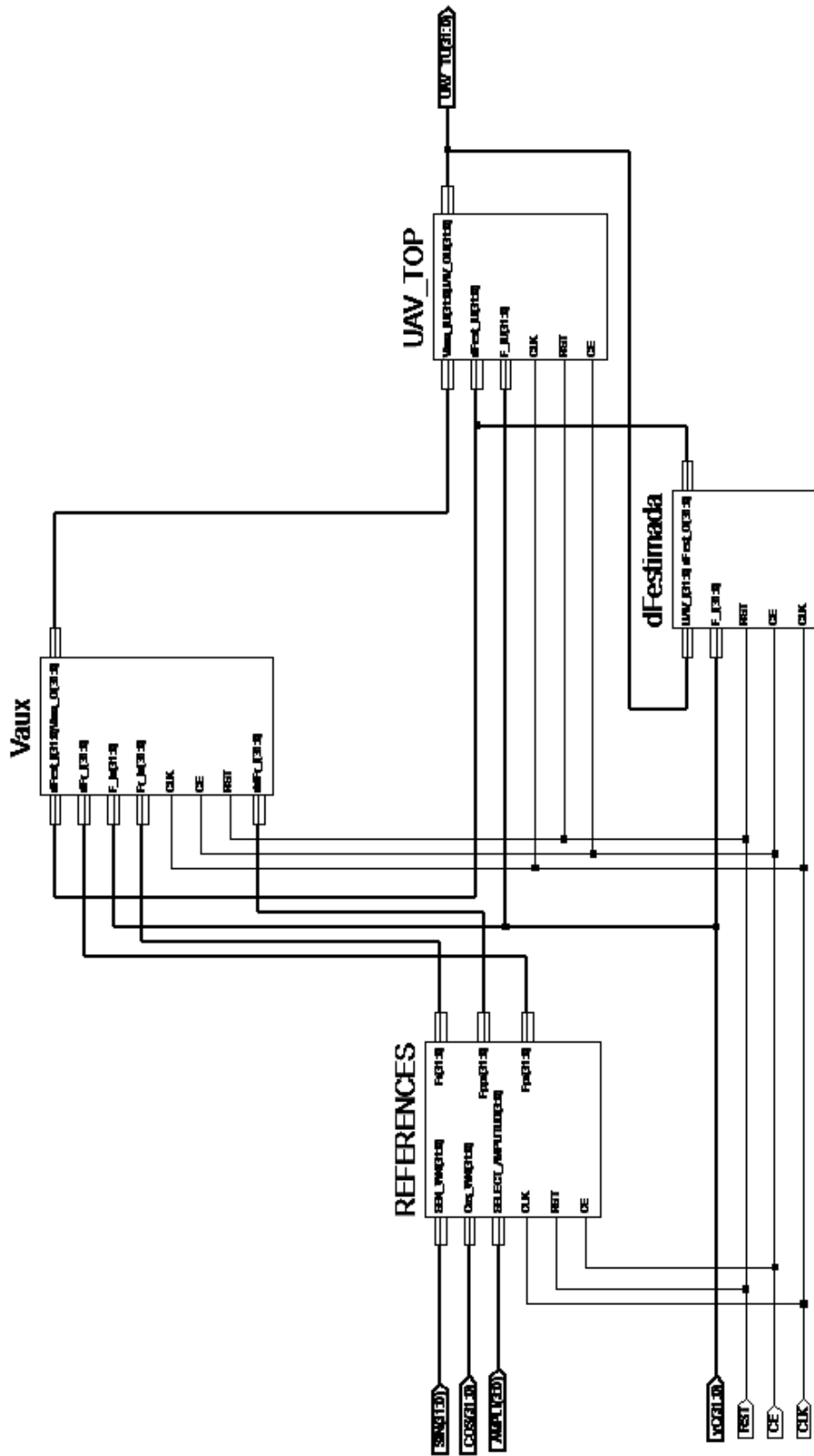


Figura E.5: Diagrama del módulo del controlador GPI.

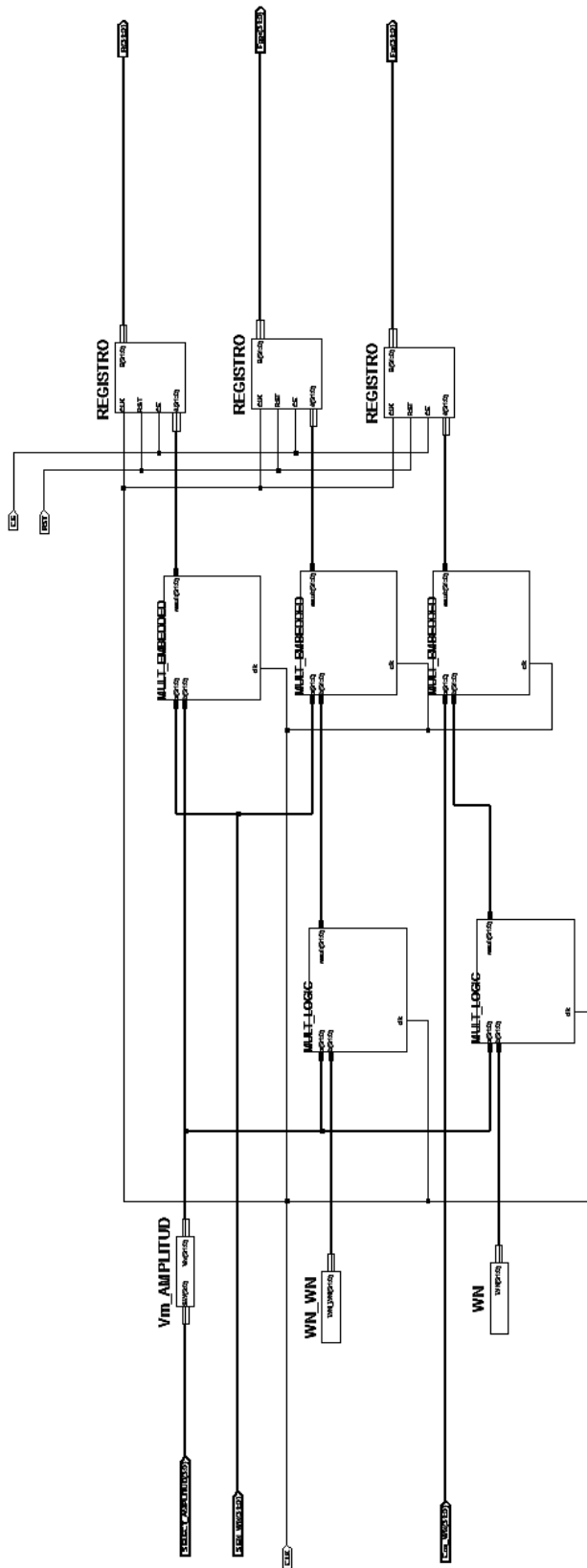


Figura E.6: Diagrama del generador de las referencias de seguimiento para el controlador GPI.

