

UNIVERSIDAD SANTO TOMÁS

CONTROL NO LINEAL PARA UN CONVERTIDOR CONMUTADO CC/CA

Realizado por

Katherin Dayanna Casallas Gutierrez

Juan Felipe Lopez Rodriguez

Proyecto de grado presentado en cumplimiento del requisito para
optar por el título de Ingeniero Electrónico



Grupo de Investigación MEM (Modelado-Electrónica-Monitoreo)

Sistemas de energía eléctrica

Facultad de Ingeniería Electrónica

División de Ingenierías

Octubre de 2023

CONTROL NO LINEAL PARA UN CONVERTIDOR CONMUTADO CC/CA

Realizado por

Katherin Dayanna Casallas Gutierrez
Juan Felipe Lopez Rodriguez

Proyecto de grado presentado en cumplimiento del requisito para
optar por el título de Ingeniero Electrónico

Dirigido por

MSc. Carlos Javier Mojica Casallas

Co-Dirigido por

PhD. Carlos Andrés Torres Pinzón

Grupo de Investigación MEM (Modelado-Electrónica-Monitoreo)

Sistemas de energía eléctrica

Facultad de Ingeniería Electrónica

División de Ingenierías

Octubre de 2023

Autoridades de la Universidad

RECTOR GENERAL

R.P. FRAY ALVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL

R.P. FRAY MAURICIO ANTONIO CORTES GALLEGU, O.P.

SECRETARIO GENERAL

Dra. INGRID LORENA CAMPOS VARGAS

DECANO DIVISIÓN DE INGENIERÍAS

R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

ING. CARLOS ANDRÉS TORRES PINZÓN

Nota de aceptación

Firma del autor

Firma del jurado

Firma del jurado

Advertencia

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Dedicatoria

Esta tesis está dedicada a mi familia, por su apoyo constante y amor incondicional en cada etapa de mi formación académica. Este trabajo es un reflejo de su influencia positiva en mi vida, su amor incondicional y la confianza que han depositado en mí. También, a mi amigo y compañero, Juan Felipe López Rodríguez, esta tesis es un testimonio de lo lejos que hemos llegado gracias a nuestro esfuerzo conjunto.

Katherin Dayanna Casallas Gutierrez

Quiero dedicar este proyecto con profundo agradecimiento a las personas que han sido mi fuente de apoyo y motivación en mi proceso de formación. A mi familia, quienes siempre han estado a mi lado brindando apoyo emocional con sus palabras de aliento. A mis amigos, con quienes compartí risas y estuvieron a mi lado en momentos difíciles. Y finalmente a mi Compañera de trabajo y amiga Katherin Dayanna Casallas Gutiérrez que gracias a su colaboración y apoyo logramos enfrentar y celebrar logros.

Juan Felipe López Rodríguez

Agradecimientos

Esta tesis es el resultado de un esfuerzo colectivo, y reconocemos profundamente la valiosa colaboración de todos aquellos que han compartido su experiencia, tiempo y conocimientos para enriquecer este trabajo. Sus contribuciones han dado forma a cada aspecto de esta investigación y han contribuido a su calidad y relevancia.

En primer lugar, queremos agradecer a nuestro director de tesis, Carlos Javier Mojica Casallas, por su orientación constante, su apoyo inquebrantable y su dedicación incansable para guiarnos a lo largo de este proceso. Sus conocimientos y sabiduría fueron fundamentales para el éxito de este trabajo. A nuestro codirector de tesis, Carlos Andrés Torres Pinzón, por su valiosa contribución y apoyo a lo largo de este proyecto. Su experiencia y orientación fueron esenciales para dar forma a esta investigación.

Por último, pero no menos importante, agradecemos a nuestras familia por su amor incondicional y su apoyo constante durante todos estos años de estudio. Su comprensión y aliento fueron un faro de luz en los momentos más exigentes. A nuestros amigos y compañeros de clase por compartir esta travesía académica con nosotros, por sus valiosos aportes y por ser una fuente constante de inspiración.

Índice general

Resumen	IX
Abstract	X
List of Figures	X
List of Tables	XIV
1. Introducción	1
1.1. Planteamiento del Problema	1
1.2. Objetivos	2
1.2.1. Objetivo General	2
1.2.2. Objetivos Específicos	2
1.3. Justificación	3
1.4. Impacto Social	4
1.5. Diseño Metodológico	5
1.5.1. Fase 1: Selección	5
1.5.2. Fase 2: Diseño	5
1.5.3. Fase 3: Validación	6
1.6. Estado del Arte	7
1.6.1. Investigaciones en el ambito del funcionamiento de los inversores monofásicos	7
1.6.2. Investigaciones en el ambito de tecnica de control	7
1.6.3. Investigaciones en el ámbito de las microrredes	9
2. Marco Teórico	10
2.1. Topología de convertidores	10
2.2. Topología de inversores	10
2.2.1. Inversor monofásico	11
2.2.2. Inversor trifásico	11
2.2.3. Inversor medio puente	12
2.2.4. Inversor puente completo con carga	12
2.2.5. Inversor multinivel	13
2.3. Técnicas de Control	13
2.3.1. Control por modulación escalar o PWM	14
2.3.2. Control de realimentación Proporcional-Integral-Derivativo	15
2.3.3. Control Predictivo basado en Modelo MPC	16
2.3.3.1. Horizonte de predicción	16
2.3.3.2. Horizonte de control	17
2.3.3.3. Función de costo	17
2.3.3.4. Optimizador	17
2.3.4. Control en modo deslizante	18
2.3.5. Superficie de deslizamiento	18
2.3.6. Coeficiente deslizante	19
2.3.7. Problema de chattering	19
2.3.8. Entrada de control	20

2.4.	Transformación DQ0	20
2.4.1.	Transformada de Clark	20
2.4.2.	Transformada de Park	21
2.5.	Transformación DQ en sistemas de una sola fase	22
3.	Caracterización de la planta	24
3.1.	Descripción de la planta general	24
3.2.	Drivers de la planta	25
3.2.1.	Protección contra desaturación	26
3.3.	Sensores de la planta	26
3.3.1.	Sensores de voltaje	26
3.3.2.	Sensores de Corriente	27
3.3.3.	Comunicación Serial	27
4.	Selección Técnica de Control	29
4.1.	Control por modulación escalar o PWM	29
4.1.1.	Características	30
4.1.2.	Ventajas y Desventajas	30
4.1.2.1.	Ventajas	30
4.1.2.2.	Desventajas	30
4.2.	Control de realimentación Proporcional-Integral-Derivativo	31
4.2.1.	Características	31
4.2.2.	Ventajas y Desventajas	31
4.2.2.1.	Ventajas	31
4.2.2.2.	Desventajas	31
4.3.	Control Predictivo basado en Modelo MPC	32
4.3.1.	Características	32
4.3.2.	Ventajas y Desventajas	33
4.3.2.1.	Ventajas	33
4.3.2.2.	Desventajas	33
4.4.	Control sliding	33
4.4.1.	Ventajas y Desventajas	34
4.4.1.1.	Ventajas	34
4.4.1.2.	Desventajas	34
5.	Modelo Matemático	36
5.1.	Análisis del sistema	36
5.2.	Modelo conmutado	38
5.3.	Modelo promediado	38
6.	Control en Modo Sliding	40
6.1.	Controlador SMC	40
6.1.1.	Comparador de ventana	41
6.1.2.	flip flop set-reset	41
6.1.3.	Control implementado	42
6.2.	Diseño	43
6.3.	Diagrama de Flujo	43
6.3.1.	Diagrama de flujo interrupción del ADC	43
6.3.2.	Diagrama de flujo interrupción del Timer	46
7.	Control de realimentación PI	48
7.1.	Transformación DQ	48
7.2.	Inversa transformada DQ	51
7.3.	Punto de equilibrio del sistema en coordenadas dq	53
7.4.	Técnica de control	56
7.5.	Espacio de estados	57

7.6. Función de transferencia	58
7.7. Diseño del compesador	59
8. Resultados y Discusión	63
8.1. Resultados en Simulación	63
8.1.1. Simulación control sliding	63
8.1.2. Implementación del control sliding en Simulink	64
8.1.3. Simulación transformada dq	67
8.1.4. Simulación transformada inversa DQ	68
8.1.5. Simulación control PI en coordenadas DQ	70
8.2. Resultados Experimentales	74
8.2.1. Implementación del control slidig	74
8.2.2. Implementación del control PI	75
8.3. Comparación Técnicas de control implementadas	79
9. Creación de proyecto en code Composer	81
9.1. Creación de proyecto con driverlib	81
9.2. Manejo de las librerías	84
9.2.1. Configuración del ePWM Trigger	84
9.2.2. Configuración del ADC	86
9.2.3. Configuración de comparadores	88
10. Conclusiones y Trabajos Futuros	90
10.1. Conclusiones	90
10.2. Trabajos Futuros	91
Anexos	92
10.3. Script Matlab	92
10.4. Codigo implementado en el control Sliding	95
10.5. Codigo implementado para control PI	115
10.5.1. Inversor.h	115
10.5.2. Inversor.c	116
10.6. Planos	134

Resumen

Este proyecto de grado formará parte de un sistema global cuyo objetivo es minimizar al máximo la potencia suministrada por la red eléctrica convencional mediante la integración de fuentes de energía renovable. El sistema completo se divide en dos bloques principales. El primero se encarga de captar la energía de los paneles solares y suministrar un bus de corriente continua (DC) variable. El segundo bloque, objeto de este trabajo, consiste en el diseño de un inversor con control de corriente que se conectará en línea con la red eléctrica, para este inversor se implementará una técnica de control no lineal y se comparará el funcionamiento vs la implementación de una técnica de control convencional PI. En la mayoría de los casos, estos convertidores se controlan utilizando técnicas de control lineal que pueden ser insuficientes para abordar perturbaciones y cambios dinámicos en el sistema. En contraste, el enfoque no lineal propuesto en este proyecto busca mejorar la respuesta del convertidor y su capacidad de adaptarse a diferentes condiciones de funcionamiento. El proyecto se divide en varias etapas, en primer lugar, se procede con la selección de una estrategia de control a través de una revisión bibliográfica para conocer diversas técnicas de control documentadas en la literatura y aplicadas previamente en otros convertidores. Posteriormente, se realiza la caracterización de la planta disponible en la Universidad Santo Tomás, para comprender su funcionamiento en detalle. Luego, se desarrolla un modelo matemático que describe el sistema de manera precisa. Y por último, se lleva a cabo el diseño e implementación de las estrategias de control seleccionadas para una potencia de salida de 200 vatios. Se emplearán simulaciones para validar el diseño y ajustar los parámetros necesarios para lograr de manera consistente y controlada la potencia de salida deseada. Una vez validado en simulaciones, se implementarán los controladores seleccionados en la planta disponible y se realizan pruebas prácticas para verificar su funcionamiento.

Se espera que los resultados de este proyecto de grado contribuyan al avance de la investigación en el campo de la electrónica de potencia y el control no lineal y además que este enfoque no lineal tenga aplicaciones prácticas en la generación de energía eléctrica, mejorando la eficiencia y la integración de fuentes de energía renovable en las redes eléctricas convencionales.

Abstract

This undergraduate project will be part of a global system whose goal is to minimize the power supplied by the conventional electrical grid through the integration of renewable energy sources. The complete system is divided into two main blocks. The first is responsible for capturing energy from solar panels and supplying a variable direct current (DC) bus. The second block, the focus of this work, involves designing an inverter with current control that will be connected in-line with the electrical grid. For this inverter, a non-linear control technique will be implemented and compared to the performance of a conventional PI control technique.

In most cases, these converters are controlled using linear control techniques, which may be insufficient to address disturbances and dynamic changes in the system. In contrast, the proposed non-linear approach in this project aims to enhance the converter's response and its ability to adapt to different operating conditions.

The project is divided into several stages. Firstly, a control strategy is selected through a literature review to explore various control techniques documented in the literature and previously applied in other converters. Subsequently, the available plant at Santo Tomás University is characterized to understand its operation in detail. Then, a mathematical model is developed to accurately describe the system. Finally, the selected control strategies are designed and implemented for an output power of 200 watts.

Simulations will be used to validate the design and adjust the necessary parameters to consistently and controllably achieve the desired output power. Once validated in simulations, the selected controllers will be implemented on the available plant, and practical tests will be conducted to verify their performance.

It is expected that the results of this undergraduate project will contribute to the advancement of research in the field of power electronics and non-linear control. Additionally, it is anticipated that this non-linear approach will have practical applications in electrical power generation, improving the efficiency and integration of renewable energy sources into conventional electrical grids.

Keywords: Alternating Current, Direct Current, Energy Conversion Converter, Mathematical Model, Non-linear Control.

Índice de figuras

1.	Metodología Fuente: Autor	5
2.	Topología inversor monofásico Fuente: Autor	11
3.	Topología inversor trifásico Fuente: Autor	12
4.	Estructura medio puente Fuente: [30]	12
5.	Estructura medio puente Fuente: [30]	13
6.	Topología inversor multinivel Fuente: Autor	13
7.	Estrategia MPC Fuente: [51]	18
8.	Superficie de deslizamiento. Fuente: [52]	19
9.	Transformada de Clarke Fuente: Adaptado de [55]	21
10.	Transformada de Park Fuente: Adaptado de [55]	21
11.	Circuito Real e Imaginario Fuente: [56]	22
12.	Circuito Real e Imaginario promediado Fuente: [56]	23
13.	Circuito transformación DQ Fuente: [56]	23
14.	Etapas de potencia de la planta Fuente: [57]	25
15.	Etapas digitales de la planta Fuente: [58]	25
16.	Driver 2ED020I12-F2 Fuente: [59]	26
17.	Sensores de Voltaje DC Fuente: [57]	27
18.	Sensores de voltaje AC Fuente: [57]	27
19.	Sensores de Corriente AC Fuente: [57]	27
20.	Comunicación Serial \textbf{Fuente: } Autor	28
21.	Circuito generador PWM Fuente: [38]	29
22.	Diagrama de bloques PID Fuente: [42]	31
23.	Estructura básica del MPC Fuente: [49]	33
24.	Inversor monofásico medio puente Fuente: Autor	36
25.	Inversor tiempo uno Fuente: Autor	37
26.	Inversor tiempo uno Fuente: Autor	37
27.	Compensador SMC Fuente: Autor	40
28.	Comparador de ventana Fuente: Autor	41
29.	Salida comparador de ventana con histéresis Fuente: Autor	42
30.	Esquema del controlador propuesto. Fuente: Adaptado de [64]	42
31.	Diagrama de flujo interrupción ADC. Fuente: Autor	44
32.	Diagrama de flujo interrupción Timer. Fuente: Autor	46
33.	Formas de onda $\alpha\beta$ Fuente: Autor	49
34.	Circuito relacionado a la variable q Fuente: Autor	53
35.	Circuito relacionado a la variable q Fuente: Autor	54
36.	Circuito relacionado a la variable q Fuente: Autor	54
37.	Circuito relacionado a la variable q Fuente: Autor	55
38.	Compensador PI Fuente: Autor	56
39.	Ley de control en discreto. Fuente: DCL User's Guide.pdf	57
40.	Polos y ceros plano z Fuente: Autor	60

41.	Respuesta del sistema sin compensador Fuente: Autor	60
42.	Respuesta del sistema con el control PI Fuente: Autor	61
43.	Esquema transformación DQ con control. Fuente: Adaptado de [64]	62
44.	Modelo conmutado inversor - Simulink Fuente: Autor	63
45.	Control sliding Fuente: Autor	64
46.	Superficie de deslizamiento Fuente: Autor	64
47.	Superficie de deslizamiento Fuente: Autor	64
48.	Señal de error Fuente: Autor	65
49.	Señales de conmutación Fuente: Autor	65
50.	Voltaje de la red Fuente: Autor	66
51.	Corriente en la bobina y Corriente de referencia Fuente: Autor	66
52.	Corriente en la red con el sistema encendido Fuente: Autor	67
53.	Diagrama de bloques i_d Fuente: Autor	67
54.	Diagrama de bloques v_d Fuente: Autor	68
55.	Diagrama de bloques i_q Fuente: Autor	68
56.	Diagrama de bloques v_q Fuente: Autor	68
57.	Diagrama de bloques transformada inversa v_α Fuente: Autor	69
58.	Diagrama de bloques transformada inversa v_β Fuente: Autor	69
59.	Diagrama de bloques transformada inversa i_α Fuente: Autor	69
60.	Diagrama de bloques transformada inversa i_β Fuente: Autor	70
61.	Diagrama de bloques transformación de Park Fuente: Autor	70
62.	Compensador coordenadas dq Fuente: Autor	70
63.	Diagrama de bloques v_q Fuente: Autor	71
64.	Compensador coordenadas dq en discreto Fuente: Autor	71
65.	Sistema lazo cerrado en coordenadas dq Fuente: Autor	72
66.	Señal de referencia Fuente: Autor	72
67.	Transformación de Park señal de referencia Fuente: Autor	73
68.	Señal de referencia Fuente: Autor	73
69.	Transformación de Park señal de referencia Fuente: Autor	73
70.	Corriente con carga de 200 Ω Fuente: Autor	74
71.	Corriente con perturbación Fuente: Autor	74
72.	Voltaje en la salida Fuente: Autor	75
73.	Corriente con carga de 210 Ω . Fuente: Autor	76
74.	Corriente con carga de 52 Ω . Fuente: Autor	76
75.	Corriente con Perturbación. Fuente: Autor	78
76.	New CCS Proyect Fuente: Autor	81
77.	Processor Options Fuente: Autor	82
78.	Include Options Fuente: Autor	82
79.	Predefined symbol Fuente: Autor	83
80.	Predefined symbol Fuente: Autor	83
81.	Predefined symbol Fuente: Autor	84
82.	Diagrama de bloques SOC Fuente: Autor	85
83.	Diagrama de bloques ADC Fuente: Autor	86
84.	Diagrama de bloques CMPSS Fuente: Autor	88
85.	Diagrama del comparador Fuente: Autor	88
86.	Pagina 1 código en Matlab para el diseño del compensador PI.	92
87.	Pagina 2 código en Matlab para el diseño del compensador PI.	93
88.	Pagina 3 código en Matlab para el diseño del compensador PI.	94
89.	Etapa Puente H Fuente: Autor	135
90.	Etapa sensores Corriente y voltaje Fuente: Autor	135
91.	Etapa sensores de voltaje Fuente: Autor	136
92.	Etapa amplificadores Fuente: Autor	136
93.	Conector DSP Fuente: Autor	137

94.	Power Branch 1A Fuente: Autor	137
95.	Power Branch 2A Fuente: Autor	138
96.	Power Branch 3A Fuente: Autor	138
97.	Power Branch 1B Fuente: Autor	139
98.	Power Branch 2B Fuente: Autor	139
99.	Power Branch 3B Fuente: Autor	140
100.	Power Source, and outputs drivers Fuente: Autor	140
101.	RDY Y FLTS Fuente: Autor	141
102.	RST SYSTEM Fuente: Autor	141
103.	Driver 1 Fuente: Autor	142
104.	Driver 2 Fuente: Autor	142
105.	Driver 3 Fuente: Autor	143
106.	Driver 4 Fuente: Autor	143
107.	Driver 5 Fuente: Autor	144
108.	Driver 6 Fuente: Autor	144

Índice de tablas

1.	Comparaciones técnicas de control	35
2.	Parámetros función de transferencia	59
3.	Parámetros	63
4.	Parámetros transformación dq	67
5.	Parámetros transformación inversa dq	69
6.	Corrientes y Potencias experimentales carga de 200 Ω	74
7.	Corrientes y Potencias experimentales con carga de 50 Ω	75
8.	Corrientes y Potencias experimentales con carga de 210	76
9.	Corrientes y Potencias experimentales con carga de 52	77
10.	Comparación técnica de control implementada Sliding	79
11.	Comparaciones técnicas de control implementada PI	80

Capítulo 1

Introducción

1.1. Planteamiento del Problema

El consumo energético se define como el gasto total de energía, sin embargo, este consumo aumenta en sistemas no eficientes, es decir a mayor consumo energético menor es la eficiencia [1]. Conforme pasan los años se evidencia un incremento en la población humana al igual que en la demanda de energía eléctrica, la cual trae como consecuencia que las fuentes de generación eléctrica existentes tengan dificultades para satisfacer las necesidades del consumidor, esto sin mencionar los efectos ambientales que se pueden producir. Por lo general, el consumo de energía eléctrica principalmente se da en la iluminación de edificios en las grandes ciudades, el uso de electrodomésticos en lugares residenciales y los procesos industriales para la producción de bienes y servicios, por lo tanto, el aumento en el consumo eléctrico promueve el uso de las fuentes de energía no convencionales, como una solución para mitigar este problema [2].

Sin lugar a duda el uso de sistemas que permitan el aprovechamiento de las energías no convencionales traen grandes beneficios al consumidor, al igual que al medio ambiente como se indica en la ley 1715 de 2014 en Colombia, teniendo en cuenta la actualización dada por la ley 2099 de 2021. Dado que esta se encarga de promover el desarrollo y uso de las fuentes de energía no convencionales en el sistema nacional para mejorar el consumo energético, y a su vez autoriza que los auto generadores a gran y pequeña escala puedan entregar sus excedentes a la red de distribución, bajo un esquema de medición bidireccional, con el fin de calcular la cantidad de energía suministrada a la red eléctrica y la energía dada por la empresa que ofrece el servicio eléctrico.

Los beneficios de emplear fuentes de energía no convencionales son innegables, y esto se vuelve aún más evidente al considerar la conversión de inversores, ya que estos desempeñan un papel crucial al transformar la corriente directa de la fuente no convencional a corriente alterna para el funcionamiento de la mayoría de los equipos electrónicos. Cabe destacar que la dinámica de los convertidores ha llevado a que se utilicen técnicas de control de realimentación lineal, que pueden provocar un funcionamiento incorrecto o un comportamiento inestable en algunos casos, porque no se tiene en cuenta la no linealidad del convertidor. Por esta razón, se desarrollan varios métodos de control para tratar de resolver los problemas de control antes mencionados considerando la no linealidad y la incertidumbre de los parámetros en los convertidores de conmutación CC/CA.[3], [4].

Para evitar pérdidas económicas como energéticas se deben implementar sistemas que realicen la interconexión de fuentes no convencionales en CC (corriente continua) con la red eléctrica CA (corriente alterna); aunque esto plantee un desafío adicional en el ámbito del control y la integración de sistemas, contemplando además, temas como lo son la estabilidad, modelado, conversión de los sistemas conmutados CC/CC, CA/CC Y CC/CA y asegurar su buen funcionamiento por medio de un control no lineal [5]. De acuerdo a esto surge la siguiente pregunta de investigación.

¿Cómo implementar una estrategia de control no lineal en un convertidor CC-CA (Inversor) monofásico de 200W?

1.2. Objetivos

1.2.1. Objetivo General

Implementar una estrategia de control no lineal para un convertidor CC/CA (Inversor) monofásico de 200W a partir de la planta disponible en la Universidad Santo Tomás.

1.2.2. Objetivos Específicos

- Seleccionar una estrategia de control no lineal que permita mejorar las prestaciones dinámicas del convertidor en cuanto al tiempo de establecimiento, relación de amortiguamiento, y un adecuado nivel de rechazo a perturbaciones, con base en una revisión bibliográfica y análisis de adaptabilidad a la planta disponible en la Universidad Santo Tomás.
- Diseñar la estrategia de control digital no lineal seleccionada, partiendo del modelado bilineal del convertidor CC/CA para obtener una potencia de salida de 200w.
- Validar experimentalmente la estrategia de control no lineal seleccionada, comparando los resultados obtenidos con respecto a los de la simulación del convertidor CC/CA para la misma potencia y prestaciones dinámicas mencionadas.

1.3. Justificación

El constante avance de la industria en la creación de nuevos sistemas o circuitos, como los inversores On Grid (sistemas conectados a la red eléctrica) y Off Grid (sistemas totalmente independientes), diseñados para integrarse con fuentes de energía no convencionales, ofrece soluciones para abordar la creciente demanda de energía procedente de la red eléctrica. Estos inversores o convertidores conmutados tienen la responsabilidad de garantizar la compatibilidad entre la energía generada por fuentes no convencionales y la red eléctrica. Lo logran mediante su capacidad para convertir la entrada de corriente continua (CC) en una salida de corriente alterna (CA), que es la forma en que funcionan la mayoría de los dispositivos electrónicos. Sin embargo, la implementación de convertidores que requieren esta interconexión plantea desafíos tanto en términos de capacidad de potencia como de distribución de energía. [2], [5], [6].

Una primera dificultad significativa surge en el diseño del control de los inversores debido a sus características no lineales. En tales casos, la práctica común implica la utilización de métodos de control lineal establecidos, sin embargo, estos métodos solo pueden asegurar la estabilidad del sistema en proximidad a un punto de equilibrio y en ciertas condiciones dinámicas, ya que perturbaciones considerables en la señal pueden provocar la inestabilidad del sistema. En consecuencia, en el contexto de la revisión bibliográfica, se identificaron múltiples estudios que comparan estrategias de control convencionales con enfoques de control no lineales en la regulación de convertidores de conmutación, con el objetivo de garantizar la estabilidad a lo largo de un amplio rango de operación.[7],[8].

Teniendo en cuenta lo mencionado en los párrafos anteriores, este documento aborda el desarrollo de una solución para un inversor On grid (en línea) conectado a una fuente de alimentación DC (que simula una fuente de alimentación a partir de energías alternativas) consistente en generar un bus AC monofásico para suministrar energía eléctrica a diferentes cargas que se encuentran conectadas en un uso residencial. Dicho inversor implementa una estrategia de control no lineal que permite evaluar y comparar las prestaciones dinámicas del sistema versus la teoría clásica del control (PI) y será implementado como parte del trabajo desarrollado al interior del grupo de investigación MEM de la Universidad Santo Tomas. Por último, el desarrollo y entendimiento de este tipo de sistemas, dará paso para una posterior implementación de un sistema trifásico de uso industrial o residencial.

1.4. Impacto Social

Actualmente se presenta una alta demanda energética a causa del rápido desarrollo industrial y el aumento de la población, provocando problemas de producción energética y por ende ambientales, los cuales ocasionan repercusiones no deseadas, como lo son la contaminación del aire, la lluvia ácida y los gases de efecto invernadero, que no solo afectan la atmósfera terrestre sino también la salud de los seres vivos [9], [10].

De acuerdo a lo anterior, para reducir la liberación de sustancias químicas, como lo son el monóxido de carbono, dióxido de carbono, hidrocarburo, óxido de nitrógeno y dióxido de azufre, se proponen las fuentes de energía no convencionales, las cuales tienen un menor impacto ambiental gracias a su energía limpia y segura. Este tipo de tecnología a diferencia de las basadas en combustibles fósiles, no genera emisiones nocivas durante el proceso de producción energética y en el caso de tener efectos negativos, estos suelen ser pequeños y pueden ser atenuados [11].

Por lo tanto, el desarrollo de este proyecto se enfoca en adaptar el voltaje producido por las fuentes no convencionales, para el aprovechamiento de las energías limpias en lugares residenciales, con el fin de reducir los costos energéticos y a su vez presentar un impacto ambiental, cumpliendo con el objetivo número siete de la ODS (Objetivos de Desarrollo Sostenible), el cual se encarga de garantizar el acceso a una energía asequible y sostenible. Esto se realizará con base a las líneas de investigación del grupo MEM, para reforzar los conocimientos relacionados a los campos de control, energías renovables y eficiencia energética, por medio del uso de una estrategia de control que permita que un convertidor CC/AC entregue potencia activa a la red a partir de las energías renovables (en este caso, a partir de una fuente DC que simula un bus DC generado a partir de energías alternativas).

1.5. Diseño Metodológico

El diseño metodológico se divide en 3 fases.



FIGURA 1: Metodología Fuente: Autor

1.5.1. Fase 1: Selección

En primer lugar, se lleva a cabo un estudio y selección de información relacionada con diversas estrategias de control no lineales. El objetivo es identificar la estrategia más apropiada para su posterior implementación en el convertidor CC/AC. Esta elección tiene como propósito mejorar las prestaciones dinámicas del sistema, incluyendo aspectos como el tiempo de establecimiento, la relación de amortiguamiento y la capacidad de rechazo de perturbaciones. Para lograr con éxito esta fase, se consideran las siguientes tareas:

- Búsqueda bibliográfica sobre los temas de la investigación.
- Comparación entre los diseños de lazos de control lineal y no-lineal.

1.5.2. Fase 2: Diseño

Con base en la estrategia de control seleccionada, se procede al diseño del lazo de control para el convertidor CC/AC. Sin embargo, para llevarlo a cabo, es esencial considerar tanto el modelado matemático del convertidor como el diseño estático, ya que estos elementos determinarán los valores de los componentes que conforman el sistema. Siguiendo esta línea, se realiza una simulación en lazo abierto y, posteriormente, una simulación en lazo cerrado utilizando el diseño desarrollado. Para cumplir con éxito esta fase, se atienden las siguientes tareas:

- Búsqueda bibliográfica sobre los temas de la investigación.
- Modelado matemático de la topología del convertidor monofásico.
- Diseño estático para determinar los elementos del convertidor CC/CA.
- Simulación del convertidor CC/CA en lazo abierto.
- Diseño del lazo de control seleccionado a partir de la revisión bibliográfica.
- Simulación del convertidor CC/CA con lazo de control.

1.5.3. Fase 3: Validación

Finalmente, se adecua el convertidor CC/AC con el lazo de control diseñado para realizar la prueba experimental que regula la potencia a 200W, donde, de acuerdo a los resultados obtenidos, se hace una comparación con respecto a la simulación del sistema mencionado, evaluando las pérdidas y efectos negativos que se presentan en la prueba experimental. Por lo tanto, para el cumplimiento de esta fase se tienen en cuenta las siguientes tareas:

- Adecuación del convertidor CC/CA para implementar el lazo de control.
- Validar la estrategia del control en el convertidor obteniendo la potencia requerida.
- Comparar la estrategia de control con los resultados obtenidos con respecto a la simulación realizada.

1.6. Estado del Arte

La necesidad de gestionar la energía eléctrica que proviene de fuentes de energía no convencionales da paso al diseño y análisis de los convertidores conmutados, estos tienen como objetivo la adaptación necesaria en niveles de tensión, corriente y frecuencia para el adecuado abastecimiento de cargas tanto en corriente continua, como en corriente alterna. Por lo tanto, la tecnología VSC (Voltage Source Converter), es un concepto que convierte la forma de onda de un voltaje de corriente continua a corriente alterna, y viceversa. Para esto se requiere utilizar dispositivos capaces de lograr dicha conversión, tales como los semiconductores y dispositivos de alta potencia [12]. A continuación, se presentan algunas investigaciones y trabajos realizados sobre el tema de los convertidores conmutados en diferentes aplicaciones:

1.6.1. Investigaciones en el ambito del funcionamiento de los inversores monofásicos

La investigación realizada por Wenjie Liu, analiza el comportamiento de los inversores conectados a la red sin el uso de un transformador, mediante la comparación de las diferentes configuraciones de enlace en corriente continua, con el fin de identificar las ventajas y desventajas de usar cada una de las estructuras en aplicaciones fotovoltaicas [13]. De manera semejante Kamran Zeb en su trabajo presenta las diferentes topologías de un inversor puente completo, contemplando los principios operativos de cada uno, al igual que las distintas estrategias de modulación, ya sean de dos niveles, de tres niveles y de tipo híbrido [12].

En ambos casos se evidencia que los inversores sin transformador tienen algunos beneficios, como lo son el costo, una menor complejidad del sistema y la eficiencia, pero a su vez los dos trabajos expresan que la mayoría de las topologías descritas cuentan con problemas relacionados con las pérdidas del sistema, los niveles de armónicos, variaciones de tensión y la corriente de fuga. Por tanto K. Geetha propone las tecnologías híbridas como solución al problema de corriente de fuga, donde se combinan las técnicas AC-bypass y DC-bypass para mejorar el rendimiento de los convertidores CC/CA [14]. En cuanto a los armónicos en la salida del sistema, se propone el uso de un filtro LCL para reducir dicho inconveniente [15].

1.6.2. Investigaciones en el ambito de tecnica de control

De acuerdo a la investigación ejecutada por Manuel Tlapa Juárez y demás autores, se plantea una metodología llamada HIL(Hardware in the loop) la cual permite la conexión del convertidor trifásico a la red eléctrica en conjunto con una técnica de control por corriente aplicada para la inyección de potencia activa y reactiva; el inversor se alimenta de una fuente de voltaje que simula la salida de voltaje de una fuente renovable. Cabe destacar que en una investigación los autores señalan que la fase, amplitud y frecuencia de la red son claves a la hora sincronizar un sistema inversor con la red eléctrica; en los inversores la detección precisa de estos factores es una parte esencial para asegurar una correcta generación de señales de referencia y que posteriormente sean utilizados en un bucle de enganche de fase (PLL) [6].

Adicionalmente los autores mencionan el diseño y corroboran en condiciones normales y en tres escenarios anormales: variaciones de amplitud, amplitud y frecuencia, frecuencia y salto de fase. Además, se desarrollan resultados de simulación con el software PSIM para confirmar la validez de la estrategia [6].

De modo similar la tesis de grado realizada por Luiz Carlos de Araujo, plantea que la generación de energía eléctrica por medio de paneles solares, necesita convertidores con niveles de tensión que puedan ser utilizados dentro los hogares. Por lo cual, presenta la implementación de un convertidor CC/CA (inversor con topología

de puente completo), utilizando circuitos elevadores de tensión con algoritmos MPPT para obtener la máxima potencia instantánea del panel solar [16].

De manera académica en la universidad politécnica de valencia, se realiza un trabajo de grado con el fin de implementar el diseño de un inversor monofásico para las prácticas de laboratorio en la materia de electrónica de potencia; el objetivo principal es que la metodología que se realiza, sea replicada y empleada de forma experimental para implementación de los estudios. A partir de una previa investigación realizan diferentes simulaciones comprobando el funcionamiento del sistema diseñado en condiciones ideales y reales. Por último, se realiza el diseño del circuito impreso del inversor y se testea de forma experimental [17].

En el ámbito del control se encuentra una investigación que plantea un integrador de Segundo orden (SOGI) utilizado en la sincronización de red para inversores conectados a la red, sin embargo, si hay un desplazamiento de CC (corriente continua) existente en los voltajes de la red, el desempeño de SOGI sufre un efecto DC generado en la señal sinusoidal. Por lo tanto, se propone una mezcla entre un integrador de segundo y tercer orden generalizado (MSTOGI) para eliminar este efecto causado por la compensación CC de los voltajes de la red. Se realiza el diseño de un PLL aplicando MSTOGI y por lo tanto establecer un MSTOGI-PLL, que se plantea más adaptable a diferentes condiciones de red y un control adaptativo de frecuencia [13].

Muhammad Waqas ofrece un documento basado en el marco de referencia DQ, el cual se caracteriza por transformar componentes variantes en el tiempo en variables de régimen permanente, denominadas como D y Q. Así mismo, se implementa un regulador proporcional integral (PI) para el control de la potencia activa y reactiva inyectada a red, junto con el PLL encargado de sincronizar la fase y frecuencia de la corriente [15]. Por otra parte, para una mejor comprensión de la transformada DQ, José Andrés López Muñoz presenta de manera detallada el procediendo matemático de la transformación de Clark y Park, para el uso de un marco estacionario ortogonal en un sistema de control [18].

Al igual que en el caso anterior, Arpit Sharma considera el uso de la transformada DQ, para el diseño de un control de modo deslizante (SMC), encargado de regular la potencia activa de acuerdo a la conmutación de los transistores, esto con el fin de evitar las variaciones de potencia producidas por condiciones externas con las que trabajan los sistemas fotovoltaicos. Por lo tanto, la estrategia usada por Arpit presenta una mejora en las respuestas de estado transitorio y estacionario, frente al rendimiento de un regulador PI convencional [19]. En el mismo orden de ideas, el trabajo "Sliding control applied to a single-phase current source inverter" plantea un sistema SMC para evitar la distorsión armónica, para ello se usa un PLL para la sincronización con la red eléctrica, al igual que dos lazos de control. De acuerdo a lo anterior, el sistema propuesto muestra un buen rendimiento en cuanto la disminución de distorsión armónica total (THD) y un factor de potencia (PF) cercano a uno a pesar de los cambios atmosféricos [20].

Por otra parte, Kamran Zeb tal como en los casos anteriores, propone una estrategia de control deslizante, pero con un bucle de bloqueo SOGI, el cual se caracteriza por su rápida respuesta y rechazo armónico. Además, se realiza una comparación entre los resultados de un regulador PI tradicional y la estrategia propuesta, demostrando que el sistema presenta una mayor eficiencia, rapidez y estabilidad frente a variaciones de corriente o voltaje [21]. Al mismo tiempo Guorong Xu presenta una nueva estrategia de control de modo aislante de retroceso adaptativo (ABSMC), que gracias a su robustez frente a incertidumbres ofrece una alta estabilidad al sistema. En cuanto al diseño del controlador, los autores realizan el modelo matemático del inversor monofásico conectado a la red teniendo en cuenta las incertidumbres [22].

1.6.3. Investigaciones en el ámbito de las microrredes

En el campo de micro redes Iván Serrano Subtil en su tesis de grado habla de la necesidad de micro redes como una alternativa para sustituir los combustibles fósiles como fuentes de energía primaria. En su trabajo realiza la integración de la energía fotovoltaica obtenida de un panel solar y un sistema de almacenamiento de energía, que será una batería como también con la red eléctrica; se estudia a fondo el modelado y la complejidad del sistema puesto que se considera que la red eléctrica opera en corriente alterna a diferencia de la batería. Por lo cual su estudio se enfoca en los convertidores de fuentes de voltaje, una investigación detallada sobre las células que componen cada módulo fotovoltaico, la eficiencia que obtienen el sistema y el flujo e intercambio de energía que hay en todo el sistema [23].

Así mismo otra investigación ofrece algunos sistemas de nanored alimentados por energía renovable fotovoltaica, donde se da a conocer que estos sistemas son fiables a partir de recursos energéticos renovables. Por lo general una nanored usa dos etapas encargadas de la conversión, aunque también es posible usar un único convertidor para las necesidades de la carga. Sin embargo, también se cuenta con un diseño completo para múltiples nanoredes interconectados, donde cada una de estas tendrá un inversor de refuerzo conmutado que por medio de su método de control de bucle cerrado que proporciona tensión de enlace de corriente continua y a su vez se incluye un método de control que proporciona un flujo de control óptimo [24].

De la misma forma según la metodología realizada por Saldaña, el uso de microrredes de CA es un campo en crecimiento en la actualidad de las redes eléctricas, las cuales permiten suministrar energías a los sistemas de distribución y permiten un soporte a las diferentes empresas electrificadoras durante la operación, mantenimiento o fallas. Se realiza un modelo y simulación de una microrred, donde se revisan diferentes estrategias de control para la regulación de potencia y finalmente se realiza la simulación validando la estrategia de control sobre la microrred [5].

No obstante un artículo tiene un enfoque de tres etapas donde se tienen presente los modelos de sistemas de generación y almacenamiento de energías, puesto que estos sistemas tienen un amplio ámbito de estudio gracias a todos sus beneficios asociados, donde a su vez se explica que por lo general la integración de los DER en las microrredes eléctricas usa normalmente un modelo conformado por una fuente de energía, los sistemas de generación o almacenamiento de energía, el convertidor encargado de dichos sistemas y el convertidor de tensión VSC. Por lo tanto, el documento presenta resultados de acuerdo al modelo planteado al integrar los sistemas de generación y almacenamiento de energía en el algoritmo DQ0 de un convertidor de fuente de tensión VSC, el cual es útil en los estudios de las microrredes [25].

Por último, con base a lo encontrado en las diferentes investigaciones, es posible concluir que el campo de acción para la integración de estos convertidores va en crecimiento con la idea de aportar en la creación de diferentes generadores de energía que su fuente no sea principalmente la red eléctrica.

Capítulo 2

Marco Teórico

La siguiente sección del documento describe brevemente los tipos de convertidores, enfocándose en las topologías de los inversores y la importancia de estos en aplicaciones. Respectivamente se aborda el tema de control lineal y no lineal dando paso a la explicación de control sliding.

2.1. Topología de convertidores

Los convertidores se clasifican dependiendo de su conversión como se menciona a continuación:

- **Convertidores Corriente Alterna - Corriente Continua (CA/CC):** Este convertidor se le conoce como rectificador, su circuito más sencillo se basa en diodos y un filtro que permite tener la señal CC [26], [27]
- **Convertidores de Corriente Continua - Corriente Continua (CC/CC):** Este convertidor permite aumentar o disminuir la tensión CC de salida respectivamente de su entrada, logrando así tener una tensión regulada [26], [28].
- **Convertidor de Corriente Alterna - Corriente Alterna (CA/CA):** Este convertidor proporciona una tensión de salida en corriente alterna regulada a partir de una tensión de entrada en corriente alterna que no se encuentra regulada.[27], [29]
- **Convertidor de Corriente Continua - Corriente Alterna (CC/CA):** Este convertidor se le conoce como inversor, este se encarga de obtener una tensión en la salida de corriente alterna a partir de una tensión en su entrada de corriente continua[29], [26].

Los convertidores desempeñan un papel crucial al proporcionar un control sobre los parámetros de salida, como la tensión, la corriente y la frecuencia, según se requiera.

2.2. Topología de inversores

Un puente H como se muestra en la figura 2, es un circuito que tiene la capacidad de transformar la energía de corriente continua a corriente alterna con magnitud y frecuencia dependiendo del diseño que se desee realizar [30], este consta de un puente y una carga que puede ser solo una resistencia, una bobina en serie

con la resistencia y en paralelo con un condensador que permite tener la señal filtrada. Los inversores son ampliamente utilizados en la conversión de energía de fuentes renovables, como los paneles solares, para su posterior inyección en la red eléctrica[29],[26].

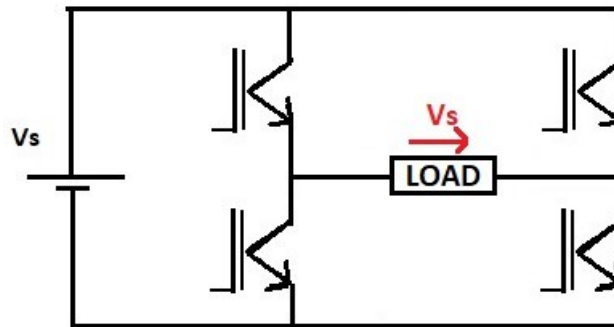


FIGURA 2: Topología inductor monofásico Fuente: Autor

Los inversores se pueden clasificar dependiendo de su señal de salida (Monofásicos y Trifásicos), por su configuración de la etapa de potencia (Medio puente, Puente completo y Multinivel) y por la técnica de control que se utiliza (Modulados y No modulados). A continuación, se da una breve explicación de cada una de sus clasificaciones

2.2.1. Inversor monofásico

Este inversor se considera Monofásico cuando en su salida solo se genera una fase de voltaje como se observa en la figura 2, este tipo de inversor está compuesto por transistores que pueden ser de baja o alta potencia, puede tener las dos configuraciones de la etapa de potencia y puede ser implementado con técnica de control por modulación o sin modulación[29]. Estos circuitos se utilizan en aplicaciones de baja potencia que requieran una alimentación de corriente alterna monofásica como son conversión de potencia, fuentes de tensión y puesta en marcha de motores monofásicos[31].

2.2.2. Inversor trifásico

Este inversor se conoce así porque puede estar conformado por 3 etapas de medio puente de inversores monofásicos conectados a la misma fuente como se observa en la figura 3[30][29]. Se debe tener en cuenta que las señales de disparo de cada etapa deben estar desfasadas 120° para lograr una tensión a la salida equilibrada [27]. Algunas de las aplicaciones de esta configuración es la alimentación de cargas trifásicas, puesta en marcha de motores y sistemas de energía renovable [31].

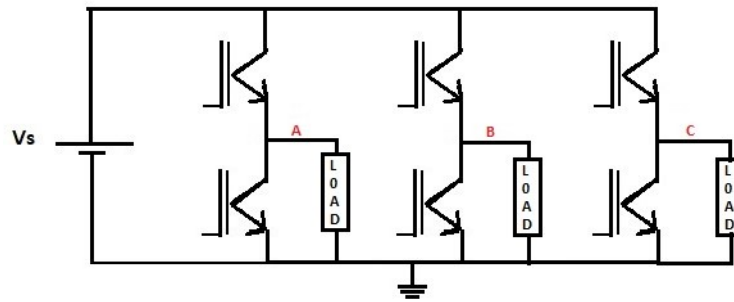


FIGURA 3: Topología inversor trifásico Fuente: Autor

2.2.3. Inversor medio puente

La configuración medio puente que se observa en la figura 4 es una de las más sencilla, está conformado por dos fuentes de tensión en serie de igual valor $\frac{V_s}{2}$ y $\frac{V_s}{2}$, por dos interruptores Q1 y Q2 y en su salida una carga como se observa en la figura 4. Su funcionamiento consta de producir una conmutación en Q1 que genera una tensión en la carga de $\pm \frac{V_s}{2}$, durante el otro semiperiodo en que se conmuta Q2, la tensión en la carga será $-\frac{V_s}{2}$. Al ser medio puente la tensión en la carga, considerando que sea solo una resistencia será una señal cuadrada con una tensión de $\frac{V_s}{2}$ y si la carga ya tiene filtros implementados (bobina y condensador) se tiene una señal seno reconstruida como se observa en la figura 4 [30].

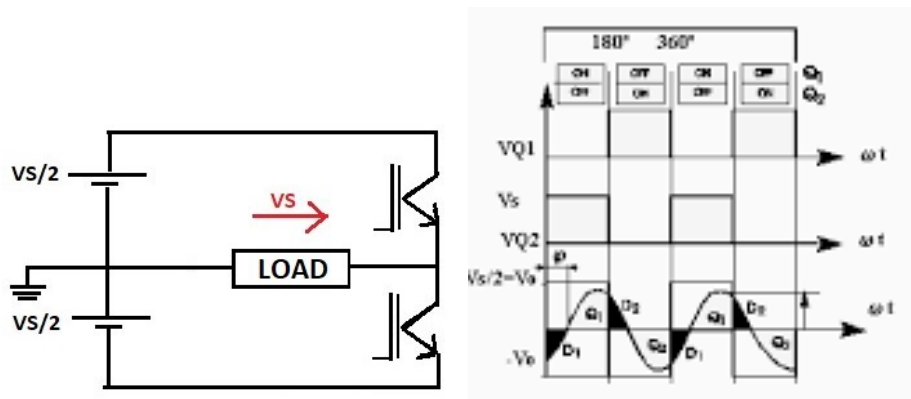


FIGURA 4: Estructura medio puente Fuente: [30]

2.2.4. Inversor puente completo con carga

El inversor puente completo que se observa en la figura 5 se compone de cuatro interruptores donde Q1 y Q2 están conectados en serie y en otra rama Q3 y Q4, así mismo estas dos ramas están conectadas en paralelo. Esta configuración se compone de 4 interruptores una fuente de voltaje Vs y una carga. En el semiciclo positivo se conmutan dos transistores siendo estos Q1 y Q4 que generan una tensión en la carga del valor de Vs y en el semiciclo negativo se conmutan los transistores Q2 y Q3 generando una tensión del valor de Vs pero invertida como se observa en la figura 5 [30].

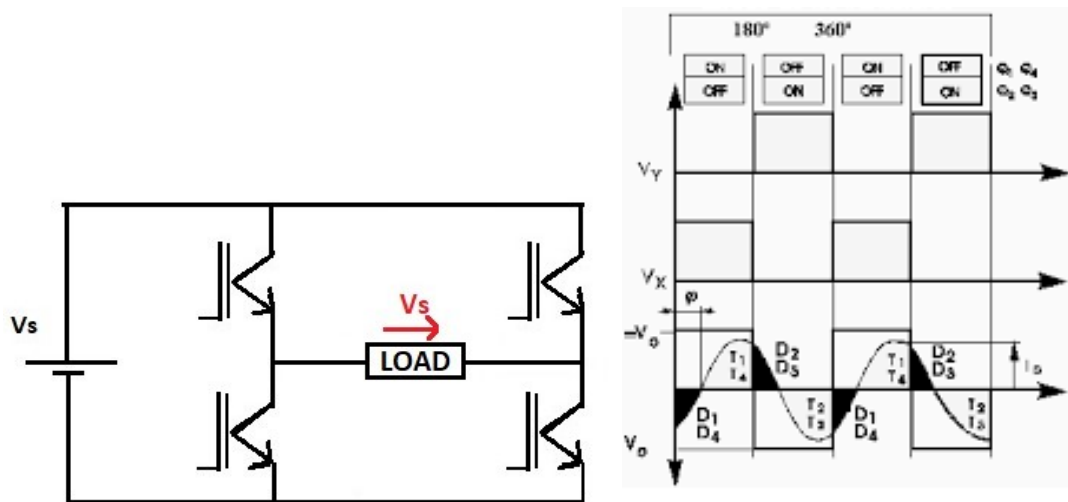


FIGURA 5: Estructura medio puente Fuente: [30]

2.2.5. Inversor multinivel

Los inversores multinivel despiertan un gran interés debido a sus notables características, que resultan de gran utilidad en la compensación de potencia. En particular, destacan por su capacidad para generar inversores de alta potencia y voltaje de manera más sencilla en comparación con otras configuraciones. Esto se debe a que al incrementar el número de niveles de voltaje en el inversor, es posible elevar su potencia nominal y, al mismo tiempo, reducir la presencia de armónicos en la señal de tensión de salida. Es ampliamente reconocido que una estructura de más de tres niveles produce una disminución significativa en la cantidad de armónicos presentes[32]. En la figura 6 se observa una configuración multinivel.

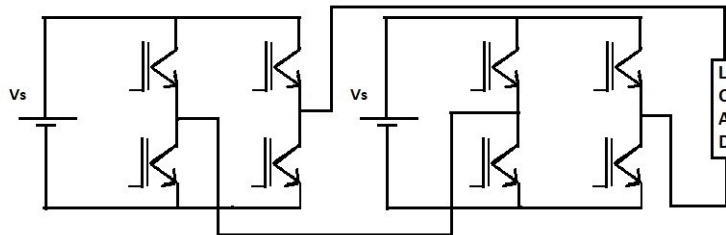


FIGURA 6: Topología inversor multinivel Fuente: Autor

El desarrollo de los inversores se ha visto impulsado por los notables avances en la tecnología de semiconductores de potencia y las fuentes de corriente continua. La correcta ejecución de las conmutaciones es fundamental, ya que esta acción permite la creación de escalonamientos en la tensión de entrada, dando lugar a una señal de voltaje con múltiples niveles.

2.3. Técnicas de Control

Los convertidores conmutados ya sean CC/CC, CC/CA y diferentes topologías que se utilizan en electrónica de potencia, utilizan diferentes técnicas de control para regular el proceso de conversión de voltaje o corriente. Este se logra mediante la implementación de técnicas de control que actúan sobre el convertidor teniendo en cuenta las características que se buscan en el inversor como propiedades de eficiencia, contenidos armónicos en la señal de salida y sus pérdidas, cada técnica tiene sus ventajas y desventajas[33][34],[35].

Las técnicas de control varían dependiendo del enfoque o la complejidad del sistema. Las técnicas lineales se aplican en sistemas que puedan aproximar sus entradas y salidas linealmente, cabe aclarar que el sistema puede tener no linealidades pero estas son pequeñas o pueden ser ignoradas[34]. Por otro lado las técnicas de control no lineales se aplican a sistemas en los cuales la relación entre sus variables de entrada y salida no son proporcionales y tienen un comportamiento no lineal, se utiliza donde las no linealidades, oscilaciones o perturbaciones grandes son significativas en el sistema[36]. Sin embargo, algunas técnicas de control no lineales se pueden utilizar en sistemas lineales para mejorar el rendimiento del sistema ya que este control tiene cierta flexibilidad para trabajar en sistemas que no presentan no linealidades lo que permite tener un control más preciso y eficiente [37].

A continuación se presentan 4 técnicas de control que se han utilizado para gestionar y estabilizar inversores monofásicos, estas técnicas son utilizadas ya sea por su simplicidad en la implementación o por su capacidad de garantizar una operación correcta de los inversores ante perturbaciones.

2.3.1. Control por modulación escalar o PWM

La modulación por ancho de pulso es una técnica de control que tiene un papel importante al controlar tensión o corriente en convertidores conmutados, esta técnica implica la generación de una señal de salida a través de la manipulación del ancho de pulsos de la señal de control[38]; esta señal es la que permite la excitación de transistores de conmutación que cambian su estado a alta velocidad[39], construyendo una onda de salida compuesta por pulsos.

La señal de control es una señal periódica que puede adoptar formas senoidales o cuadradas. La manipulación del ciclo de trabajo de esta señal desempeña un papel fundamental en el control de la cantidad de energía que se obtiene en la salida, su expresión matemática en la ecuación 2.1.

$$D = \frac{\tau}{T} \quad (2.1)$$

- D : ciclo de trabajo
- τ = ancho de pulso
- T periodo de la señal

Las técnicas de Modulación por Ancho de Pulso (PWM) se dividen en diversas categorías que incluyen:

- PWM de Amplitud AMPWM: Se modifica la amplitud de la señal de control sin alterar la frecuencia y el periodo [40].
- PWM Senoidal: La señal PWM se modula para que los pulsos adapten una forma senoidal[38] [41].
- PWM de Frecuencia FMPWM: La frecuencia de la señal se modifica sin alterar la amplitud de esta [40].
- PWM de fase: La fase de la señal PWM se modifica con el fin sincronizar la señal de salida en relación con la referencia [40].

La implementación de este control consta de un microcontrolador o cualquier dispositivo en el cual se pueda realizar la comparación entre una señal diente de sierra y la señal moduladora. En la salida del comparador se genera una señal con la frecuencia de la diente de sierra y con el ciclo de trabajo de la señal moduladora [38] [39]. Una de las principales desventajas de este control es que puede presentar ruido o interferencias que necesitan solucionarse con la utilización de filtros.

2.3.2. Control de realimentación Proporcional-Integral-Derivativo

El control PID, es una de las técnica mas utilizadas en el campo del control automático por su flexibilidad y capacidad, teniendo una gran habilidad para regular sistemas en un estado deseado. Esta técnica es ampliamente utilizada en inversores, regulando la tensión de salida del inversor como también llega a ser utilizado como un control de corriente [42]. Esta técnica se implementa por medio de tres ganancias que se presentan a continuación:

- Proporcional (P): Esta constante produce una señal de control proporcional al error que se tiene entre la salida deseada y la salida actual del sistema [43] [44].

Esta acción de control es directamente proporcional a la señal de error, cuando el error es grande la señal proporcional también lo es, no obstante hay implicaciones que se deben tener en cuenta si esta ganancia proporcional aumenta considerable, puesto que puede mejorar la respuesta del sistema pero también aumentar la inestabilidad del sistema [43].

$$Kp = Kp * e(t) \quad (2.2)$$

- Integral (I): Esta constante integra la variable del error con el fin de lograr una señal de error constante en estado estacionario, garantizando que la salida converja al valor deseado [43] .

Esta integral se define como la acumulación de la señal de error, se van sumando pequeños errores con el fin de que la acción integral sea mayor. La etapa integral consigue reducir el error en régimen permanente, aumentar esta constante puede aumentar la inestabilidad del sistema pero aumentar la respuesta del sistema [42].

$$Ki = Ki \int_0^t e(t) dt \quad (2.3)$$

- Derivativo (D): Este termino tiene la función de calcular la tasa de cambio de error y producir una señal de control a la velocidad de cambio del error, en otras palabras es la derivada del error multiplicada por una constante; con el fin de reducir la respuesta transitoria y evitar oscilaciones. Un problema que presenta utilizar la acción derivativa es que amplifica señales con alta frecuencia como lo es el ruido y este se puede presentar en la señal de error provocando acciones del control no deseados [44] [42].

$$Kd = Kd \frac{d(e)}{dt} \quad (2.4)$$

La ecuación general del controlador se observa en la ecuación

$$c(t) = Kp * e(t) + Ki \int_0^t e(t) dt + Kd \frac{d(e)}{dt} \quad (2.5)$$

En diversas aplicaciones, la combinación de solamente de la acción proporcional y la acción integral permite lograr una regulación precisa y la estabilidad de los sistemas. En el contexto de inversores, esta estrategia garantiza la generación de una señal senoidal constante y la capacidad de adaptarse eficazmente a diversas cargas y perturbaciones[44].

Es importante tener en cuenta que en algunos sistemas, existen limitaciones inherentes que pueden restringir la capacidad del controlador para alcanzar la salida deseada. A pesar de los esfuerzos de sintonización ya sea por el metodo de Ziegler y Nichols o por algoritmos de busqueda [45], el controlador puede llegar a una condición de saturación, lo que provoca un comportamiento inadecuado en el sistema [44] **pid2** [43].

2.3.3. Control Predictivo basado en Modelo MPC

El Control Predictivo es una técnica de control avanzada que ha desempeñado un papel fundamental en el campo de la electrónica de potencia, transformando la forma en que se gestionan los convertidores de potencia y otros dispositivos que requieren precisión y eficiencia[46]. A pesar de que muchos procesos reales son no lineales, en ocasiones es posible aproximarlos como sistemas lineales dentro de un rango operativo específico. En estos casos, los enfoques de Control Predictivo basado en Modelo (MPC) lineal se convierten en una herramienta valiosa, especialmente cuando se enfrentan desafíos de predicción debido a discrepancias entre el modelo y el proceso real. Estos controladores MPC lineales aprovechan el principio de superposición del álgebra lineal, lo que simplifica los cálculos y garantiza velocidad en el control [47] [48].

Sin embargo, cuando los modelos lineales no logran representar con suficiente precisión las no linealidades inherentes al proceso real, se recurre a enfoques alternativos. En ciertos casos, se aplican transformaciones a las variables del proceso antes o después de su modelado para mitigar las no linealidades. Asimismo, se puede optar por el uso de MPC no lineal, donde el modelo no lineal se emplea directamente para el control del proceso. El MPC abarca diversos métodos y algoritmos, todos destinados a anticipar y asegurar una señal de control que responda a perturbaciones o cumpla con requisitos de rendimiento en tiempo real[48] [49] [50].

Este control es la agrupación de diferentes algoritmos acogidos de algunas estrategias de control desarrolladas en torno a controladores lineales, por lo que el MPC tiene las siguientes características:

- Para predecir el comportamiento de las salidas en el futuro, el MPC utiliza el modelo matemático del sistema[51] [46].
- Debe conocer las condiciones iniciales del sistema para el control y la predicción[51].
- Su tarea principal es minimizar la función de costos [46].
- Este control tiene dos conceptos fundamentales que son el horizonte de control y el horizonte de predicción[46].

2.3.3.1. Horizonte de predicción

Una de las características particulares del MPC es el horizonte de predicción N . Este horizonte, denotado como N y actualizado en cada instante t , constituye el intervalo de tiempo durante el cual se analizan las decisiones de control presentes y futuras. En cada paso, se utiliza este horizonte para anticipar el comportamiento de las salidas del sistema[48]

Teniendo en cuenta lo anterior, empleando el horizonte de predicción se hallan las salidas futuras utilizando el modelo matemático de la planta que es también llamado modelo de predicción como se observa en la ecuación 2.7. El comportamiento de las salidas futuras dependen de los valores conocidos hasta el instante t y de las señales de control[48].

$$\gamma(t) + 1(t), \gamma(t) + 2(t), \gamma(t) + 1(t), \dots, (t) + N(t) \quad (2.6)$$

La ecuación 2.7 termina siendo una secuencia que se calcula mediante la información de la evolución de la planta en cada instante t [51]. Es decir:

- Los valores pasados de entradas o respuestas del proceso[51].
- Los valores pasados y presentes de las salidas de la planta[51].

2.3.3.2. Horizonte de control

Por otro lado se tiene el horizonte de control N_c , este determina cuantas veces en el futuro se debe calcular la señal de control en cada intervalo de muestreo, se representa la ecuación ??; este horizonte es un numero positivo que representa la cantidad de intervalos en los cuales se va a evaluar la señal de control[51]. La elección de este numero es crucial puesto que este determina cuantos pasos hacia adelante se deben tener en cuenta para determinar la señal de control actual.

$$u(t), u(t + 1t), u(t + 2t), \dots, u(t + N_c - 1t) \quad (2.7)$$

Se debe tener en cuenta que entre mas amplio sea el horizonte de control mejora el rendimiento a largo plazo pero esto implicaría mas procesos computacionales que podrían afectar la respuesta del control [51] [48].

2.3.3.3. Función de costo

La secuencia de señales de control encontradas por medio del horizonte de control, buscan minimizar la función de costos dada por la ecuación 2.8 [51], con el fin de mantener la salida de la planta lo mas cercana a la trayectoria de referencias futuras. Este criterio en ocasiones es una función cuadrática entre el error en la salida previamente predicha y la trayectoria de referencias futuras[51] [48].

$$\omega(t + k) \quad (2.8)$$

Con base a lo anterior, el control predictivo basado en un modelo esta constituido por el horizonte predictivo, el horizonte de control y la función de costo, estos conceptos se ilustran en la figura 7

2.3.3.4. Optimizador

La herramienta en la cual se realizan los cálculos de las señales de control con el fin de disminuir la función de costo se le conoce como optimizador. El objetivo del optimizador es disminuir la función de costo, teniendo

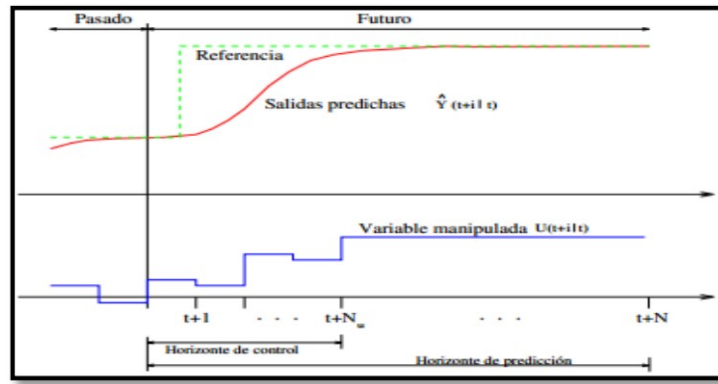


FIGURA 7: Estrategia MPC Fuente: [51]

en cuenta que el sistema puede presentar restricciones y dinámicas del mismo, por lo cual esta herramienta utiliza el horizonte de predicción, el modelo de predicción para determinar la secuencia que logra minimizar la función de costos [47] [50].

Existen diferentes tipos de optimizadores, y la elección depende de las características del problema, la disponibilidad de los tiempos de ejecución y las restricciones propias del problema. Algunos se presentan a continuación:

- Métodos de Programación Lineal (PL): Los problemas que se presentan en MPC llegan a ser programación lineales o programación cuadrática y se pueden resolver con algoritmos como el método Simplex o el de barrera interior [47].
- Métodos de Programación No Lineal (NLP): Por el contrario si el problema es no lineal, se utilizan algoritmos para resolver el problema como el método de Newton-Raphson [50].
- Métodos de Optimización Numérica: Este método consta de usar diferentes herramientas como MATLAB's `fmincon` o Python's `scipy.optimize` que ofrecen herramientas para resolver problemas numéricos como el MPC [47] [50].

2.3.4. Control en modo deslizante

El control deslizante se caracteriza por ser una de las estrategias de control más eficientes y populares en los convertidores de potencia, puesto que la principal ventaja de este método es su poca dependencia de un modelo exacto del sistema a controlar, así mismo tiene una baja sensibilidad con respecto a las perturbaciones que puedan presentarse en la planta. Por lo tanto, el control de modo deslizante consiste en forzar un sistema a realizar un movimiento sobre una superficie deslizante, la cual es una ruta que el sistema debe seguir de manera forzada para alcanzar la estabilidad [52] [53].

2.3.5. Superficie de deslizamiento

Es la trayectoria que permite llevar al sistema que está siendo controlado al punto de equilibrio "O", esto quiere decir que, sin importar la condición inicial del sistema a controlar, este siempre tenderá a tener una trayectoria hacia la superficie deslizante. Con base a esto, se dice que el control de modo deslizante se encuentra constituido por dos fases, conocidas como fase de alcance y fase deslizante, las cuales se presentan en la figura

8. De acuerdo a esto, la primera fase se caracteriza por ser aquella donde la trayectoria “S” es forzada a dirigirse hacia la superficie deslizante, y una vez que se encuentre en ella, entra en juego la segunda fase, donde “S” sigue la trayectoria de la superficie hasta asentarse en el punto de equilibrio [52] [53].

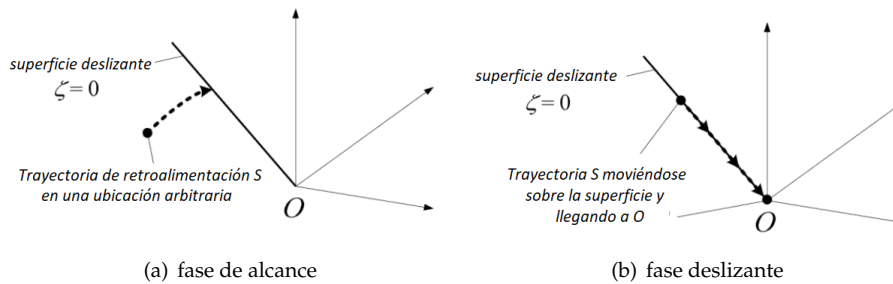


FIGURA 8: Superficie de deslizamiento. **Fuente:** [52]

En este orden de ideas, la superficie de deslizamiento “S” se selecciona dependiendo del orden y la complejidad del sistema a controlar. Donde la expresión de la ecuación 2.9 es la superficie de deslizamiento más comúnmente usada cuando se tiene un sistema lineal. En este contexto se dice que λ_n representa el coeficiente de deslizamiento, x_2 que es el error del sistema y x_1 la derivada temporal del error. Además, cuando se trata de un sistema de orden superior se puede utilizar la expresión de la ecuación 2.10 [53].

$$S(x) = \lambda_1 x_1(t) + x_2 \quad (2.9)$$

$$S(x) = \lambda_1 x_1(t) + \lambda_2 x_2(t) + \dots + \lambda_n x_n(t) \quad (2.10)$$

Sin embargo, también se tiene en cuenta una superficie de deslizamiento para sistemas no lineales como se presenta en la ecuación 2.11, donde γ debe tener un valor entre cero y uno para mantener la estabilidad del sistema.

$$S(x) = \lambda_1 x_1^\gamma(t) + x_2 \quad (2.11)$$

2.3.6. Coeficiente deslizante

Este es el encargado de determinar la respuesta dinámica del sistema que esta siendo controlado, donde se dice que entre mas grande sea el coeficiente, más rápida es la respuesta dinámica del sistema. Sin embargo, si se tiene un coeficiente muy grande, se reduce la región de existencia, lo que causa inestabilidad. De acuerdo esto el coeficiente de deslizamiento se selecciona en algunos casos por medio del ensayo y error, o también puede ser determinado con la raíz de los polos en bucle cerrado, así mismo es importante tener en cuenta que los coeficientes usados en la superficie de deslizamiento deben ser mayores que cero para mantener la estabilidad del sistema [52].

2.3.7. Problema de chattering

A pesar de que el control deslizante se caracteriza por ser una técnica sencilla, con una buena respuesta dinámica y buenas propiedades de regulación, la estrategia en algunos casos, presenta problemas de vibración (chattering). Dicho inconveniente es un obstáculo para la aplicación del control deslizante, puesto que, a causa de la rápida conmutación del controlador, la no linealidad del sistema aumenta provocando ruido en el laso

de control, dado que el chattering hace que se exciten todos los elementos del circuito que no se tuvieron en cuenta para el modelado usado en diseño del control deslizante [54].

2.3.8. Entrada de control

La entrada de control es aquella que permite garantizar que las variables de estado sean forzadas a estar sobre la superficie de deslizamiento, por lo tanto una forma de definir la entrada de control es por medio del control equivalente, que se determina usando la expresión $S(x) = \dot{S}(t) = 0$. De acuerdo a esto la expresión usada para realizar el diseño de la entrada de control se presenta en la ecuación 2.12 [53].

$$\dot{S}(t) = \frac{\partial S}{\partial x} f(x, t) + \frac{\partial S}{\partial x} g(x, t) u_{eq}(t) = 0 \quad (2.12)$$

Sin embargo, el uso del control equivalente requiere de dos funciones a evaluar, que en este caso son $f(x, t)$ y $g(x, t)$, esto quiere decir que, si no es posible aplicar este tipo de entrada en el control deslizante, otra opción a usar es el método de velocidad constante, que es una alternativa muy usada en este tipo de control. Por otra parte, es importante tener presente que este método a pesar de ser usado con frecuencia trae consigo problemas de chattering. El método de velocidad constante se expresa como se presenta en la ecuación 2.13, donde K hace referencia a la velocidad de convergencia [53].

$$\dot{S}(t) = -K \text{sign}(S(x)) \quad (2.13)$$

2.4. Transformación DQ0

El marco giratorio dq se caracteriza por ser una técnica comúnmente usada en el diseño de compensadores para sistemas trifásicos, como los son motores eléctricos e inversores. Esto se debe a su capacidad para transformar los voltajes o corrientes de un sistema trifásico a coordenadas rotativas que son invariantes en el tiempo, conocidas como d y q o también llamadas componente directa y de cuadratura. De acuerdo a esto, la transformación simplifica y facilita el análisis de un sistema eléctrico, puesto que ya no se evalúan las tres variables referentes a las fases (a , b y c) sino únicamente d y q , las cuales contarán con un compensador independiente que generalmente suele ser un PID convencional. Por lo tanto para realizar este procedimiento se deben considerar dos transformaciones, conocidas como la transformada de Clarke y Park..

2.4.1. Transformada de Clark

Esta transformación se caracteriza por obtener una coordenada bifásica conocida como $\alpha\beta$ a partir un sistema de tres fases (abc), de acuerdo a esto, para realizar esta transformación se usa la expresión presentada en la ecuación 2.14.

$$\begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (2.14)$$

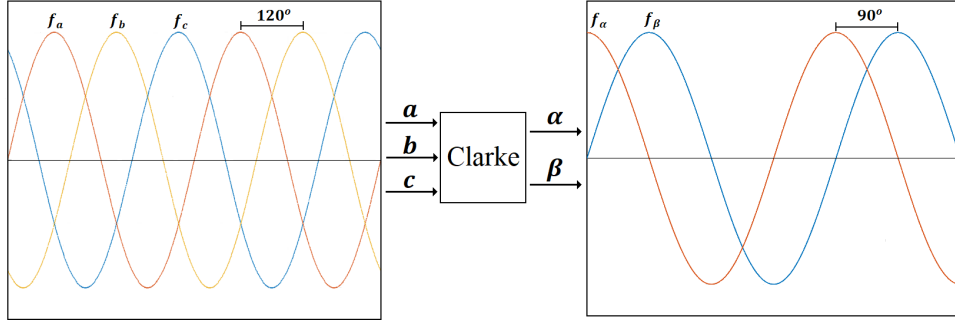


FIGURA 9: Transformada de Clarke Fuente: Adaptado de [55]

2.4.2. Transformada de Park

A diferencia de la transformada de Clark que aun da como resultado $\alpha\beta$ que son variantes en el tiempo, la transformada de Park se encarga de tomar las coordenadas bifásicas y convertirlas en el marco giratorio DQ, las cuales permanecen con un valor constante. Por lo tanto, la expresión usada para realizar esta transformación se presenta en la ecuación 2.15

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \begin{bmatrix} \cos(\omega * t) & \sin(\omega * t) \\ -\sin(\omega * t) & \cos(\omega * t) \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} \quad (2.15)$$

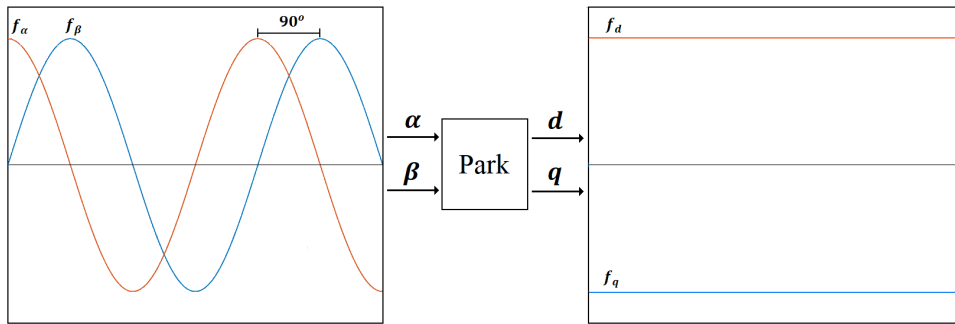


FIGURA 10: Transformada de Park Fuente: Adaptado de [55]

De acuerdo a las ecuaciones 2.14 y 2.15, la expresión general para la transformada DQ0 se encuentra representada en la ecuación 2.16 junto con su inversa en la ecuación 2.17, donde la relación entre el sistema de tres fases y la matriz de transformación dq0 es: $X_{dq0} = T_\theta * X_{abc}$ y $X_{abc} = T_\theta^{-1} * X_{dq0}$, así mismo la multiplicación directa entre las matrices da como resultado la matriz identidad $T_\theta * T_\theta^{-1} = T_\theta^{-1} * T_\theta = I_{3 \times 3}$.

$$T_\theta = \frac{2}{3} \begin{bmatrix} \cos(\omega * t) & \cos(\omega * t - \frac{2\pi}{3}) & \cos(\omega * t + \frac{2\pi}{3}) \\ -\sin(\omega * t) & -\sin(\omega * t - \frac{2\pi}{3}) & -\sin(\omega * t + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.16)$$

$$T_\theta^{-1} = \begin{bmatrix} \cos(\omega * t) & -\sin(\omega * t) & 1 \\ \cos(\omega * t - \frac{2\pi}{3}) & -\sin(\omega * t - \frac{2\pi}{3}) & 1 \\ \cos(\omega * t + \frac{2\pi}{3}) & -\sin(\omega * t + \frac{2\pi}{3}) & 1 \end{bmatrix} \quad (2.17)$$

2.5. Transformación DQ en sistemas de una sola fase

Puesto que este método requiere de un sistema de tres fases, no es usual que se use en convertidores monofásicos, dado que no existe una relación directa entre un sistema de una única fase y la transformación al marco giratorio DQ. Sin embargo, es posible emular una segunda fase y de esta manera tener un sistema bifásico al cual aplicar la transformada de Park para obtener las variables de régimen permanente D y Q. Por lo tanto, para la construcción de un sistema con coordenadas bifásicas $\alpha\beta$ a partir de un sistema de una sola fase, se tiene en cuenta un circuito imaginario, exactamente igual que el convertidor que se esté trabajando, donde la única diferencia es un desfase de 90 grados. De acuerdo a lo anterior en la figura 11 se presenta la emulación de la segunda fase de un inversor monofásico [56].

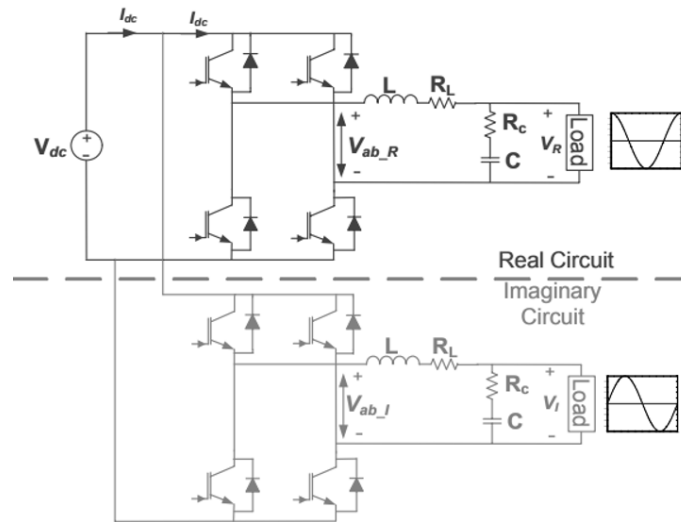


FIGURA 11: Circuito Real e Imaginario Fuente: [56]

De acuerdo a la figura 11, los voltajes de salida del sistema se definirían como se presentan en la ecuación 2.18 y 2.19.

$$v_R(t) = A_R \cos(\omega * t + \phi) \quad (2.18)$$

$$v_I(t) = A_I \sin(\omega * t + \phi) \quad (2.19)$$

Por lo tanto, para realizar la transformación al marco giratorio DQ, inicialmente se debe obtener el modelo conmutado de todo el sistema presentado en la figura 11 y representar las ecuaciones diferenciales de voltajes y corrientes del circuito real e imaginario. Una vez determinadas las ecuaciones diferenciales, se usa la técnica de promediado, la cual convierte la ecuación diferencial en una media móvil, que es usada cuando se parte de que la frecuencia de conmutación es lo suficientemente alta para que la dinámica del sistema sea contante. De acuerdo a lo anterior, la expresión para determinar el promediado se encuentra representada por la ecuación 2.20 [56].

$$\bar{x}(t) = \int_{t-T}^t x(\tau) d\tau \quad (2.20)$$

Con base al promediado del sistema, el circuito de la figura 11 puede ser representado por la figura 12, donde $d(t)$ es el ciclo de trabajo y la corriente I_{dc} es igual a la corriente I_R , puesto que I_{dc} es la corriente del circuito real.

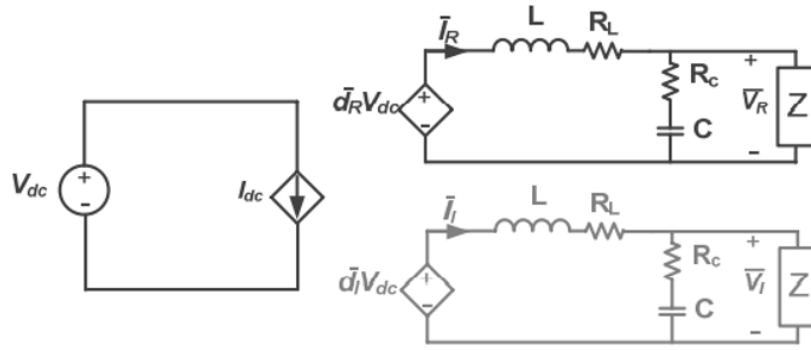


FIGURA 12: Circuito Real e Imaginario promediado **Fuente:** [56]

Puesto que ya se tiene el modelo promediado del sistema completo, ya es posible realizar la transformación al marco giratorio DQ usando la expresión $X_{\alpha\beta} = T^{-1} * X_{dq}$, y de esta manera obtener la matriz en variables de estado con respecto al marco giratorio. Sin embargo, al reemplazar dicha expresión dentro de las variables de estado, se obtiene una derivada con dos funciones que están siendo multiplicadas $((fg)'(x))$, la cual se resuelve usando la regla del producto, que se presenta en la ecuación 2.21 [56].

$$\frac{d(uv)}{dt} = v \frac{d(u)}{dt} + u \frac{d(v)}{dt} \quad (2.21)$$

De tal modo que una vez realizada la transformación DQ el circuito de la figura 11 puede ser representado por la figura 13.

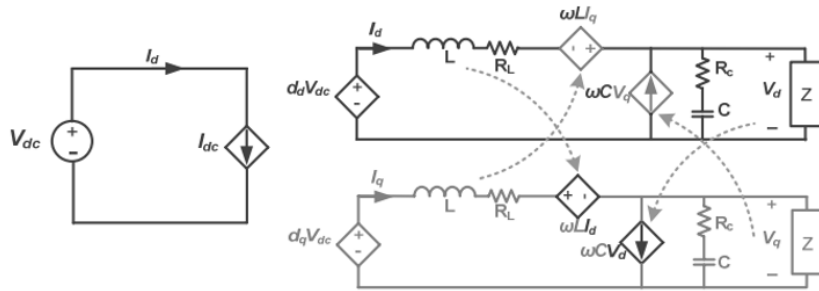


FIGURA 13: Circuito transformación DQ **Fuente:** [56]

Capítulo 3

Caracterización de la planta

3.1. Descripción de la planta general

El sistema implementado por el grupo MEM que se utiliza para la implementación del control cuenta con dos etapas para su funcionamiento. La primera es la etapa de potencia observada en la figura 14, esta se encuentra constituida por una topología de seis interruptores IGBT junto con un driver encargado de la medición de temperatura para el enfriamiento del mismo que permite regular la velocidad de un ventilador de 12 voltios. En la sección de anexos 10.6 se encuentra la hoja de datos.

Adicionalmente el sistema permite realizar la medición de corrientes y tensiones, en este caso se miden dos corrientes, dos voltajes en corriente alterna y dos voltajes en el bus DC; que se pueden observar en la sección de anexos 10.6.

En cuanto a la segunda etapa que se observa en la figura 15, esta es la encargada de trabajar toda la parte digital, como lo son las señales de disparo de la etapa de potencia con ayuda de una tarjeta DSP de Texas instruments, de acuerdo esto el sistema cuenta con seis etapas de “controlador driver’s” con la capacidad de manejar 12 dispositivos de conmutación [18]. En el anexo 10.6 se encuentran los planos de la planta.