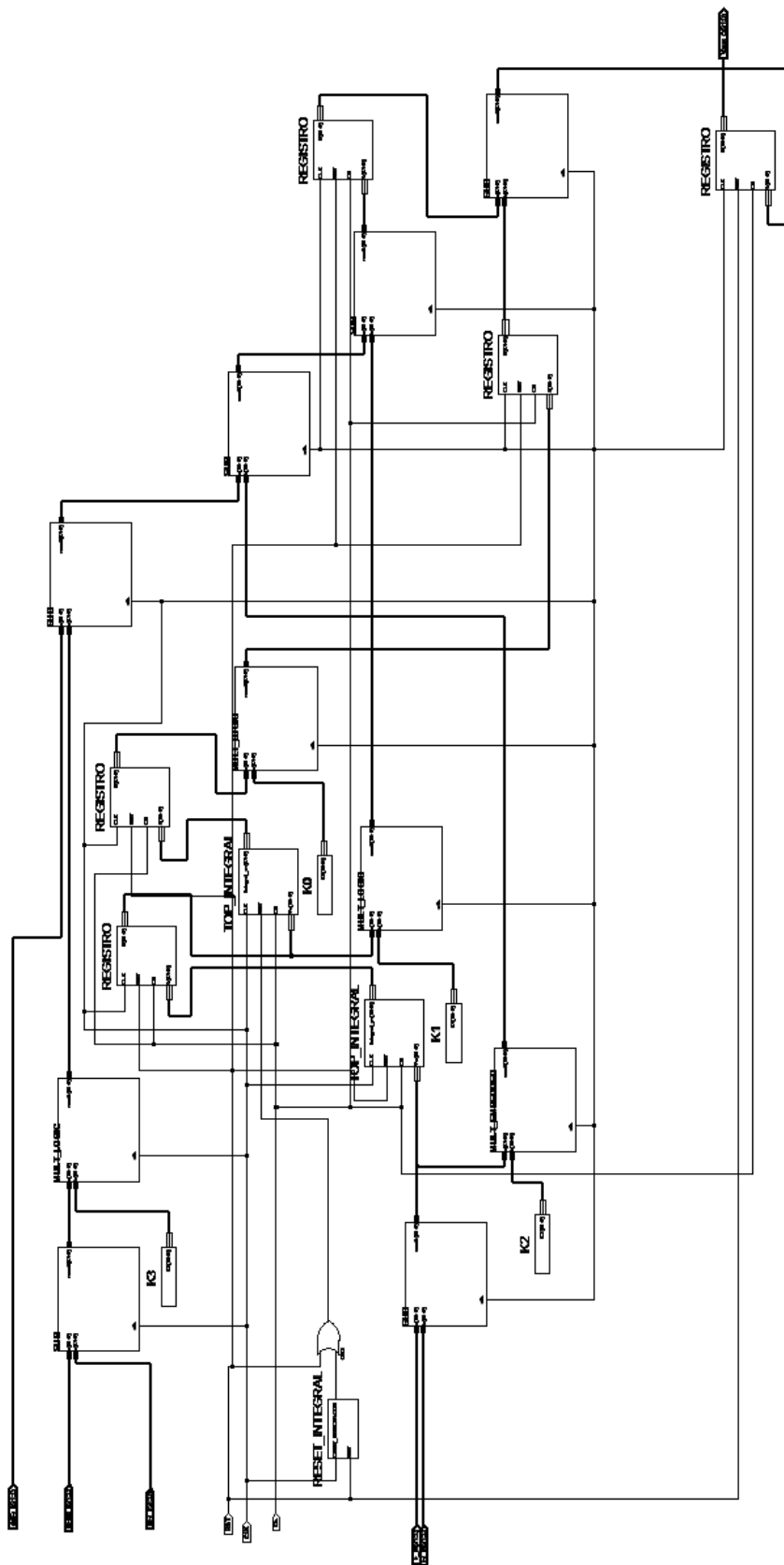


Figura E.7: Diagrama de la entrada de control artificial del controlador GPI.

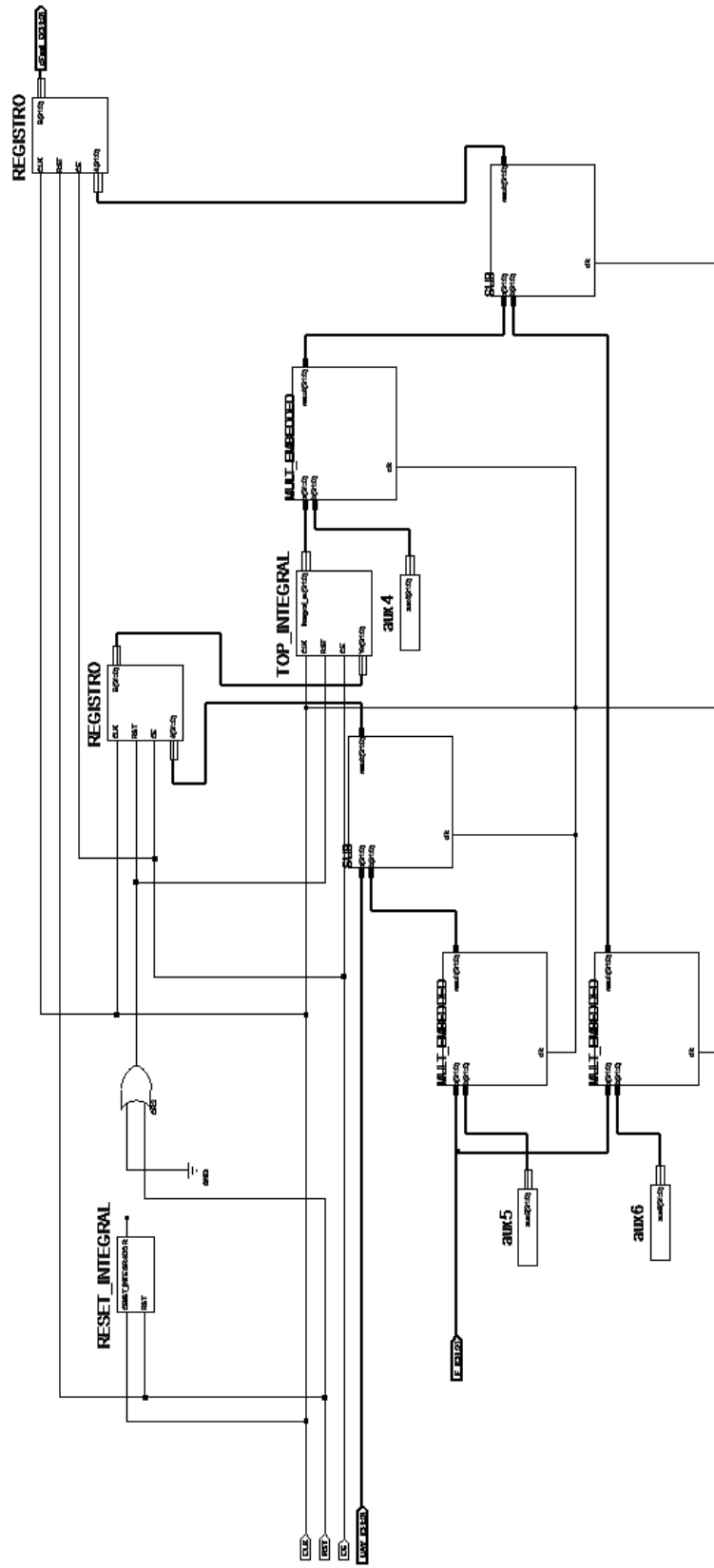


Figura E.8: Diagrama del reconstructor de estado del controlador GPI.