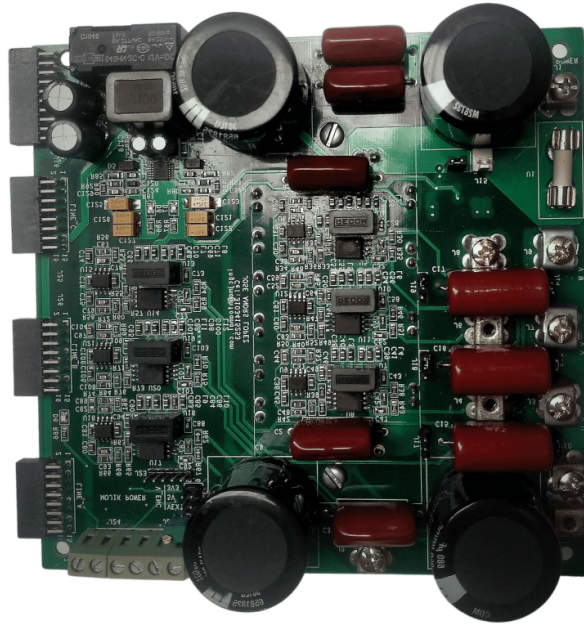


FIGURA 14: Etapa de potencia de la planta Fuente: [57]

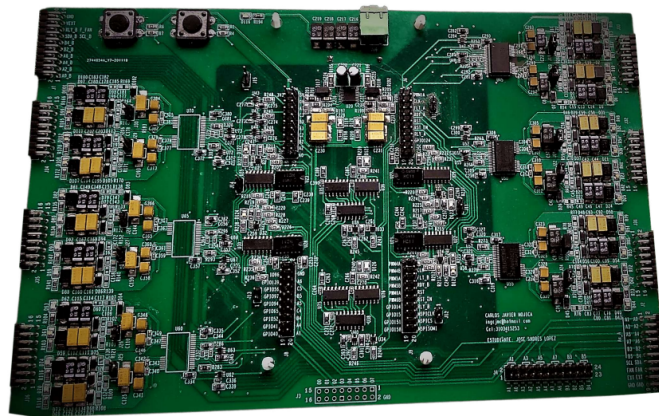


FIGURA 15: Etapa digital de la planta Fuente: [58]

3.2. Drivers de la planta

En el diseño se implementa un driver 2ED020I12-F2, el cual se caracteriza por ser un excitador de doble puerta IGBT, que proporciona dos salidas de controlador totalmente independientes y a su vez puede ser usado para excitar dispositivos MOS de potencia. Una de las ventajas de usar este driver son las funciones de control y protección que se encuentran incluidas, dado que de esta manera se realizan sistemas de alta fiabilidad, entre dichas funciones se tiene la proyección por saturación del IGBT con salidas de estado FAULT y dos salidas de estado READY que informan si el dispositivo está alimentado y funciona correctamente. A continuación, se presenta un esquema del Driver 2ED020I12-F2 [59].

Bloqueo por subtensión (UVLO) Como se mencionó el driver cuenta con funciones de protección para garantizar que los IGBT conmuten de manera adecuada, por lo que se tiene presente el bloqueo por subtensión, que establece una baja tensión en la polarización tanto para la parte digital como para la parte de tensión de disparo del transistor para todas las salidas del excitador. Esto quiere decir que el driver apaga las salidas de los IGBT y envía una señal al DSC para que este enterado [59]. **Salida de estado ready (LISTO)** La salida de estado READY se caracteriza por tener tres protecciones internas, las cuales se presentan a continuación:

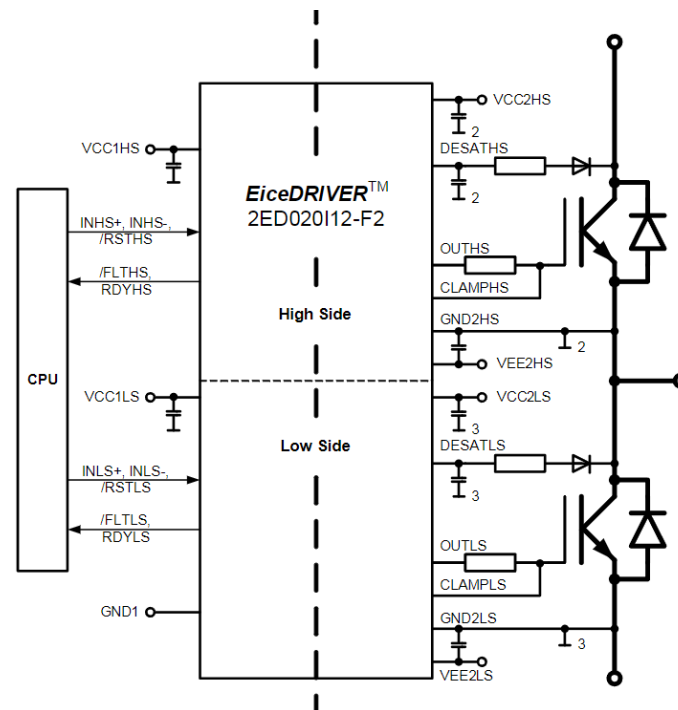


FIGURA 16: Driver 2ED020I12-F2 Fuente: [59]

- UVLO del chip de entrada.
- UVLO del chip de salida tras un breve retardo.
- Transmisión de señal interna tras un breve retardo.

3.2.1. Protección contra desaturación

Esta protección es usada cuando se presenta un caso de cortocircuito, esto quiere decir que la corriente en el transistor aumenta y por ende, lo mismo pasa con la tensión colector emisor del IGBT, esta situación es sensada por el driver, deshabilitando las salidas para proteger el transistor, e indicando al microcontrolador de lo sucedido para que este ejecute un reset al driver, realice un conteo de fallas y posteriormente apague el sistema, o simplemente apague inmediatamente el sistema por detección de corto circuito [59].

3.3. Sensores de la planta

Como se mencionó con anterioridad, el sistema cuenta con seis sensores, dos de corriente, dos de voltaje recibiendo señales en el lado de corriente alterna y dos de voltaje en el bus DC. Adicionalmente se puede seleccionar si el sensado de la tensión se realiza por fase o por línea, así mismo cada sensor cuenta con una etapa de filtrado para trabajar con las señales medidas de manera digital [18], a continuación, se presentan los esquemas usados para la implementación de los seis sensores mencionados.

3.3.1. Sensores de voltaje

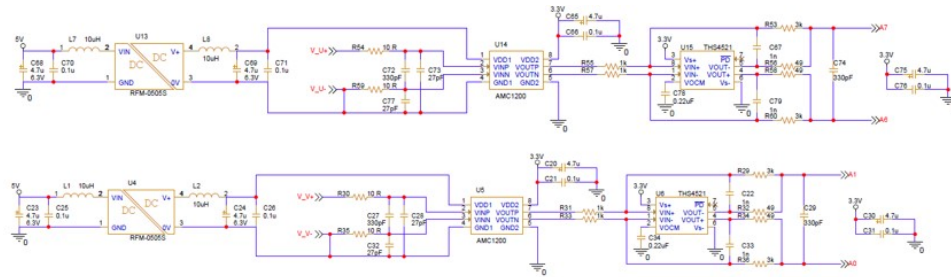


FIGURA 17: Sensores de Voltaje DC Fuente: [57]

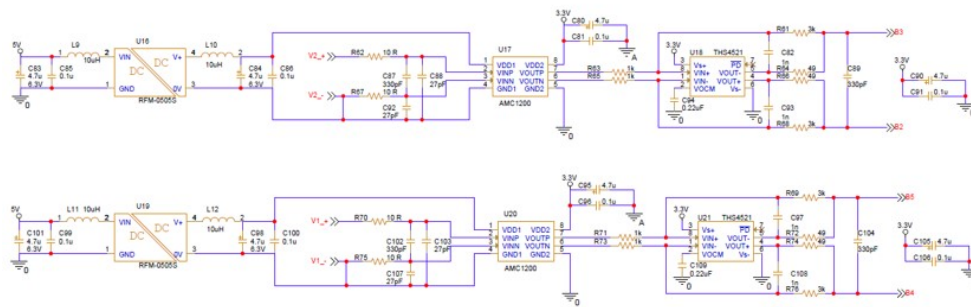


FIGURA 18: Sensores de voltaje AC Fuente: [57]

3.3.2. Sensores de Corriente

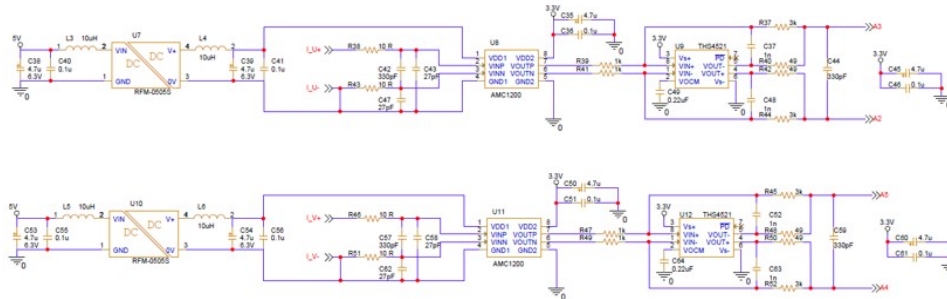


FIGURA 19: Sensores de Corriente AC Fuente: [57]

3.3.3. Comunicación Serial

Para evidenciar si el sistema funciona de manera adecuada se usa un programa conocido como Tera Term, el cual se encarga de permitir la comunicación serial con la DSP y de esta manera visualizar todas las configuraciones del dispositivo por medio de mensajes presentados desde la terminal, y a su vez observar si la planta se funciona correctamente de acuerdo a las señales de READY o FAULT dadas por los drivers que activan los IGBT. En la figura 20 se presenta un mensaje dado por Tera Term una vez que la planta es encendida.

Capítulo 4

Selección Técnica de Control

En este capítulo se abordan las técnicas presentadas en la sección 2.3, cada una con una breve descripción, características, ventajas, desventajas y algunas de las aplicaciones en las que se han utilizado. La correcta selección de la técnica de control es esencial para garantizar el funcionamiento de la planta disponible en la Universidad Santo Thomas. Así mismo, realizar la comparación de estas técnicas nos da paso para seleccionar la mejor opción, evaluar la complejidad y el costo de la implementación de cada una de ellas.

4.1. Control por modulación escalar o PWM

Esta técnica lineal es utilizada en inversores y rectificadores, se basa en la comparación de una señal de referencia modular y una señal portadora de forma triangular o diente de sierra como se observa en la figura 21. Esta comparación logra regular la tensión o la corriente variando el ancho de pulso de la señal de conmutación [39] [41] [40]; el PWM trabaja ajustando el tiempo en el cual se aplica una señal de ya sea de tensión o de corriente a la carga en ciclos de conmutación. Este control ha sido esencial y una parte fundamental en la electrónica de potencia [38].

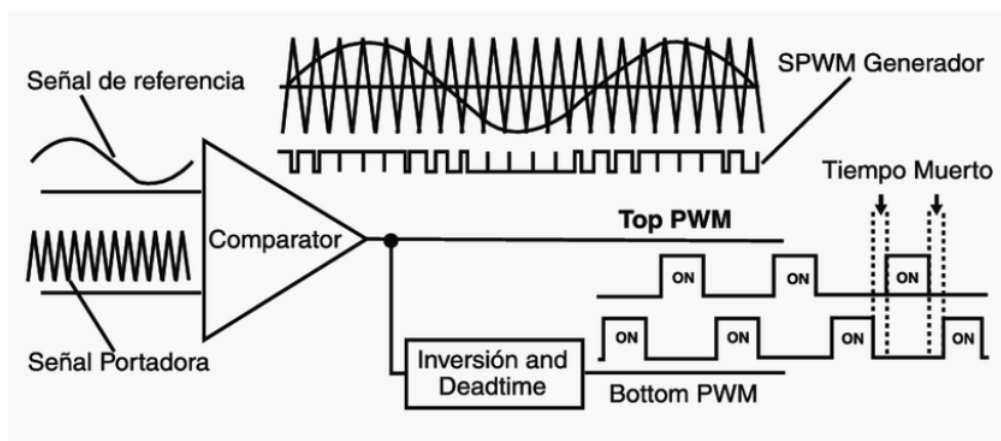


FIGURA 21: Circuito generador PWM Fuente: [38]

4.1.1. Características

Las características de este control se basan en los atributos que hacen que esta técnica sea ampliamente utilizada en en muchas aplicaciones. Se presentan algunas de estas.

- Tiene una gran capacidad para regular potencia de manera precisa y controlada, dado que al variar el PWM también varía la cantidad de potencia [40].
- Este control exhibe una respuesta específica gracias a su capacidad para variar rápidamente el ciclo de trabajo del PWM. Esta habilidad le permite ajustarse de manera eficiente y precisa, adaptándose a las demandas del sistema de forma ágil y efectiva.[40].
- Requiere conocimientos del modelo del sistema, para que la señal de modulación se adapte a las características del inversor [39].
- Contribuye a una mayor eficiencia energética, ya que la regulación de potencia es precisa sin necesidad de disipar calor de forma adicional [38].

4.1.2. Ventajas y Desventajas

4.1.2.1. Ventajas

- Este control permite una regulación precisa ya sea de tensión, corriente o frecuencia de salida del inversor [39].
- Permite la variación de frecuencia y salida de tensión,dado que puede operar en diferentes dispositivos eléctricos con diferentes requerimientos [41].
- Compatibilidad con control digital: Este control se puede implementar de forma análoga como digital [39].

4.1.2.2. Desventajas

- Necesidad de componentes de conmutación rápida: En sistemas de electrónica de potencia al operar con PWM se requieren dispositivos de conmutación rápida que pueden llegar a ser costosos y que se deban trabajar en situaciones térmicas altas [41].
- La implementación ya sea análoga o digital puede llegar a tener inconvenientes en sistemas que generen altas corrientes y tensiones, lo que requiere dispositivos que pueden llegar a ser costosos [41].
- Se generan armónicos de alta frecuencia, lo que puede requerir filtros adicionales [40].

El control PWM se utiliza comúnmente para regular la velocidad de motores eléctricos, como los motores de corriente continua (DC) y motores de inducción, en aplicaciones industriales y de automoción [38], así como también en la conversión de energía eléctrica, como en inversores utilizados en sistemas de energía solar y eólica, para generar corriente alterna a partir de corriente continua [39].

4.2. Control de realimentación Proporcional-Integral-Derivativo

Esta técnica de control con retroalimentación, utiliza la función de transferencia del sistema para describir el comportamiento de la salida del sistema con respecto a su referencia[44]. La finalidad de este control es hacer que el error de estado estacionario sea cero siendo la figura 22 el diagrama de bloques del sistema de control mas básico en lazo cerrado; como su nombre lo indica se conforma de las siguientes características:

4.2.1. Características

- Un control PID puede adaptarse a las características de la planta [42].
- Si solo se utilizan las ganancias Proporcional e Integral, el control funciona, pero su respuesta puede ser lenta [43] [44].
- Son muy versátiles y son ampliamente aplicables [35].
- Existen algoritmos de sintonización que encuentran cada una de las tres ganancias
- Son adaptables a la hora de aplicaciones que necesiten respuestas rápidas y precisas [43] [35].

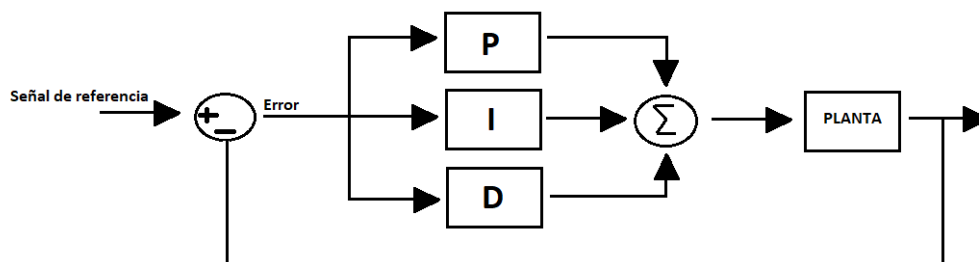


FIGURA 22: Diagrama de bloques PID Fuente: [42]

4.2.2. Ventajas y Desventajas

4.2.2.1. Ventajas

- Este control es simple de entender e implementar a comparación de otras técnicas de control avanzadas [44].
- Cuando se tienen las ganancias del controlador, este proporciona un control estable en condiciones de operación variable [44].
- La acción derivativa proporciona una respuesta rápida a cambios en la señal de error [44].

4.2.2.2. Desventajas

- Este control es mas adecuado en sistemas en los que se presentan no linealidades que pueden ser ignoradas, en sistemas no lineales no funciona de forma correcta [42].

- A cambios abruptos pueden requerir ajustes periódicos en las ganancias [43].
- La selección de las constantes puede ser un reto, dado que en sistemas mas complejos deben ser mas exactas y requerirá un ajuste [43].

El PID se utiliza en aplicaciones donde se deba controlar la tensión, la corriente y la frecuencia como lo son en convertidores de energía, como inversores, convertidores CC/CC y rectificadores [42]. También en sistemas de energía renovable, como la conversión de energía solar o eólica [35]. Además, en aplicaciones de electrónica de potencia como la regulación de la tensión en fuentes de alimentación conmutadas y la corrección del factor de potencia en equipos electrónicos[18].

4.3. Control Predictivo basado en Modelo MPC

Es una técnica avanzada de control y su estructura básica se observa en la figura 108; su característica principal radica en su capacidad para anticipar el comportamiento de un sistema y, en consecuencia, tomar decisiones en tiempo real basadas en estas predicciones[51]. El control predictivo es un conjunto de métodos que se ha desarrollado en base a los siguientes principios

- Modelo Dinámico del sistema: Se utiliza el modelo matemático del sistema puede ser lineal o no lineal que describa el comportamiento dinámico, este se utiliza para predecir cómo responde a diferentes entradas de control[49] [46].
- Su habilidad de predicción lo hace compensar intrínsecamente los tiempos muertos[48].
- Aparte del horizonte de predicción, este control utiliza un horizonte de control, este tiene la función de obtener la cantidad de muestras hacia el futuro que se implementarán con el fin de ajustar la señal de control en un corto tiempo [48] [51].
- Para que el MPC logre su objetivo este debe lograr minimizar su función de costo, esta función tiene términos que buscan minimizar el error que se obtenga entre la salida deseada y la salida real [46] [49].
- Si el sistema debe operar entre límites específicos este control puede manejar restricciones que ajustan la señal de control [46] [48].

Estos principios le permiten al MPC trabajar de forma dinámica las señales de control con el fin de cumplir su objetivo dentro de restricciones operativas.

4.3.1. Características

- Este control utiliza un modelo dinámico para predecir cambios en las variables que de salida del sistema ante cambios en las variables de entrada [46].
- Considera variables independientes que no son controladas como perturbaciones [46] [49] [51].
- Esta técnica de control es avanzada por lo que tiene la capacidad de manejar no linealidades que tenga un sistema [46].
- Tiene la capacidad de predecir y dar una respuesta ante perturbaciones [49] [48].

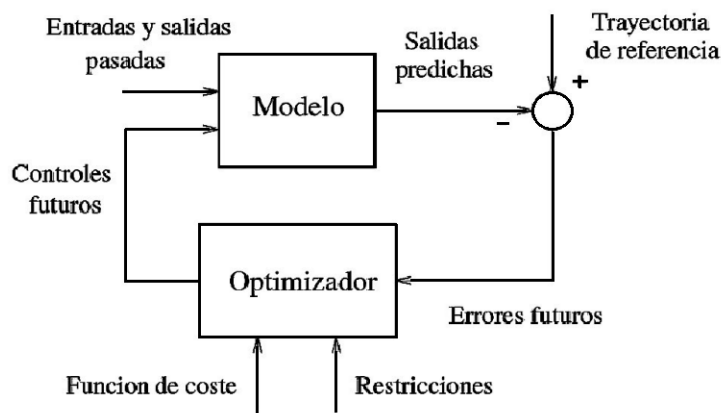


FIGURA 23: Estructura basica del MPC Fuente: [49]

4.3.2. Ventajas y Desventajas

4.3.2.1. Ventajas

- Maneja sistemas multivariables y no variables [46] [51].
- Este control es puede ser controlado en sistemas de fase no-mínima, es decir que tienen retardos de fase a altas frecuencias [46].
- Capacidad de trabajar con sistemas de múltiples entradas y salidas [49] [46].

4.3.2.2. Desventajas

- Requiere un modelo matemático muy preciso [48].
- El hardware debe ser especializado a la hora de implementar este control [48].
- Su implementación puede llegar a ser costosa en comparación con otros métodos de comparación [49].

El control MPC se aplica en la gestión de sistemas energéticos, como la generación y distribución de energía eléctrica, para optimizar la producción y el consumo de energía [51]. Además es ideal para sistemas complejos y altamente dinámicos.

4.4. Control sliding

El control deslizante o control sliding es una estrategia de control utilizada en sistemas dinámicos para mantener una variable de interés dentro de un límite o superficie de deslizamiento deseada es decir, el control deslizante se basa en la idea de llevar el sistema hacia un estado deseado (el "deslizante") en un espacio de estados específico [60]. Esto se logra mediante una ley de control que lleva al sistema hacia la superficie de deslizamiento donde se estabiliza [61]. Además, a menudo no requiere un conocimiento detallado del modelo del sistema, lo que facilita su implementación en situaciones donde la identificación del modelo puede ser difícil o costosa. Algunas de sus características son:

- Independencia del modelo, ya que no depende del modelo matemático del sistema [62].

- Es capaz de seguir una referencia deseada [63].
- Es una estrategia que ajusta la acción de control en tiempo real[61] [60].
- Tiene una respuestas rápida ante cambios de referencia ya que logra tener una gran capacidad para corrección rápida de errores [62] [61].

4.4.1. Ventajas y Desventajas

4.4.1.1. Ventajas

- El concepto de este control es muy fácil de comprender y tiene una facilidad de implementación [60] [62].
- Fácil adaptabilidad ante las necesidades específicas del sistema [63].
- Tiene un seguimiento preciso de la referencia y una rápida respuesta a perturbaciones [61] [60].

4.4.1.2. Desventajas

- Este control tiene sensibilidad al ruido en las señales de medición [60].
- En algunos casos se presenta el fenómeno conocido como "chattering", que es la presencia de oscilaciones en la señal cerca a la superficie de deslizamiento [63].
- El rendimiento del control depende la selección correcta de la superficie de deslizamiento [61].

El control sliding se ha utilizado en una variedad de campos, desde el control de robots y sistemas de navegación hasta aplicaciones en ingeniería eléctrica y electrónica de potencia [61] [62]. Su capacidad para proporcionar un seguimiento preciso y una rápida corrección de errores lo convierte en una estrategia de control atractiva en situaciones donde la estabilidad y el rendimiento son críticos [60] [63].

A demás de las características, ventajas y desventajas de cada control se debe tener en cuenta las prestaciones dinámicas que se tienen en cuanto al tiempo de establecimiento, relación de amortiguamiento y un adecuado nivel de rechazo a perturbaciones que se han obtenido en su implementación en otros inversores monofásicos con el fin de conocer la viabilidad de su uso en la planta disponible. En la tabla 1 se compara cada una de estas prestaciones dinámicas con base

Técnica de Control	Tiempo de Establecimiento	Relación de Amortiguamiento.	Rechazo a Perturbaciones
PWM	El tiempo que tarda el sistema para llegar a su estado estacionario es determinado por la frecuencia de la señal de modulación [39] [41].	Esta relación se ve influenciada en la forma en la que se ajusta el control, la frecuencia de conmutación y las características físicas de la planta [39] [40].	Se debe ajustar el ciclo de trabajo de la señal PWM en respuesta a las perturbaciones para mantener la salida del sistema [41] [38].

Técnica de Control	Tiempo de Establecimiento	Relación de Amortiguamiento.	Rechazo a Perturbaciones
PID	El tiempo de respuesta depende de las tres ganancias implementadas pero la ganancia que tiene mas influencia es la ganancia proporcional esta tiene la capacidad de aumentar la velocidad de respuesta, no obstante si esta llega a ser demasiado alta, el sistema se puede volver inestable [35] [42].	La acción integral ajusta la señal de control con el fin de eliminar cualquier error cuando se presentan perturbaciones pero un aumento de esta ganancia puede aumentar la respuesta transitoria o el overshoot [43] [44].	Este control monitorea continuamente la salida del sistema obteniendo una señal de error entre esta y la referencia, cuando se tuvo una perturbación el control busca volver a tener el sistema en estado estacionario [35] [45].
MPC	El tiempo de establecimiento está vinculado al horizonte de control, que se refiere al período de tiempo en el futuro durante el cual se establece el comportamiento del sistema [48]. Un horizonte de control prolongado puede influir en el tiempo que se necesita para alcanzar el estado estacionario del sistema [49].	La relación depende de la capacidad del MPC para predecir la salida del sistema y las características físicas del sistema puesto que este control necesita un modelo preciso del mismo [50] [47] [51].	El horizonte de predicción define cuantos pasos en el futuro se va a predecir el comportamiento del sistema, entre mas largo permite mejor respuesta ante perturbaciones [49] [47] [46].
Sliding	Depende de varios factores, incluyendo la naturaleza del sistema, la elección de la superficie de deslizamiento y la sintonización de los parámetros de control [61] [62] .	La superficie de deslizamiento actúa como una "barrera." "límite" que el sistema trata de seguir. Cuando la variable controlada se desvía de esta superficie, se aplica una acción de control para forzar al sistema a regresar a la superficie de deslizamiento [63] [60] .	El control deslizante conocido por su capacidad de rechazo de perturbaciones efectivo, puesto que gracias a la superficie de deslizamiento se tiene una acción de control rápida ante perturbaciones [63] [62].

TABLA 1: Comparaciones técnicas de control

Capítulo 5

Modelo Matemático

5.1. Analisis del sistema

Para modelar un sistema, se debe comprender cuál es su comportamiento y que estados va tener a lo largo del tiempo, por lo que inicialmente se debe comprender como actúan las corrientes y voltajes del convertidor con respecto a la conmutación de los transistores. De acuerdo a esto, para el análisis matemático se consideran dos tiempos, los cuales representan el estado de corte (circuito abierto) y saturación (corto circuito) del transistor. Por lo tanto, en la figura 24 se presenta el convertidor que se desea trabajar.

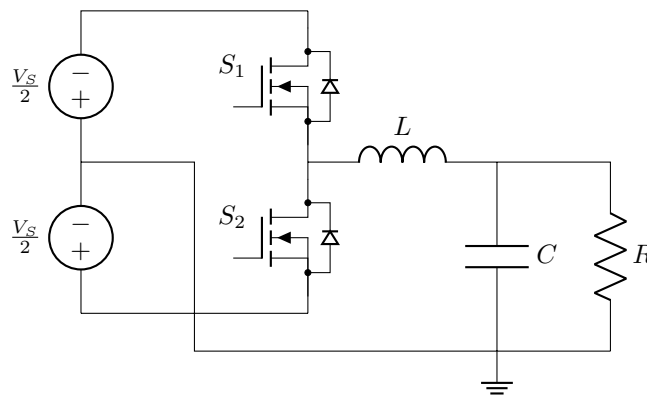
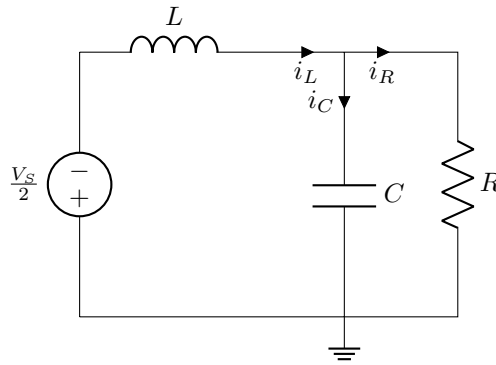


FIGURA 24: Inversor monofasico medio puente **Fuente: Autor**

Dado que el convertidor cuenta con dos transistores estos actuaran de manera complementaria, esto quiere decir que en el tiempo uno se considera que "S1" está en corte y "S2" en saturación. Por lo tanto, al tener estas consideraciones el circuito queda de la siguiente manera:

FIGURA 25: Inversor tiempo uno **Fuente: Autor**

Al tomar el circuito presentado en la figura 25 y al analizar todas las corrientes que entran y salen del nodo con ayuda de la ley de kirchhoff de corriente se tiene:

$$i_L(t) = i_C(t) + i_R(t) \quad (5.1)$$

$$i_L(t) = C \frac{dv_o(t)}{dt} + \frac{v_o(t)}{R} \quad (5.2)$$

$$\frac{dv_o(t)}{dt} = \frac{i_L(t)}{C} - \frac{v_o(t)}{R * C} \quad (5.3)$$

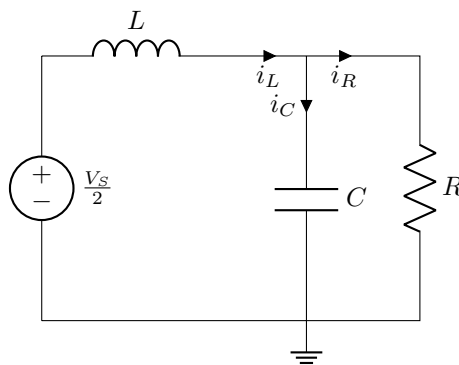
Y al realizar analisis de mallas con la ley de kirchhoff de corriente se obtiene:

$$\frac{v_s}{2} = v_L + v_o \quad (5.4)$$

$$\frac{v_s}{2} = L \frac{di_L}{dt} + v_o \quad (5.5)$$

$$\frac{di_L}{dt} = \frac{v_s}{2 * L} - \frac{v_o}{L} \quad (5.6)$$

Para el tiempo dos se considera el caso contrario donde "S1" está en saturación y "S2" en corte. Por lo tanto, el circuito con estas consideraciones es:

FIGURA 26: Inversor tiempo uno **Fuente: Autor**

Al usar la ley de kirchhoff de corrientes:

$$i_L(t) = i_c(t) + i_R(t) \quad (5.7)$$

$$i_L(t) = C \frac{dv_o(t)}{dt} + \frac{v_o(t)}{R} \quad (5.8)$$

$$\frac{dv_o(t)}{dt} = \frac{i_L(t)}{C} - \frac{v_o(t)}{R * C} \quad (5.9)$$

Y al realizar la ley de kirchhoff de voltajes se obtiene:

$$\frac{v_s}{2} + v_L + v_o = 0 \quad (5.10)$$

$$v_L = -\frac{v_s}{2} - v_o \quad (5.11)$$

$$L \frac{di_L}{dt} = -\frac{v_s}{2} - v_o \quad (5.12)$$

$$\frac{di_L}{dt} = -\frac{v_s}{2 * L} - \frac{v_o}{L} \quad (5.13)$$

5.2. Modelo conmutado

El modelo conmutado es la representación matemática del funcionamiento del inversor monofásico, donde se consideran los tiempos de trabajo de cada uno de los transistores, que siempre deben ser complementarios. Por lo tanto, este modelo es la relación entre el comportamiento del inversor en el tiempo uno que tiene un ciclo útil "u" y el tiempo dos "1-u".

$$\frac{dv_o}{dt} = \left[\frac{i_L}{C} - \frac{v_o}{R * C} \right] u + \left[\frac{i_L}{C} - \frac{v_o}{R * C} \right] (1 - u) \quad (5.14)$$

$$\frac{di_L}{dt} = \left[\frac{v_s}{2 * L} - \frac{v_o}{L} \right] u + \left[-\frac{v_s}{2 * L} - \frac{v_o}{L} \right] (1 - u) \quad (5.15)$$

5.3. Modelo promediado

Este modelo en la representación del promedio del modelo conmutado, donde se eliminan las discontinuidades producidas por el tiempo uno y dos. Por lo tanto, tomando las ecuaciones 5.14 y 5.15 se tiene:

$$\frac{d}{dt} \bar{i}_L = \frac{V_s}{L} \left(\bar{d} - \frac{1}{2} \right) - \frac{\bar{V}_c}{L} \quad (5.16)$$

Para simplificar la expresión se considera que:

$$\bar{D} = \bar{d} - \frac{1}{2} \quad (5.17)$$

Por lo tanto:

$$\frac{d}{dt}\bar{i}_L = \frac{V_s}{L}\bar{D} - \frac{\bar{V}_c}{L} \quad (5.18)$$

$$\frac{d}{dt}\bar{V}_c = \frac{\bar{i}_L}{C} - \frac{\bar{V}_c}{RC} \quad (5.19)$$

Capítulo 6

Control en Modo Sliding

Basándose en la revisión bibliográfica detallada en la sección 4, y tras considerar las características de las cuatro estrategias de control analizadas, se opta por la implementación del control deslizante. Esta elección se fundamenta en la destacada capacidad del control deslizante para adaptarse eficazmente tanto a sistemas lineales como no lineales, incluso cuando estos enfrentan perturbaciones significativas. En comparación con las otras tres estrategias de control evaluadas, el control deslizante se destaca por no requerir información detallada sobre el modelo del sistema, lo que lo convierte en una estrategia versátil con aplicaciones en la electrónica de potencia en una amplia gama de convertidores.

6.1. Controlador SMC

El control de modo deslizante se caracteriza por obligar que la respuesta de un sistema se establezca sobre la superficie de deslizamiento seleccionada, por ello el compensador está basado en histéresis, puesto que de esta manera se ajustan dos niveles de referencia que permiten guiar la dinámica del sistema hacia el punto deseado. En este caso se consideró que la superficie de deslizamiento es el error producido por la realimentación del sistema, esto quiere decir que los niveles de referencia dados por un límite superior e inferior no permiten que el error supere dicho umbral y se encuentre cercano a cero, permitiendo que la señal a controlar siga la referencia deseada. De acuerdo a esto en la figura 27 se presenta la topología del controlador de modo deslizante que es usado.

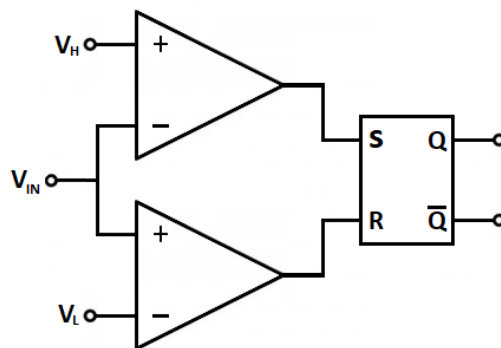


FIGURA 27: Compensador SMC Fuente: Autor

6.1.1. Comparador de ventana

Como se presenta en la figura 27 el controlador cuenta con un comparador de ventana, puesto que este es el encargado de determinar si el error del sistema en lazo cerrado se encuentra dentro del rango definido por los dos niveles de referencia. De acuerdo a esto, la topología cuenta con un comparador para el límite superior y otro para el inferior, donde cada uno toma una decisión binaria si el error del sistema supera cualquiera de los dos niveles. Esto quiere decir que se tendrá un “1” cuando la señal de entrada supere la referencia de cualquiera de los comparadores y un “0” en el caso contrario. Por lo tanto, en la figura 28 se presenta el comportamiento del comparador de ventana, Donde la imagen (a) representa la salida del comparador superior y la (b) la del inferior.

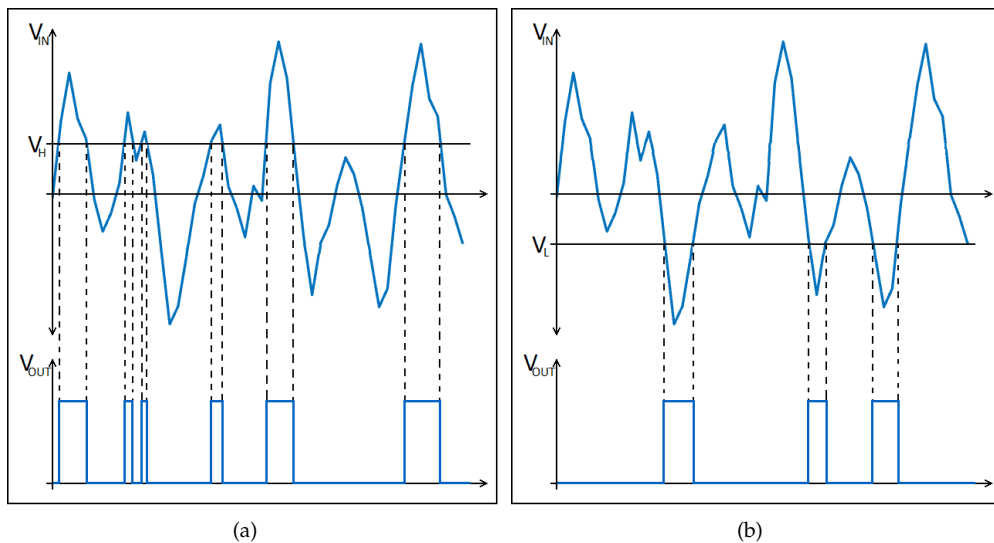


FIGURA 28: Comparador de ventana **Fuente: Autor**

6.1.2. flip flop set-reset

Dado que el comparador de ventana presenta conmutaciones rápidas cuando hay fluctuaciones sobre cualquiera de los niveles de referencia, se tiene como solución el uso de un flip flop set-reset como se puede observar en la figura 27. Puesto que de esta manera el comparador contará con una banda de histéresis, la cual permite que su salida únicamente cambie de estado cuando la entrada supere cualquiera de los límites. Por lo tanto, la salida del comparador superior se encarga de setear el flip flop y de esta manera mantener un uno lógico hasta que el límite del comparador inferior sea superado, puesto que este resetea la compuerta. De acuerdo a esto, en la figura 29 se presenta el comportamiento del comparador de ventana con histéresis, donde la salida Q es la modulación PWM encargada de controlar la dinámica del sistema.

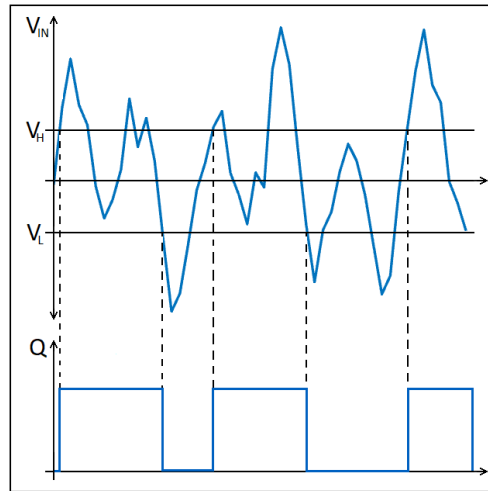


FIGURA 29: Salida comparador de ventana con histéresis **Fuente: Autor**

6.1.3. Control implementado

En este caso se realiza un control de corriente, puesto que el inversor al ser on-grid ya cuenta con un voltaje fijo en su salida. Por otra parte, la referencia se obtiene de la carga con la que cuenta el circuito, por lo que se decide controlar la corriente que fluye a través de la bobina. Por ende, cuando la corriente de la bobina sea igual a corriente que consume la carga, la corriente que es inyectada a la red es igual o cercana a cero, esto quiere decir que el control permite que el inversor supla la potencia que solicita la carga sin inyectar nada a la red eléctrica y sin consumir nada de la misma. En el figura 30 se presenta un esquemático del inversor con el controlador.

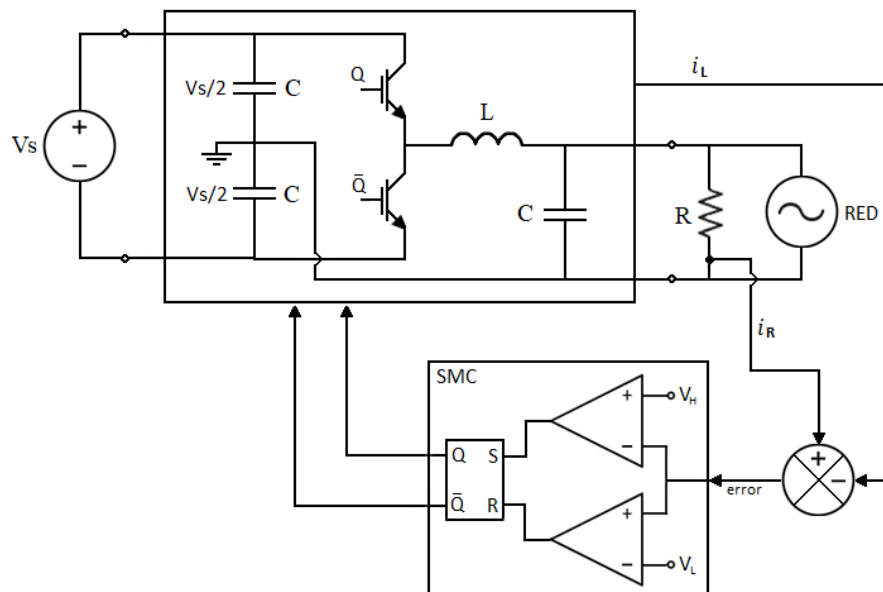


FIGURA 30: Esquema del controlador propuesto. **Fuente: Adaptado de [64]**

6.2. Diseño

6.3. Diagrama de Flujo

A partir de la comprensión del concepto y el funcionamiento del control sliding, se procede a realizar los diagramas de flujo que describen cómo se llevará a cabo la implementación del control deslizante digital. Estos diagramas ofrecen una representación gráfica del control deslizante; uno de ellos se refiere a la interrupción del ADC, como se detalla en la sección 6.3.1, mientras que el otro está relacionado con la generación de una interrupción por Timer, tal como se describe en la sección 6.3.2

6.3.1. Diagrama de flujo interrupción del ADC

En la figura 31 se tiene el diagrama de flujo donde se inicia el proceso adquiriendo la señal de error (ADC_VAL). Luego, esta señal se somete a una estructura condicional donde se compara con el límite superior establecido (REF+) si se obtiene una respuesta de no se continúa a comparar con el límite inferior (REF-). Si la respuesta es si en la comparación con el límite superior, se procede a verificar si la variable A tiene un valor igual a 1. En caso de que si, indica que el estado no ha cambiado, y no se ejecuta ninguna acción adicional. En contraste, si la respuesta es no, señala un cambio en el estado, lo que conduce a la configuración de la variable A en 1. A continuación, se reinicia la interrupción del Timer y se habilita para generar la banda muerta, tal como se explica en la sección 6.3.2. Además, las salidas denominadas Q y QNEG se ponen en cero. Este mismo proceso se replica en la condición correspondiente al límite inferior, aunque en este caso se utiliza la variable B."