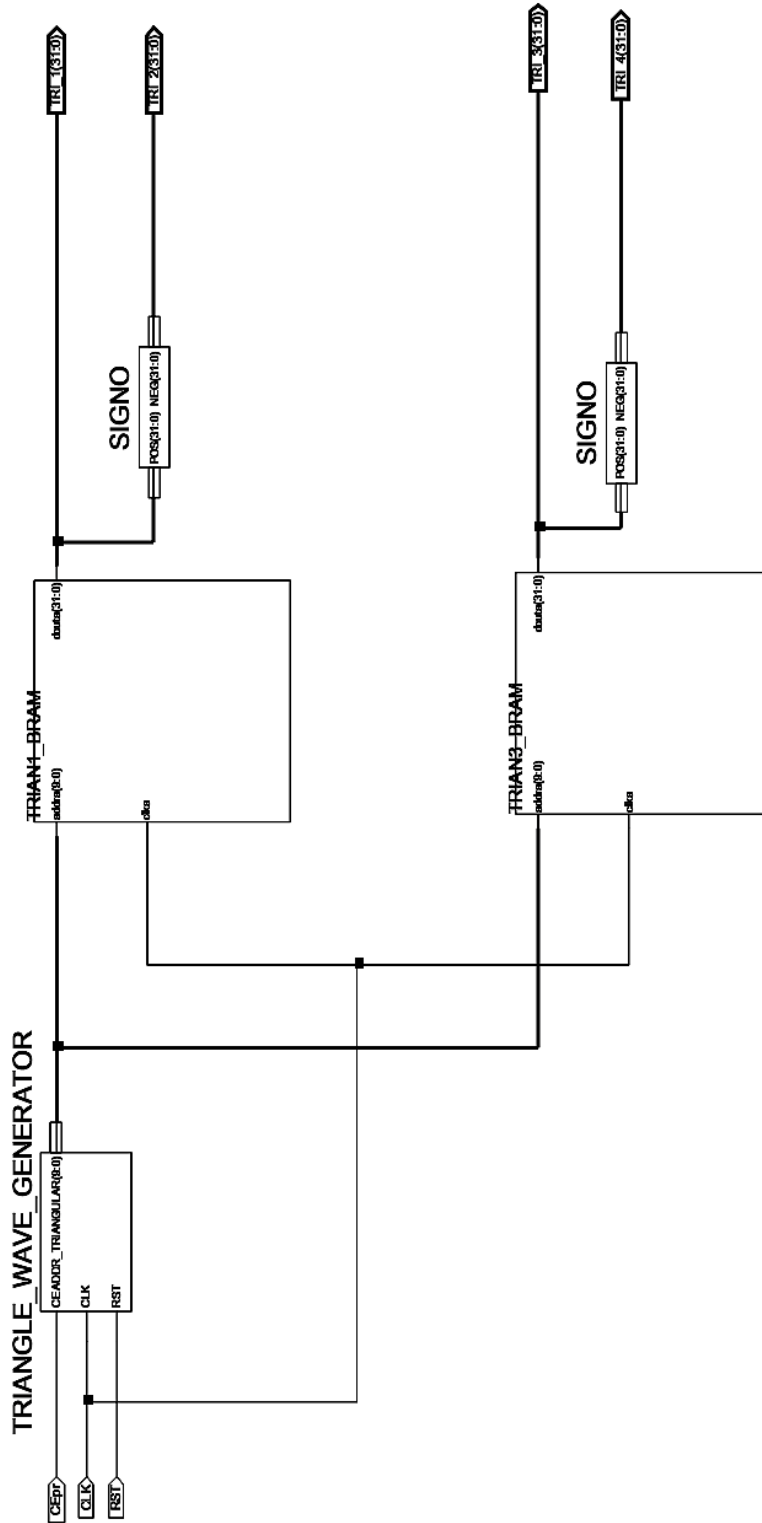


Figura E.9: Diagrama del Generador triangular del Modulador **SPWM**.

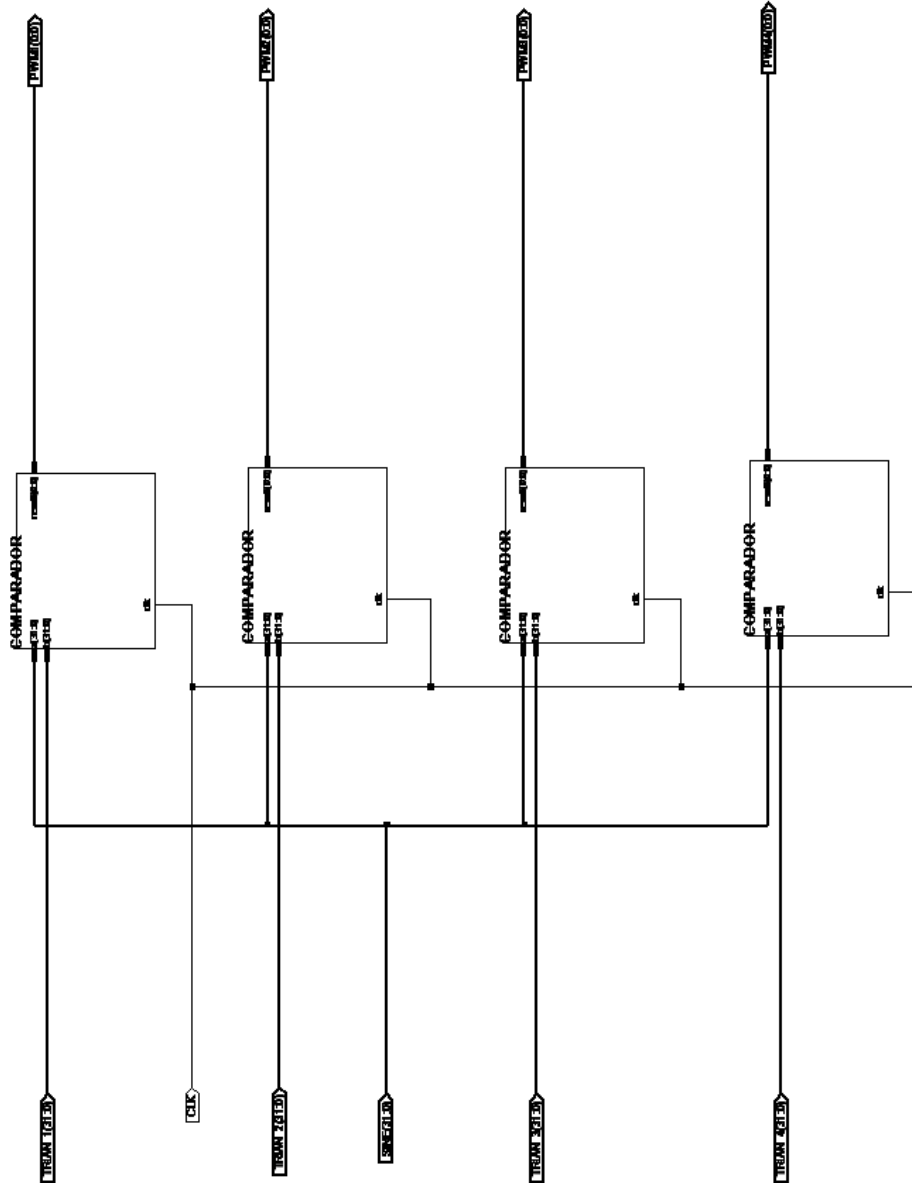


Figura E.10: Diagrama de los comparadores de magnitud del Modulador **SPWM**.