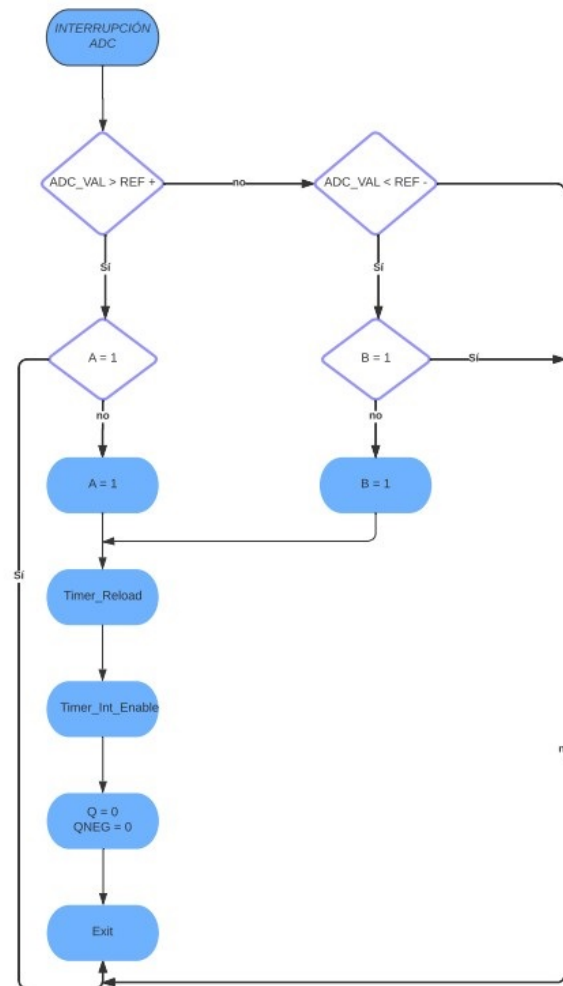


FIGURA 31: Diagrama de flujo interrupción ADC. **Fuente: Autor**

Teniendo en cuenta el diagrama de flujo de la interrupción del ADC y la información obtenida del procesador de señales TMS320F28379D de Texas Instruments en la sección 9.2.2. Se conoce que la DSP puede utilizar el modo de adquisición analógica-digital diferencial a 16 bits, que involucra el uso de dos pines para esta adquisición. Internamente, realiza la resta entre los valores analógicos de la corriente de referencia y la corriente de la carga. Una vez obtenida la resta, que se convierte en valores digitales dentro del rango de 0 a 65535 mediante la interrupción del ADC, se inicia la descripción del control deslizante a través de una serie de condicionales que se observan a continuación.

```

CurrentMeaLoad = (float32)(65535-(ADC_readResult(ADCCRESULT_BASE, ADC_SOC_NUMBER4)));
limiteSuperior = Offset + 15;
limiteInferior = Offset - 15;

if(CurrentMeaLoad > limiteSuperior)
{
    if(A == 1)
    {
    }
    else
    {
        A = 1;
        B = 0;
        GPIO_WritePin(160, 1);
        Interrupt_enable(INT_TIMER0);
        CPUTimer_reloadTimerCounter(CPUTIMER0_BASE);
        CPUTimer_startTimer(CPUTIMER0_BASE);
        GPIO_WritePin(0, 0);
        GPIO_WritePin(1, 0);
    }
}
else
{
    if(CurrentMeaLoad < limiteInferior)
    {
        if(B == 1)
        {
        }
        else
        {
            B = 1;
            A = 0;
            GPIO_WritePin(160, 1);
            Interrupt_enable(INT_TIMER0);
            CPUTimer_reloadTimerCounter(CPUTIMER0_BASE);
            CPUTimer_startTimer(CPUTIMER0_BASE);
            GPIO_WritePin(0, 0);
            GPIO_WritePin(1, 0);
        }
    }
}

```

Una vez adquirida la señal digital, se establece un valor de offset de 32,580, que representa el centro de la señal. Luego, durante la interrupción, se suma 15 al límite superior y se resta 15 al límite inferior de este valor. Este procedimiento se realiza con el propósito de limitar la señal de error dentro de ese rango específico que define nuestra superficie de deslizamiento.

Cabe aclarar que se emplean dos variables cruciales (A y B) cuyos valores, cuando se igualan a uno, adquieren un papel crucial en la sección 6.3.2

6.3.2. Diagrama de flujo interrupción del Timer

En la figura 32 se observa el diagrama de flujo para generar banda muerta puesto que cuando se realiza la transición de un transistor desde el estado apagado (no conductor) al estado encendido (conductor) o viceversa, puede ocurrir brevemente una situación en la que ambos transistores, el que se apaga y el que se enciende, estén parcialmente activos al mismo tiempo. Para evitar esta problemática, se implementa una interrupción de temporizador que activa o desactiva el transistor de acuerdo si hubo cambios previos en las señales de conmutación, es decir, si cualquiera de las variables A o B alcanza el valor de 1, se reconoce como una señal de cambio. En este punto, durante la interrupción del ADC, se envía un valor de cero a los dos pines encargados de controlar los transistores, lo que resulta en su apagado. Simultáneamente, se activa la interrupción del temporizador que aguarda un período de $2\ \mu s$, creando de esta manera la banda muerta antes de enviar una señal de salida en alto dependiendo de cada variable si A tiene un valor de 1 se envía un alto en la salida Q y si se tiene un valor de 1 en la variable B se envía un alto en la salida QNEG.

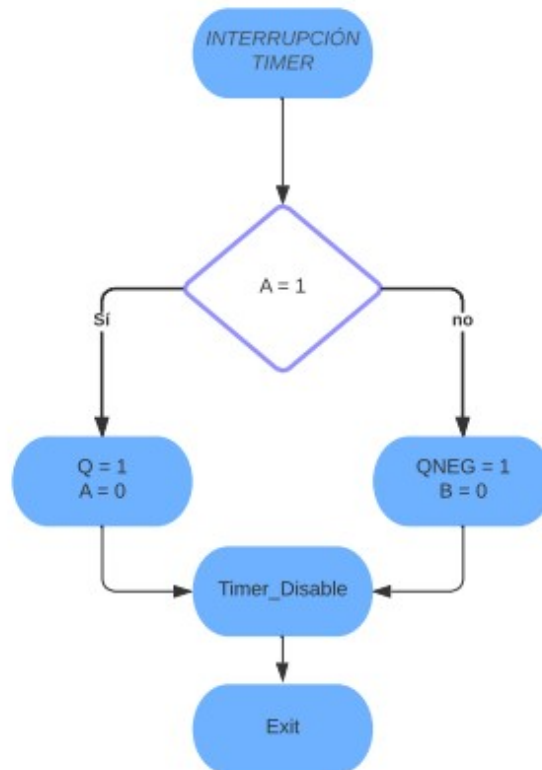


FIGURA 32: Diagrama de flujo interrupción Timer. **Fuente: Autor**

Basándose en el diagrama de flujo previo, se implementa el siguiente código que se encarga de generar la banda muerta en la DSP.

```
__interrupt void cpuTimer0ISR(void)
{
    cpuTimer0IntCount++;
    if(A == 1)
    {
        GPIO_WritePin(0, 0);
        GPIO_WritePin(1, 1);
    }
    else
    {
        GPIO_WritePin(0, 1);
        GPIO_WritePin(1, 0);
    }
    Interrupt_disable(INT_TIMER0);
    CPUTimer_stopTimer(CPUTIMER0_BASE);
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP1);
}
```

Al final de la interrupción esta misma se deshabilita y se detiene el conteo para que halla un reinicio cuando se necesite en la interrupción del ADC.

Capítulo 7

Control de realimentación PI

7.1. Transformación DQ

Una vez que se tiene el modelo promediado del inversor monofásico en la sección 5, se realiza la transformación al marco giratorio dq . Para ello, se tiene presente el circuito imaginario que representa el desfase de 90 grados con respecto al sistema original. De acuerdo a esto, en las ecuaciones 7.1 y 7.2 se presenta el modelo promediado del circuito real e imaginario.

$$\frac{d}{dt} \begin{bmatrix} \bar{i}_{L_R} \\ \bar{i}_{L_I} \end{bmatrix} = \frac{V_s}{L} \begin{bmatrix} \bar{D}_R \\ \bar{D}_I \end{bmatrix} - \frac{1}{L} \begin{bmatrix} \bar{V}_{c_R} \\ \bar{V}_{c_I} \end{bmatrix} \quad (7.1)$$

$$\frac{d}{dt} \begin{bmatrix} \bar{V}_{c_R} \\ \bar{V}_{c_I} \end{bmatrix} = \frac{1}{C} \begin{bmatrix} \bar{i}_{L_R} \\ \bar{i}_{L_I} \end{bmatrix} - \frac{1}{RC} \begin{bmatrix} \bar{V}_{c_R} \\ \bar{V}_{c_I} \end{bmatrix} \quad (7.2)$$

Dado que en este caso ya se tienen dos fases, las variables relacionadas al sistema real se consideran como alpha (α) y las del sistema imaginario como beta (β), las cuales permiten realizar la transformada de Park. Por lo tanto las ecuaciones 7.1 y 7.2 pueden ser representadas como:

$$\frac{d}{dt} \begin{bmatrix} \bar{i}_{\alpha} \\ \bar{i}_{\beta} \end{bmatrix} = \frac{V_s}{L} \begin{bmatrix} \bar{D}_{\alpha} \\ \bar{D}_{\beta} \end{bmatrix} - \frac{1}{L} \begin{bmatrix} \bar{V}_{\alpha} \\ \bar{V}_{\beta} \end{bmatrix} \quad (7.3)$$

$$\frac{d}{dt} \begin{bmatrix} \bar{V}_{\alpha} \\ \bar{V}_{\beta} \end{bmatrix} = \frac{1}{C} \begin{bmatrix} \bar{i}_{\alpha} \\ \bar{i}_{\beta} \end{bmatrix} - \frac{1}{RC} \begin{bmatrix} \bar{V}_{\alpha} \\ \bar{V}_{\beta} \end{bmatrix} \quad (7.4)$$

Y su representación de manera vectorial es:

$$\frac{d}{dt} \vec{i}_{\alpha\beta} = \frac{V_s}{L} \vec{D}_{\alpha\beta} - \frac{1}{L} \vec{V}_{\alpha\beta} \quad (7.5)$$

$$\frac{d}{dt} \vec{V}_{\alpha\beta} = \frac{1}{C} \vec{i}_{\alpha\beta} - \frac{1}{RC} \vec{V}_{\alpha\beta} \quad (7.6)$$

Sin embargo, para realizar la transformada de Park, en este caso no se usa la matriz convencional presentada en la ecuación 2.15, puesto que esta matriz se usa cuando $\alpha(\alpha)$ y $\beta(\beta)$ tienen una forma de onda como se presenta en la imagen (a) de la figura 33, pero como en este caso el circuito real tiene como salida una senoidal y su desfase de 90 grados da como resultado una cosenoidal negativa, $\alpha(\alpha)$ y $\beta(\beta)$ se encontraran representadas por la imagen (b) de la figura 33. Por lo tanto, para realizar la transformación se usa la expresión presentada en la ecuación 7.7 y de esta manera obtener las variables en las coordenadas dq .

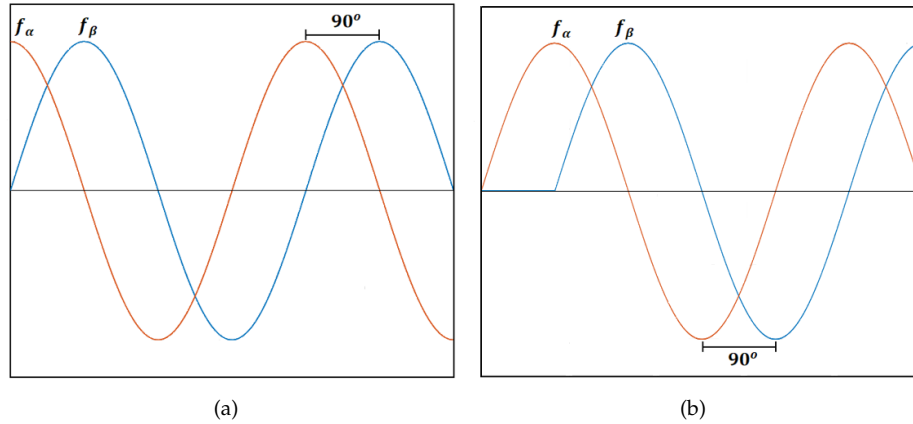


FIGURA 33: Formas de onda $\alpha\beta$ Fuente: Autor

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = \begin{bmatrix} \sin(\omega * t) & -\cos(\omega * t) \\ \cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} \quad (7.7)$$

De acuerdo a esto, para realizar la transformada de Park se deben dejar las ecuaciones 7.5 y 7.6 en términos de d y q , para ello se toma como base la representación vectorial de la ecuación 7.7 como se presenta a continuación:

$$X_{dq} = T * X_{\alpha\beta} \quad (7.8)$$

Donde T es la matriz de Park.

$$T = \begin{bmatrix} \sin(\omega * t) & -\cos(\omega * t) \\ \cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \quad (7.9)$$

Al despejar el vector $X_{\alpha\beta}$ de la ecuación 7.8 se tiene:

$$X_{\alpha\beta} = T^{-1} * X_{dq} \quad (7.10)$$

Donde T^{-1} es la inversa de la matriz de Park.

$$T^{-1} = \begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \quad (7.11)$$

Por lo tanto, al tomar las ecuaciones 7.5 y 7.6 y sustituir todos los vectores relacionados con α (α) y β (β) por la ecuación 7.10 se obtiene.

$$\frac{d}{dt} \left(T^{-1} \vec{i}_{dq} \right) = \frac{V_s}{L} \left(T^{-1} \vec{D}_{dq} \right) - \frac{1}{L} \left(T^{-1} \vec{V}_{dq} \right) \quad (7.12)$$

$$\frac{d}{dt} \left(T^{-1} \vec{V}_{dq} \right) = \frac{1}{C} \left(T^{-1} \vec{i}_{dq} \right) - \frac{1}{RC} \left(T^{-1} \vec{V}_{dq} \right) \quad (7.13)$$

Sin embargo, se puede observar que en las dos ecuaciones se presenta la derivada del producto de dos funciones, que en este caso son la matriz T^{-1} y el vector $X_{\alpha\beta}$, por lo que es necesario usar la expresión de la ecuación 7.1 para resolver dichas derivadas, como se presenta a continuación:

$$\frac{d}{dt} \left(T^{-1} \vec{i}_{dq} \right) = T^{-1} \frac{d}{dt} \left(\vec{i}_{dq} \right) + \vec{i}_{dq} \frac{d}{dt} (T^{-1}) \quad (7.14)$$

$$\frac{d}{dt} \left(T^{-1} \vec{V}_{dq} \right) = T^{-1} \frac{d}{dt} \left(\vec{V}_{dq} \right) + \vec{V}_{dq} \frac{d}{dt} (T^{-1}) \quad (7.15)$$

Una vez que se han resuelto las derivadas estas se igualan con las ecuaciones 7.12 y 7.13 con el fin de obtener:

$$T^{-1} \frac{d}{dt} \left(\vec{i}_{dq} \right) + \vec{i}_{dq} \frac{d}{dt} (T^{-1}) = \frac{V_s}{L} \left(T^{-1} \vec{D}_{dq} \right) - \frac{1}{L} \left(T^{-1} \vec{V}_{dq} \right) \quad (7.16)$$

$$T^{-1} \frac{d}{dt} \left(\vec{V}_{dq} \right) + \vec{V}_{dq} \frac{d}{dt} (T^{-1}) = \frac{1}{C} \left(T^{-1} \vec{i}_{dq} \right) - \frac{1}{RC} \left(T^{-1} \vec{V}_{dq} \right) \quad (7.17)$$

Y al realizar la matemática correspondiente se tiene:

$$\frac{d}{dt} \left(\vec{i}_{dq} \right) + \vec{i}_{dq} * T \frac{d}{dt} (T^{-1}) = \frac{1}{L} \left(V_s \vec{D}_{dq} - \vec{V}_{dq} \right) \quad (7.18)$$

$$\frac{d}{dt} \left(\vec{V}_{dq} \right) + \vec{V}_{dq} * T \frac{d}{dt} (T^{-1}) = \frac{1}{C} \left(\vec{i}_{dq} - \frac{\vec{V}_{dq}}{R} \right) \quad (7.19)$$

donde:

$$T \frac{d}{dt} (T^{-1}) = \begin{bmatrix} \sin(\omega * t) & -\cos(\omega * t) \\ \cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \frac{d}{dt} \left(\begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \right) \quad (7.20)$$

Y se tiene como resultado:

$$T \frac{d}{dt} (T^{-1}) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \quad (7.21)$$

Por lo tanto, al reemplazar la expresión de la ecuación 7.21 se obtienen las siguientes expresiones:

$$\frac{d}{dt} \left(\vec{i}_{dq} \right) = \frac{1}{L} \left(V_s \vec{D}_{dq} - \vec{V}_{dq} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{i}_{dq} \quad (7.22)$$

$$\frac{d}{dt}(\vec{V}_{dq}) = \frac{1}{C} \left(\vec{i}_{dq} - \frac{\vec{V}_{dq}}{R} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{V}_{dq} \quad (7.23)$$

Que también pueden ser representadas de manera matricial como se presenta a continuación:

$$\frac{d}{dt} \begin{bmatrix} i_d \\ i_q \end{bmatrix} = \frac{1}{L} \left(V_s \begin{bmatrix} D_d \\ D_q \end{bmatrix} - \begin{bmatrix} V_d \\ V_q \end{bmatrix} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \quad (7.24)$$

$$\frac{d}{dt} \begin{bmatrix} V_d \\ V_q \end{bmatrix} = \frac{1}{C} \left(\begin{bmatrix} i_d \\ i_q \end{bmatrix} - \frac{1}{R} \begin{bmatrix} V_d \\ V_q \end{bmatrix} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix} \quad (7.25)$$

Finalmente, con base a las matrices de las ecuaciones 7.24 y 7.25, se obtienen las ecuaciones de estado de las cuatro variables (v_d, i_d, v_q, i_q) y a su vez se reemplaza D_x por la expresión de la ecuación 5.17. A continuación, se presentan las cuatro expresiones.

$$\frac{d}{dt} i_d = \frac{1}{L} \left(V_s \left(d_d - \frac{1}{2} \right) - V_d \right) + \omega i_q \quad (7.26)$$

$$\frac{d}{dt} i_q = \frac{1}{L} \left(V_s \left(d_q - \frac{1}{2} \right) - V_q \right) - \omega i_d \quad (7.27)$$

$$\frac{d}{dt} V_d = \frac{1}{C} \left(i_d - \frac{1}{R} V_d \right) + \omega V_q \quad (7.28)$$

$$\frac{d}{dt} V_q = \frac{1}{C} \left(i_q - \frac{1}{R} V_q \right) - \omega V_d \quad (7.29)$$

7.2. Inversa transformada DQ

Para obtener nuevamente las variables de estado en términos de alpha (α) y beta (β), se realiza la transformada inversa de park tomando como base la expresión presentada en la ecuación 7.8. De acuerdo a esto, se toman las ecuaciones 7.22 y 7.23 reemplazando los vectores que estén en términos de d y q por la ecuación 7.8, a excepción del vector $D_{\alpha\beta}$ y $V_{\alpha\beta}$ de la ecuación 7.22, puesto que de esta manera el sistema tendría como entrada la modulación $D_{\alpha\beta}$ usada para la transformación de Park y los voltajes d y q . Por lo tanto, a continuación se presentan la expresiones que se tienen al remplazar los términos mencionados.

$$\frac{d}{dt} (T \vec{i}_{\alpha\beta}) = \frac{1}{L} (V_s \vec{D}_{dq} - \vec{V}_{dq}) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} T \vec{i}_{\alpha\beta} \quad (7.30)$$

$$\frac{d}{dt} (T \vec{V}_{\alpha\beta}) = \frac{1}{C} \left(\vec{i}_{dq} - \frac{T \vec{V}_{\alpha\beta}}{R} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} T \vec{V}_{\alpha\beta} \quad (7.31)$$

Como se puede observar, las dos ecuaciones cuentan con la derivada del producto de dos funciones, que en este caso son la matriz T y el vector $X_{\alpha\beta}$, por lo que es necesario aplicar la ecuación n y de esta manera obtener las expresiones:

$$\frac{d}{dt} (T \vec{i}_{\alpha\beta}) = T \frac{d}{dt} (\vec{i}_{\alpha\beta}) + \vec{i}_{\alpha\beta} \frac{d}{dt} (T) \quad (7.32)$$

$$\frac{d}{dt} (TV_{\alpha\beta}) = T \frac{d}{dt} (\vec{V}_{\alpha\beta}) + \vec{V}_{\alpha\beta} \frac{d}{dt} (T) \quad (7.33)$$

Una vez que se tiene la solución de las derivadas, se igualan las ecuaciones 7.30 y 7.31 con las dos expresiones anteriores y de esta manera obtener:

$$T \frac{d}{dt} (\vec{i}_{\alpha\beta}) + \vec{i}_{\alpha\beta} \frac{d}{dt} (T) = \frac{1}{L} (V_s D_{dq} - V_{dq}) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} T \vec{i}_{\alpha\beta} \quad (7.34)$$

$$T \frac{d}{dt} (\vec{V}_{\alpha\beta}) + \vec{V}_{\alpha\beta} \frac{d}{dt} (T) = \frac{1}{C} \left(i_{dq} - \frac{TV_{\alpha\beta}}{R} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} T \vec{V}_{\alpha\beta} \quad (7.35)$$

Al dividir las dos expresiones por T .

$$\frac{d}{dt} (\vec{i}_{\alpha\beta}) + \vec{i}_{\alpha\beta} T^{-1} \frac{d}{dt} (T) = \frac{1}{L} T^{-1} (V_s D_{dq} - V_{dq}) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{i}_{\alpha\beta} \quad (7.36)$$

$$\frac{d}{dt} (\vec{V}_{\alpha\beta}) + \vec{V}_{\alpha\beta} T^{-1} \frac{d}{dt} (T) = \frac{1}{C} \left(T^{-1} i_{dq} - \frac{V_{\alpha\beta}}{R} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{V}_{\alpha\beta} \quad (7.37)$$

Donde:

$$T^{-1} \frac{d}{dt} (T) = \begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \frac{d}{dt} \left(\begin{bmatrix} \sin(\omega * t) & -\cos(\omega * t) \\ \cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \right) \quad (7.38)$$

Y al realizar la matemática correspondiente se tiene:

$$T^{-1} \frac{d}{dt} (T) = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \quad (7.39)$$

Luego se toman las ecuaciones 7.36 y 7.37, se despejan las derivadas y se reemplaza en cada expresión el resultado presentado en la ecuación 7.39.

$$\frac{d}{dt} (\vec{i}_{\alpha\beta}) = \frac{1}{L} T^{-1} (V_s D_{dq} - V_{dq}) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{i}_{\alpha\beta} - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{i}_{\alpha\beta} \quad (7.40)$$

$$\frac{d}{dt} (\vec{V}_{\alpha\beta}) = \frac{1}{C} \left(T^{-1} i_{dq} - \frac{V_{\alpha\beta}}{R} \right) - \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \vec{V}_{\alpha\beta} - \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \vec{V}_{\alpha\beta} \quad (7.41)$$

Al restar las matrices en ambas ecuaciones se tiene:

$$\frac{d}{dt} (\vec{i}_{\alpha\beta}) = \frac{1}{L} (V_s T^{-1} D_{dq} - T^{-1} V_{dq}) \quad (7.42)$$

$$\frac{d}{dt} (\vec{V}_{\alpha\beta}) = \frac{1}{C} \left(T^{-1} i_{dq} - \frac{V_{\alpha\beta}}{R} \right) \quad (7.43)$$

Finalmente, se reemplazan todos los términos relacionados con la la matriz T^{-1} por la expresión de la ecuación 7.11, y de esta manera se tiene la transformada inversa de Park.

$$\frac{d}{dt} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{1}{L} \left(V_s \begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \begin{bmatrix} D_d \\ D_q \end{bmatrix} - \begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix} \right) \quad (7.44)$$

$$\frac{d}{dt} \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{1}{C} \left(\begin{bmatrix} \sin(\omega * t) & \cos(\omega * t) \\ -\cos(\omega * t) & \sin(\omega * t) \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} - \frac{1}{R} \begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} \right) \quad (7.45)$$

Que también pueden ser representadas de la siguiente manera:

$$\frac{d}{dt} i_\alpha = \frac{1}{L} \left(\sin(\omega * t) \left(V_s \left(d_d - \frac{1}{2} \right) - V_d \right) + \cos(\omega * t) \left(V_s \left(d_q - \frac{1}{2} \right) - V_q \right) \right) \quad (7.46)$$

$$\frac{d}{dt} i_\beta = \frac{1}{L} \left(\cos(\omega * t) \left(V_d - V_s \left(d_d - \frac{1}{2} \right) \right) + \sin(\omega * t) \left(V_s \left(d_q - \frac{1}{2} \right) - V_q \right) \right) \quad (7.47)$$

$$\frac{d}{dt} V_\alpha = \frac{1}{L} \left(\sin(\omega * t) i_d + \cos(\omega * t) i_q - \frac{1}{R} V_\alpha \right) \quad (7.48)$$

$$\frac{d}{dt} V_\beta = \frac{1}{L} \left(\sin(\omega * t) i_q - \cos(\omega * t) i_d - \frac{1}{R} V_\beta \right) \quad (7.49)$$

7.3. Punto de equilibrio del sistema en coordenadas dq

Para el análisis del sistema, se realiza su representación circuital tomando como base las variables de estado en coordenadas dq , las cuales se encuentra presentadas en las ecuaciones 7.26, 7.27, 7.28 y 7.29. Por lo tanto, para realizar el esquemático presentado en la figura 34 se usan las ecuaciones 7.26 y 7.28, puesto que corresponden a d y para la figura 35 las ecuaciones 7.27 y 7.29 que representan a q .

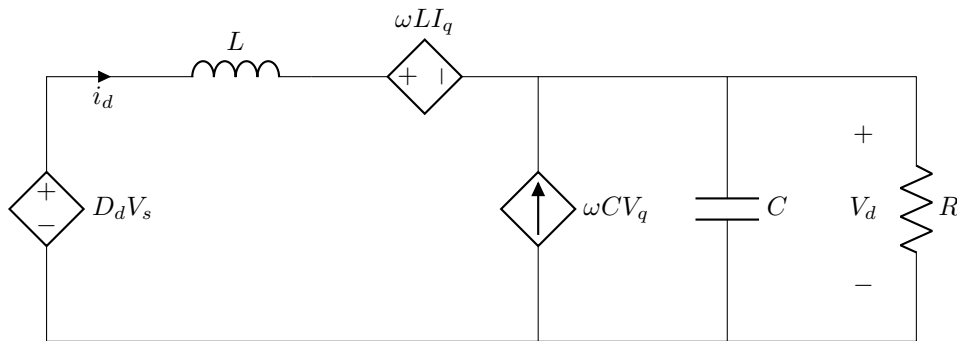
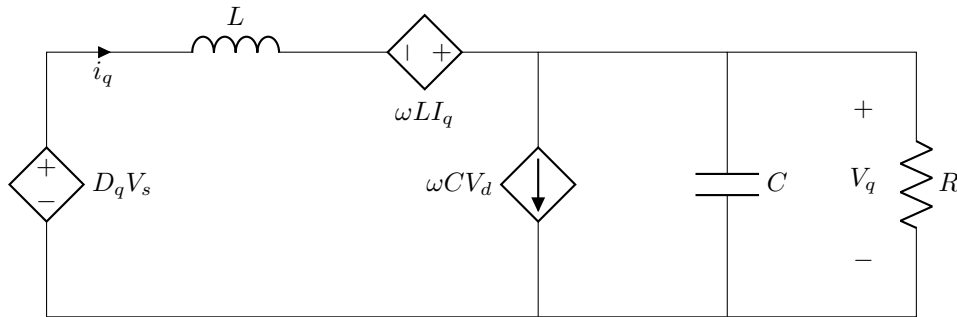
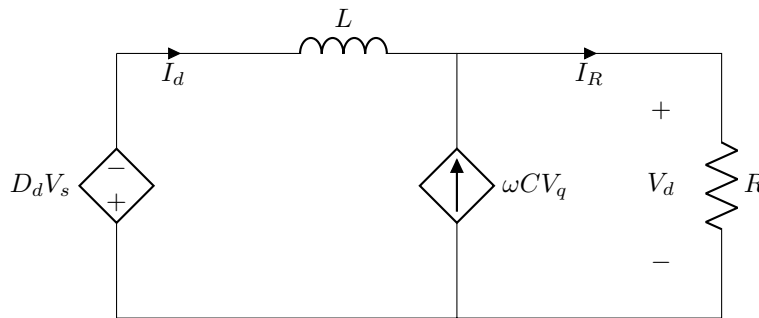


FIGURA 34: Circuito relacionado a la variable q **Fuente: Autor**

Por lo tanto, como los parámetros dq son en continuo se debe analizar el sistema en su punto de equilibrio, esto quiere decir que al trabajar en CC el inductor actúa como un corto circuito y el capacitor como un circuito abierto. De acuerdo a esto, a continuación se presenta el análisis matemático.

FIGURA 35: Circuito relacionado a la variable q Fuente: AutorFIGURA 36: Circuito relacionado a la variable q Fuente: Autor

Al analizar el circuito de la figura 36 se identifica que hay dos mallas por lo que se usa la ley de kirchhoff de voltajes. Sin embargo, también se puede evidenciar que las dos mallas comparten una fuente de corriente, esto quiere decir que es posible aplicar el método de super mallas para reducir el numero de ecuaciones. Por lo tanto, al usar el método mencionado se tiene:

$$-D_d V_s - \omega L I_q + V_d = 0 \quad (7.50)$$

En este caso se desea despejar la variable D_d que representa la entrada del sistema.

$$D_d V_s = V_d - \omega L I_q \quad (7.51)$$

$$D_d = \frac{V_d - \omega L I_q}{V_s} \quad (7.52)$$

Luego se reemplaza D_d por la expresión de la ecuación 5.17 y de esta manera tener:

$$D_d = \frac{V_d - \omega L I_q}{V_s} + \frac{1}{2} \quad (7.53)$$

Por otra parte, al usar la ley de kirchhoff de corrientes en el nodo del circuito se tiene la siguiente expresión:

$$I_d + \omega C V_q = I_R \quad (7.54)$$

Sin embargo, en este caso para reducir la expresión se considera que V_q se estable en cero.

$$I_d = I_R \quad (7.55)$$

Al usar la ley de ohm se tiene:

$$I_R = \frac{V_d}{R} \quad (7.56)$$

Con respecto al circuito relacionado a la variable q , se realiza el mismo análisis que el circuito anterior y de esta manera despejar la entrada D_q y la corriente I_q .

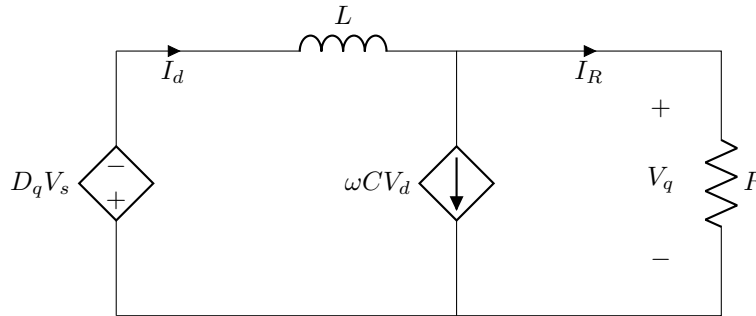


FIGURA 37: Circuito relacionado a la variable q Fuente: Autor

Al realizar el método de súper mallas al circuito presentado en la figura 37 se obtiene la siguiente expresión:

$$-D_q V_s + \omega L I_d + V_q = 0 \quad (7.57)$$

Se despeja la variable D_q y se tiene presente la consideración de que V_q es igual a cero.

$$D_q V_s = \omega L I_d \quad (7.58)$$

$$D_q = \frac{\omega L I_d}{V_s} \quad (7.59)$$

Se reemplaza la entrada D_q por la expresión de la ecuación 5.17.

$$D_q = \frac{\omega L I_d}{V_s} + \frac{1}{2} \quad (7.60)$$

Para hallar la corriente I_q se usa la ley de kirchhoff de corrientes y de esta manera obtener:

$$I_q = \omega C V_d + I_R \quad (7.61)$$

Dado que en esta caso la corriente en la resistencia depende del voltaje V_q la corriente I_R es igual a cero.

$$I_q = \omega C V_d \quad (7.62)$$

Finalmente se reemplaza la expresión de la ecuación 7.62 dentro de la ecuación 7.53, y la ecuación 7.56 dentro de la ecuación 7.60 y de esta manera obtener:

$$D_d = \frac{V_d(1 - \omega^2 LC)}{V_s} + \frac{1}{2} \quad (7.63)$$

$$D_q = \frac{\omega L V_d}{V_s R} + \frac{1}{2} \quad (7.64)$$

7.4. Técnica de control

En este caso se considera realizar un controlador convencional PI, el cual se constituye por una componente proporcional e integral encargadas de ajustar y mantener el comportamiento del sistema en un estado deseado. Donde la componente proporcional se encarga de actuar frente al error actual entre la referencia y la salida del sistema, a diferencia de la componente integral que trabaja con los errores pasados usando la corrección acumulativa.

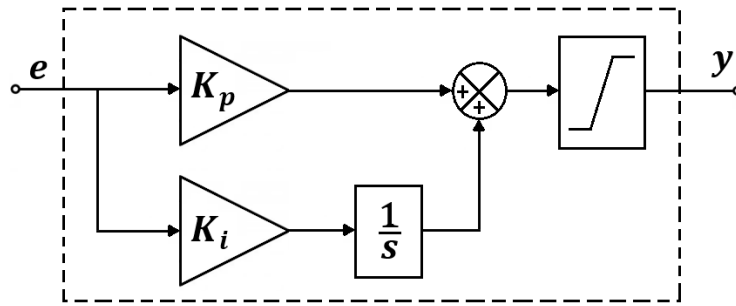


FIGURA 38: Compensador PI Fuente: Autor

y su función de transferencia es:

$$\frac{y(s)}{x(s)} = K_p + K_i * \frac{1}{s} \quad (7.65)$$

Dado que el diseño del compensador se realiza de manera digital, la discretización de la ecuación 7.65 con el método de tustin es la siguiente:

$$\frac{y(z)}{x(z)} = K_p + K_i * \frac{T_s}{2} * \frac{(z+1)}{(z-1)} \quad (7.66)$$

Al realizar la matemática correspondiente se tiene:

$$\frac{y(z)}{x(z)} = \frac{K_p(z-1) + K_i * \frac{T_s}{2}(z+1)}{z-1} \quad (7.67)$$

Y al factorizar:

$$\frac{y(z)}{x(z)} = \frac{z(K_p + K_i * \frac{T_s}{2}) - (K_p - K_i * \frac{T_s}{2})}{z-1} \quad (7.68)$$

Sin embargo, se debe tener presente que el compensador al contar con la acción integral puede generar un comportamiento no deseado, puesto que en la salida del controlador se tendrá una restricción dada por un saturador. Esto quiere decir, que cuando la señal de control se encuentra limitada y sale de dicho rango el integrador sigue acumulando error y trata de corregirlo causando respuestas no deseadas. Por lo tanto, como solución a dicho problema se usa un método anti-windup, el cual consiste en anular la acción integral temporalmente cuando se presente la saturación y de esta manera evitar que se acumule el error. Con base a esto, en la figura 39 se presenta el esquemático que es usado en la implementación digital.

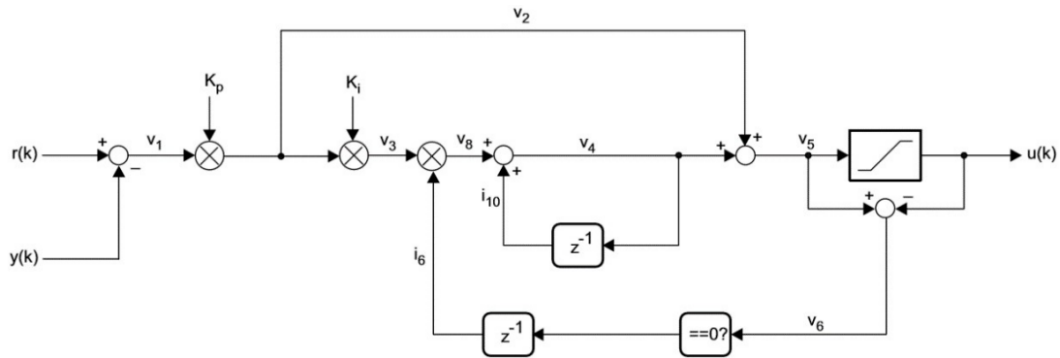


FIGURA 39: Ley de control en discreto. Fuente: DCL User's Guide.pdf

7.5. Espacio de estados

Para el diseño del compensador del inversor monofásico, se considera el marco estacionario ($dq0$), puesto que este se caracteriza por facilitar el diseño del controlador al no trabajar con componentes variantes en el tiempo. Esto quiere decir que para el diseño del controlador PI se toman como base las variables de estado presentadas en las ecuaciones 7.26, 7.28, 7.27 y 7.29, las cuales hacen referencia al voltaje y corriente del sistema. De acuerdo a esto, inicialmente se representan las variables de estado de forma matricial, usando la ecuación presentada a continuación.

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (7.69)$$

Donde (x) es el vector que hace referencia a las variables de estado y (u) el vector de las entradas, esto quiere decir que la expresión de la ecuación 7.74 es comúnmente usada para modelar como el estado de un sistema (x) cambia con respecto a una entrada (u) . Por lo tanto, la representación matricial de las variables de estado es:

$$\begin{bmatrix} \frac{d}{dt} i_d \\ \frac{d}{dt} v_d \\ \frac{d}{dt} i_q \\ \frac{d}{dt} v_q \end{bmatrix} = \begin{bmatrix} 0 & -1/L & \omega & 0 \\ 1/C & -1/RC & 0 & \omega \\ -\omega & 0 & 0 & -1/L \\ 0 & -\omega & 1/C & -1/RC \end{bmatrix} \begin{bmatrix} i_d \\ v_d \\ i_q \\ v_q \end{bmatrix} + \begin{bmatrix} V_s/L & 0 \\ 0 & 0 \\ 0 & V_s/L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_d & d_q \end{bmatrix} \quad (7.70)$$

En cuanto a la salida del sistema, para modelarla de manera matricial se usa la siguiente expresión.

$$y(t) = Cx(t) + Du(t) \quad (7.71)$$

Con respecto a la salida del sistema, el inversor al ser on-grid ya cuenta con un voltaje fijo en su salida, por lo que es necesario controlar la corriente y de esta manera obtener la potencia deseada. Sin embargo, como se está trabajando en coordenada $dq0$, el compensador cuenta con un lazo para cada constante, esto quiere decir que se debe diseñar un controlador tanto para la corriente i_d como para la i_q , de tal manera que usando la expresión de la ecuación 7.71 las salidas quedan expresadas de la siguiente manera.

$$y_d = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ v_d \\ i_q \\ v_q \end{bmatrix} \quad (7.72)$$

$$y_q = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_d \\ v_d \\ i_q \\ v_q \end{bmatrix} \quad (7.73)$$

7.6. Función de transferencia

Una vez que se tiene la representación matricial en variables de estado, es necesario determinar la función de transferencia del sistema, puesto que esta es la encargada de encontrar la relación entre la entrada y salida del mismo, pero en el dominio de Laplace. De acuerdo a esto, inicialmente se realiza la transformada de Laplace a las ecuaciones 7.74 y 7.71 considerando condiciones iniciales nulas, como se presenta a continuación.

$$\mathcal{L}\{\dot{x}(t)\} = \mathcal{L}\{Ax(t) + Bu(t)\} \quad (7.74)$$

$$\mathcal{L}\{y(t)\} = \mathcal{L}\{Cx(t) + Du(t)\} \quad (7.75)$$

Por lo tanto, se tiene:

$$sX(s) = AX(s) + BU(s) \quad (7.76)$$

$$Y(s) = CX(s) + DU(s) \quad (7.77)$$

Al factorizar la ecuación 7.76:

$$sX(s) - AX(s) = BU(s) \quad (7.78)$$

Dado que es una operación entre matrices, "s" se multiplica por una matriz identidad.

$$(sI - A)X(s) = BU(s) \quad (7.79)$$

$$X(s) = (sI - A)^{-1}BU(s) \quad (7.80)$$

Al sustituir la ecuación 7.80 en la ecuación 7.77 y al despejar la relación entre la salida y entrada se tiene:

$$Y(s) = C(sI - A)^{-1}BU(s) + DU(s) \quad (7.81)$$

$$\frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \quad (7.82)$$

Como se mencionó anteriormente, el sistema va contar con dos compensadores, por lo que es necesario identificar dos funciones de transferencia. Por lo tanto, para obtener la salida i_d frente a la entrada d_d se usa la expresión de la ecuación 7.82 con la matriz de estados B y C como:

$$\frac{i_d(s)}{d_d(s)} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \left[\begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & -\frac{1}{L} & \omega & 0 \\ \frac{1}{C} & -\frac{1}{RC} & 0 & \omega \\ -\omega & 0 & 0 & -\frac{1}{L} \\ 0 & -\omega & \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \right]^{-1} \begin{bmatrix} \frac{V_s}{L} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (7.83)$$

En cuanto a la salida i_q con respecto a la entrada d_q se usan las matrices B y C como:

$$\frac{i_q(s)}{d_q(s)} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & -\frac{1}{L} & \omega & 0 \\ \frac{1}{C} & -\frac{1}{RC} & 0 & \omega \\ -\omega & 0 & 0 & -\frac{1}{L} \\ 0 & -\omega & \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \right]^{-1} \begin{bmatrix} 0 \\ 0 \\ \frac{V_s}{L} \\ 0 \end{bmatrix} \quad (7.84)$$

Una vez que se opera tanto la ecuación 7.83 como la 7.84, se obtienen las siguientes funciones de transferencia.

$$\frac{i_d(s)}{d_d(s)} = \frac{V_s \left(\frac{1}{L}s^3 + \frac{2}{RLC}s^2 + \left(\frac{\omega^2}{L} + \frac{1}{L^2C} + \frac{1}{R^2LC^2} \right) s + \frac{1}{RL^2C^2} \right)}{s^4 + \frac{2}{RC}s^3 + \left(2\omega^2 + \frac{2}{LC} + \frac{1}{R^2C^2} \right) s^2 + \left(\frac{2\omega^2}{RC} + \frac{2}{RLC^2} \right) s + \omega^4 - \frac{2\omega^2}{LC} + \frac{\omega^2}{R^2C^2} + \frac{1}{L^2C^2}} \quad (7.85)$$

$$\frac{i_q(s)}{d_q(s)} = \frac{V_s \left(\frac{1}{L}s^3 + \frac{2}{RLC}s^2 + \left(\frac{\omega^2}{L} + \frac{1}{L^2C} + \frac{1}{R^2LC^2} \right) s + \frac{1}{RL^2C^2} \right)}{s^4 + \frac{2}{RC}s^3 + \left(2\omega^2 + \frac{2}{LC} + \frac{1}{R^2C^2} \right) s^2 + \left(\frac{2\omega^2}{RC} + \frac{2}{RLC^2} \right) s + \omega^4 - \frac{2\omega^2}{LC} + \frac{\omega^2}{R^2C^2} + \frac{1}{L^2C^2}} \quad (7.86)$$

7.7. Diseño del compesador

Para el diseño de los compensadores se deben tomar como base las funciones de transferencia presentadas en las ecuaciones 7.87 y 7.86, las cuales demuestran la dinámica de las constantes d y q . Sin embargo, como las dos son exactamente iguales se diseña el mismo compensador para cada lazo de control, para ello se usa la herramienta "sisotool" de Matlab, la cual permite realizar el diseño ajustando los parámetros del controlador, como lo son las ganancias y los polos para cumplir con los parámetros de diseño que se deseen. Por lo tanto, inicialmente se la ecuación 7.87 y se rempazan los valores presentados en la tabla 2.

V_s	$L(H)$	$C(F)$	$R(\Omega)$	$f(Hz)$
420	27mH	2μF	70Ω	60Hz

TABLA 2: Parámetros función de transferencia

Y de esta manera obtener:

$$\frac{i_d(s)}{d_d(s)} = \frac{(1,556 * 10^4)s^3 + (2,222 * 10^8)s^2 + (1,084 * 10^{12})s + 2,058 * 10^{15}}{s^4 + (1,429 * 10^3)s^3 + (8,834 * 10^7)s^2 + (2,666 * 10^{11})s + 3,449 * 10^{14}} \quad (7.87)$$

Dado que el diseño del controlador se realizará a partir de un sistema discreto, se discretiza la ecuación anterior por el método de tustin.

$$\frac{y(z)}{x(z)} = \frac{0,1553z^4 - 0,2681z^3 - 0,03847z^2 + 0,2683z - 0,1167}{z^4 - 3,72z^3 + 5,193z^2 - 3,225z + 0,7515} \quad (7.88)$$

Con base a la función de transferencia anterior, en la figura 40 se presenta la ubicación de los polos y ceros del sistema a controlar en el plano z.

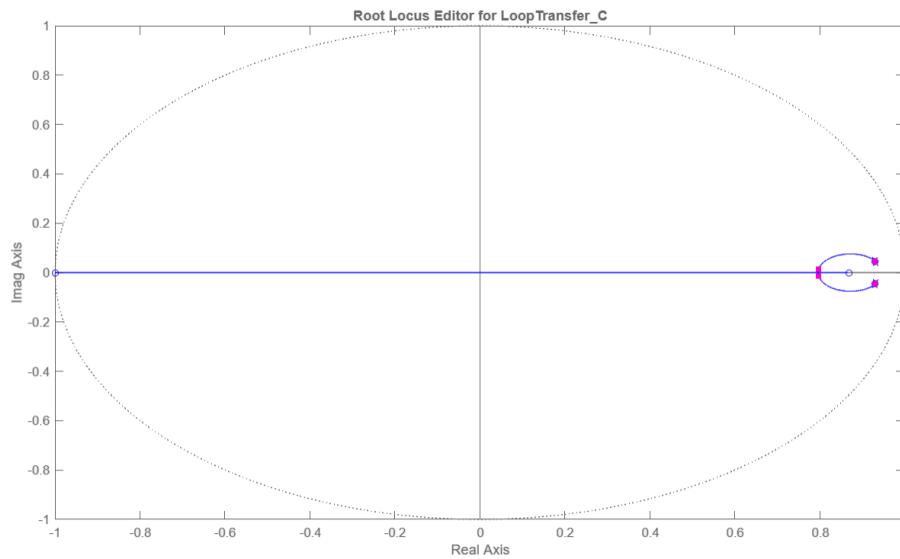


FIGURA 40: Polos y ceros plano z **Fuente: Autor**

Adicionalmente “sisotool” también permite tener una evaluación de la estabilidad del sistema dando una perturbación escalón, de tal manera que en la figura 41 se presenta la respuesta del sistema sin un compensador.

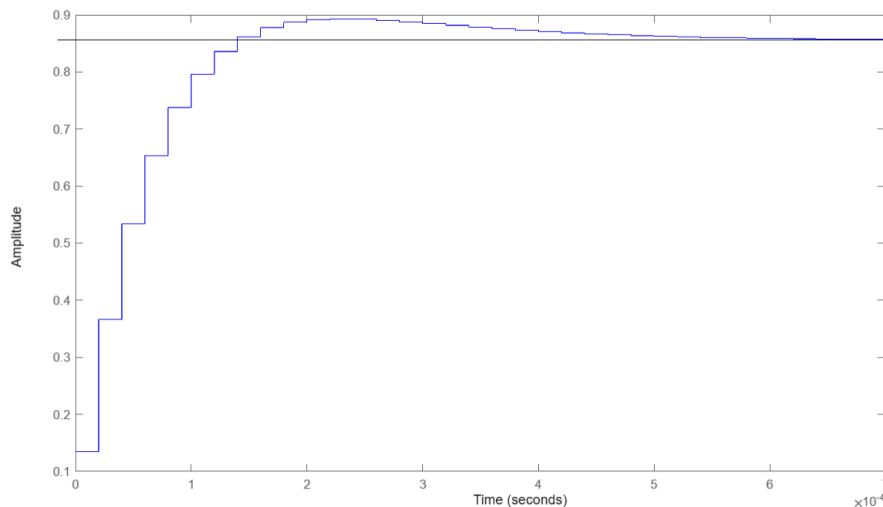


FIGURA 41: Respuesta del sistema sin compensador **Fuente: Autor**

En este caso el sistema no presenta inestabilidad, pero si error de estado estacionario. De acuerdo a esto se usa la herramienta “sisotool” para ajustar las ganancias proporcional e integral y de esta manera optimizar el rendimiento de la planta, una vez que se han ajustado las ganancias se obtiene el siguiente compensador.

$$\frac{y(z)}{x(z)} = 4,5966 * \frac{z - 0,975}{z - 1} \quad (7.89)$$

Por lo tanto la respuesta del sistema en lazo cerrado con el compensador diseñado se presenta en la figura 42, donde se evidencia que se ha eliminado el error de estado estacionario.

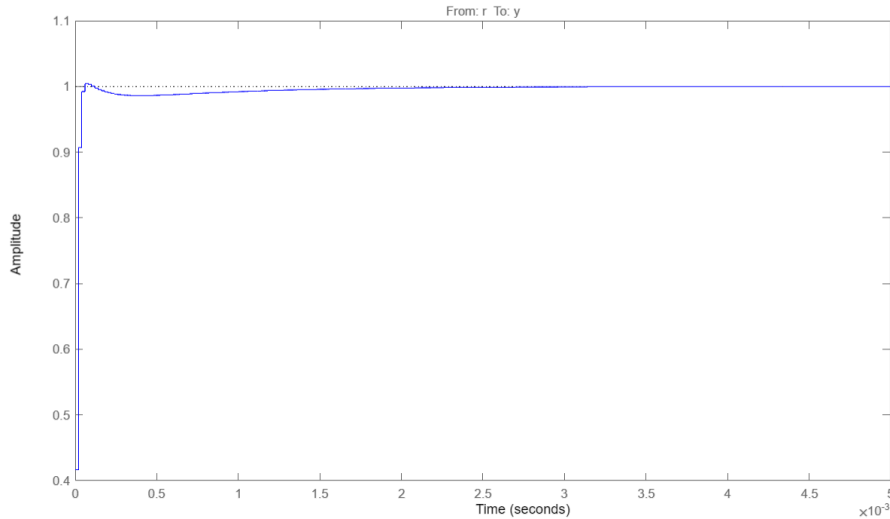


FIGURA 42: Respuesta del sistema con el control PI **Fuente: Autor**

Sin embargo, en la ecuación 7.89 no se evidencia cual es la ganancia proporción y la integral, por lo que es necesario usar la ecuación 7.66 para igualar los términos y de esta manera despejar dichas ganancias, a continuación se presenta la matematica.

$$K_p + K_i * \frac{T_s}{2} = 4,5966 \quad (7.90)$$

$$K_p - K_i * \frac{T_s}{2} = 4,481685 \quad (7.91)$$

Inicialmente se reemplaza la ecuación 7.91 dentro de la ecuación 7.90 y de esta manera se despeja la ganancia proporcional.

$$K_p + K_p - 4,481685 = 4,5966 \quad (7.92)$$

$$K_p = \frac{4,5966 + 4,481685}{2} \quad (7.93)$$

$$K_p = 4,5391425 \quad (7.94)$$

Para hallar la ganancia integral se usa la ecuación 7.91 y se despeja como se presenta a continuación:

$$K_i = \frac{2 * (K_p - 4,481685)}{T_s} \quad (7.95)$$

Al reemplazar la ganancia proporcional se tiene:

$$K_i = \frac{2 * (4,5391425 - 4,481685)}{T_s} \quad (7.96)$$

$$K_i = 2,2983 \times 10^{-6} \quad (7.97)$$

Una vez que han sido diseñados los compensadores para las corrientes i_d e i_q , se usan las salidas d_d y d_q para realizar la transformada inversa de park y de esta manera obtener alfa y beta. Sin embargo, en este caso solo se usa alfa para generar la modulación EPWM, esto se debe a que alfa es la representación del sistema real y beta del sistema imaginario, por lo que no se tiene en cuenta. Para una mejor comprensión en la figura 43 se presenta el sistema en lazo cerrado usando los compensadores para las constantes d y q .

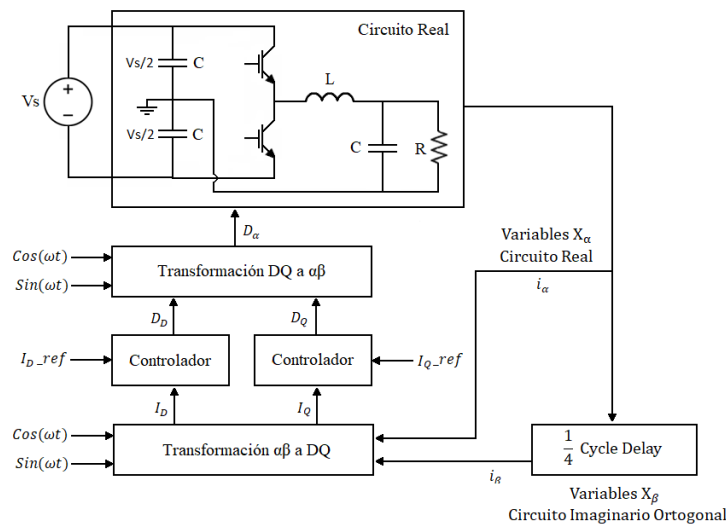


FIGURA 43: Esquema transformación DQ con control. Fuente: Adaptado de [64]

Capítulo 8

Resultados y Discusión

8.1. Resultados en Simulación

8.1.1. Simulación control sliding

Con respecto al apartado anterior, se emplea el modelo de la planta que se observa en la figura 44 en el software de Simulink para verificar el control antes de su implementación. En la simulación, se utilizan los valores de los parámetros que figuran en la tabla 3.

V_s	$L(H)$	$C(F)$	$R(\Omega)$	$f(Hz)$
340	$27mH$	$2\mu F$	200Ω	$60Hz$

TABLA 3: Parametros

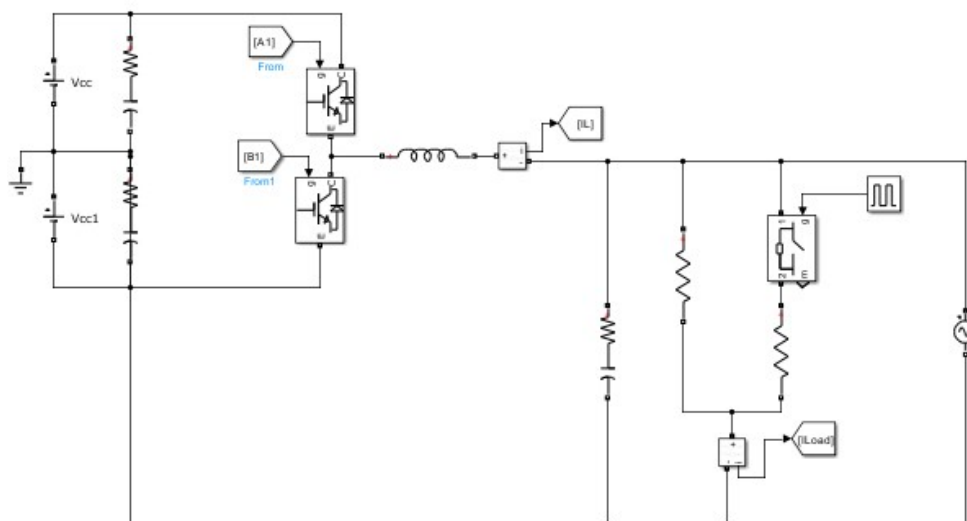


FIGURA 44: Modelo conmutado inversor - Simulink **Fuente: Autor**

Para implementar el control de corriente en la salida del inversor, se conecta un voltaje de red de 120V RMS. Además, la referencia de este control se obtiene a través de la corriente que fluye en la carga. El objetivo es que cuando el sistema se encienda, sea capaz de suministrar la corriente necesaria a la carga y la corriente de la red se aproxime a cero.

8.1.2. Implementación del control sliding en Simulink

En la figura 45 se presentan las etapas que conforman la estrategia de control que determina la señal de conmutación de los transistores.

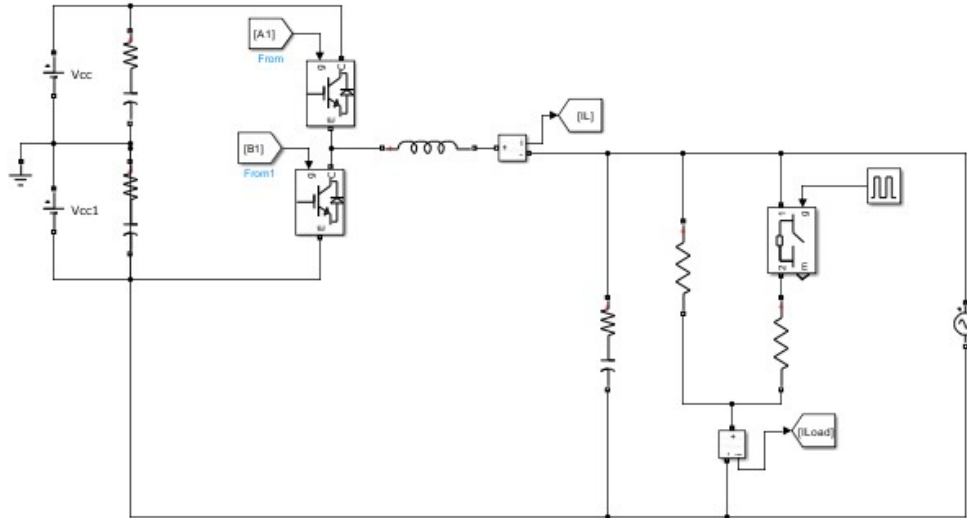


FIGURA 45: Control sliding **Fuente: Autor**

Para iniciar el control deslizante, se realiza una resta entre la corriente de referencia, que se obtiene de la carga, y la corriente de la bobina. Esta resta constituye nuestra superficie de deslizamiento.

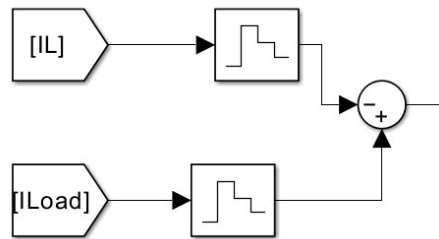


FIGURA 46: Superficie de deslizamiento **Fuente: Autor**

La operación de resta describe el comportamiento deseado del sistema. En caso de que el sistema se desvíe de esta referencia, se recurre a comparadores como se observa en la figura 47 que contrastan el valor del error con la superficie de deslizamiento. Cada uno de estos comparadores genera una señal que indica si el sistema se encuentra por encima o por debajo de la superficie en cada momento.

Además, se emplea un flip-flop RS a la salida de los comparadores con el propósito de prevenir cambios de estado no deseados y proporcionar un comportamiento de conmutación más estable a las señales del comparador.

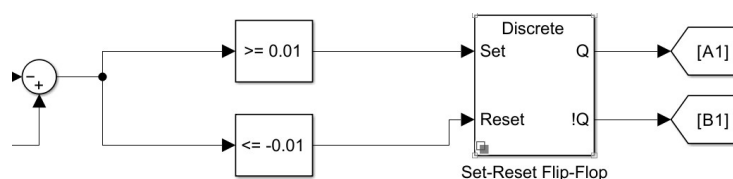


FIGURA 47: Superficie de deslizamiento **Fuente: Autor**

Las señales resultantes son las que determinan el comportamiento del inversor.

En la figura 48 se puede observar cómo la señal varía a lo largo de la superficie de deslizamiento, con una tendencia a converger hacia cero.

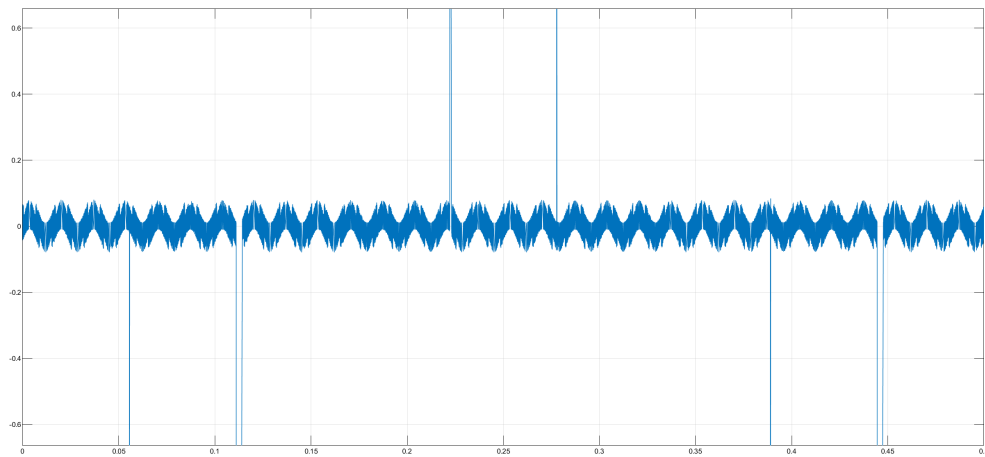


FIGURA 48: Señal de error **Fuente: Autor**

La señal resultante de los flip-flops regula la conmutación de los transistores, cabe aclarar que estas señales son complementarias entre sí como se observa en la figura 49.

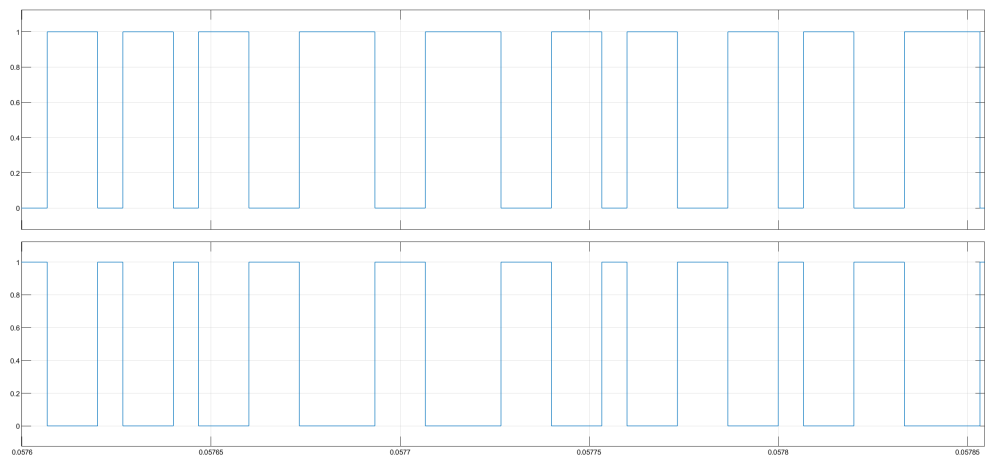


FIGURA 49: Señales de conmutación **Fuente: Autor**

Mientras que el inversor tenga la capacidad de suministrar la potencia requerida a la carga, cuando inicia la conmutación, la red mantiene el voltaje constante como se observa en la figura 50.

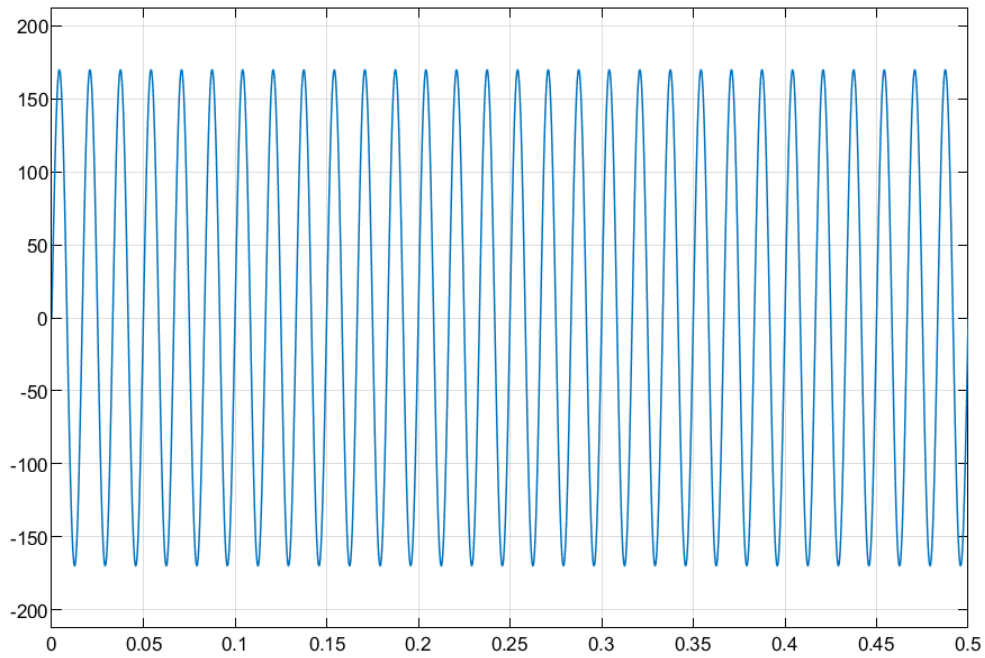


FIGURA 50: Voltaje de la red **Fuente: Autor**

En este punto, es el propio sistema el encargado de suministrar la corriente, lo que hace que la potencia suministrada por la red disminuya casi a cero. En la figura 51 se observa el seguimiento de corriente que se genera en la bobina con respecto a la corriente de referencia. Con el fin de analizar el funcionamiento de la estrategia de control con perturbaciones, mediante una conmutación se conecta una carga en paralelo, que modifica el valor de la carga por lo cual cambia la corriente de referencia.

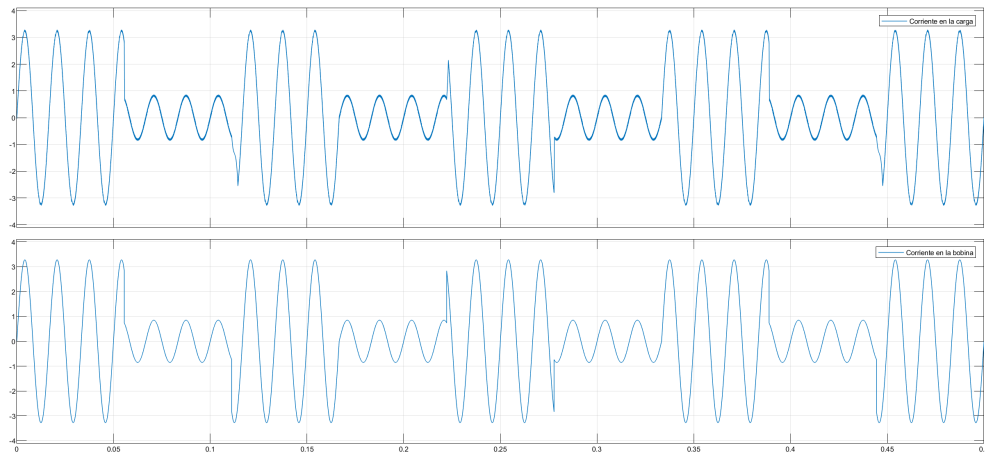


FIGURA 51: Corriente en la bobina y Corriente de referencia **Fuente: Autor**

Cuando el sistema está apagado, se sabe que la red es la que suministra la corriente requerida por la carga. En la simulación se observó que cuando el inversor se enciende, la corriente proveniente de la red es de aproximadamente 2 miliamperios como se observa en la figura 52, lo que nos garantiza que el sistema tiene la capacidad de suplantar a la red al suministrar la potencia requerida a la carga.

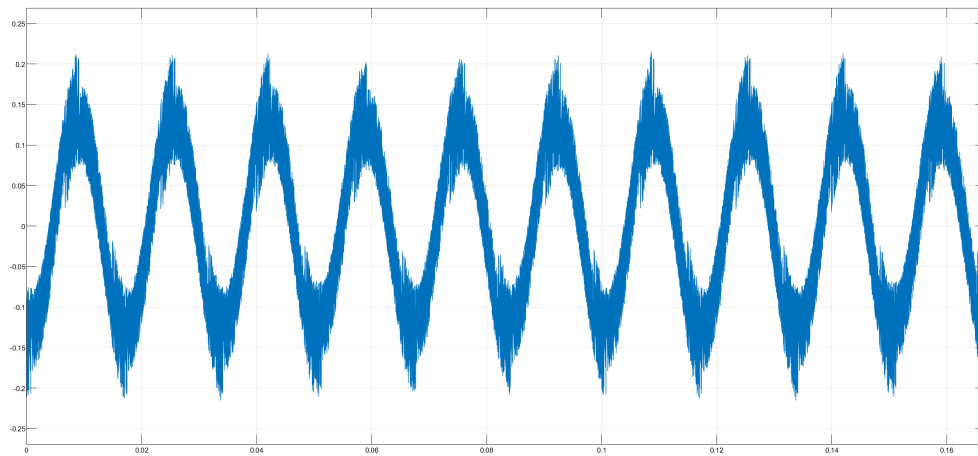


FIGURA 52: Corriente en la red con el sistema encendido **Fuente: Autor**

8.1.3. Simulación transformada dq

Para realizar la simulación del sistema en coordenadas dq se toman las ecuaciones 7.26, 7.27, 7.28 y 7.29, y se modelan en diagramas de bloques, con el fin de evaluar si efectivamente se obtienen variables de régimen permanente. En este caso se consideran los parámetros presentados en la tabla 4, donde d_d y d_q se determinan usando las ecuaciones 7.63 y 7.63 , considerando $V_d = 170$ y $V_q = 0$.

V_s	$L(H)$	$C(F)$	$R(\Omega)$	$f(Hz)$	d_d	d_q
340	27mH	2 μF	50 Ω	60Hz	0.99616	0.60178

TABLA 4: Parámetros transformación dq

Como se puede observar en la figura 53, al simular la representación en coordenadas dq de la corriente I_d se tiene como resultado una señal que tiene como punto de equilibrio 3.4 amperios.

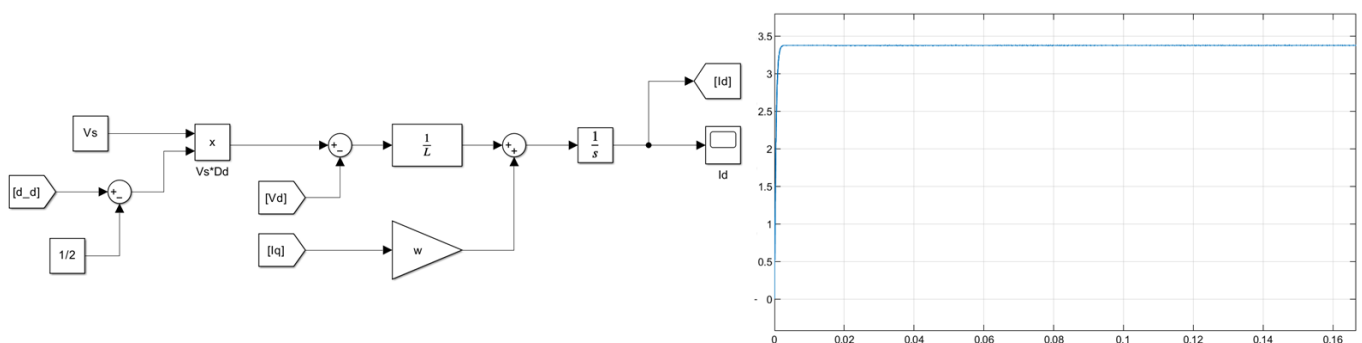
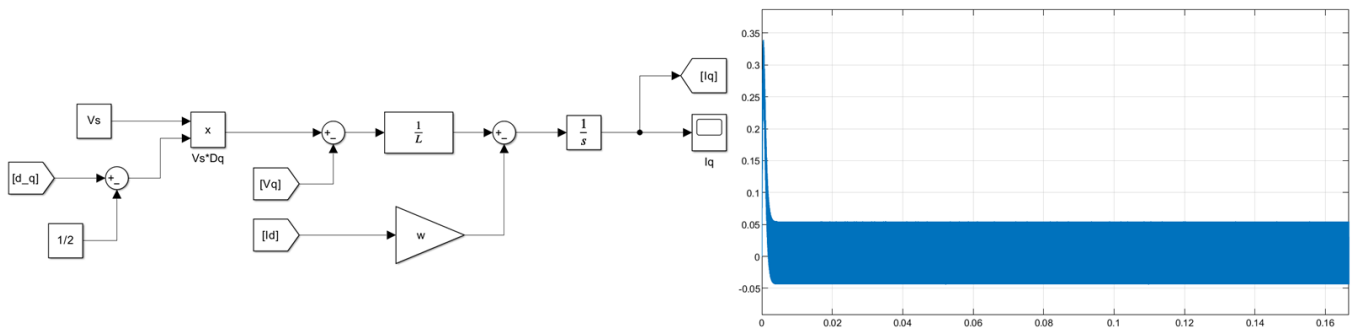
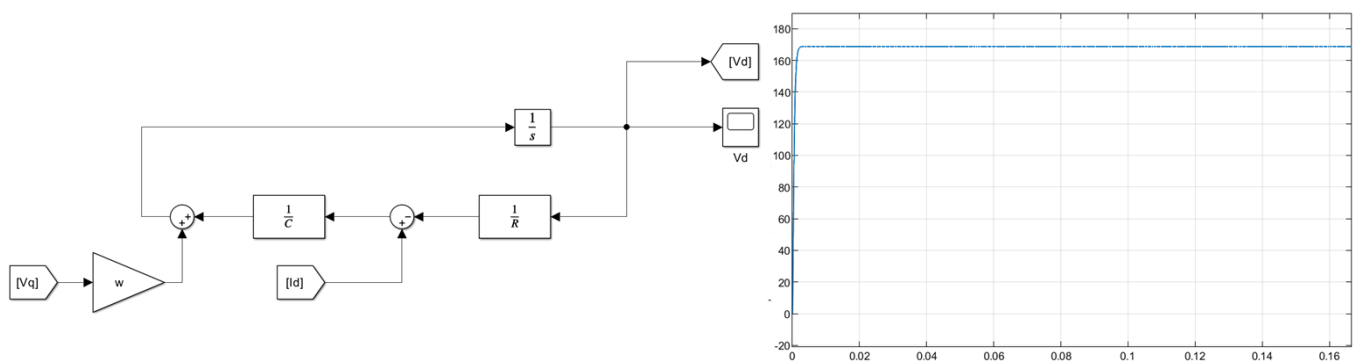


FIGURA 53: Diagrama de bloques i_d **Fuente: Autor**

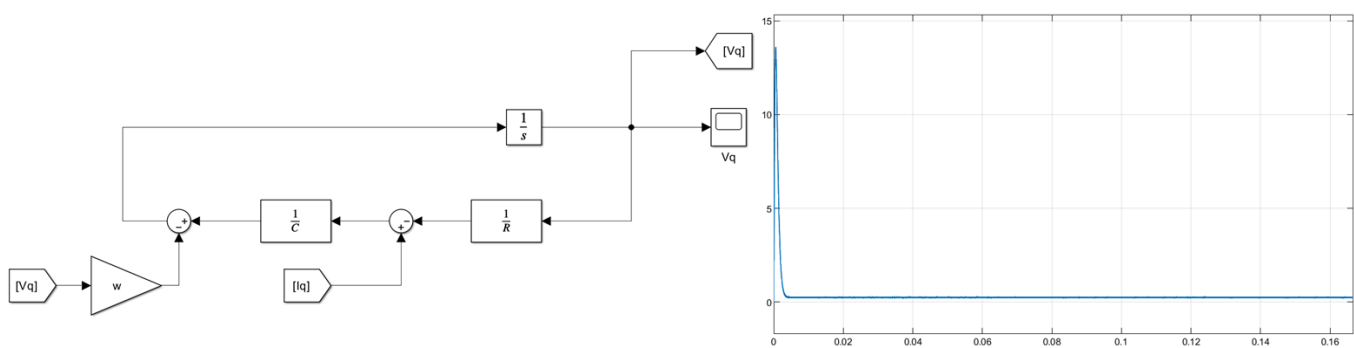
A diferencia de la señal presentada en la figura anterior, la corriente I_q oscila cerca al cero como se puede observar en la figura 54.

FIGURA 54: Diagrama de bloques v_d Fuente: Autor

Puesto que para el calculo de la entra d_d se considero que V_d es igual a 170, al realizar la transformada de Park la señal tendra como punto de dicho valor como se presenta en la figura 53.

FIGURA 55: Diagrama de bloques i_q Fuente: Autor

Al igual que en el caso anterior, como se considero que V_q tendrá como punto de equilibrio 0, al realizar la transformación de Park se tiene como resultado la señal presentada en la figura 53.

FIGURA 56: Diagrama de bloques v_q Fuente: Autor

8.1.4. Simulación transformada inversa DQ

Para realizar las simulaciones correspondientes a la transformada inversa de Park se toman como base las ecuaciones 7.46, 7.47, 7.48 y 7.49, , y de esta manera realizar su representación en diagrama de bloques. En este caso se considera que las entradas del sistema para realizar la transformación son V_d , V_q , d_d y d_q , estoy con el

fin de obtener alfa (α) y beta (β) tanto de los voltajes como de las corrientes, descuerdo a esto para realizar la simulación se consideran los datos presentados en la tabla 5.

V_d	V_q	$L(H)$	$C(F)$	$R(\Omega)$	$f(Hz)$	d_d	d_q
170	0	27mH	2 μ F	50 Ω	60Hz	0.99616	0.60178

TABLA 5: Parámetros transformación inversa dq

Al realizar la transformada inversa a la corriente I_d se tiene como resultado la señal presentada en la figura 57, donde se observa que tiene como valor pico I_d .

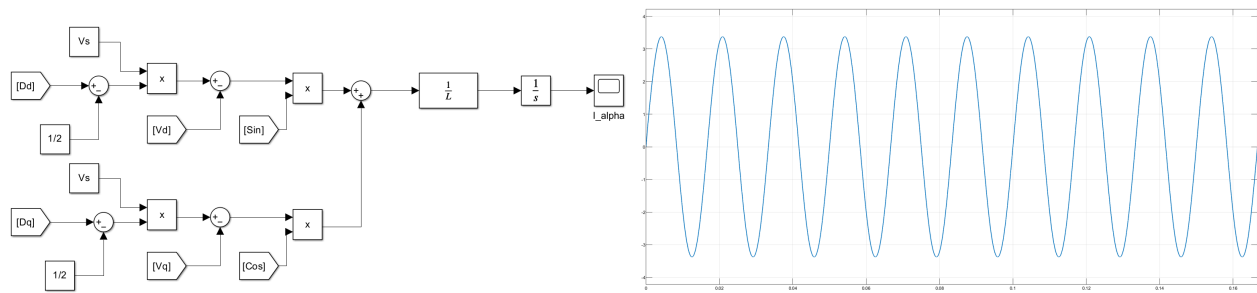


FIGURA 57: Diagrama de bloques transformada inversa v_α Fuente: Autor

Con respecto a la señal I_{beta} , en la figura 58 se observa que es igual que la señal presentada en la figura 57 pero desfasada 90 grados.

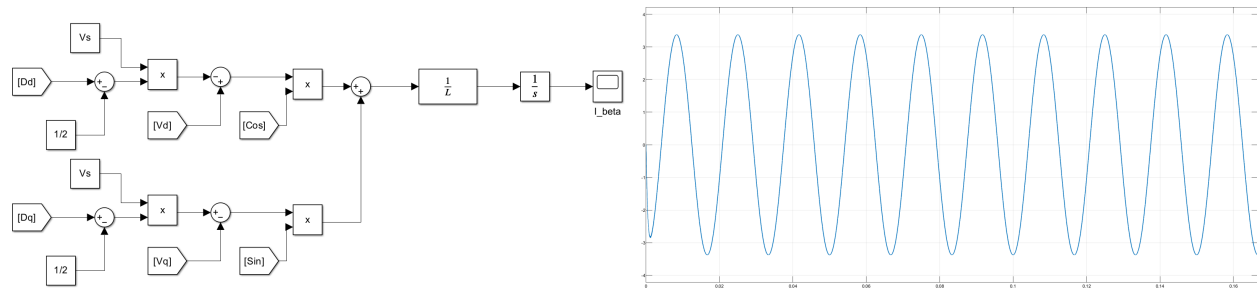


FIGURA 58: Diagrama de bloques transformada inversa v_β Fuente: Autor

En la figura 60 se presenta la respuesta del sistema al realizar la transformada inversa de V_d y de esta manera obtener V_α , como se puede observar el valor pico de la señal es 170 lo que corresponde al valor que tenía V_d en la figura .