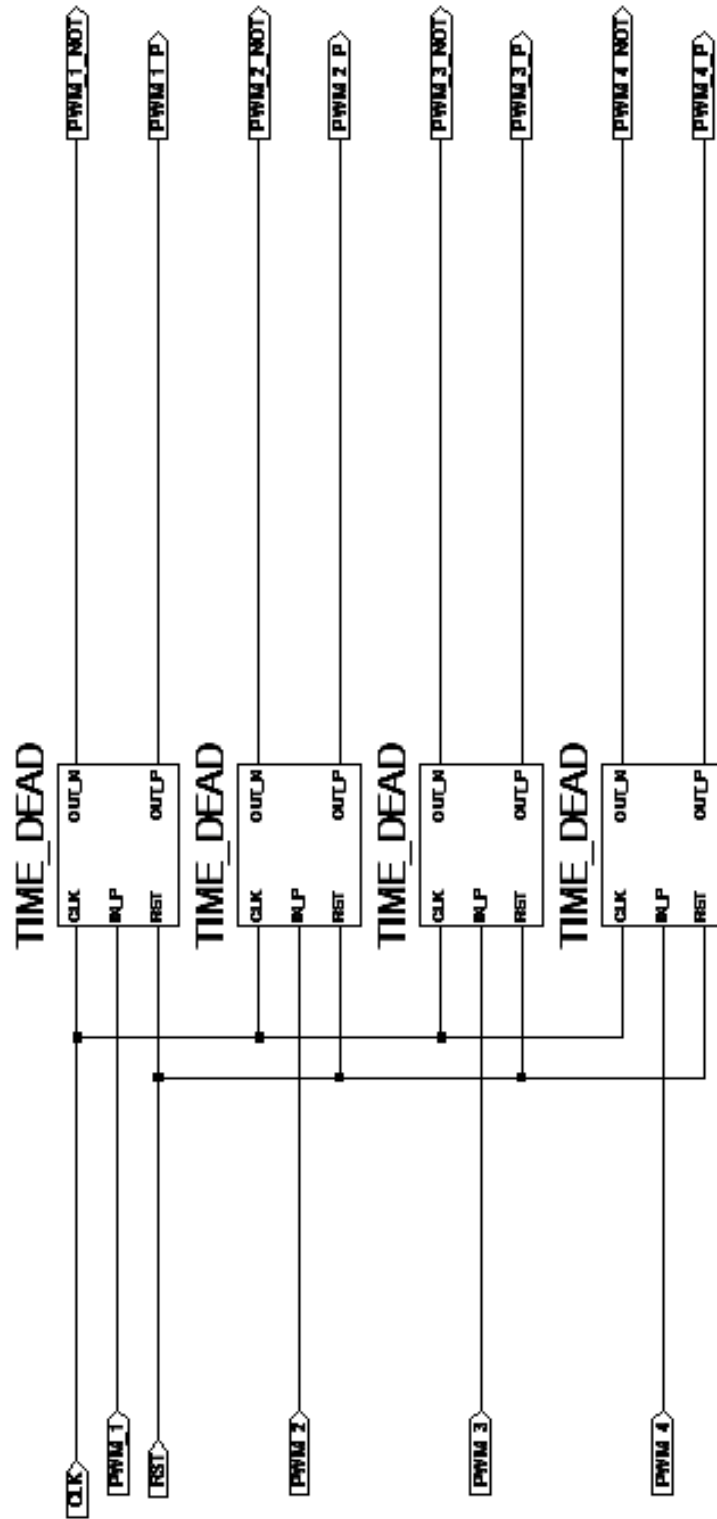


Figura E.11: Diagrama del generador de tiempo muerto del Modulador SPWM.

Apéndice F

Modulo de comunicación USB PC-FPGA y datos extraídos de la arquitectura propuesta

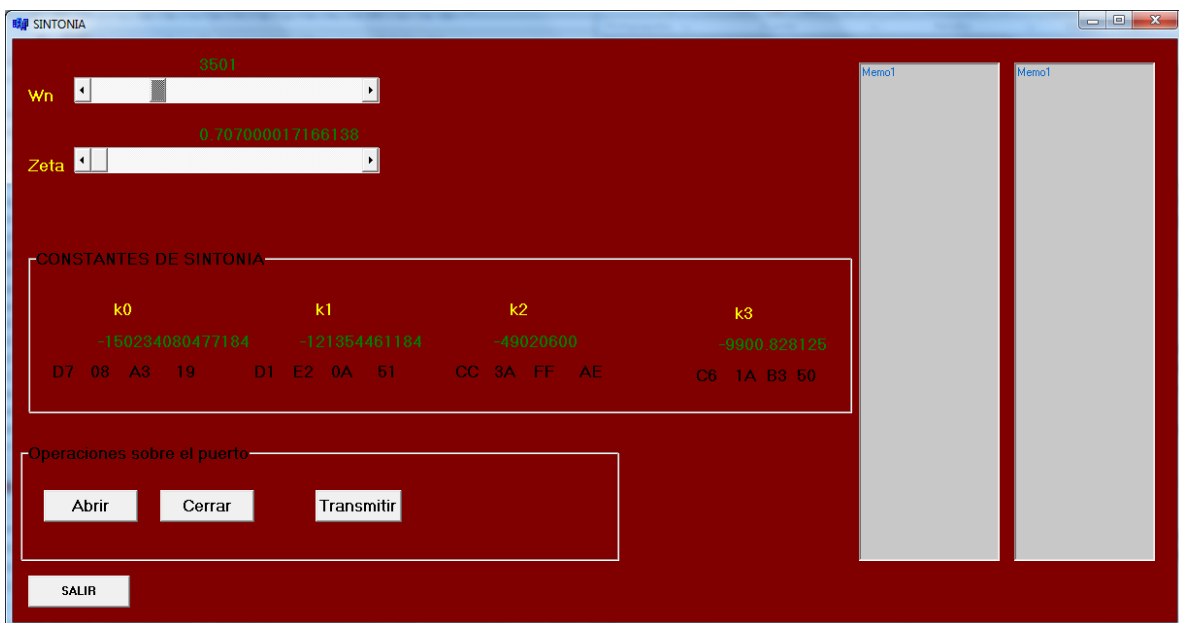


Figura F.1: Interfaz gráfica de transmisión/recepción de datos vía USB por medio del **DLP-USB245M**.

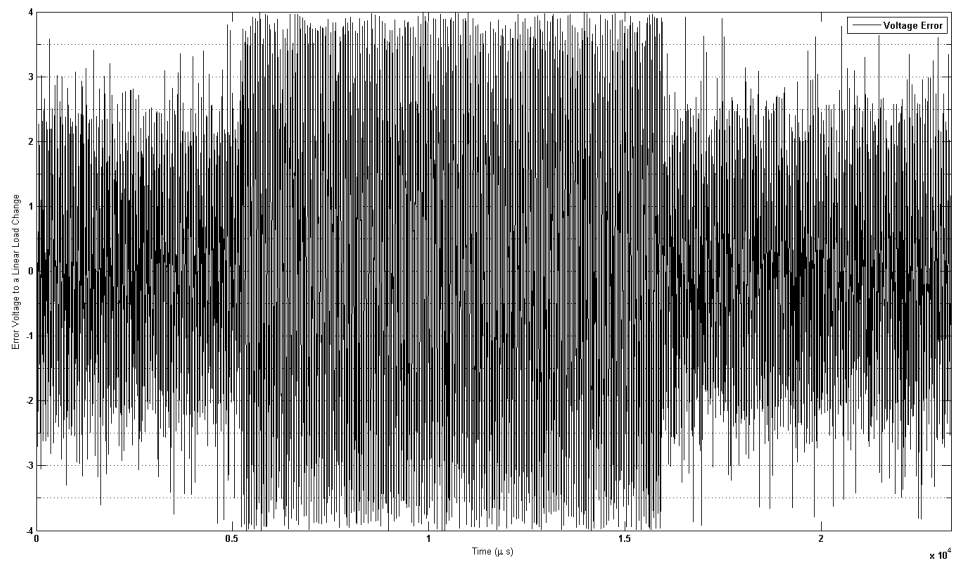


Figura F.2: Error de voltaje obtenido vía USB por medio del **DLP-USB245M**.