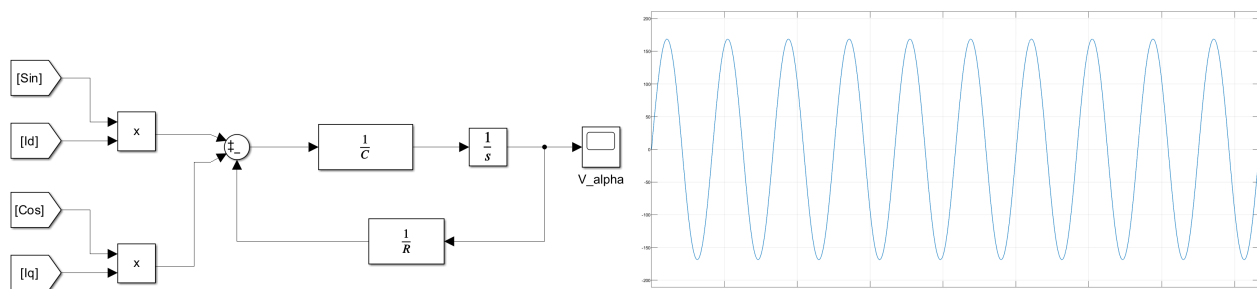


FIGURA 59: Diagrama de bloques transformada inversa i_α Fuente: Autor

Como se observa en la figura 60, al realizar la transformada inversa de V_q se tiene como resultado la misma señal que V_α pero desfasada 90 grados.

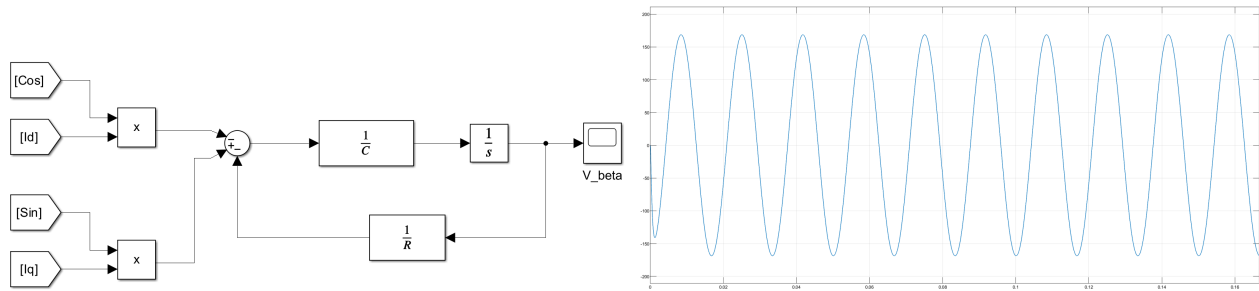


FIGURA 60: Diagrama de bloques transformada inversa i_β Fuente: Autor

8.1.5. Simulación control PI en coordenadas DQ

Para realizar la simulación en coordenada dq inicialmente se debe diseñar un bloque encargado de ejecutar la transformación de Park, tomando como base la expresión presentada en la ecuación 7.7. Por otra parte, como el sistema necesita tanto α (α) como β (β) para la transformación, se tiene en cuenta un retardo con respecto a la señal original y de esta manera obtener las dos señales con un desfase de 90 grados, de acuerdo a esto en la figura 61 se presenta el diagrama de bloques usado.

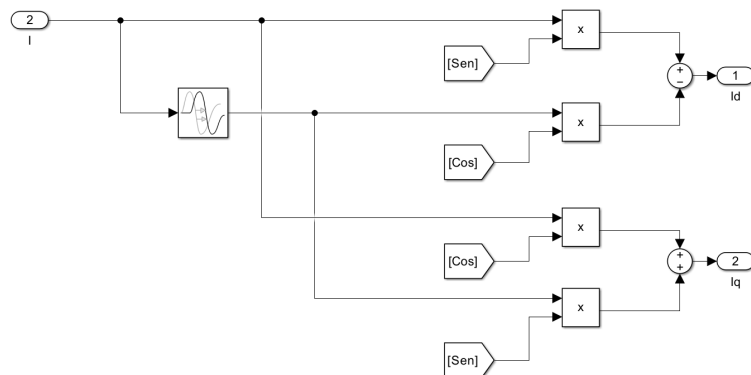


FIGURA 61: Diagrama de bloques transformación de Park Fuente: Autor

En este caso se usa el diagrama de bloques presentado en la figura 61 para realizar la transformación a la corriente de referencia que en este caso es la carga y la corriente de la bobina, puesto que es la señal a controlar. Luego se realiza la implementación del controlador PI, que es igual tanto para la corriente i_d como i_q . De acuerdo a esto a continuación se presenta el diagrama de bloques del controlador PI en coordenadas dq .

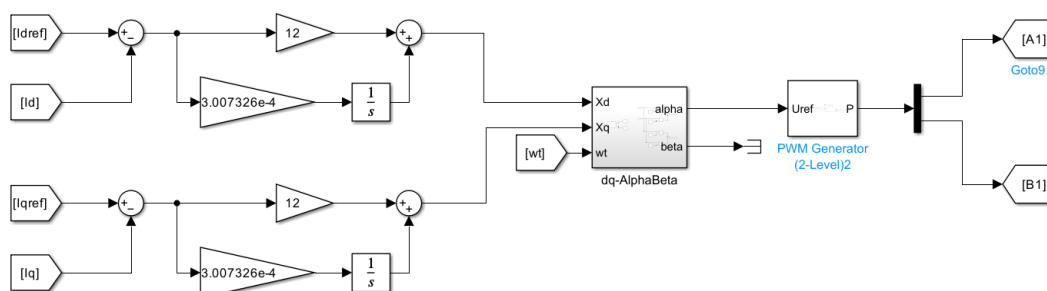


FIGURA 62: Compensador coordenadas dq Fuente: Autor

Como se puede observar en la figura 62, luego de que los compensadores de d y q ajustan su salida para eliminar el error, se realiza la transformada inversa Park tomando como base las salidas antes mencionadas y de esta manera obtener nuevamente α (α) y β (β), donde únicamente se usa α (α) que representa al circuito real para generar la modulación PWM y de esta manera controlar la dinámica del sistema. Por otra parte, para realizar la transformación inversa se toma como base la expresión de la ecuación 7.10 con el fin de obtener el diagrama de bloques presentado en la figura 63.

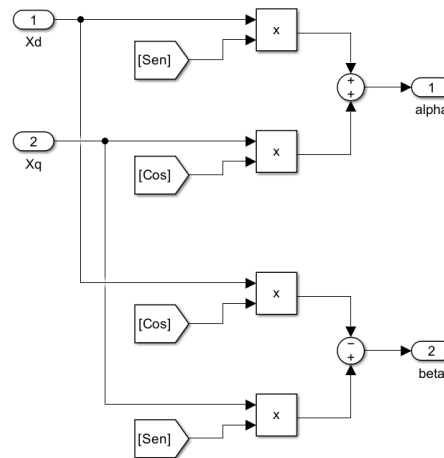


FIGURA 63: Diagrama de bloques v_q Fuente: Autor

Adicionalmente, también se considero realizar el control de manera discreta usando los bloques presentados en la figura 64, donde también se tiene presente el método de clamping para evitar el efecto windup, de acuerdo a esto los niveles de saturación son entre 0.95 y -0.95.

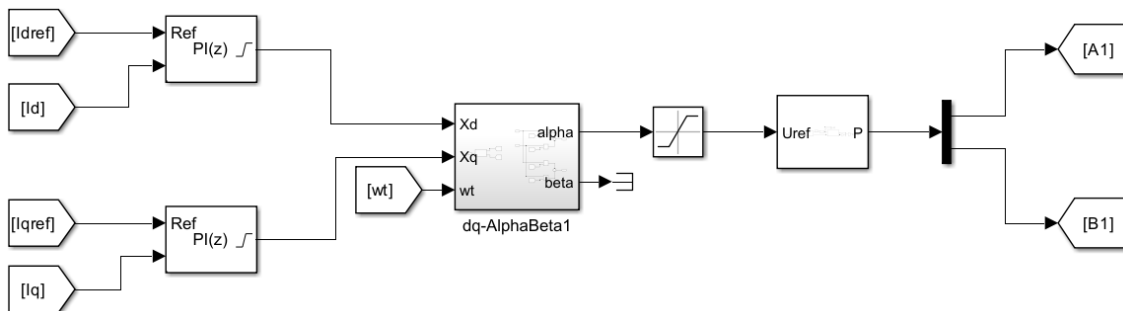


FIGURA 64: Compensador coordenadas dq en discreto Fuente: Autor

Por lo tanto teniendo en cuenta lo mencionado anteriormente, la el sistema en lazo cerrado se encuentra representado de la siguiente manera:

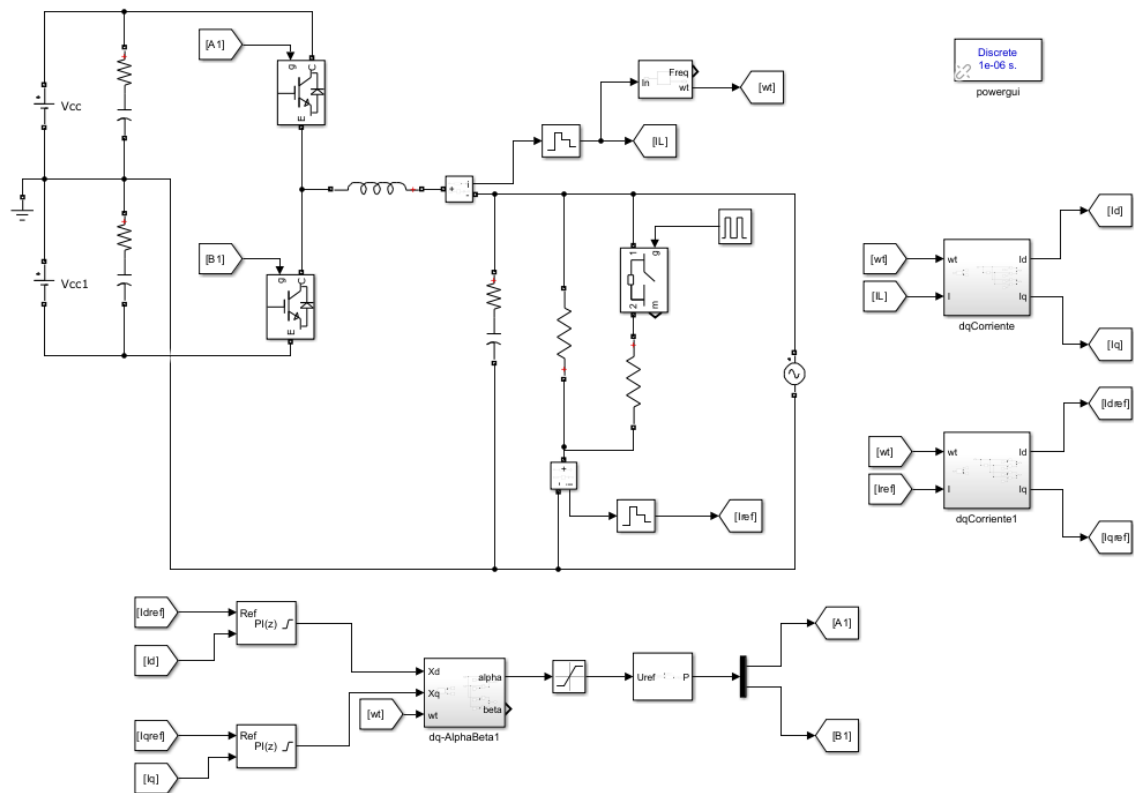


FIGURA 65: Sistema lazo cerrado en coordenadas dq Fuente: Autor

Al simular el sistema presentado en la figura 65 tomando como base los datos presentados en la tabla n, se tiene que la señal de referencia al realizar la conmutación para habilitar y deshabilitar el paralelo entre las dos resistencias tiene como respuesta lo presentado en la figura 66 y en la figura 67 se muestra el resultado de realizar la transformación de Park, donde la señal de color azul hace alusión a d y la naranja a q .

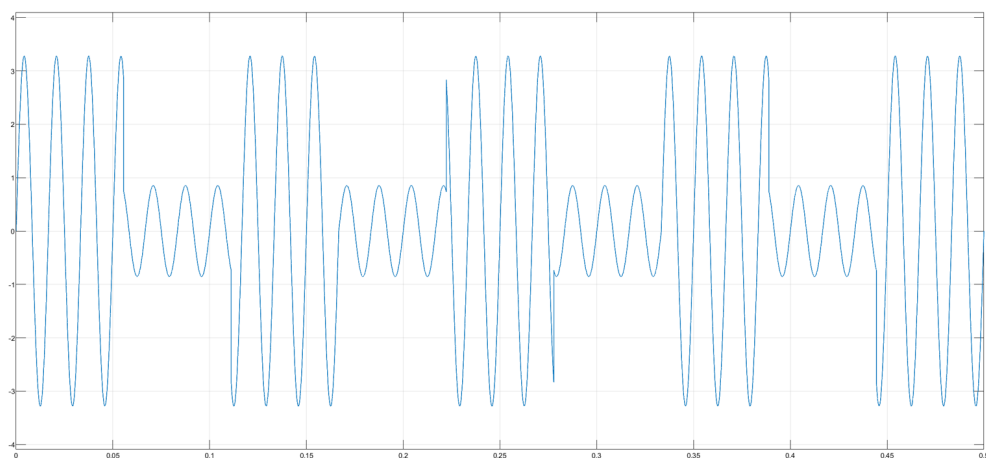


FIGURA 66: Señal de referencia Fuente: Autor

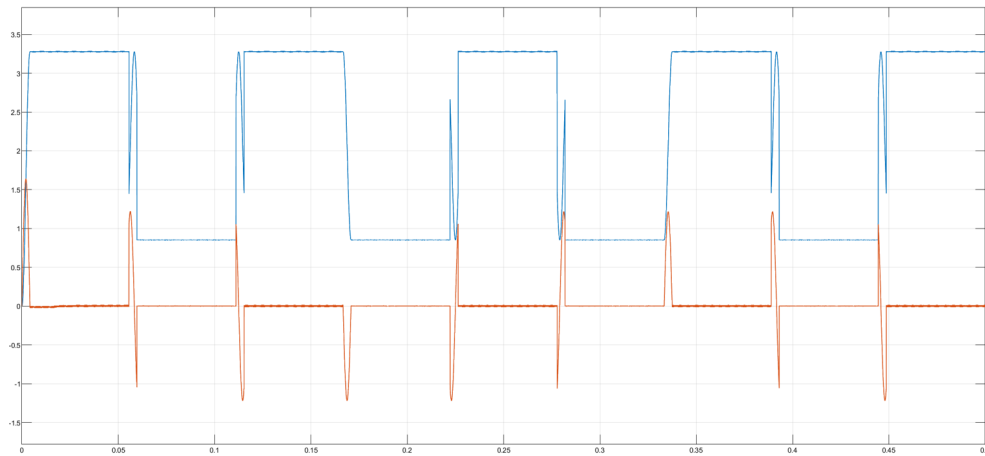


FIGURA 67: Transformación de Park señal de referencia **Fuente: Autor**

Con respecto a la corriente en la bobina que es la señal que se desea controlar se tiene la respuesta presentada en la figura 68 y su representación en coordenadas dq luego de la transformación de Park en la figura 69.

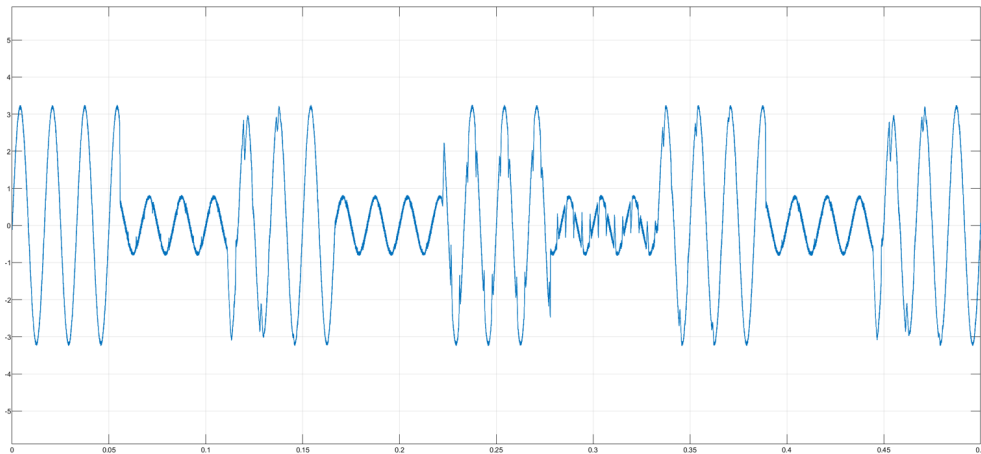


FIGURA 68: Señal de referencia **Fuente: Autor**

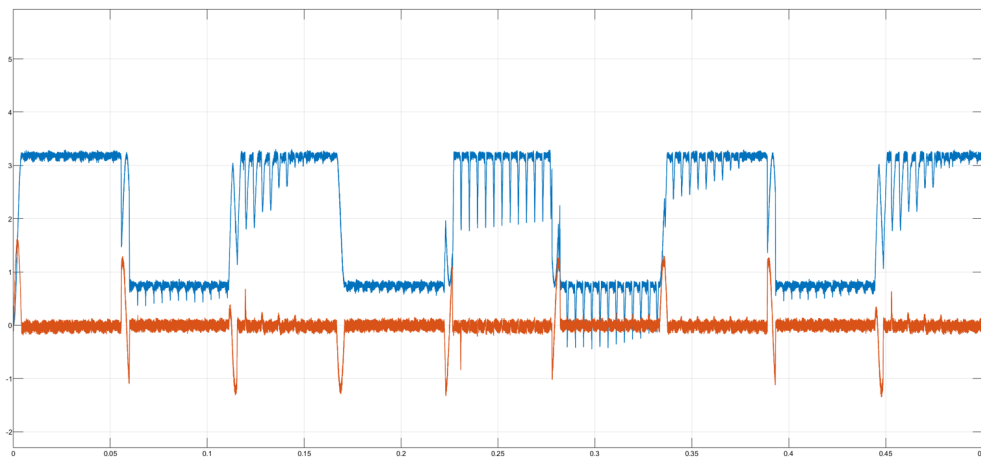


FIGURA 69: Transformación de Park señal de referencia **Fuente: Autor**

8.2. Resultados Experimentales

8.2.1. Implementación del control slidig

La validación práctica implica la implementación de los dos diagramas de flujos presentados en la sección 6.3. Siguiendo este enfoque, se realiza una prueba inicial con una carga de $200\ \Omega$ para analizar el rendimiento del sistema con corrientes por debajo de 1 amperio. En la figura 70 se puede apreciar que debido a la superficie de deslizamiento, la corriente generada en la bobina mantiene la misma forma que la corriente de referencia presente en la carga.

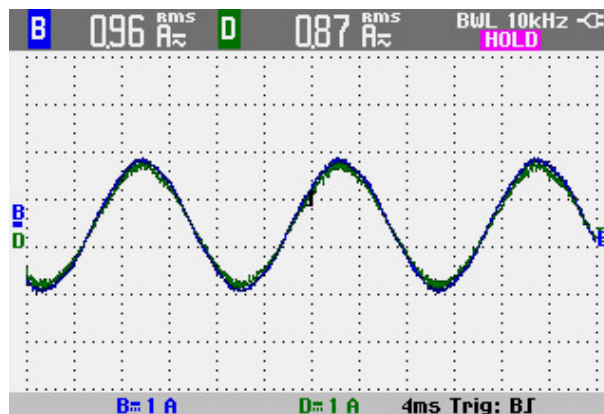


FIGURA 70: Corriente con carga de $200\ \Omega$ Fuente: Autor

La tabla 6 presenta los resultados obtenidos en términos de potencia y corriente tanto en el inversor como en la red eléctrica.

$I_{delared}$	$I_{delsistema}$	$P_{delared}$	$P_{delsistema}$
0,1A	0,87A	15	100

TABLA 6: Corrientes y Potencias experimentales carga de $200\ \Omega$

Con el fin de evaluar la respuesta del sistema ante una perturbación, se lleva a cabo una conmutación utilizando una carga de $70\ \Omega$ que se conecta cada 0.1 segundos en paralelo con la carga de $200\ \Omega$, con el propósito de inducir una perturbación y generar una corriente superior a 2 amperios, como se observa en la siguiente figura 71. Es evidente que la superficie de deslizamiento permite el seguimiento de la corriente a pesar de la presencia de perturbaciones.

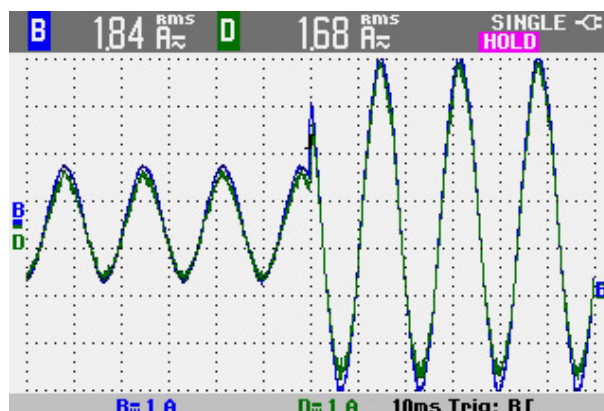


FIGURA 71: Corriente con perturbación Fuente: Autor

La tabla 7 presenta los resultados obtenidos en la planta en términos de potencia y corriente tanto en el inversor como en la red eléctrica.

$I_{delared}$	$I_{delsistema}$	$P_{delared}$	$P_{delsistema}$
1,84A	1,68A	20	186

TABLA 7: Corrientes y Potencias experimentales con carga de $50\ \Omega$

Además de la introducción de perturbaciones o cambios en la carga, la señal de voltaje que se observa en la figura 72 no se ve afectada, ya que la estrategia de control se enfoca en la corriente. Es importante destacar que en ambos escenarios, la forma de onda es sinusoidal en todas las pruebas y que, a pesar de los cambios en la referencia o la aparición de perturbaciones, el control deslizante garantiza que la corriente permanezca dentro de la superficie de deslizamiento.

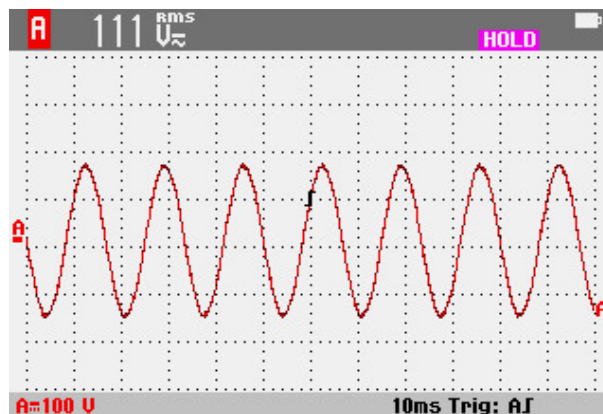


FIGURA 72: Voltaje en la salida Fuente: Autor

8.2.2. Implementación del control PI

Para la validación práctica se implementa de manera digital el compensador presentado en la figura 39, el cual cuenta con el método anti-windup para prevenir problemas por parte de la acción integral que conforma al compensador PI. Por otra parte, se tienen como ganancias proporcional e integral los valores presentados en las ecuaciones 7.94 y 7.97, los cuales fueron determinados de manera teoría. Por lo tanto, inicialmente se realizan pruebas experimentales con una carga de $210\ \Omega$ y de esta manera evidenciar el comportamiento del sistema con corrientes a baja estala como se presenta en la figura 73.

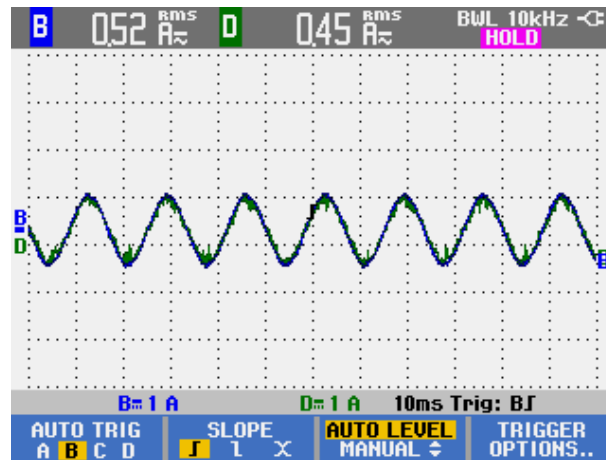


FIGURA 73: Corriente con carga de 210Ω . Fuente: Autor

Como se puede observar en la figura anterior, el sistema en lazo cerrado usando el mismo compensador tanto para la variable d como q luego de realizar la transformada de Park no tiene problemas de estabilidad y no cuanta con error de estado estacionario, sin embargo, si presenta distorsión armónica a causa de las conmutaciones que controlan la dinámica del sistema. Así mismo, en la tabla n se presentan los datos con relación a la corriente de referencia, la corriente controlada, la potencia que aporta la red eléctrica a la carga que se esta trabajando y la potencia generada por el sistema.

$I_{d,delared}$	$I_{d,delsistema}$	$P_{d,delared}$	$P_{d,delsistema}$
0,52A	0,45A	7	50

TABLA 8: Corrientes y Potencias experimentales con carga de 210Ω

Luego se realizan pruebas a alta escala, con el fin de obtener la potencia propuesta de 200 watts, en este caso se usa una carga de 52Ω para evaluar el comportamiento del sistema con el PI.

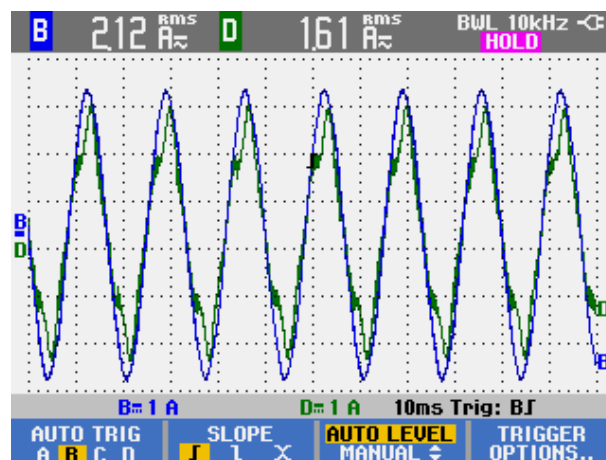


FIGURA 74: Corriente con carga de 52Ω . Fuente: Autor

Como se puede observar en la figura 74, La respuesta por parte del sistema presenta error de estado estacionario con respecto a señal de referencia, puesto que a pesar de tener 440 voltios en la entrada del sistema este no es suficiente para satisfacer las necesidades del circuito, por ende, el compensador no corrige en su totalidad la señal a controlar. De acuerdo a esto en la tabla n se presentan los datos correspondientes a las mediciones realizadas con una corriente a alta escala.

<i>I.delared</i>	<i>I.delsistema</i>	<i>P.delared</i>	<i>P.delsistema</i>
2,12A	1,61A	53	180

TABLA 9: Corrientes y Potencias experimentales con carga de 52

Finalmente se evalúa la estabilidad del sistema frente a una perturbación, para ello, al igual que en el control de modo deslizante, se realiza la conmutación entre dos cargas en paralelo. Esto quiere decir que el sistema siempre tendrá fija la carga de $210\ \Omega$ en su salida y eventualmente cada 0.1 segundos se conectará o desconectará el paralelo de la carga de $70\ \Omega$ para generar la perturbación. De acuerdo a esto en la figura 75 se presenta el comportamiento del sistema frente a una perturbación usando el compensador PI.

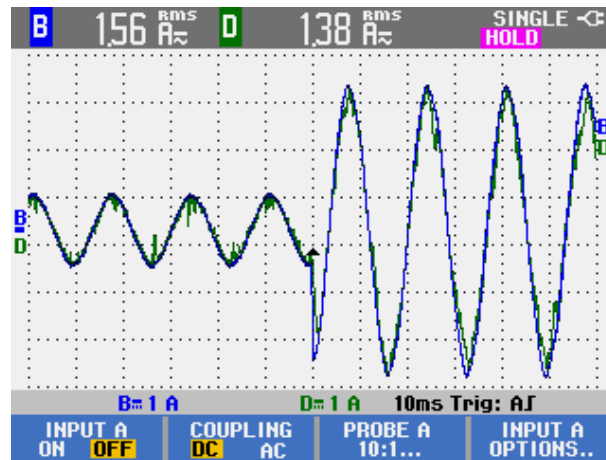


FIGURA 75: Corriente con Perturbación. Fuente: Autor

Con respecto a la figura 75, se observa que el sistema no presenta sobre picos cuando hay cambio de carga y tiene un buen comportamiento únicamente cuando se tienen corrientes bajas, puesto que como se mencionó, cuando el sistema requiere una corriente muy alta se debe tener un voltaje de entrada muy alto para que se elimine el error de estado estacionario.

8.3. Comparación Técnicas de control implementadas

A continuación, se presenta en la tabla 10 y en la tabla 11, ventajas y desventajas observadas durante la implementación y el diseño tanto del control PI como del control deslizante.

Técnica de Control	Ventajas	Desventajas
Sliding	■ Una de las ventajas sobresalientes del control deslizante, en comparación con el control PI, radica en su capacidad para operar sin requerir un modelo matemático del sistema. El control deslizante demostró su capacidad para adaptarse y controlar eficazmente el inversor sin depender de un modelo detallado. ■ La implementación del control deslizante digital, como se pudo apreciar en la sección 8.2.1, se caracteriza por su sencillez, especialmente cuando se comprende a fondo cómo funciona el principio del sliding ■ En el caso de un inversor, por lo general, se necesita un voltaje de entrada al menos el doble de la tensión de salida deseada. No obstante, durante la implementación del control sliding, no fue necesario aumentar la tensión de entrada previamente establecida para generar la corriente deseada en la salida.	■ Una característica fundamental del control deslizante es su enfoque en mantener el sistema dentro de una región deslizante específica para gestionar eficazmente las perturbaciones. Sin embargo, cuando esta región deslizante no se define con la suficiente precisión, conduce a la aparición de un error de estado estacionario en el sistema controlado. ■ La elección de la superficie de deslizamiento en el control deslizante implica una investigación exhaustiva sobre el funcionamiento tanto del control sliding como del sistema al que se pretende implementar. Una selección inapropiada puede dar lugar a comportamientos no deseados o incluso a la inestabilidad del sistema. Por lo tanto, esta elección debe garantizar un rendimiento óptimo del sistema bajo el control deslizante.

TABLA 10: Comparación técnica de control implementada Sliding

Técnica de Control	Ventajas	Desventajas
PI	■ Ofrece una respuesta rápida ante variaciones en el valor de referencia y, además, evita la generación de sobrepicos en el momento de la perturbación. Esto se debe a que el controlador proporcional-integral (PI) se destaca por su capacidad para prevenir oscilaciones o picos no deseados en el sistema, lo que lo convierte en una opción valiosa cuando se busca una respuesta suave y controlada. ■ El compensador no presenta un error de estado estacionario, ya que la componente integral del mismo trabaja para minimizar dicho error y mantener la respuesta del sistema en el valor deseado de manera constante, sin experimentar oscilaciones, lo que contribuye a preservar su estabilidad.	■ Necesita que el voltaje de entrada aumente de manera proporcional con respecto a la corriente de referencia, puesto que, si la corriente en la carga aumenta considerablemente y el voltaje de entrada no es lo suficientemente grande, la corriente a controlar que en este caso es la bobina presentara error de estado estacionario. ■ Para el diseño se debe comprender de manera adecuada el comportamiento y dinámica del sistema, por lo que es necesario realizar correctamente el modelado del circuito. Como en este caso se realiza el compensador en coordenadas dq, el proceso a realizar es largo y aumenta las ecuaciones de estado, dado que por cada variable de estado se generan dos más, que hacen referencia a las coordenadas d y q.

TABLA 11: Comparaciones técnicas de control implementada PI

Capítulo 9

Creación de proyecto en code Composer

9.1. Creación de proyecto con driverlib

Debido a la complejidad del dispositivo F28379D, este permite la creación y depuración de proyectos en dos tipos de software de desarrollo, API de biblioteca de controladores y por estructuras de campo de bits. Estos softwares llegan a ser compatibles entre sí o se pueden trabajar por separado; en este caso se crea el proyecto con el fin de que se pueda programar de los dos modos de operación como se presenta a continuación:

- 1. Desde la ventana principal se selecciona Archivo ->Nuevo ->Proyecto CSS, en este se selecciona Target como genérico, se le asigna un nombre y ubicación. Por último se da click en finalizar y se crea el proyecto.

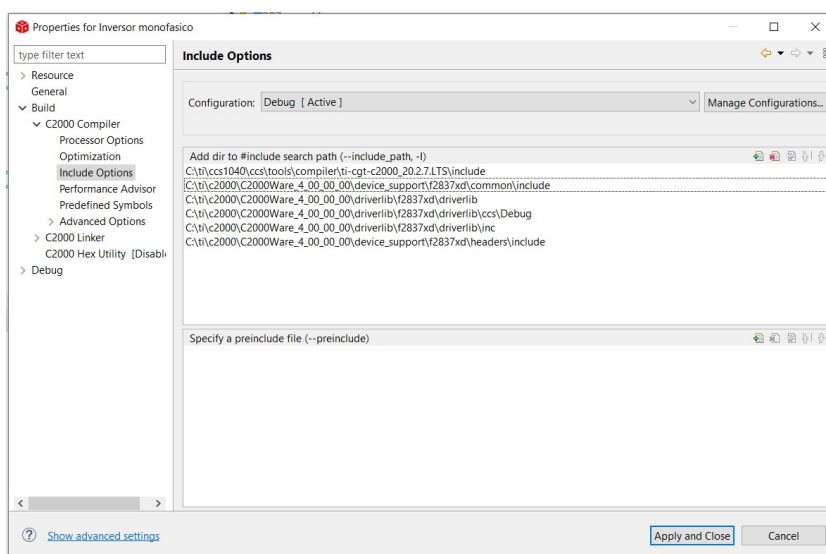


FIGURA 76: New CCS Project **Fuente: Autor**

- 2. Se deben configurar algunas especificaciones de construcción del proyecto, para ello se da click derecho en el proyecto, se selecciona propiedades y en esta ventana se debe buscar la opción "Processor Options" donde se deben completar las especificaciones como se observa en la figura 76

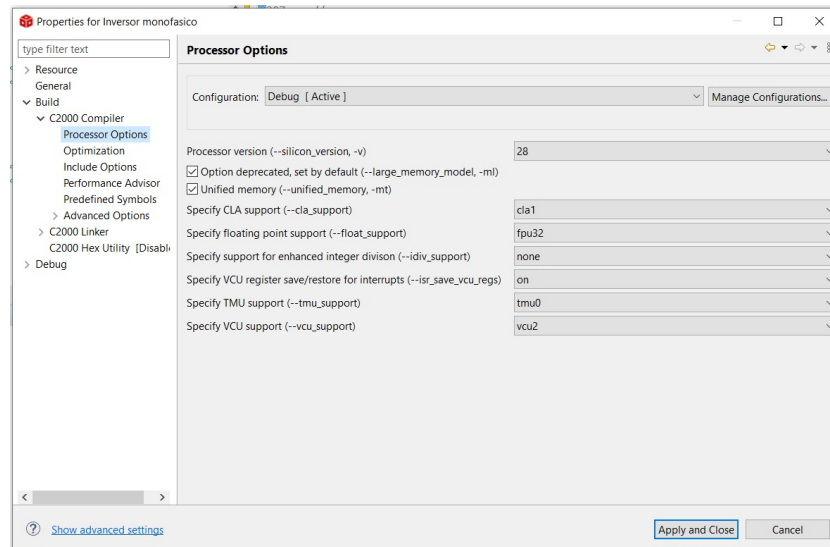


FIGURA 77: Processor Options Fuente: Autor

- 3. En la pestaña de C2000 Compiler se selecciona “Include Options”, en esta ventana se deben agregar las rutas de archivos que se presentan en la figura 81 los cuales permiten trabajar con los dos modos de operación.

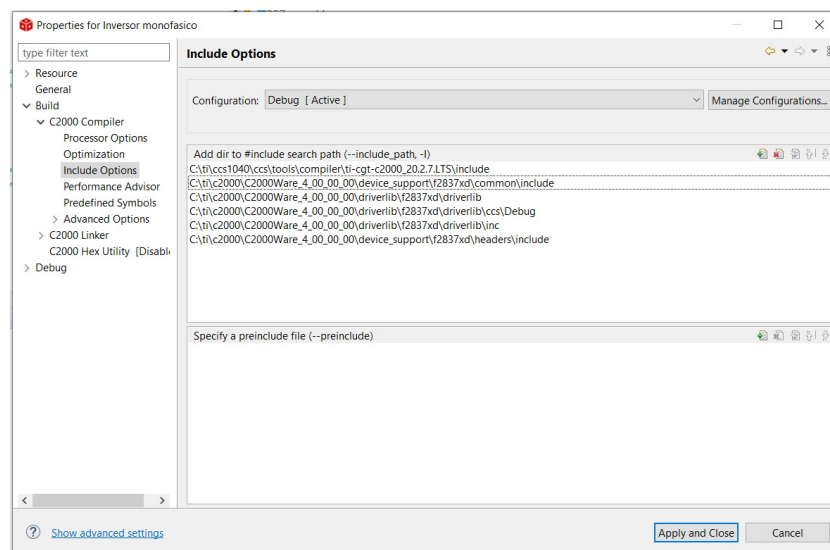


FIGURA 78: Include Options Fuente: Autor

- 4. Se debe expandir la opción “Advanced options” y buscar la opción “Predefined symbol”, en él se deben agregar los símbolos predefinidos DUAL_HEADERS, CPU1 para programar en los dos modos de operación y si se usa Launchpad se debe agregar el símbolo predefinido _"LAUNCHXLF28379D_ con el fin de configurar el reloj adecuado.

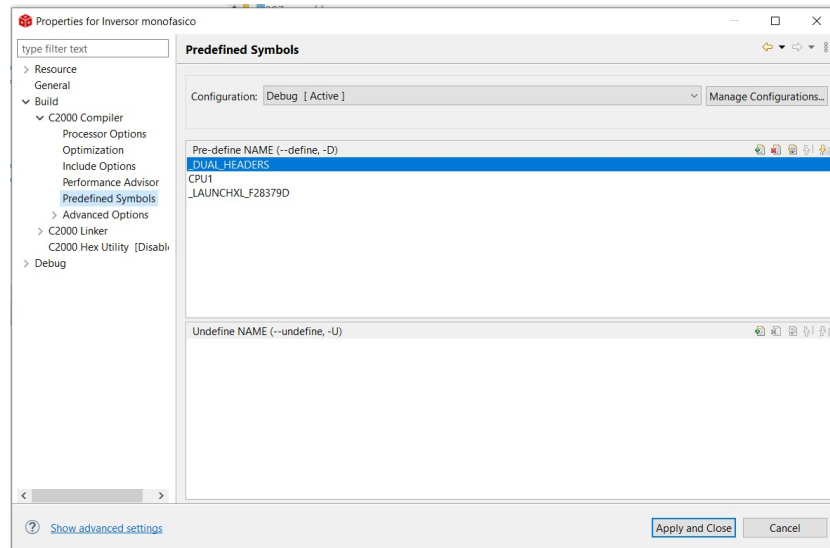


FIGURA 79: Predefined symbol Fuente: Autor

- 5. En la pestaña C2000 Linker se busca “File Search Path” donde se deben agregar los siguientes archivos: rts2800_fpu.lib, 2837xD_RAM_Ink_cpu1.cmd, common\cmd, headers\cmd y F2837xD-Headers_nonBios. Se debe eliminar libc.a, porque se usara rts2800_fpu.lib como biblioteca para el soporte de ejecución.

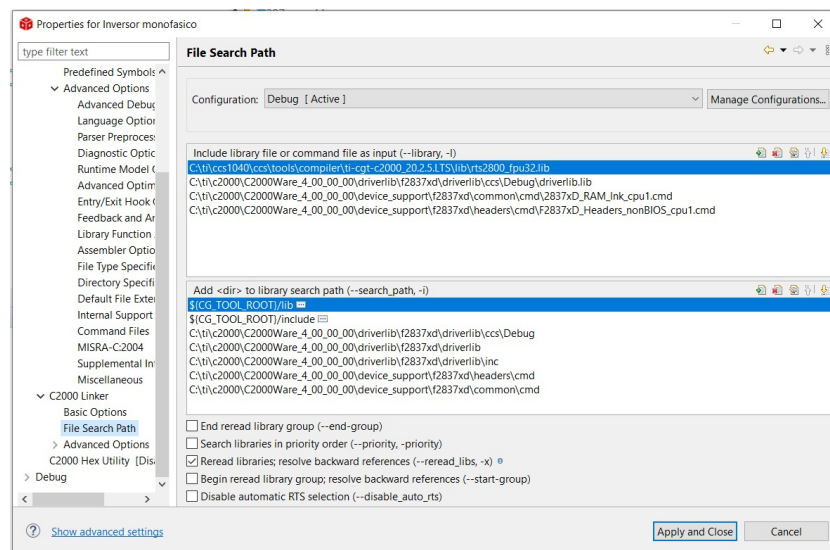


FIGURA 80: Predefined symbol Fuente: Autor

- 6. Mientras este en la pestaña de C2000 Linker se debe seleccionar en “Advanced options” en el cual se deben desplegar opciones donde una de ellas es “Symbol Management”. Donde se debe especificar que el punto de inicio del programa va a ser `code_start` como se observa a continuación y con esto se finalizan todas las configuraciones.

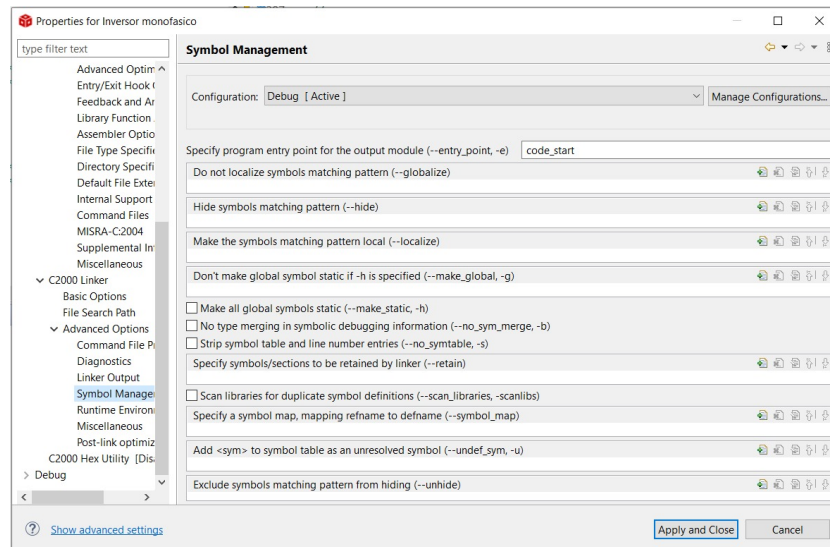


FIGURA 81: Predefined symbol **Fuente: Autor**

- 7. Por ultimo se deben vincular algunos archivos que se utilizan como `headers_files`, para esto se debe dar click derecho en el proyecto y seleccionar agregar archivos. Los archivos que se deben vincular son los siguientes:

- `common\source\F2837xD_CodeStartBranch.asm`
- `common\source\F2837xD_usDelay.asm`
- `common\source\F2837xD_SysCtrl.c`
- `common\source\F2837xD_Gpio.c`
- `common\source\F2837xD_Ipc.c`

NOTA: En el manual técnico de referencia al final de cada capítulo se encuentran las librerías

9.2. Manejo de las librerías

Las librerías son archivos ".c" que tienen vinculados archivos de cabecera, estos son principalmente "driverlib.h" "device.h". Estas librerías presentan un ejemplo base de cada uno de los periféricos que se desea configurar. A continuación se presenta una breve descripción de la configuración del ADC, el PWM, el DAC y los comparadores que fueron base en el desarrollo del proyecto.

9.2.1. Configuración del ePWM Trigger

El periférico (ePWM) Modulador de ancho de pulso mejorado es un elemento para el control de sistemas electrónicos que se encuentran en equipos industriales o comerciales; especialmente este periférico tiene un uso en control de motores, fuentes de alimentación conmutada y conversión de energía.

El ePWM se configura con la idea de lograr una única conversión en el ADCINx cuando el temporizador del ePWM alcance su período establecido en el TBPRD ,para esto, se debe configurar con alguno de los 16 SOC (Start Of Conversion) que Se observan en la figura 82.

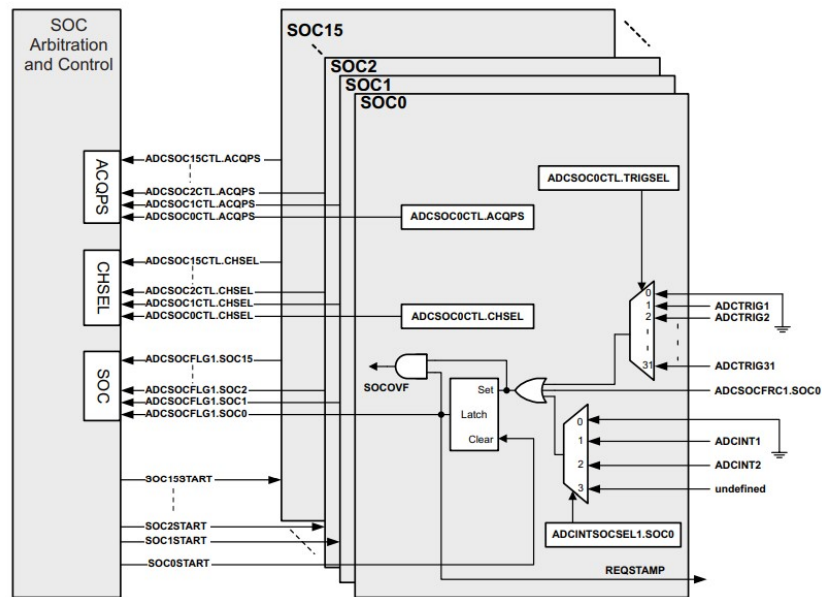


FIGURA 82: Diagrama de bloques SOC Fuente: Autor

La activación del ADC y la secuencia de activación se logran haciendo la correcta configuración del SOC, cada SOC es un conjunto de herramientas que definen como se va a realizar la conversión. A continuación se presenta la configuración que se realizó para el ePWM Trigger.

```
void initEPWM2(void)
{
    // Configuración del SOC para que se produzca en el primer evento de recuento

    EPWM_disableADCTrigger(EPWM8_BASE, EPWM_SOC_A);

    //En este caso se utiliza el ePWM8 para iniciar la conversión cuando el contador de este llegue a 0

    EPWM_setADCTriggerSource(EPWM8_BASE, EPWM_SOC_A, EPWM_SOC_TBCTR_ZERO);

    EPWM_setADCTriggerEventPrescale(EPWM8_BASE, EPWM_SOC_A, 1);

    //Se le asigna el valor del TBPRD en 1000 al PWM8

    EPWM_setTimeBasePeriod(EPWM8_BASE, 1000);

    //Se le asigna el valor a los prescalizadores en 1 dado que no se necesita escalar el valor de recuento

    EPWM_setClockPrescaler(EPWM8_BASE, EPWM_CLOCK_DIVIDER_1, EPWM_HSCLOCK_DIVIDER_1);

    //El conteo del TBPRD va hacer en modo arriba - abajo

    EPWM_setTimeBaseCounterMode(EPWM8_BASE, EPWM_COUNTER_MODE_UP_DOWN);
}
```

Para el valor que se le asigna a la función `EPWM_setTimeBasePeriod()` se tiene en cuenta la siguiente ecuación, en la cual se usa la frecuencia de muestreo previamente establecida en 50Khz.

$$TBPRD = \frac{1}{2} * \left(\frac{Fsysclockout}{Fpwm * clkdiv * hspclkdiv} \right) \quad (9.1)$$

- `Fsysclockout`: Frecuencia de reloj a la que trabaja el pwm 100Mhz
- `Fpwm`: Frecuencia a la cual se quiere realizar el muestreo en nuestro caso se utiliza 50Khz
- `clkdiv`: Valor del divisor bajo que se establece en 1
- `hspclkdiv`: Valor del divisor alto que se establece en 1

La ecuación da como resultado el valor asignado al `TBPRD`

$$TBPRD = \frac{1}{2} * \left(\frac{100Mhz}{50Mhz * 1 * 1} \right) = 1000 \quad (9.2)$$

9.2.2. Configuración del ADC

El periférico ADC permite la conversión analógica - digital con resolución seleccionable de 16 bits (modo diferencial) o 12 bits (modo singular).Este dispositivo cuenta con 15 canales de adquisición como se observa en la figura 83 y cada uno de estos módulos consta de un solo circuito de muestreo y retención.Este modulo esta diseñado con la posibilidad de ser duplicado en el mismo chip lo que permite un muestreo simultáneo o el uso de ADC independientes.

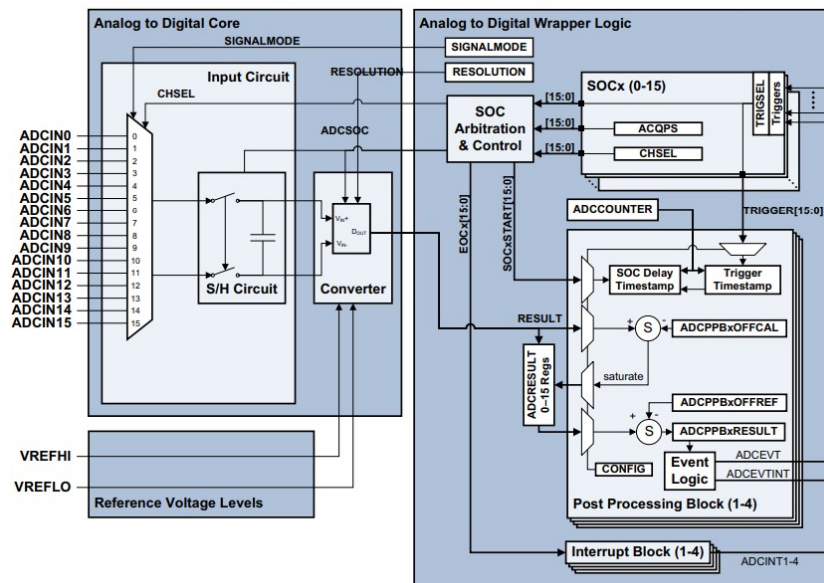


FIGURA 83: Diagrama de bloques ADC Fuente: Autor

En el programa se configura el `ADCA` y el `ADCC`, para los dos se establece la misma inicialización de la siguiente forma:

```

void initADC(void)
{
    //División del reloj de la dsp en 4 puesto que el conversor análogo - digital no funciona a una

    ADC_setPrescaler(ADCA_BASE, ADC_CLK_DIV_4_0);

    //Se configura la resolución del ADC y el modo en que se va a utilizar

    ADC_setMode(ADCA_BASE, ADC_RESOLUTION_16BIT, ADC_MODE_DIFFERENTIAL);

    //Esta función configura el pulso que da señal del fin de la conversión generado en el ADC, este

    ADC_setInterruptPulseMode(ADCA_BASE, ADC_PULSE_END_OF_CONV);

    //Habilita la conversión en el ADC seleccionado

    ADC_enableConverter(ADCA_BASE);
}

```

Para tener control de la planta se realiza la siguiente configuración de todos los sensores que se tienen ella, tanto de voltaje como de corriente.

Esta función configura el soc que se va utilizar, el pwm que va a realizar el disparo, en que modo se realiza la adquisición (modo diferencial) y la venta de adquisición que sera de 64

```

void initADCSOC(void)
{
    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER0, ADC_TRIGGER_EPWM8_SOCA,
                ADC_CH_ADCIN0_ADCIN1, 64);
    //Channel V_U

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER1, ADC_TRIGGER_EPWM8_SOCA,
                ADC_CH_ADCIN14_ADCIN15, 64);
    //Channel V_V

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER2, ADC_TRIGGER_EPWM8_SOCA,
                ADC_CH_ADCIN2_ADCIN3, 64);
    //Channel I_U

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER3, ADC_TRIGGER_EPWM8_SOCA,
                ADC_CH_ADCIN4_ADCIN5, 64);
    //Channel I_V

    ADC_setupSOC(ADCC_BASE, ADC_SOC_NUMBER4, ADC_TRIGGER_EPWM8_SOCA,
                ADC_CH_ADCIN2_ADCIN3, 64);
    //Channel I_Load
}

```

9.2.3. Configuración de comparadores

Este dispositivo cuenta con un subsistema de comparadores (CMPSS) como se observa en la figura 84 estos son comparadores analógicos que son útiles en aplicaciones de control, en sistemas conmutados, en monitoreo de disparo de voltaje, entre otros. Este módulo con el fin de ver la posibilidad de utilizar este submódulo en el control deslizante.

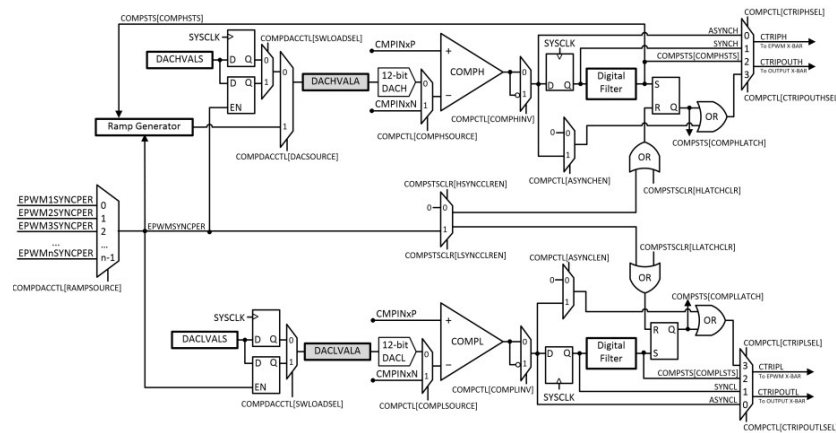


FIGURA 84: Diagrama de bloques CMPSS Fuente: Autor

Este subsistema se construye alrededor de varios módulos, cada uno de estos se conforma de dos entradas en configuración DAC (conversión digital - analógica) de 12 bits y dos filtros digitales para cada una de las salidas de los comparadores denotadas como "H" (high) y "L" (low). También la entrada positiva del comparador se activa desde un pin externo y la entrada negativa se puede controlar por el DAC de 12 bits o también por un pin externo; cabe aclarar que a la salida de cada comparador se puede utilizar un filtro digital si así se desea tener una señal filtrada. Así mismo cada comparador genera una salida digital indicando si su voltaje en la entrada positiva es mayor que en la entrada negativa como se observa en la figura 85.

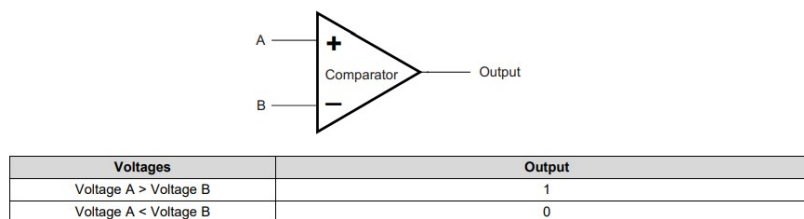


FIGURA 85: Diagrama del comparador Fuente: Autor

. Para activar el submódulo de comparadores se realiza la siguiente configuración

En el main del programa se debe enrutar los pines para que sean las salidas del comparador con el X_BAR correspondiente al modulo de comparadores que se va a utilizar

```
GPIO_setPinConfig(GPIO_14_OUTPUTXBAR3);
GPIO_setPinConfig(GPIO_15_OUTPUTXBAR4);
```

En una función se debe configurar los siguientes parámetros

```
void initCMPSS (void)
{
```

```

//Se establece el modulo 4 de comparadores
CMPSS_enableModule(CMPSS4_BASE);

//Se configura una de las entradas del comparador tanto el alto como el bajo que sea
//controlado por el DAC

CMPSS_configHighComparator(CMPSS4_BASE, CMPSS_INSRC_DAC);
CMPSS_configLowComparator(CMPSS4_BASE, CMPSS_INSRC_DAC);

//Se realiza las adecuaciones para utilizar el DAC en una de las entradas

CMPSS_configDAC(CMPSS4_BASE, CMPSS_DACREF_VDDA | CMPSS_DACSRC_SHDW);

// De forma digital se debe establecer a que valor se va a realizar la comparación
// para los dos comparadores

CMPSS_setDACValueHigh(CMPSS4_BASE, 2098);
CMPSS_setDACValueLow(CMPSS4_BASE, 1998);

//Se inicializa el filtro de salida

CMPSS_initFilterHigh(CMPSS4_BASE);
CMPSS_initFilterLow(CMPSS4_BASE);

//Para el filtro se debe disponer que sea asincrono, y la salida del comparador bajo sea invertida

CMPSS_configOutputsHigh(CMPSS4_BASE, CMPSS_TRIP_ASYNC_COMP |
                        CMPSS_TRIPOUT_ASYNC_COMP);

CMPSS_configOutputsLow(CMPSS4_BASE, CMPSS_TRIP_ASYNC_COMP |
                      CMPSS_TRIPOUT_ASYNC_COMP | CMPSS_INV_INVERTED);

//Se termina de enrutar la salida de los Gpio 14 y 15 como las salidas de los comparadores
//del modulo 4

XBAR_setOutputMuxConfig(XBAR_OUTPUT3, XBAR_OUT_MUX06_CMPSS4_CTRIPOUTH );

XBAR_enableOutputMux(XBAR_OUTPUT3, XBAR_MUX06);

XBAR_setOutputMuxConfig(XBAR_OUTPUT4, XBAR_OUT_MUX07_CMPSS4_CTRIPOUTL );

XBAR_enableOutputMux(XBAR_OUTPUT4, XBAR_MUX07);
}

```

Capítulo 10

Conclusiones y Trabajos Futuros

10.1. Conclusiones

- La elección de la técnica de control depende de las necesidades específicas del sistema, y cada una de las técnicas: Control PWM, Control MPC, Control PID y Control Sliding, presenta características únicas que las hacen adecuadas para diferentes situaciones. El Control PWM se destaca por su simplicidad y eficiencia en aplicaciones de potencia, mientras que el Control PID ofrece versatilidad y aplicabilidad en una amplia variedad de sistemas. El Control MPC proporciona precisión y es idóneo para sistemas complejos, aunque su implementación puede ser más exigente. Por otro lado, el Control Sliding sobresale por su capacidad de adaptación en sistemas altamente no lineales sin necesidad de un modelo preciso del sistema.
- Los resultados experimentales parten del modelo matemático y de la predicción del funcionamiento del sistema en la simulación. Sin embargo, se observa que los resultados en la implementación práctica pueden variar debido a factores como el ruido, las características específicas de los dispositivos utilizados en la planta, la adquisición de las señales, entre otros.
- El control deslizante ha demostrado ser una estrategia de control efectiva para aplicaciones en inversores de potencia. Esto se evidencia al mantener la corriente de salida dentro de una superficie de deslizamiento deseada como se observa en la figura 70 y en la figura 71, este control garantizó un seguimiento preciso de la referencia y aseguró el cumplimiento de la potencia objetivo de 200 vatios, incluso al presentar perturbaciones. Su simplicidad a la hora de la implementación lo convierte en una técnica de control llamativa y que sea utilizada en diferentes aplicaciones.
- El controlador PI en coordenadas dq ofrece una buena estabilidad frente a perturbaciones y no presenta error de estado estacionario, como se presenta en la figura 73. Sin embargo, es importante mencionar que este control requiere de altas tensiones en la entrada del sistema, puesto que, si se tiene una corriente de referencia muy alta, se requiere de más voltaje para que la señal a controlar siga la referencia. Esto se comprobó cuando se establece como entrada 440 voltios, y no fue suficiente para que la corriente de la bobina (señal controlada) alcanzara el valor de la referencia dado por la carga, como se aprecia en la figura 74. Por ende, el control de modo deslizante tiene como ventaja que no necesita de altas tensiones en su entrada, dado que con 380 voltios la señal controlada siempre sigue la referencia, a diferencia del compensador PI que tiene como limitación el voltaje de entrada.

- En el control sliding y en el control PI se pudo observar que cuando el sistema se enciende y no cuenta con la capacidad de corriente suficiente para abastecer la carga, recibe asistencia de la red eléctrica, esta asistencia es muy baja en comparación a lo que aporta el inversor.
- La implementación tanto del control de modo deslizante como del control proporcional-integral (PI) aporta notables beneficios en términos de eficiencia energética en una residencia. Ambos enfoques permiten al inversor generar una potencia muy cercana a la que consume la carga conectada a la red eléctrica, lo que se traduce en una reducción del consumo de energía en el hogar. Sin embargo, es importante destacar que entre las dos estrategias de control implementadas, se observó que el sistema con el compensador PI aporta ligeramente más potencia a la carga con respecto al control de modo deslizante, pero esto solo se evidencia cuando la corriente de la carga no es muy alta, puesto que de ser el control de modo deslizante tiene un mejor aporte.
- El hardware presentado en la sección 3 tuvo un origen inicial en otro proyecto, lo que planteó ciertas dificultades, como la necesidad de comprender los planos, distinguir entre las etapas de potencia y las etapas digitales, así como comprender las conexiones y el funcionamiento. Para adaptar tanto el control Sliding como el control PI del inversor, fue necesario incorporar una tarjeta de sensores para el sensado de las señales. Esto se debió a la necesidad de garantizar una calibración igual en la adquisición de señales, ya que no se disponía de un sensor independiente en la tarjeta para medir la corriente de referencia. Además, para generar la conmutación de la carga, se implementó un puente en H y se añadió un relé externo al hardware.

10.2. Trabajos Futuros

- La implementación del control sliding en un inversor monofásico abre la puerta a futuros desarrollos. Uno de los trabajos futuros sería la adaptación de esta técnica para su implementación en un inversor trifásico. Esto permitiría mayor versatilidad en el manejo y conversión de la energía eléctrica, lo que podría tener aplicaciones relevantes en sistemas de energía más complejos.
- Se adjuntan los planos tanto de la etapa de potencia como de la etapa digital de la tarjeta, con el propósito de facilitar futuras mejoras en la misma teniendo en consideración el tipo de sistema en el que se planea realizar la implementación sobre esta.
- Es importante tener en cuenta que, en este trabajo, se utilizó una fuente de corriente continua (DC) como fuente primaria. No obstante, en futuros trabajos se debe contemplar la posibilidad de diseñar el bus DC a partir de una fuente de energía renovable, lo que contribuiría a una mayor sostenibilidad y eficiencia del sistema.

ANEXOS

10.3. Script Matlab

```
%Elementos del sistema
L = 27e-3
L = 0.0270
R = 70
R = 70
C = 2e-6
C = 2.0000e-06
w = 2*pi*60
w = 376.9911
Vs = 420
Vs = 420

%%Matrices para determinar la funcion de transferencia
A = [0 -1/L w 0; 1/C -1/(R*C) 0 w; -w 0 0 -1/L; 0 -w 1/C -1/(R*C)]
A = 4x4
10^5 x
      0    -0.0004    0.0038      0
    5.0000   -0.0714      0    0.0038
   -0.0038      0      0   -0.0004
      0   -0.0038    5.0000   -0.0714

B = [Vs/L; 0; 0; 0]
B = 4x1
10^4 x
    1.5556
      0
      0
      0

Bq = [0; 0; Vs/L; 0]
Bq = 4x1
10^4 x
      0
      0
    1.5556
      0

Cd = [1 0 0 0]
Cd = 1x4
      1      0      0      0

Cq = [0 0 1 0]
```

FIGURA 86: Pagina 1 código en Matlab para el diseño del compensador PI.


```

Cq = 1x4
      0      0      1      0

%funcion de transferencia Id/Dd
[num_d, den_d] = ss2tf(A,B,Cd,0)

num_d = 1x5
1015 x
      0      0.0000      0.0000      0.0011      2.0576
den_d = 1x5
1014 x
      0.0000      0.0000      0.0000      0.0027      3.4494

G_d = tf(num_d,den_d)

G_d =

      1.556e04 s^3 + 2.222e08 s^2 + 1.084e12 s + 2.058e15
-----
      s^4 + 1.429e04 s^3 + 8.834e07 s^2 + 2.666e11 s + 3.449e14

Continuous-time transfer function.
Model Properties

%funcion de transferencia Iq/Dd
[num_q, den_q] = ss2tf(A,Bq,Cq,0)

num_q = 1x5
1015 x
      0      0.0000      0.0000      0.0011      2.0576
den_q = 1x5
1014 x
      0.0000      0.0000      0.0000      0.0027      3.4494

G_q = tf(num_q,den_q)

G_q =

      1.556e04 s^3 + 2.222e08 s^2 + 1.084e12 s + 2.058e15
-----
      s^4 + 1.429e04 s^3 + 8.834e07 s^2 + 2.666e11 s + 3.449e14

Continuous-time transfer function.
Model Properties

%Ganancias seleccionadas para el control de cada funcion de transferencia
Ts = 1/50000

Ts = 2.0000e-05

sysd = c2d(G_d,Ts,'tustin')

sysd =

      0.1553 z^4 - 0.2681 z^3 - 0.03847 z^2 + 0.2683 z - 0.1167

```

FIGURA 87: Pagina 2 código en Matlab para el diseño del compensador PI.

```
-----  
      z^4 - 3.72 z^3 + 5.193 z^2 - 3.225 z + 0.7515  
  
Sample time: 2e-05 seconds  
Discrete-time transfer function.  
Model Properties
```

```
%%sisotool(sysd)
```

FIGURA 88: Pagina 3 código en Matlab para el diseño del compensador PI.

10.4. Codigo implementado en el control Sliding

```

#include "Six_Drivers.h"
#include "F28x_Project.h"
#include "F2837xD_device.h"
#include "Inversor.h"
#include "driverlib.h"
#include "device.h"
#include "stdio.h"
// #include "SCI_UART.c"

//
// Defines
//
#define Offset                32580
#define RESULTS_BUFFER_SIZE  833
#define EX_ADC_RESOLUTION    16
#define ISR_FREQUENCY        50000
#define GRID_FREQ            60
#define PI                   3.1416
#define Kml                   5.14e-3
int cont = 0;
uint16_t adcAResults[RESULTS_BUFFER_SIZE]; // Buffer for results
uint16_t State_ON;
uint16_t limiteSuperior;
uint16_t limiteInferior;
char *msg; // Index into result buffer
float K = 10;
// BUFFER
float Buff1[RESULTS_BUFFER_SIZE];
float Buff2[RESULTS_BUFFER_SIZE];
float BuffDelay[RESULTS_BUFFER_SIZE];

// Function Prototypes
//
void initADC(void);
void initADCSOC(void);
void ConfigGPIO(void);
void Cfg_ON_INT(void);
void initCPUTimers(void);
void configCPU_TIMER(uint32_t, float, float);
__interrupt void adcAISR(void);
__interrupt void ON_Sys_isr(void);
__interrupt void OFF_Sys_isr(void);
__interrupt void RDY_Drivers_isr(void);
__interrupt void Fault_Drivers_isr(void);
__interrupt void cpuTimer0ISR(void);
void setupEPWMActiveHighComplementary(uint32_t base);
void setupEPWMActiveHigh(uint32_t base);
void initEPWM2(void);

```

```

void Config_UART_GPIO(void);
void scia_echoback_init(void);

uint16_t adcAResult0;
uint16_t adcAResult1;
uint16_t cpuTimer0IntCount;
static uint16_t A;
static uint16_t B;

//
// Main
//
void main(void)
{
    //
    // Initialize device clock and peripherals//Inicializar el reloj del dispositivo y los periféricos
    //
    Device_init();
    //GPIO_SetupPinMux(42, GPIO_MUX_CPU1, 15);
    //
    // Disable pin locks and enable internal pullups.//Desactiva los bloqueos de los pines y activa pullups
    //
    Device_initGPIO();
//    EALLOW;
//    ClkCfgRegs.LOSPCP.all = 0;    // Divide por 1 sysclkout for SCI at 115200
//    EDIS;

    Config_UART_GPIO();
    scia_echoback_init();    // Initialize SCI for echoback

    msg = "\r\f!!! HELLO, THIS IS A INVERSOR !!! \n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    //
    // Initialize PIE and clear PIE registers. Disables CPU interrupts.
    //
    Interrupt_initModule();

    //
    // Initialize the PIE vector table with pointers to the shell Interrupt
    // Service Routines (ISR).
    //
    //InitPieVectTable();

    Interrupt_initVectorTable();

    //

```

```

// Interrupts that are used in this example are re-mapped to ISR functions
// found within this file. // Las interrupciones que se utilizan en este ejemplo se reasignan
//

Interrupt_register(INT_TIMER0, &cpuTimer0ISR);
Interrupt_register(INT_ADCA1, &adcA1ISR);
Interrupt_register(INT_XINT1, &RDY_Drivers_isr);
Interrupt_register(INT_XINT2, &Fault_Drivers_isr);
Interrupt_register(INT_XINT3, &ON_Sys_isr);
Interrupt_register(INT_XINT4, &OFF_Sys_isr);

initCPUTimers();

//
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 1 second Period (in uSeconds)
//
configCPUTimer(CPUTIMER0_BASE, DEVICE_SYSCCLK_FREQ, 2);
//
// To ensure precise timing, use write-only instructions to write to the
// entire register. Therefore, if any of the configuration bits are changed
// in configCPUTimer and initCPUTimers, the below settings must also
// be updated.
//
CPUTimer_enableInterrupt(CPUTIMER0_BASE);

//Encoder_sys();
Sys_Flags.all = 0;
//
// Set up the ADC and the ePWM and initialize the SOC//Configurar el ADC y el ePWM e inicializar
//

initADC();
Cfg_SWs_GPIO();
Cfg_Leds_GPIO();
Cfg_Relay_GPIO();
Cfg_2E020I12_A_GPIOs();
Cfg_Addrs_GPIOs();
Cfg_RSTs_GPIO();
IC2E020I12_RDYs_OK();
ConfigGPIO();
initEPWM2();
initADCSOC();

//
// Initialize PLL singles phase
//
//

```

```

// Initialize results buffer
//
//
// Enable ADC interrupt
//

Interrupt_enable(INT_ADCA1);
Interrupt_enable(INT_XINT1);
Interrupt_enable(INT_XINT2);
Interrupt_enable(INT_XINT3);
Interrupt_enable(INT_XINT4);

//
// Set up COMP1H
//

//
// Configure GPIO14 to output CTRIPOUT1H (routed through XBAROUTPUT3)
//
GPIO_setPinConfig(GPIO_14_OUTPUTXBAR3);
GPIO_setPinConfig(GPIO_15_OUTPUTXBAR4);
GPIO_setPinConfig(GPIO_29_OUTPUTXBAR6);
GPIO_setPinConfig(GPIO_16_OUTPUTXBAR7);

//
// Enable Global Interrupt (INTM) and realtime interrupt (DBGM)//Activar la interrupción global
//
EINT;
ERTM;

//
// Loop indefinitely//Bucle indefinido
//

//GPIO_setPadConfig(42, GPIO_PIN_TYPE_STD);
Turn_Sys_OFF();

Turn_Slow_Charge_OFF();
EPWM_enableADCTrigger(EPWM8_BASE, EPWM_SOC_A);
EPWM_setTimeBaseCounterMode(EPWM8_BASE, EPWM_COUNTER_MODE_UP_DOWN);
while(1);

}

/*****
/*****
__interrupt void adcA1ISR(void)
{

```

```

static uint16_t index = 0;
static uint16_t indexdelay = 0;
static float32 Voltage;
static float32 CurrentMeaLoad;;

```

```

/////////////////////////////////CONVERSIÓN/////////////////////////////////
Voltage = (float32) (65535-(ADC_readResult(ADCARESLT_BASE, ADC_SOC_NUMBER0)));

```

```

CurrentMeaLoad = (float32) (65535-(ADC_readResult(ADCCRESULT_BASE, ADC_SOC_NUMBER4)));
limiteSuperior = Offset + 15;
limiteInferior = Offset - 15;

```

```

if(CurrentMeaLoad > limiteSuperior)
{
    //32202

    //32776
    if(A == 1)
    {

    }
    else
    {
        A = 1;
        B = 0;

        GPIO_WritePin(160, 1);
        Interrupt_enable(INT_TIMER0);
        CPUTimer_reloadTimerCounter(CPUTIMER0_BASE);
        CPUTimer_startTimer(CPUTIMER0_BASE);
        GPIO_WritePin(0, 0);
        GPIO_WritePin(1, 0);
    }
}
else
{
    if(CurrentMeaLoad < limiteInferior)
    {
        //32198

        //32768
        if(B == 1)
        {

        }
        else
        {

```

```

        B = 1;
        A = 0;

        GPIO_WritePin(160, 1);
        Interrupt_enable(INT_TIMER0);
        CPUTimer_reloadTimerCounter(CPUTIMER0_BASE);
        CPUTimer_startTimer(CPUTIMER0_BASE);
        GPIO_WritePin(0, 0);
        GPIO_WritePin(1, 0);
    }
}

}

Buff1[index] = CurrentMeaLoad;

indexdelay = index + (RESULTS_BUFFER_SIZE/4);
if(indexdelay >= RESULTS_BUFFER_SIZE)
    indexdelay = indexdelay - RESULTS_BUFFER_SIZE;
//BuffDelay[indexdelay] = splll.AC_input;

if(++index >= RESULTS_BUFFER_SIZE)
    index = 0;

// Clear the interrupt flag
//
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
//
// Check if overflow has occurred
//
if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
{
    ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
//
// Acknowledge the interrupt
//
Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP1);
}

/*****
/*****
__interrupt void cpuTimer0ISR(void)
{

    cpuTimer0IntCount++;

    if(A == 1)

```

```

{

    GPIO_WritePin(0, 0);
    GPIO_WritePin(1, 1);
}
else
{

    GPIO_WritePin(0, 1);
    GPIO_WritePin(1, 0);
}

Interrupt_disable(INT_TIMER0);
CPUTimer_stopTimer(CPUTIMER0_BASE);

//
// Acknowledge this interrupt to receive more interrupts from group 1
//
Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP1);
}
/*****
/*****
__interrupt void ON_Sys_isr(void)
{
    Turn_Sys_ON();
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP12);
    //PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
}
/*****
/*****
__interrupt void OFF_Sys_isr(void)
{
    Turn_Sys_OFF();
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP12);
}
/*****
/*****
__interrupt void RDY_Drivers_isr(void)
{
//    msg = "\r\n- VOLTAGE 2ED020I12 (RDYs) FAILs!!! \n\0";
//    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
//    DELAY_US(10000);
//    Sys_Flags.bit.FAIL_RDYs = 1;                //Source Register Flag
//    GPIO_disableInterrupt(GPIO_INT_XINT1);
//    GPIO_disableInterrupt(GPIO_INT_XINT2);
//
//    //Source Fault Identification Sequence

```

```

//    IC2E020I12_RDYS_OK();
//    Turn_Sys_OFF();
//    GPIO_enableInterrupt (GPIO_INT_XINT1);
//    GPIO_enableInterrupt (GPIO_INT_XINT2);
//    Interrupt_clearACKGroup (INTERRUPT_ACK_GROUP1);
}
/*****
*****/
__interrupt void Fault_Drivers_isr(void)
{
    msg = "\r\n- Fail DESATURATION 2ED020I12 !!! \n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    Sys_Flags.bit.FAIL_FLTs = 1;                //Fail Register Flag
    GPIO_disableInterrupt (GPIO_INT_XINT2);      //disable FLT Interrupt
    GPIO_disableInterrupt (GPIO_INT_XINT1);      //disable RDY Interrupt
    //Fault Identification Sequence
    IC2E020I12_FLT_RST();
    GPIO_enableInterrupt (GPIO_INT_XINT2);      //enable FLT Interrupt
    GPIO_enableInterrupt (GPIO_INT_XINT1);      //enable RDY Interrupt
    Interrupt_clearACKGroup (INTERRUPT_ACK_GROUP1);
}
/*****
*****/
void initADC(void)
{
    //
    // Set ADCDLK divider to /4 //Poner el divisor ADCDLK a /4
    //
    ADC_setPrescaler(ADCA_BASE, ADC_CLK_DIV_4_0);
    ADC_setPrescaler(ADCC_BASE, ADC_CLK_DIV_4_0);
    //
    // Set resolution and signal mode (see #defines above) and load
    // corresponding trims. // Establecer la resolución y el modo de señal (ver #defines arriba) y c
    //
    #if(EX_ADC_RESOLUTION == 12)
        ADC_setMode(ADCA_BASE, ADC_RESOLUTION_12BIT, ADC_MODE_SINGLE_ENDED);
    #elif(EX_ADC_RESOLUTION == 16)
        ADC_setMode(ADCA_BASE, ADC_RESOLUTION_16BIT, ADC_MODE_DIFFERENTIAL);
        ADC_setMode(ADCC_BASE, ADC_RESOLUTION_16BIT, ADC_MODE_DIFFERENTIAL);
    #endif
    //
    // Set pulse positions to late // Ajustar las posiciones de los pulsos a la tarde
    //
    ADC_setInterruptPulseMode(ADCA_BASE, ADC_PULSE_END_OF_CONV);
    ADC_setInterruptPulseMode(ADCC_BASE, ADC_PULSE_END_OF_CONV);
    //
    // Power up the ADC and then delay for 1 ms //Encender el ADC y retardar 1 ms
    //
    ADC_enableConverter(ADCA_BASE);

```

```

    ADC_enableConverter(ADCC_BASE);
    DEVICE_DELAY_US(1000);
}

/*****
*****/

void initEPWM2(void)
{
    // Configuración del SOC para que se produzca en el primer evento de recuento
    EPWM_disableADCTrigger(EPWM8_BASE, EPWM_SOC_A);

    //En este caso se utiliza el ePWM8 para iniciar la conversión cuando el contador de este llegue a 1000

    EPWM_setADCTriggerSource(EPWM8_BASE, EPWM_SOC_A, EPWM_SOC_TBCTR_ZERO);
    EPWM_setADCTriggerEventPrescale(EPWM8_BASE, EPWM_SOC_A, 1);

    //Se le asigna el periodo de conteo en 1000 al PWM8

    EPWM_setTimeBasePeriod(EPWM8_BASE, 1000);

    EPWM_setClockPrescaler(EPWM8_BASE, EPWM_CLOCK_DIVIDER_1, EPWM_HSCLOCK_DIVIDER_1);

    //El conteo del TBPRD va hacer en modo arriba - abajo

    EPWM_setTimeBaseCounterMode(EPWM8_BASE, EPWM_COUNTER_MODE_UP_DOWN);

    EPWM_setActionQualifierAction(EPWM8_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_OUTPUT_HIGH, EPWM_AQ_OUTPUT_ON_T
    EPWM_setActionQualifierAction(EPWM8_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_OUTPUT_LOW, EPWM_AQ_OUTPUT_ON_T

}

/*****
*****/

void initADCSOC(void)
{
    //
    // Configure SOC0 of ADCA to convert pin A0. The EPWM1SOCA signal will be
    // the trigger.
    // - For 12-bit resolution, a sampling window of 15 (75 ns at a 200MHz
    //   SYSCLK rate) will be used. For 16-bit resolution, a sampling window
    //   of 64 (320 ns at a 200MHz SYSCLK rate) will be used.
    // - NOTE: A longer sampling window will be required if the ADC driving
    //   source is less than ideal (an ideal source would be a high bandwidth
    //   op-amp with a small series resistance). See TI application report
    //   SPRACT6 for guidance on ADC driver design.
    //

    #if(EX_ADC_RESOLUTION == 12)
        ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER0, ADC_TRIGGER_EPWM1_SOCA,

```

```

        ADC_CH_ADCIN0, 15);
#elif (EX_ADC_RESOLUTION == 16)
    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER0, ADC_TRIGGER_EPWM8_SOCA,
        ADC_CH_ADCIN0_ADCIN1, 64);
    msg = "\r\n- Channel A0 (V_V) trigger ePWM 8A\n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER1, ADC_TRIGGER_EPWM8_SOCA,
        ADC_CH_ADCIN14_ADCIN15, 64);
    msg = "\r\n- Channel A0 (V_U) trigger ePWM 8A\n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER2, ADC_TRIGGER_EPWM8_SOCA,
        ADC_CH_ADCIN2_ADCIN3, 64);
    msg = "\r\n- Channel A0 (I_U) trigger ePWM 8A\n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER3, ADC_TRIGGER_EPWM8_SOCA,
        ADC_CH_ADCIN4_ADCIN5, 64);
    msg = "\r\n- Channel A0 (I_V) trigger ePWM 8A\n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    ADC_setupSOC(ADCC_BASE, ADC_SOC_NUMBER4, ADC_TRIGGER_EPWM8_SOCA,
        ADC_CH_ADCIN2_ADCIN3, 64);
    msg = "\r\n- Channel A0 (I_Load) trigger ePWM 8A\n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

#endif
    //
    // Set SOC0 to set the interrupt 1 flag. Enable the interrupt and make
    // sure its flag is cleared.
    //
    ADC_setInterruptSource(ADCA_BASE, ADC_INT_NUMBER1, ADC_SOC_NUMBER0);
    ADC_enableInterrupt(ADCA_BASE, ADC_INT_NUMBER1);
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
/*****
/*****
void setupEPWMActiveHighComplementary(uint32_t base)
{
    //
    // Use EPWMA as the input for both RED and FED
    //
    EPWM_setRisingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);
    EPWM_setFallingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);

    //
    // Set the RED and FED values
    //