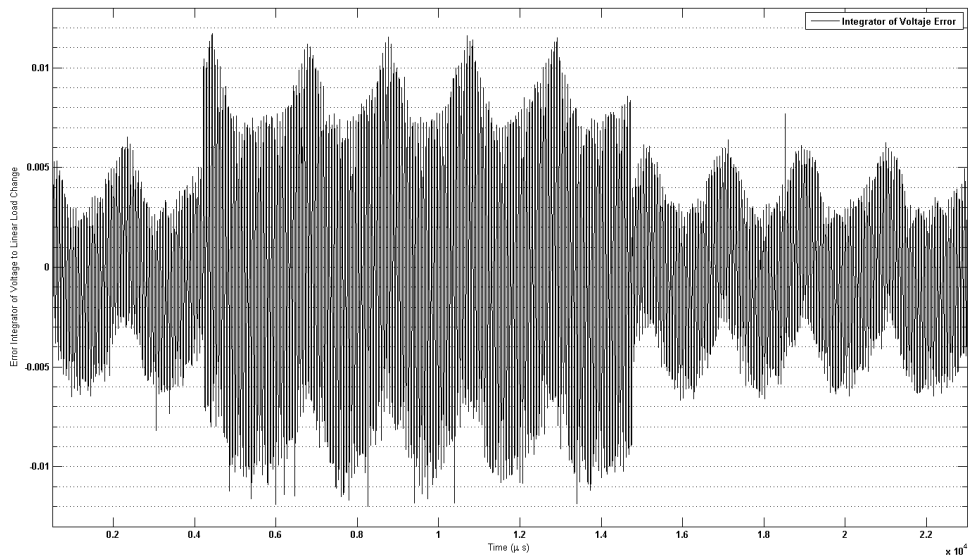


Figura F.3: Integral del Error de voltaje obtenido vía USB por medio del **DLP-USB245M**.

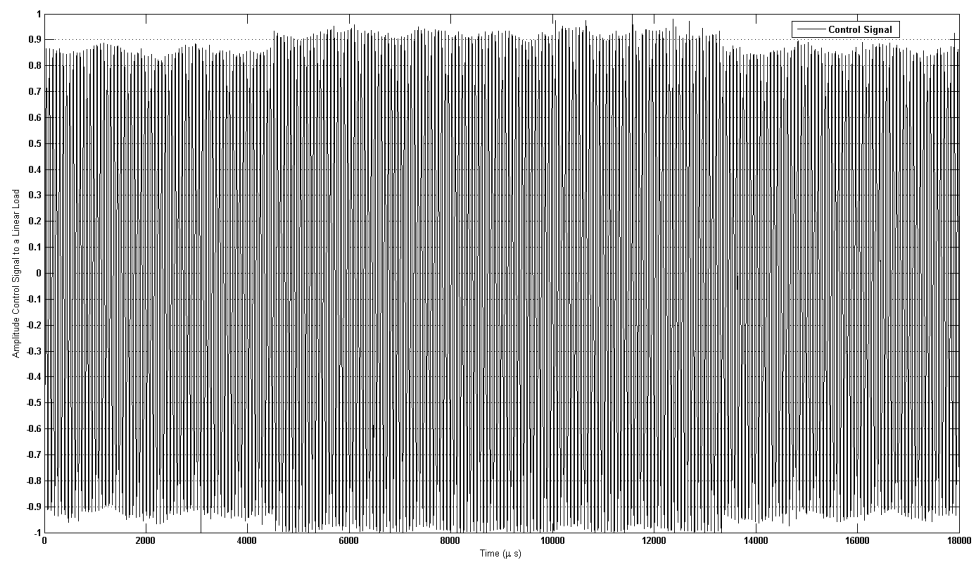


Figura F.4: Salida de control obtenida vía USB por medio del **DLP-USB245M**.

Apéndice G

Artículos Publicados

Bibliografía

- [1] Franquelo, L.G.; Rodriguez, J.; Leon, J.I.; Kouro, S.; Portillo, R.; Prats, M.A.M.; “The age of multilevel converters arrives ”, *Industrial Electronics Magazine, IEEE* , vol.2, no.2, pp.28-39, June 2008.
- [2] Christopher, E.; Sumner, M.; “An Interface for Renewable Energy Sources using a Multilevel Rectifier with Enhanced DC Link Control” , *Power Electronics, Machines and Drives*, 2006. The 3rd IET International Conference on, vol., no., pp.383-387, Mar. 2006.
- [3] Khomfoi, S.; Praisuwan, N.; “A hybrid cascaded multilevel inverter for interfacing with renewable energy resources” , *Power Electronics Conference (IPEC), 2010 International* , vol., no., pp.2912-2917, 21-24 June 2010.
- [4] Rodriguez, J.; Jih-Sheng Lai; Fang Zheng Peng; “Multilevel inverters: a survey of topologies, controls, and applications” , *Industrial Electronics, IEEE Transactions on* , vol.49, no.4, pp. 724- 738, Aug 2002.
- [5] Nabae, Akira; Takahashi, Isao; Akagi, Hirofumi; “A New Neutral-Point-Clamped PWM Inverter” , *Industry Applications, IEEE Transactions on* , vol.IA-17, no.5, pp.518-523, Sept. 1981.
- [6] M. Carpita, S. Teconi, “A Novel Multilevel Structure for voltage source inverter” , in *Proc. EPE 1991*, pp. 90-9.
- [7] Meynard, T.A.; Foch, H.; , “Multilevel conversion: high voltage choppers and voltage-source inverters” , *Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE*, vol., no., pp.397-403 vol.1, 29 Jun-3 Jul 1992.
- [8] Fang Zheng Peng; Jih-Sheng Lai; McKeever, J.; VanCoevering, J.; “A multilevel voltage-source inverter with separate DC sources for static VAR generation” , *Industry Applications Conference, 1995. Thirtieth IAS Annual Meeting, IAS '95., Conference Record of the 1995 IEEE*, vol.3, no., pp.2541-2548 vol.3, 8-12 Oct 1995.
- [9] Fang Zhang; Peng, F.Z.; Zhaoming Qian; , “Study of the multilevel converters in DC-DC applications” , *Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual*, vol.2, no., pp. 1702- 1706 Vol.2, 20-25 June 2004.
- [10] Wei Qian; Peng, F.Z.; Miaosen Shen; Tolbert, L.M.; , “3X DC-DC Multiplier/Divider for HEV Systems” , *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, vol., no., pp.1109-1114, 15-19 Feb. 2009.
- [11] A. K. Verma, P. R. Thakura, K.C.Jana y G.Buja, Fellow “Cascaded Multilevel Inverter for Hybrid Electric Vehicles” , *IEEE Power Electronics (IICPE), 2010 India International*, ISBN: 978-1-4244-7883-5, Mar. 2011.

- [12] LIEBHERR-ELEKTRONIK. "Onboard Power." dSPACE Magazine Vol. 3 (2010): 32-35.
- [13] Coronado, Ixtláhuatl, Pável Zúñiga y Juan M. Ramírez. "FACTS: Soluciones Modernas para la Industria Eléctrica." Avance y Perspectiva, UNAM 20 (2002): 235-244.
- [14] Peng, F.Z.; Jin Wang; "A universal STATCOM with delta-connected cascade multilevel inverter", Power Electronics Specialists Conference, 2004. PESC 04. 2004 IEEE 35th Annual , vol.5, no., pp. 3529- 3533 Vol.5, 20-25 June 2004.
- [15] Jin Wang; Peng, F.Z.; "Unified power flow controller using the cascade multilevel inverter", Power Electronics, IEEE Transactions on , vol.19, no.4, pp. 1077- 1084, July 2004.
- [16] Panagis, P.; Stergiopoulos, F.; Marabeas, P.; Manias, S.; "Comparison of state of the art multilevel inverters", Power Electronics Specialists Conference, 2008. PESC 2008. IEEE , vol., no., pp.4296-4301, 15-19 June 2008.
- [17] Fang Zheng Peng; "A generalized multilevel inverter topology with self voltage balancing", Industry Applications, IEEE Transactions on , vol.37, no.2, pp.611-618, Mar/Apr 2001.
- [18] Ceglia, G.; Guzman, V.; Sanchez, C.; Ibanez, F.; Walter, J.; Gimenez, M.I.; "A New Simplified Multilevel Inverter Topology for DC/AC Conversion", Power Electronics, IEEE Transactions on , vol.21, no.5, pp.1311-1319, Sept. 2006.
- [19] Gupta, R.; Ghosh, A.; Joshi, A.; "Multiband Hysteresis Modulation and Switching Characterization for Sliding-Mode-Controlled Cascaded Multilevel Inverter", Industrial Electronics, IEEE Transactions on , vol.57, no.7, pp.2344-2353, July 2010.
- [20] Dixon, J.; Moran, L.; Rodriguez, J.; Domke, R.; "Reactive Power Compensation Technologies: State-of-the-Art Review", Proceedings of the IEEE , vol.93, no.12, pp.2144-2164, Dec. 2005 "Recent advances in multilevel converter/inverter topologies and applications", Power Electronics Conference (IPEC), 2010 International , vol., no., pp.492-501, 21-24 June 2010.
- [21] Haiwen Liu; Tolbert, L.M.; Ozpineci, B.; Zhong Du; "Comparison of fundamental frequency and PWM methods applied on a hybrid cascaded multilevel inverter", Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE , vol., no., pp.3233-3237, 10-13 Nov. 2008.
- [22] Chinnaiyan, V. Kumar, y otros. "Control Techniques For Multilevel Voltage Source Inverters." The 8th International Power Engineering Conference (IPEC 2007) (2004): 1023-1028.
- [23] Liqiao Wang; Weiyang Wu; "FPGA Based Multichannel PWM Pulse Generator for Multi-modular Converters or Multilevel Converters", Power Electronics and Motion Control Conference, 2006. IPEMC 2006. CES/IEEE 5th International , vol.1, no., pp.1-5, 14-16 Aug. 2006.
- [24] Eunsoo Jung; Seung-Ki Sul; "Implementation of Grid-connected Single-phase Inverter Based on FPGA", Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE , vol., no., pp.889-893, 15-19 Feb. 2009.

- [25] Zhang Yang; Yin Zhong-dong; Shan Ren-zhong; Huang Tao; “Research on Cascade Multilevel Inverter Based on FPGA Control”, Energy and Environment Technology, 2009. ICEET '09. International Conference on , vol.2, no., pp.225-227, 16-18 Oct. 2009.
- [26] Azli, N.A.; Teng, L.Y.; Lim, P.Y.; “Implementation of a Single-carrier Multilevel PWM Technique Using Field Programmable Gate Array (FPGA)”, Power Electronics and Drive Systems, 2007. PEDS '07. 7th International Conference on , vol., no., pp.836-841, 27-30 Nov. 2007.
- [27] Grout, Ian . Digital Systems Design with FPGAs and CPLDs. United States of America:Elsevier Ltd, 2008.
- [28] Chu, Pong P. FPGA prototyping by VHDL examples Xilinx Spartan-3 Version.United States of America: John Wiley & Sons, 2008.
- [29] R. Wilson, Peter. Design Recipes for FPGAs. Great Britain: MPG Books Ltd, 2007.
- [30] Kilts, Steve. Advanced FPGA Design Architecture, Implementation and Optimization. United States of America: John Wiley & Sons, 2007.
- [31] Design Automation Standards Subcommittee of the Design Automation Technical Committee of the IEEE Computer Society. “IEEE Std 1076-1987. IEEE Standard VHDL Language Reference Manual”. Institute of Electrical and Electronics Engineers. 1988.
- [32] Design Automation Standards Committee of the IEEE Computer Society. “IEEE Std 1364-1995. IEEE Standard Hardware Description Language Based on the Verilog Hardware Description Language”. Institute of Electrical and Electronics Engineers. 1995.
- [33] Faa-Jeng Lin; Li Tao Teng; Chih-Kai Chang; “Adaptive Backstepping Control for Linear Induction Motor Drive Using FPGA”, IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on , vol., no., pp.1269-1274, 6-10 Nov. 2006.
- [34] Miranda, H.; Cardenas, V.; Espinosa-Perez, G.; Noriega-Pineda, D.; “Multilevel Cascade Inverter with Voltage and Current Output Regulated Using a Passivity - Based Controller”, Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE , vol.2, no., pp.974-981, 8-12 Oct. 2006.
- [35] Gregoire, L.-A.; Al-Haddad, K.; Ounejjar, Y.; “A new method of control for multilevel converter implemented on FPGA”, Electrical Power & Energy Conference (EPEC), 2009 IEEE , vol., no., pp.1-6, 22-23 Oct. 2009.
- [36] Rech, C.; Pinheiro, J.R.; “Hybrid Multilevel Converters: Unified Analysis and Design Considerations”, Industrial Electronics, IEEE Transactions on , vol.54, no.2, pp.1092-1104, April 2007.
- [37] Manjrekar, M.D.; Steimer, P.K.; Lipo, T.A.; “Hybrid multilevel power conversion system: a competitive solution for high-power applications”, Industry Applications, IEEE Transactions on , vol.36, no.3, pp.834-841, May/Jun 2000.
- [38] Navabi, Zainalabedin . Embedded core design with FPGAs. Estados Unidos de America: McGraw-Hill Professional, 2006, PP-433.

- [39] Hyde, Randall. Chapter Twelve Calculation Via Table Lookups. 2006. Noviembre de 2010 <<http://webster.cs.ucr.edu/AoA/Windows/HTML/TableLookups.html>>.
- [40] J. Linares Flores, A. Antonio Garcia, A. Orantes Molina; “Smooth Starterfor a DC Machine through a DC-to-DC buck converter, Revista deIngeniería Investigación y Tecnología. ISSN 1405-7743 FI-UNAM, Vol.XII, No. 2, to appear in 2011.
- [41] Zurita-Bustamante, E.; Linares-Flores, J.; Guzman-Ramirez, E.; Sira-Ramirez, H.; “A comparison between the GPI and the PID controllers for the stabilization of a DC-DC ”buckconverter: A Field Programmable Gate Array implementation.”, Industrial Electronics, IEEE Transactions on , vol.PP, no.99, pp.1, 0.
- [42] Naouar, M-W.; Monmasson, E.; Naassani, A.A.; Slama-Belkhodja, I.; Patin, N.; “FPGA-Based Current Controllers for AC Machine Drives—A Review”, Industrial Electronics, IEEE Transactions on , vol.54, no.4, pp.1907-1925, Aug. 2007.
- [43] Monmasson, E.; Cirstea, M.N.; “FPGA Design Methodology for Industrial Control Systems—A Review”, Industrial Electronics, IEEE Transactions on , vol.54, no.4, pp.1824-1842, Aug. 2007.
- [44] Monmasson, E.; Idkhajine, L.; Naouar, M.W.; “FPGA-based Controllers”, Industrial Electronics Magazine, IEEE , vol.5, no.1, pp.14-26, March 2011.
- [45] Slotine J. J., W. Li. Applied Nonlinear Control. Prentice-Hall, 1991.
- [46] Quiang, Lu; Nonlinear Control Systems and Power Systems Dinamics; Kluwer Academic Publisher;2001.
- [47] Evangelisti, E.; Controllability and Observability, series: C.I.M.E. Summer Schools, 46; Ed. Springer; pp-284, 2010.
- [48] Batinn, R.H.; “Theodore Von Karman Lecture: Some Funny things happened on the way to the Moon;27th Aerospace Sciences Meeting, Reno, Nevada, AIAA-89-0861,1989.
- [49] Rugh, W.J; Linear System Theory, 2nd ed.; Prentice-Hall, Englewood Cliffs, N.J.,1996
- [50] Fliess, M., Lévine, J. ,Martin, P. , and Rouchon, P.; Flatness and defect of non-linear systems: Introductory theory and examples; International Journal of Control, 61(6) :1327-1361, 1995.
- [51] Murray,R. M., Rathinam,M., and Sluis,W. M.;Differential flatness of mechanical control systems. In Proceedings ASME International Congress and Exposition,1995.
- [52] Anritter, F.; “On the relations between different flatness based design methods for tracking controllers, American Control Conference, 2008 , vol., no., pp.1740-1745, 11-13 June 2008.
- [53] H. Sira-Ramirez and S. K. Agrawal, Differentially Flat Systems. New York: Marcel Dekker, 2004.
- [54] Fliess, M.; “Some basic structural propierties of generalized linear systems”, Systems and Control Letters, Vol.15, pp. 391-369,1990.

- [55] Linares-Flores, J; Antonio-García, A y Orantes-Molina, A.; “Arranque suave para un motor de CD a través de un convertidor reductor CD-CD. *Ing. invest. y tecnol.* [online]. 2011, vol.12, n.2 [citado 2012-01-19], pp. 137-148. ISSN 1405-7743.
- [56] Fliess, M.; “State Feddbacks without Asymtotic Observers and Generalized PID Regulators Nonlinear Control in the year 200; *Lecture Notes in Control and Information Sciences*; London; 2001.
- [57] Sira-Ramirez; *GPI Control of Linear and Nonlinear Systems*; Notas de curso; 2010.
- [58] Sira-Ramírez, H. ,Silva-Ortigoza, R.; *Control design techniques in power electronics devices*; Springer-Verlag, London, Limited, 2006.
- [59] E. W. Zurita-Bustamante, J. Linares-Flores, E. Guzmán-Ramírez, H. Sira-Ramírez; “A comparison between the GPI and the PID controllers for the stabilization of a DC-DC “buck” converter: A Field Programmable Gate Array implementation, *IEEE Transactions on Industrial Electronics*, Issue 99, 2011. This article has been accepted for publication in a future issue of this journal.
- [60] M. Fliess, R. Marquez, E. Delaleau and H. Sira-Ramírez; “Correcteurs Proportionnels-Intégraux Généralisés”, *ESAIM: Control, Optimization and Calculus of Variations*, Vol. 7, No. 2, pp. 23-41, 2002.
- [61] *Programmable Logic ICs Market Shares and Forecasts Worldwide, 2010 to 2016*, <http://www.electronics.ca/publications/>,2011.
- [62] Rodríguez-Andina, J.J.; Moure, M.J.; Valdés, M.D.; “Features, Design Tools, and Application Domains of FPGAs; *Industrial Electronics, IEEE Transactionson* , vol.54, no.4, pp.1810-1823, Aug. 2007.
- [63] Xilinx Staff; “Celebrating 20 years ofinnovation”, *Xcell J.*, no. 48, Spring 2004.
- [64] *Programmable Logic ICs Market Shares and Forecasts Worldwide, 2010 to 2016*; *electronics.ca publications.* [Online]. *Available : http : //www.electronics.ca/publications/products/.*
- [65] J.Munoz-Castaner, R. Asorey-Cacheda, F. J. Gil-Castineira, F. J. Gonzalez-Castano, and P. S. Rodriguez-Hernandez, “A review of aeronautical electronics and its parallelism with automotive electronics”, *IEEE Trans. Ind. Electron.*, vol. 54, no. 99, p. 1, Apr. 2010.
- [66] P. Conmy and I. Bate; “Component-based safety analysis of FPGAs”, *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, p. 195, May 2010.
- [67] F. Salewski and S. Kowalewski; “Hardware/software design considerations for automotive embedded systems”, *IEEE Trans. Ind. Informat.*, vol. 4, no. 3, p. 56, Aug. 2008.
- [68] Monmasson, E.; Idkhajine, L.; Bahri, I.; Naouar, M.-W.; Charaabi, L.; “Design methodology and FPGA-based controllers for Power Electronics and drive applications”, *Industrial Electronics and Applications (ICIEA)*, 2010 the 5th IEEE Conference on , vol., no., pp.2328-2338, 15-17 June 2010.
- [69] Monmasson, E.; Idkhajine, L.; Cirstea, M.N.; Bahri, I.; Tisan, A.; Naouar, M.W.; “FPGAs in Industrial Control Applications”, *Industrial Informatics, IEEE Transaction*, vol.7, no.2, pp.224-243, May 2011.

- [70] A. V. Deshmukh, Microcontrollers, Theory and Applications, ser.The Companies, Computer Engineering Series. New York: Mac-Graw-Hill, 2007.
- [71] P. Lapsley, DSP Processor Fundamentals: Architecture and Features. Piscataway, NJ: IEEE Press, 1997.
- [72] W. H. Wolf, FPGA-Based System Design. Englewood Cliffs, NJ: Prentice-Hall, 2004.
- [73] Wael Badawy, Graham Jullien, System-on-chip for real-time applications. Kluwer. ISBN 1402072546, 9781402072543. 465 pages,2003.
- [74] Actel on-line literature. Online. *Available : www.actel.com*
- [75] Xilinx on-line literature. Online. *Available : www.xilinx.com/support/documentation/white_papers/wp312NextGen28nmOverview.pdf*
- [76] Altera on-line literature. Online. *Available : www.altera.com/corporate/newsroom/releases/2011/products/nr_portfolio.html*
- [77] Xilinx on-line literature. Online. *Available : www.xilinx.com*
- [78] Altera on-line literature. Online. *Available : www.altera.com*
- [79] Altera on-line literature. Online. *Available : www.altera.com/corporate/newsroom/releases/2011/products/nr_portfolio.html*
- [80] Idkhajine, E. Monmasson, M.-W. Naouar, A. Prata, and K. Bouallaga; “Fully integrated FPGA-based controller for synchronous motor drives”; IEEE Trans. Ind. Electron., vol. 56, no. 10, pp. 4006-4017, Oct. 2009.
- [81] K. Eshraghian; “SoC emerging technologies, IEEE Proc., vol. 94, no.6, pp. 1197-1213, Jun. 2006.; G. Martin, .overview of the MPSoC design challenge, in Proc.DAC’06 Conf., 2006, CD-ROM.; R. Kumar, D. Tullsen, and N. Jouppi, “Core architecture optimization for heterogeneous chip multiprocessors”, in Proc. PACT’2006 Conf., Seattle, Apr. 2006, pp. 23-32.
- [82] R. Kumar, D. Tullsen, and N. Jouppi; “Core architecture optimization for heterogeneous chip multiprocessors, in Proc. PACT’2006 Conf., Seattle, Apr. 2006, pp. 23-32.; C. A. Zeferino, M. E. Kreutz, L. Carro, and A. A. Susin, “A study on communication issues for systems-on-chip”, in Proc. SBCCI Conf., 2002, pp. 121-126.
- [83] System C on-line literature. [Online]. *Available : <http://www.systemc.org>*
- [84] J. J. Labrosse, MicroC/OS II: The Real Time Kernel, 2nd ed. Lawrence, KA: CMP Books, 2002.
- [85] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes; “Features, design tools, and application domains of FPGAs, IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1810-1823, Aug. 2007.
- [86] H. A. Toliyat, S. G. Campbell; “DSP-Based electromechanical motion control”, CRC Press, 2004.

- [87] S. N. Murthy, W. Alvis, R. Shirodkar, K. Valavanis, W. Moreno; "Methodology for implementation of unmanned vehicle control on FPGA using system generator", in Proc. IEEE ICCDCS'08 Conf., 2008, CD-ROM.
- [88] E. Monmasson and M. Cirstea; "FPGA design methodology for industrial control systems-A review", IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1824-1842, Aug. 2007.
- [89] J. S. Beeckler, W. J. Gross; "A Methodology for Prototyping Flexible Embedded Systems in Proc. CCECE'07 Conf., 2007, CD-ROM.
- [90] M.-W. Naouar, E. Monmasson, A. A. Naassani, I. Slama-Belkhouja and N. Patin; "FPGA-based current controllers for AC machine drives-A review", IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1907-1925, Aug. 2007.
- [91] Monmasson, E.; Idkhajine, L.; Naouar, M.W.; "FPGA-based Controllers", Industrial Electronics Magazine, IEEE , vol.5, no.1, pp.14-26, March 2011.
- [92] Breittfelder, K. and Messina, D., editors, IEEE 100: The Authoritative Dictionary of IEEE Standard Terms (7th edn.), Institute of Electrical and Electronics Engineers, Inc., New York, 2000.
- [93] J. Espinoza and G. Joós; "Power converter system simulation using high level languages", in Conf. Record IEEE 4th Workshop Computers in Power Electronics, Trois-Rivieres, Canada, Aug. 1994, pp. 79-84.
- [94] T.A. Meynard, M. Fadel, and N. Aouda; "Modeling of multilevel converters", IEEE Trans. Ind. Electron., vol. 44, no. 3, pp. 356-364, June 1997.
- [95] G. Gateau, M. Fadel, P. Maussion, R. Bensaid, and T. Meynard; "Multicell converters: Active control and observation of flying-capacitor voltages", IEEE Trans. Ind. Electron., vol. 49, no. 5, pp. 998-1008, Oct. 2002.
- [96] C.T. Rim, D.Y. Hu, and G.H. Cho; "Transformers as equivalent circuits for switches: General proofs and D-Q transformation-based analyses", IEEE Trans. Ind. Applicat., vol. 26, no. 4, pp. 777-785, July 1990.
- [97] S.B. Han, N.S. Choi, C.T. Rim, and G.H. Cho; "Modeling and analysis of static and dynamic characteristics for buck-type three-phase PWM rectifier by circuit DQ transformation", IEEE Trans. Power Electron., vol. 13, no. 2, pp. 323-336, Mar. 2001.
- [98] R.C. Portillo, M.M. Prats, J.I. Leon, J.A. Sanchez, J.M. Carrasco, E. Galvan, and L.G. Franquelo; "Modeling strategy for back-to-back three-level converters applied to high-power wind turbines, IEEE Trans. Ind. Electron., vol. 53, no. 5, pp. 1483-1491, Oct. 2006.
- [99] Ounejjar, Y.; Al-Haddad, K.; Alolah, A.I.; "Averaged model of the 31-level packed U cells converter", Industrial Electronics (ISIE), 2011 IEEE International Symposium on , vol., no., pp.1831-1836, 27-30 June 2011.
- [100] G. Escobar, J. Leyva-Ramos, J.M. Carrasco, E. Galvan, R. Portillo, M.M. Prats, and L.G. Franquelo; "Modeling of a three level converter used in asynchronous rectifier application", in Proc. IEEE 35th Ann. Power Electronics Specialists Conf.(PESC2004), vol. 6, Aachen, Germany, pp.4306-4311.

- [101] J. Rodriguez, S. Bernet, B. Wu, J. O. Pontt, and S. Kouro; “Multilevel voltage-source-converter topologies for industrial medium-voltage drives, IEEE Transactions on Industrial Electronics, 54(6), 2930-2945, December 2007.
- [102] Comision Federal de Electricidad. Online. Available : [http : //www.cfe.gob.mx](http://www.cfe.gob.mx)
- [103] R. H. Baker and L. H. Bannister; “Electric Power Converter, Ü.S. Patent 3 867 643, Feb. 1975.
- [104] R. H. Baker; “Bridge Converter Circuit, Ü.S. Patent 4 270 163, May 1981.
- [105] P. W. Hammond; “Medium Voltage PWM Drive and Method, Ü.S. Patent 5 625 545, Apr. 1977.
- [106] F. Z. Peng and J. S. Lai; “Multilevel Cascade Voltage-source Inverter with Separate DC source, Ü.S. Patent 5 642 275, June 24, 1997.
- [107] P. W. Hammond; “Four-quadrant AC-AC Drive and Method, Ü.S. Patent 6 166 513, Dec. 2000.
- [108] M. F. Aiello, P. W. Hammond, and M. Rastogi; “Modular Multi-level Adjustable Supply with Series Connected Active Inputs, Ü.S. Patent 6 236 580, May 2001.
- [109] J. P. Lavieville, P. Carrere, and T. Meynard; “Electronic Circuit for Converting Electrical Energy and a Power Supply Installation Making Use Thereof, Ü.S. Patent 5 668 711, Sept. 1997.
- [110] T. Meynard, J.-P. Lavieville, P. Carrere, J. Gonzalez, and O. Bethoux; “Electronic Circuit for Converting Electrical Energy, Ü.S. Patent 5 706 188, Jan. 1998.
- [111] D. Noriega-Pineda et. al (ref: D. Noriega-Pineda, G. Espinosa-Pérez, V. Cardenas, and H. Miranda; “On the Passivity-based Control for Multilevel Inverters”, 10th IEEE International Power Electronics Congress, pp. 1-6, Oct. 2006.
- [112] R. Ortega, A. Loría, P. J. Nicklasson, and H. Sira-Ramírez; Passivity-based Control of Euler-Lagrange Systems, Springer-Verlag London 2010.
- [113] H. Sira-Ramírez and S. Agrawal; *Differentially Flat Systems*, Marcel Dekker, New York, USA, 2004.
- [114] J. Lévine; *Analysis and Control of Nonlinear Systems: A Flatness-based Approach*, Springer-Verlag Berlin Heidelberg, 2009.
- [115] Franco-González, A., R. Marquez 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, pp. 209-212, 2007.
- [116] Linares-Flores J., H. Sira-Ramírez; “Sliding Mode Delta Modulation GPI Control of a DC Motor through a Buck Converter”, 2nd IFAC Symposium on System, Structure and Control, pp. 442-447, Oaxaca (México), Dec 2004.

- [117] E. W. Zurita-Bustamante, J. Linares-Flores, E. Guzmán-Ramírez, H. Sira-Ramírez; A comparison between the GPI and the PID controllers for the stabilization of a DC-DC “buck” converter: A Field Programmable Gate Array implementation, IEEE Transactions on Industrial Electronics, Issue 99, 2011. This article has been accepted for publication in a future issue of this journal.
- [118] SIMNON on-line literature. Online. *Available* : www.mpassociates.gr/software/catalog/sci/simnon/simnon.htm
- [119] Pagina del producto on-line. Online. *Available* : www.mathworks.com/products/simulink/index.html
- [120] Pagina del producto on-line. Online. *Available* : <http://www.powersimtech.com/index.php?name=home>
- [121] Pagina del producto on-line. Online. *Available* : http://www.psim-europe.com/psim_simcoupler.php
- [122] Palnitkar S., A guide to digital design and synthesis. 2nd ed. USA:Prentice-Hall, 2003.
- [123] IEEE Computer Society. IEEE Standard for Binary Floating-Point Arithmetic, IEEE Std 754-1985. 1985.
- [124] Xilinx Inc. Spartan-3E FPGA Family: Complete Data Sheet, Xilinx.2008.
- [125] Isermann, R.: Digital Control Systems, Springer Verlag - 1981.
- [126] U. Ascher, L. Petzold; *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM: Society for Industrial and Applied Mathematics. 1998.
- [127] E. Kreyszig; *Advanced Engineering Mathematics*. John Wiley & Sons. 2006.
- [128] Pagina del producto on-line. Online. *Available* : <http://www.microchip.com/>
- [129] L. Idkhajine, E. Monmasson; “Design Methodology for Complex FPGA based Controllers Application to an EKF Sensorless AC Drive”, XIX International Conference on Electrical Machines, ICEM, 2010.
- [130] Pong P. Chu, FPGA PROTOTYPING BY VHDL EXAMPLES Xilinx Spartan-3 Version , Wiley Interscience, 2008.
- [131] Pagina del producto on-line. Online. *Available* : <http://www.fdti.com>
- [132] Dorf, Richard C., Bishop, Robert H.; Modern Control Systems. Pearson Prentice Hall. 2007.
- [133] Bárcenas, Bárcenas, E., Análisis y desarrollo de un inversor multinivel, Tesis de maestría, CENIDET, Departamento de Electrónica, Cuernavaca-Morelos, 2002.
- [134] Mclyman, Colonel Wm. T., Transformer And Inductor Design Handbook, Tercera edición, Editorial Dekker, 2004.