```

```

EPWM_setFallingEdgeDelayCount(base, 200);
EPWM_setRisingEdgeDelayCount(base, 200);

//
// Invert only the Falling Edge delayed output (AHC)
//
EPWM_setDeadBandDelayPolarity(base, EPWM_DB_RED, EPWM_DB_POLARITY_ACTIVE_HIGH);
EPWM_setDeadBandDelayPolarity(base, EPWM_DB_FED, EPWM_DB_POLARITY_ACTIVE_LOW);

//
// Use the delayed signals instead of the original signals
//
EPWM_setDeadBandDelayMode(base, EPWM_DB_RED, true);
EPWM_setDeadBandDelayMode(base, EPWM_DB_FED, true);

//
// DO NOT Switch Output A with Output B
//
EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_A, false);
//EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_B, false);

} /*****
/*****
void setupEPWMActiveHigh(uint32_t base)
{
    //
    // Use EPWMA as the input for both RED and FED
    //
    EPWM_setRisingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);
    EPWM_setFallingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);

    //
    // Set the RED and FED values
    //
    EPWM_setFallingEdgeDelayCount(base, 200);
    EPWM_setRisingEdgeDelayCount(base, 200);

    //
    // Do not invert the delayed outputs (AH)
    //
    EPWM_setDeadBandDelayPolarity(base, EPWM_DB_RED, EPWM_DB_POLARITY_ACTIVE_HIGH);
    EPWM_setDeadBandDelayPolarity(base, EPWM_DB_FED, EPWM_DB_POLARITY_ACTIVE_HIGH);

    //
    // Use the delayed signals instead of the original signals
    //
    EPWM_setDeadBandDelayMode(base, EPWM_DB_RED, true);
    EPWM_setDeadBandDelayMode(base, EPWM_DB_FED, true);

    //

```

```

// DO NOT Switch Output A with Output B
//
EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_A, false);
EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_B, false);

}

/*****
*****/

void IC2E020I12_RDYs_OK(void) // esta fubcion al correr el programa se usa para forzar los PWMs en b
{
    uint16_t Addr_RDY, Branch_RDY;

    Set_Addr_Bus(0);

    msg = "\r\n\n===== ";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    if(GPIO_readPin(RDY_A_Pin) == 0)
    {
        for(Addr_RDY = 1; Addr_RDY < 7; Addr_RDY++)
        {
            Set_Addr_Bus(Addr_RDY);
            if(GPIO_ReadPin(RDY_A_Pin) == 0)
            {
                if(Addr_RDY == 1 || Addr_RDY == 4)
                {
                    //EPWM_setActionQualifierContSWForceAction(EPWM1_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_S
                    //setupEPWMActiveHigh(EPWM1_BASE);
                    GPIO_writePin(160, 0);
                    GPIO_writePin(125, 0);
                }

                if(Addr_RDY < 4)
                {
                    Branch_RDY = Addr_RDY;
                    msg = "\r\n- A_RDY_H is BAD!!!\n\0";
                }
                else
                {
                    Branch_RDY = Addr_RDY-3;
                    msg = "\r\n- A_RDY_L is BAD!!!\n\0";
                }
                msg[11] = 0x30|Branch_RDY;
                SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
            }
        }
    }

    Set_Addr_Bus(0);

```

```

    DELAY_US(20);
}
/*****
/*****
void IC2E020I12_FLT_RST(void)
{
    uint16_t Addr_FLT, Branch_FLT;

    Set_Addr_Bus(0);

    msg = "\r\n\n===== ";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

    if(GPIO_readPin(FLT_A_Pin) == 0)
    {
        for(Addr_FLT = 1; Addr_FLT < 7; Addr_FLT++)
        {
            Set_Addr_Bus(Addr_FLT);
            if(GPIO_readPin(FLT_A_Pin) == 0)
            {
                if(Addr_FLT < 4)
                {
                    Branch_FLT = Addr_FLT;
                    msg = "\r\n- A_FLT_H is BAD!!!\n\0";
                }
                else
                {
                    Branch_FLT = Addr_FLT-3;
                    msg = "\r\n- A_FLT_L is BAD!!!\n\0";
                }
                msg[11] = 0x30|Branch_FLT;
                SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

                GPIO_writePin(RST_A_Pin, 1);
                DELAY_US(2);
                GPIO_writePin(RST_A_Pin, 0);
                DELAY_US(1);

                Turn_Sys_OFF();
            }
        }
    }
    Set_Addr_Bus(0);
    DELAY_US(20);
}
/*****
/*****
void Turn_Sys_ON(void)
{
    State_ON = 1;

```

```

//Change State Interrupts
GPIO_disableInterrupt (GPIO_INT_XINT3); //Interrupt_enable(INT_XINT3);
//XintRegs.XINT3CR.bit.ENABLE = 0;           // Disable ON Interrupt
GPIO_enableInterrupt (GPIO_INT_XINT4);
//XintRegs.XINT4CR.bit.ENABLE = 1;           // Enable OFF Interrupt
//Change State Indicators
GPIO_writePin(RST_EN_A_Pin, 1);
GPIO_writePin(LED_ON_OFF, LED_ON);
GPIO_writePin(LED_ERROR, LED_OFF);
//Change State PWM's
//setupEPWMActiveHighComplementary (EPWM1_BASE);
//EPWM_setActionQualifierContSWForceAction (EPWM1_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_DISABLED);
EPWM_setActionQualifierContSWForceAction (EPWM2_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
EPWM_setActionQualifierContSWForceAction (EPWM3_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
setupEPWMActiveHigh (EPWM2_BASE);
setupEPWMActiveHigh (EPWM3_BASE);
msg = "\r\n- SYSTEM ON!!! \n\0";
SCI_writeCharArray (SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****
void Turn_Sys_OFF (void)
{
    State_ON = 0;
    //Change State Interrupts
    GPIO_disableInterrupt (GPIO_INT_XINT4);
    GPIO_enableInterrupt (GPIO_INT_XINT3);           // Enable ON Interrupt
    //Reset All Drivers
    Reset_All();
    GPIO_writePin(0, 0);
    GPIO_writePin(1, 0);
    GPIO_writePin(LED_ON_OFF, LED_OFF);
    //Change State PWM's
    EPWM_setActionQualifierContSWForceAction (EPWM2_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    EPWM_setActionQualifierContSWForceAction (EPWM3_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);

    msg = "\r\n- SYSTEM OFF \n\0";
    SCI_writeCharArray (SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****
void Turn_Slow_Charge_ON (void)
{
    //Change State Indicators
    GPIO_writePin(RELAY_A_Pin, RELAY_OFF);
    msg = "\r\n- Active Slow Charge - Series Resistance \n\0";
    SCI_writeCharArray (SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****/

```

```

void Turn_Slow_Charge_OFF(void)
{
    //Change State Indicators
    GPIO_writePin(RELAY_A_Pin, RELAY_ON);
    msg = "\r\n- Inactive Slow Charge - Normal Operation \n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****

void Reset_All(void)
{
    GPIO_writePin(RST_A_Pin, 0);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 0);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 1);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 0);
}

/*****
/*****

void Set_Addr_Bus(uint16_t addr)
{
    //Uint32 tmp = 0x0, Old_Data = GpioDataRegs.GPADAT.all;
    Uint32 tmp = 0x0, Old_Data = GPIO_readPortData(GPIO_PORT_A);

    uint16_t Addr_TMP = addr;
    //Array address pines
    const int Address_Pin [ADDR_LENGTH] = {ADDR_A_PIN, ADDR_B_PIN, ADDR_C_PIN}; // LSB - MSB
    //Move data to address directions
    int idx;
    for ( idx = 0; idx < ADDR_LENGTH; idx++)
    {
        //Clean address register values
        Old_Data &= ~((Uint32)1 << Address_Pin[idx]);
        // Move data to the address bit
        tmp |= (Uint32)((Addr_TMP >> idx) & 1) << Address_Pin[idx];
    }
    //Send to GPIO's Data
    //GpioDataRegs.GPADAT.all = Old_Data | tmp;
    GPIO_writePortData(GPIO_PORT_A, Old_Data | tmp);
    DELAY_US(2);
}

/*****
/*****

void Cfg_SWs_GPIO(void)
{
    //Configuracion SW_ON_Pin
    GPIO_setPadConfig(SW_ON_Pin, GPIO_PIN_TYPE_PULLUP);

```

```

//GPIO_setQualificationMode(SW_ON_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
GPIO_setQualificationMode(SW_ON_Pin, GPIO_QUAL_6SAMPLE);
GPIO_setPinConfig(GPIO_60_GPIO60);
GPIO_setDirectionMode(SW_ON_Pin, GPIO_DIR_MODE_IN);
GPIO_setQualificationPeriod(SW_ON_Pin, 510); // Each sampling window is 510*SYSCLKOUT
GPIO_setInterruptType(GPIO_INT_XINT3, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
GPIO_setInterruptPin(SW_ON_Pin, GPIO_INT_XINT3);
GPIO_enableInterrupt(GPIO_INT_XINT3); //Interrupt_enable(INT_XINT3);
msg = "\r\n- SW_ON_Pin INTERRUPT 3 configured \n\0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);

//Configuracion SW_OFF_Pin
GPIO_setPadConfig(SW_OFF_Pin, GPIO_PIN_TYPE_PULLUP);
//GPIO_setQualificationMode(SW_OFF_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
GPIO_setQualificationMode(SW_OFF_Pin, GPIO_QUAL_6SAMPLE);
GPIO_setPinConfig(GPIO_65_GPIO65);
GPIO_setDirectionMode(SW_OFF_Pin, GPIO_DIR_MODE_IN);
GPIO_setQualificationPeriod(SW_OFF_Pin, 510); // Each sampling window is 510*SYSCLKOUT
GPIO_setInterruptType(GPIO_INT_XINT4, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
GPIO_setInterruptPin(SW_OFF_Pin, GPIO_INT_XINT4);
GPIO_enableInterrupt(GPIO_INT_XINT4); //Interrupt_enable(INT_XINT3);
msg = "\r\n- SW_OFF_Pin INTERRUPT 4 configured \n\0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****
void Cfg_2E020I12_A_GPIOs(void)
{
    //Configuracion RST_EN_A_Pin
    GPIO_setPadConfig(RST_EN_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_writePin(RST_EN_A_Pin, 0);
    GPIO_setPinConfig(GPIO_58_GPIO58);
    GPIO_setDirectionMode(RST_EN_A_Pin, GPIO_DIR_MODE_OUT);

    //Configuracion RST_A_Pin
    GPIO_setPadConfig(RST_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_writePin(RST_A_Pin, 0);
    GPIO_setPinConfig(GPIO_61_GPIO61);
    GPIO_setDirectionMode(RST_A_Pin, GPIO_DIR_MODE_OUT);

    //Configuracion RDY_A_Pin
    GPIO_setPadConfig(RDY_A_Pin, GPIO_PIN_TYPE_PULLUP);
    //GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
    GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_6SAMPLE);
    GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_ASYNC);
    GPIO_setPinConfig(GPIO_122_GPIO122);
    GPIO_setDirectionMode(RDY_A_Pin, GPIO_DIR_MODE_IN);
    GPIO_setQualificationPeriod(RDY_A_Pin, 510); // Each sampling window is 510*SYSCLKOUT
    GPIO_setInterruptType(GPIO_INT_XINT1, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
    GPIO_setInterruptPin(RDY_A_Pin, GPIO_INT_XINT1);

```

```

GPIO_enableInterrupt (GPIO_INT_XINT1); //Interrupt_enable(INT_XINT3);
msg = "\r\n- RDY_Pin INTERRUPT 1 configured \n\0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg,40);

//Configuracion FLT_A_Pin
GPIO_setPadConfig(FLT_A_Pin, GPIO_PIN_TYPE_PULLUP);
//GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_6SAMPLE);
GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_ASYNC);
GPIO_setPinConfig(GPIO_123_GPIO123);
GPIO_setDirectionMode(FLT_A_Pin, GPIO_DIR_MODE_IN);
GPIO_setQualificationPeriod(FLT_A_Pin,510); // Each sampling window is 510*SYSCLKOUT
GPIO_setInterruptType(GPIO_INT_XINT2, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
GPIO_setInterruptPin(FLT_A_Pin,GPIO_INT_XINT2);
GPIO_enableInterrupt (GPIO_INT_XINT2); //Interrupt_enable(INT_XINT3);
msg = "\r\n- FLT_A_Pin INTERRUPT 2 configured \n\0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg,40);
}

/*****
/*****
void Cfg_Addrs_GPIOs(void)
{
    GPIO_SetupPinMux(ADDR_A_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinMux(ADDR_B_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinMux(ADDR_C_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinOptions(ADDR_A_PIN, GPIO_OUTPUT, GPIO_PULLUP);
    GPIO_SetupPinOptions(ADDR_B_PIN, GPIO_OUTPUT, GPIO_PULLUP);
    GPIO_SetupPinOptions(ADDR_C_PIN, GPIO_OUTPUT, GPIO_PULLUP);
    msg = "\r\n- GPIO Address configured \n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg,40);
}

/*****
/*****
void Cfg_Relay_GPIO(void)
{
    //Configuracion RELAY_A_Pin
    GPIO_setPadConfig(RELAY_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_59_GPIO59);
    GPIO_setDirectionMode(RELAY_A_Pin, GPIO_DIR_MODE_OUT);
    msg = "\r\n- GPIO Relay's configured \n\0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg,40);
}

/*****
/*****
void Cfg_RSTs_GPIO(void)
{
    //Configuracion RST_A_Pin
    GPIO_setPadConfig(RST_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_61_GPIO61);

```

```

GPIO_setDirectionMode(RST_A_Pin, GPIO_DIR_MODE_OUT);

//Configuracion RST_EN_A_Pin
GPIO_setPadConfig(RST_EN_A_Pin, GPIO_PIN_TYPE_PULLUP);
GPIO_setPinConfig(GPIO_58_GPIO58);
GPIO_setDirectionMode(RST_EN_A_Pin, GPIO_DIR_MODE_OUT);

GPIO_writePin(RST_A_Pin, 0);
GPIO_writePin(RST_EN_A_Pin, 0);
msg = "\r\n- GPIO Reset configured \n\n0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
}
/*****
/*****
void Cfg_Leds_GPIO(void)
{
    uint16_t i;

    //Configuracion LED_ON_OFF
    GPIO_setPadConfig(LED_ON_OFF, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_31_GPIO31);
    GPIO_setDirectionMode(LED_ON_OFF, GPIO_DIR_MODE_OUT);

    //Configuracion LED_ERROR
    GPIO_setPadConfig(LED_ERROR, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_34_GPIO34);
    GPIO_setDirectionMode(LED_ERROR, GPIO_DIR_MODE_OUT);

    for(i = 0; i < 3; i++)
    {
        DELAY_US(500000);
        GPIO_writePin(LED_ON_OFF, LED_ON);
        GPIO_writePin(LED_ERROR, LED_ON);
        DELAY_US(500000);
        GPIO_writePin(LED_ON_OFF, LED_OFF);
        GPIO_writePin(LED_ERROR, LED_OFF);
    }
    msg = "\r\n- GPIO Leds configured \n\n0";
    SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
}
/*****
/*****
void ConfigGPIO(void)
{
    GPIO_setPadConfig(0, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO0
    GPIO_setPadConfig(1, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO1
    GPIO_setPinConfig(GPIO_0_GPIO0);                //SALIDAS GPIO0-GPIO1
    GPIO_setPinConfig(GPIO_1_GPIO1);
    GPIO_setDirectionMode(0, GPIO_DIR_MODE_OUT);
    GPIO_setDirectionMode(1, GPIO_DIR_MODE_OUT);

```

```

GPIO_WritePin(0, 0);
GPIO_WritePin(1, 0);

/// PINES PARA APAGAR EL SISTEMA

GPIO_setPadConfig(160, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO0
GPIO_setPadConfig(125, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO1
GPIO_setPinConfig(GPIO_160_GPIO160);               //SALIDAS GPIO0-GPIO1
GPIO_setPinConfig(GPIO_125_GPIO125);
GPIO_setDirectionMode(160, GPIO_DIR_MODE_OUT);
GPIO_setDirectionMode(125, GPIO_DIR_MODE_OUT);
GPIO_WritePin(160, 0);
GPIO_WritePin(125, 0);

//      GPIO_setPadConfig(PWM_A_H1_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO0
//      GPIO_setPadConfig(PWM_A_L1_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO1
//      GPIO_setPinConfig(GPIO_0_EPWM1A);                          // GPIO0 = PWM1A
//      GPIO_setPinConfig(GPIO_1_EPWM1B);

GPIO_setPadConfig(PWM_A_H2_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO0
GPIO_setPadConfig(PWM_A_L2_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO1
GPIO_setPinConfig(GPIO_2_EPWM2A);                          // GPIO0 = PWM1A
GPIO_setPinConfig(GPIO_3_EPWM2B);

GPIO_setPadConfig(PWM_A_H3_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO0
GPIO_setPadConfig(PWM_A_L3_Pin, GPIO_PIN_TYPE_PULLUP);      // Enable pullup on GPIO1
GPIO_setPinConfig(GPIO_4_EPWM3A);                          // GPIO0 = PWM1A
GPIO_setPinConfig(GPIO_5_EPWM3B);

GPIO_setPinConfig(GPIO_159_EPWM8A);

msg = "\r\n- GPIO ePWMs configured \n\0";
SCI_writeCharArray(SCIA_BASE, (uint16_t*)msg, 40);
}

/*****
/*****
void initCPUTimers(void)
{
    //
    // Initialize timer period to maximum
    //
    CPUTimer_setPeriod(CPUTIMER0_BASE, 0xFFFFFFFF);

    //
    // Initialize pre-scale counter to divide by 1 (SYSCLKOUT)
    //
    CPUTimer_setPreScaler(CPUTIMER0_BASE, 0);

```

```

//
// Make sure timer is stopped
//
CPUTimer_stopTimer(CPUTIMER0_BASE);

//
// Reload all counter register with period value
//
CPUTimer_reloadTimerCounter(CPUTIMER0_BASE);

//
// Reset interrupt counter
//
cpuTimer0IntCount = 0;
}
/*****
/*****
void configCPUTimer(uint32_t cpuTimer, float freq, float period)
{
    uint32_t temp;

    //
    // Initialize timer period:
    //
    temp = (uint32_t)(freq / 1000000 * period);
    CPUTimer_setPeriod(cpuTimer, temp);

    //
    // Set pre-scale counter to divide by 1 (SYSCLKOUT):
    //
    CPUTimer_setPreScaler(cpuTimer, 0);

    //
    // Initializes timer control register. The timer is stopped, reloaded,
    // free run disabled, and interrupt enabled.
    // Additionally, the free and soft bits are set
    //
    CPUTimer_stopTimer(cpuTimer);
    CPUTimer_reloadTimerCounter(cpuTimer);
    CPUTimer_setEmulationMode(cpuTimer,
                               CPUTIMER_EMULATIONMODE_STOPAFTERNEXTDECREMENT);
    CPUTimer_enableInterrupt(cpuTimer);

    //
    // Resets interrupt counters for the three cpuTimers
    //
    if (cpuTimer == CPUTIMER0_BASE)
    {
        cpuTimer0IntCount = 0;
    }
}

```

```
}
```

10.5. Codigo implementado para control PI

10.5.1. Inversor.h

```
*
```

```
* Created on: 28/12/2020
* Author: Katherin Casallas y Felipe Lopez
*/
```

```
#ifndef INVERSOR_H_
#define INVERSOR_H_
```

```
// ADC Max Values Measures
```

```
#define Gain_Voltage    36.2      // Max Voltage Measurement ADC
#define Gain_Current    50        // Max Current Measurement ADC
```

```

/*****
*****/

```

```
void Cfg_Leds_GPIO(void);
void Cfg_Relay_GPIO(void);
void Cfg_2E020I12_A_GPIOs(void);
void Cfg_PWMs_A_GPIO(void);
void Cfg_RSTs_GPIO(void);
void Cfg_RDY_INT(void);
void Cfg_FLT_INT(void);
void Cfg_SWs_GPIO(void);
void Cfg_ON_INT(void);
void Cfg_OFF_INT(void);
void Cfg_ADC_Epwm(void);
void IC2E020I12_RDYs_OK(void);
void IC2E020I12_FLT_RST(void);
void Set_Addr_Bus(Uint16 addr);
void Cfg_Addrs_GPIOs(void);
void Turn_Sys_ON(void);
void Turn_Sys_OFF(void);
void Turn_Slow_Charge_ON(void);
void Turn_Slow_Charge_OFF(void);
void Reset_All(void);
void initADCSensor(void);
void initADCSENSOR(void);
```

```

//__interrupt void RDY_Drivers_isr(void);
//__interrupt void Fault_Drivers_isr(void);
//__interrupt void ON_Sys_isr(void);
//__interrupt void OFF_Sys_isr(void);

void enceder_sys(void);
void apagar_sys(void);
void Check_RDY_2E020I12_UVWX(Uint16 See_OK);
void RST_2E020I12_UVWX(void);

/*****
*****/
#define PI_i_d_DEFAULTS { 0.0, /* Ref*/ \
    0.0, /* Fdb*/ \
    0.0, /* Err*/ \
    12, /* Kp */ \
    0.00003, /* Ki */ \
    0.0, /* Up */ \
    0.0, /* Ui */ \
    0.0, /* OutPresat*/ \
    0.95, /* OutMax*/ \
    -0.95, /* OutMin*/ \
    0.0, /* Out*/ \
    1.0 /* U6 */ }

/*****
*****/

#endif /* inversor_H_ */

```

10.5.2. Inversor.c

```

#include "F28x_Project.h"
#include "F2837xD_device.h"
#include "driverlib.h"
#include "device.h"
#include "Six_Drivers.h"
#include "Inversor.h"
#include "Solar_F.h"
//#include "IQmathLib.h"
#include "stdio.h"

typedef struct
{
    float Ref; /* Input: Reference input
    float Fdb; /* Input: Feedback input

```



```

    float Err;           // Variable: Error
    float Kp;            // Parameter: Proportional gain
    float Ki;            // Parameter: Integral gain
    float Up;            // Variable: Proportional output
    float Ui;            // Variable: Integral output
    float OutPreSat;     // Variable: Pre-saturated output
    float OutMax;        // Parameter: Maximum output
    float OutMin;        // Parameter: Minimum output
    float Out;           // Output: PID output
    float U6;            // Sat, No_Sat indicator
} PI_REGS;

```

```

#define EPWM_BASE      EPWM1_BASE
#define PI              3.1416
#define Fs              50000
#define Fa              60
#define RESULTS_BUFFER_SIZE  833
#define RESULTS_BUFFER_SIZE_D 208
#define ARG             2*PI*Fa/Fs
#define EX_ADC_RESOLUTION 16
#define TBPRD_A         1000
#define TBPRD_B         500
#define Ts              1 / Fs
#define kp              5
#define ki              30
#define k1              2*kp + ki*Ts
#define k2              ki*Ts - 2*kp
#define k3              2
#define k4              2

```

```

SPLL_1ph_F spll1;
SPLL_1ph_F_NOTCH_COEFF spll_notch_coef1;

```

```

PARK_F park_ref;
PARK_F park_load;

```

```

iPARK_F ipark;

```

```

//
// Function Prototypes
//
void ConfigGPIO(void);
void Cfg_SWs_GPIO(void);
void Cfg_Leds_GPIO(void);
void Cfg_Relay_GPIO(void);
void Cfg_2E020I12_A_GPIOs(void);
void Cfg_Addrs_GPIOs(void);
void Cfg_RSTs_GPIO(void);
void IC2E020I12_RDYs_OK(void);
void Turn_Sys_ON(void);

```

```

void Turn_Sys_OFF(void);
void Turn_Slow_Charge_OFF(void);
void Set_Addr_Bus(Uint16 addr);
void Reset_All(void);
void IC2E020I12_FLT_RST(void);
void setupEPWMActiveHigh(uint32_t base);
void setupEPWMActiveHighComplementary(uint32_t base);
void initEPWM(void);
void Cfg_ON_INT(void);
void setupEPWMActiveHighComplementary(uint32_t base);
void initADC(void);
void initADCSOC(void);
void configureDAC(void);

void Init_PI_Outs(PI_REGS *p, float ref);
void DCL_PIp(PI_REGS *v);

__interrupt void adcA1ISR(void);
__interrupt void ON_Sys_isr(void);
__interrupt void OFF_Sys_isr(void);
__interrupt void RDY_Drivers_isr(void);
__interrupt void Fault_Drivers_isr(void);

uint16_t adcAResults[RESULTS_BUFFER_SIZE];    // Buffer for results
float Buff1[RESULTS_BUFFER_SIZE];
float Buff2[RESULTS_BUFFER_SIZE];
float Buff3[RESULTS_BUFFER_SIZE];
float Buff4[RESULTS_BUFFER_SIZE];

PI_REGS PI_i_d = PI_i_d_DEFAULTS;
PI_REGS PI_i_q = PI_i_d_DEFAULTS;

uint16_t State_ON, tmp_g;
float arg_t;
float k = 2.5;//2.5

void main(void)
{
    //
    // Initialize device clock and peripherals//Inicializar el reloj del dispositivo y los periféricos
    //
    tmp_g = TBPRD_A;
    arg_t = ARG;
    Device_init();
    Device_initGPIO();
    Interrupt_initModule();
    Interrupt_initVectorTable();

    Interrupt_register(INT_ADCA1, &adcA1ISR);

```

```

Interrupt_register(INT_XINT1,&RDY_Drivers_isr);
Interrupt_register(INT_XINT2,&Fault_Drivers_isr);
Interrupt_register(INT_XINT3,&ON_Sys_isr);
Interrupt_register(INT_XINT4,&OFF_Sys_isr);

Sys_Flags.all = 0;

ConfigGPIO();
Cfg_SWs_GPIO();
Cfg_Leds_GPIO();
Cfg_Relay_GPIO();
Cfg_2E020I12_A_GPIOs();
Cfg_Addrs_GPIOs();
Cfg_RSTs_GPIO();
IC2E020I12_RDYs_OK();
configureDAC();
initEPWM();
initADC();
initADCSOC();

PARK_F_init(&park_ref);
PARK_F_init(&park_load);

iPARK_F_init(&ipark);

SPLL_1ph_F_init(Fa, ((float) (1.0/Fs)), &spll1);
SPLL_1ph_F_notch_coeff_update(((float) (1.0/Fs)), (float) (2*PI*Fa*2), (float) 0.00001, (float) 0.1, &spll1);

Init_PI_Outs(&PI_i_d, 0);
Init_PI_Outs(&PI_i_q, 0);

Interrupt_enable(INT_ADCA1);
Interrupt_enable(INT_XINT1);
Interrupt_enable(INT_XINT2);
Interrupt_enable(INT_XINT3);
Interrupt_enable(INT_XINT4);
//
// Enable Global Interrupt (INTM) and realtime interrupt (DBGM)//Activar la interrupción global
//
EINT;
ERTM;

Turn_Sys_OFF();
Turn_Slow_Charge_OFF();

EPWM_enableADCTrigger(EPWM_BASE, EPWM_SOC_A);
EPWM_setTimeBaseCounterMode(EPWM_BASE, EPWM_COUNTER_MODE_UP_DOWN);
while(1)
{

```

```

    }
}

/*****
/*****
__interrupt void adcA1ISR(void)
{
    static uint16_t index = 0;
    //static float32 Voltage;
    static uint16_t indexdelay = 0;
    static float32 Signal_delay_Ref;
    static float32 Signal_delay_iL;
    static float32 zero = 0;
    static float32 Current_Ref;
    static float32 Current_iL;
    int16_t tmp;
    uint16_t tmp1;

    ///////////////////////////////////MEDICIONES////////////////////////////////////
    //Voltage = (float32) (ADC_readResult (ADCARERESULT_BASE, ADC_SOC_NUMBER0));
    //CurrentMeaLoad = (float32) (ADC_readResult (ADCCRESULT_BASE, ADC_SOC_NUMBER4));
    //CurrentBobi = (float32) (ADC_readResult (ADCCRESULT_BASE, ADC_SOC_NUMBER5));

    ///////////////////////////////////SEÑALES_ESCALIZADAS////////////////////////////////////

    Current_Ref = (float32) (k*(((9.15e-5)*(ADC_readResult (ADCCRESULT_BASE, ADC_SOC_NUMBER4))-3)
    Current_iL = (float32) (k*(((9.15e-5)*(ADC_readResult (ADCCRESULT_BASE, ADC_SOC_NUMBER5))-3)
    spll1.AC_input = Current_Ref;

    ///////////////////////////////////PLL////////////////////////////////////
    SPL1_1ph_F_FUNC(&spl11);
    spll1.theta[0] *= 6.28318530718;

    ///////////////////////////////////TRANSFORMADA_PARK_REF////////////////////////////////////
    //Generar señal desfasada 90 grados
    indexdelay = (index + RESULTS_BUFFER_SIZE_D) % RESULTS_BUFFER_SIZE;
    Buff1[index] = spll1.AC_input;
    Signal_delay_Ref = Buff1[indexdelay];

    //Transformada de park
    park_ref.beta = spll1.AC_input;
    park_ref.alpha = Signal_delay_Ref;
    park_ref.zero = zero;
    park_ref.sin = sin(spll1.theta[0]);
    park_ref.cos = cos(spll1.theta[0]);
    PARK_F_FUNC(&park_ref);

    ///////////////////////////////////TRANSFORMADA_PARK_LOAD////////////////////////////////////
    Buff2[index] = Current_iL;
    Signal_delay_iL = Buff2[indexdelay];

```

```

//Transformada de park
park_load.beta = Current_iL;
park_load.alpha = Signal_delay_iL;
park_load.zero = zero;
park_load.sin = sin(spll1.theta[0]);
park_load.cos = cos(spll1.theta[0]);
PARK_F_FUNC(&park_load);

/*-----
/*----- COMPENSADOR DE CORRIENTE ID -----
/*-----

    PI_i_d.Ref = park_ref.d;
    PI_i_d.Fdb = park_load.d;
    DCL_PIp(&PI_i_d);

/*-----
/*----- COMPENSADOR DE CORRIENTE IQ -----
/*-----

    PI_i_q.Ref = park_ref.q;
    PI_i_q.Fdb = park_load.q;
    DCL_PIp(&PI_i_q);

//////////////////TRANSFORMADA_INVERSA_PARK//////////////////

    ipark.d = PI_i_d.Out;
    ipark.q = PI_i_q.Out;
    ipark.zero = zero;
    ipark.sin = sin(spll1.theta[0]);
    ipark.cos = cos(spll1.theta[0]);
    iPARK_F_FUNC(&ipark);

//////////////////SATURADOR//////////////////

    if(ipark.beta >= 0.95)
        ipark.beta = 0.95;
    if(ipark.beta <= -0.95)
        ipark.beta = -0.95;

//////////////////COMPARADOR_PWM//////////////////

    DAC_setShadowValue(DACB_BASE, (uint16_t)(spll1.theta[0]*400) + 1500);
    tmp = TBPRD_B * ipark.beta;
//    tmp = TBPRD_B * Current_Ref;
    if(tmp > TBPRD_B)
        tmp = TBPRD_B;
    if(tmp < -TBPRD_B)
        tmp = -TBPRD_B;

    tmp1 = TBPRD_B + tmp;
    EPWM_setCounterCompareValue(EPWM_BASE, EPWM_COUNTER_COMPARE_A, tmp1);

//////////////////GRAFICAR//////////////////

    //Buff3[index] = (float)tmp;

```

```

    Buff3[index] = park_ref.d;
    //Buff4[index] = park_load.d;
    //Buff4[index] = (float)tmp1;
    Buff4[index] = park_load.d;
    //////////////////////////////////////
    if(++index >= RESULTS_BUFFER_SIZE)
        index = 0;

    // Clear the interrupt flag
    //
    ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
    //
    // Check if overflow has occurred
    //
    if(true == ADC_getInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1))
    {
        ADC_clearInterruptOverflowStatus(ADCA_BASE, ADC_INT_NUMBER1);
        ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
    }
    //
    // Acknowledge the interrupt
    //
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP1);
}
/*****
*****/
__interrupt void Fault_Drivers_isr(void)
{
    Sys_Flags.bit.FAIL_FLTs = 1;           //Fail Register Flag
    GPIO_disableInterrupt(GPIO_INT_XINT2); //disable FLT Interrupt
    GPIO_disableInterrupt(GPIO_INT_XINT1); //disable RDY Interrupt
    //Fault Identification Sequence
    IC2E020I12_FLT_RST();
    GPIO_enableInterrupt(GPIO_INT_XINT2);  //enable FLT Interrupt
    GPIO_enableInterrupt(GPIO_INT_XINT1);  //enable RDY Interrupt
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP1);
}
/*****
*****/
__interrupt void ON_Sys_isr(void)
{
    Turn_Sys_ON();
    Interrupt_clearACKGroup(INTERRUPT_ACK_GROUP12);
    //PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
}
/*****
*****/
__interrupt void OFF_Sys_isr(void)
{
    Turn_Sys_OFF();
}

```

```

    Interrupt_clearACKGroup (INTERRUPT_ACK_GROUP12);
}
/*****
/*****
__interrupt void RDY_Drivers_isr(void)
{
    DELAY_US(10000);
    Sys_Flags.bit.FAIL_RDYs = 1;           //Source Register Flag
    GPIO_disableInterrupt (GPIO_INT_XINT1);
    GPIO_disableInterrupt (GPIO_INT_XINT2);

    //Source Fault Identification Sequence
    IC2E020I12_RDYs_OK();
    Turn_Sys_OFF();
    GPIO_enableInterrupt (GPIO_INT_XINT1);
    GPIO_enableInterrupt (GPIO_INT_XINT2);
    Interrupt_clearACKGroup (INTERRUPT_ACK_GROUP1);
}
/*****
/*****
void initEPWM(void)
{
    //
    // Disable SOCA
    //
    EPWM_disableADCTrigger (EPWM_BASE, EPWM_SOC_A);
    //
    // Configure the SOC to occur on the first up-count event // Configurar el SOC para que se produ
    //
    EPWM_setADCTriggerSource (EPWM_BASE, EPWM_SOC_A, EPWM_SOC_TBCTR_ZERO);
    EPWM_setADCTriggerEventPrescale (EPWM_BASE, EPWM_SOC_A, 1);
    //
    // Set the compare A value to 1000 and the period to 1999
    // Assuming ePWM clock is 100MHz, this would give 50kHz sampling
    // 50MHz ePWM clock would give 25kHz sampling, etc.
    // The sample rate can also be modulated by changing the ePWM period
    // directly (ensure that the compare A value is less than the period).
    //

    //EPWM_setCounterCompareValue (EPWM1_BASE, EPWM_COUNTER_COMPARE_A, 500);
//    EPWM_setTimeBasePeriod (EPWM1_BASE, 1999);
    EPWM_setTimeBasePeriod (EPWM_BASE, TBPRD_A);
    //
    // Set the local ePWM module clock divider to /1
    //
    EPWM_setClockPrescaler (EPWM_BASE, EPWM_CLOCK_DIVIDER_1, EPWM_HSCLOCK_DIVIDER_1);
    //
    // Freeze the counter
    //
//    EPWM_setTimeBaseCounterMode (EPWM1_BASE, EPWM_COUNTER_MODE_STOP_FREEZE);

```

```

    EPWM_setTimeBaseCounterMode(EPWM_BASE, EPWM_COUNTER_MODE_UP_DOWN);
    EPWM_setActionQualifierAction(EPWM_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_OUTPUT_HIGH, EPWM_AQ_OUTPUT_ON_T
    EPWM_setActionQualifierAction(EPWM_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_OUTPUT_LOW, EPWM_AQ_OUTPUT_ON_T
    EPWM_setActionQualifierAction(EPWM_BASE, EPWM_AQ_OUTPUT_B, EPWM_AQ_OUTPUT_LOW, EPWM_AQ_OUTPUT_ON_T
    EPWM_setActionQualifierAction(EPWM_BASE, EPWM_AQ_OUTPUT_B, EPWM_AQ_OUTPUT_HIGH, EPWM_AQ_OUTPUT_ON_T
}
/*****
/*****
void initADC(void)
{
    //
    // Set ADCDLK divider to /4 //Poner el divisor ADCDLK a /4
    //
    ADC_setPrescaler(ADCA_BASE, ADC_CLK_DIV_4_0);
    ADC_setPrescaler(ADCC_BASE, ADC_CLK_DIV_4_0);
    //
    // Set resolution and signal mode (see #defines above) and load
    // corresponding trims. // Establecer la resolución y el modo de señal (ver #defines arriba) y c
    //
    #if(EX_ADC_RESOLUTION == 12)
        ADC_setMode(ADCA_BASE, ADC_RESOLUTION_12BIT, ADC_MODE_SINGLE_ENDED);
    #elif(EX_ADC_RESOLUTION == 16)
        ADC_setMode(ADCA_BASE, ADC_RESOLUTION_16BIT, ADC_MODE_DIFFERENTIAL);
        ADC_setMode(ADCC_BASE, ADC_RESOLUTION_16BIT, ADC_MODE_DIFFERENTIAL);
    #endif
    //
    // Set pulse positions to late // Ajustar las posiciones de los pulsos a la tarde
    //
    ADC_setInterruptPulseMode(ADCA_BASE, ADC_PULSE_END_OF_CONV);
    ADC_setInterruptPulseMode(ADCC_BASE, ADC_PULSE_END_OF_CONV);
    //
    // Power up the ADC and then delay for 1 ms //Encender el ADC y retardar 1 ms
    //
    ADC_enableConverter(ADCA_BASE);
    ADC_enableConverter(ADCC_BASE);
    DEVICE_DELAY_US(1000);
}
/*****
/*****
void initADCSOC(void)
{
    //
    // Configure SOC0 of ADCA to convert pin A0. The EPWM1SOCA signal will be
    // the trigger.
    // - For 12-bit resolution, a sampling window of 15 (75 ns at a 200MHz
    //   SYSCLK rate) will be used. For 16-bit resolution, a sampling window
    //   of 64 (320 ns at a 200MHz SYSCLK rate) will be used.
    // - NOTE: A longer sampling window will be required if the ADC driving
    //   source is less than ideal (an ideal source would be a high bandwidth
    //   op-amp with a small series resistance). See TI application report

```

```

//  SPRACT6 for guidance on ADC driver design.
//

#if(EX_ADC_RESOLUTION == 12)
    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER0, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN0, 15);
#elif(EX_ADC_RESOLUTION == 16)
    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER0, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN0_ADCIN1, 64);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER1, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN14_ADCIN15, 64);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER2, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN2_ADCIN3, 64);

    ADC_setupSOC(ADCA_BASE, ADC_SOC_NUMBER3, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN4_ADCIN5, 64);

    ADC_setupSOC(ADCC_BASE, ADC_SOC_NUMBER4, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN2_ADCIN3, 64);

    ADC_setupSOC(ADCC_BASE, ADC_SOC_NUMBER5, ADC_TRIGGER_EPWM1_SOCA,
                  ADC_CH_ADCIN4_ADCIN5, 64);

#endif
//
// Set SOC0 to set the interrupt 1 flag. Enable the interrupt and make
// sure its flag is cleared.
//
ADC_setInterruptSource(ADCA_BASE, ADC_INT_NUMBER1, ADC_SOC_NUMBER0);
ADC_enableInterrupt(ADCA_BASE, ADC_INT_NUMBER1);
ADC_clearInterruptStatus(ADCA_BASE, ADC_INT_NUMBER1);
}
/*****
/*****
void ConfigGPIO(void)
{
    GPIO_setPadConfig(PWM_A_H1_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO0
    GPIO_setPadConfig(PWM_A_L1_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO1
    GPIO_setPinConfig(GPIO_0_EPWM1A);                          // GPIO0 = PWM1A
    GPIO_setPinConfig(GPIO_1_EPWM1B);

    GPIO_setPadConfig(PWM_A_H2_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO0
    GPIO_setPadConfig(PWM_A_L2_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO1
    GPIO_setPinConfig(GPIO_2_EPWM2A);                          // GPIO0 = PWM1A
    GPIO_setPinConfig(GPIO_3_EPWM2B);

    GPIO_setPadConfig(PWM_A_H3_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO0
    GPIO_setPadConfig(PWM_A_L3_Pin, GPIO_PIN_TYPE_PULLUP);    // Enable pullup on GPIO1

```

```

    GPIO_setPinConfig(GPIO_4_EPWM3A);                // GPIO0 = PWM1A
    GPIO_setPinConfig(GPIO_5_EPWM3B);
}
/*****
/*****
void Cfg_SWs_GPIO(void)
{
    //Configuracion SW_ON_Pin
    GPIO_setPadConfig(SW_ON_Pin, GPIO_PIN_TYPE_PULLUP);
    //GPIO_setQualificationMode(SW_ON_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
    GPIO_setQualificationMode(SW_ON_Pin, GPIO_QUAL_6SAMPLE);
    GPIO_setPinConfig(GPIO_60_GPIO60);
    GPIO_setDirectionMode(SW_ON_Pin, GPIO_DIR_MODE_IN);
    GPIO_setQualificationPeriod(SW_ON_Pin, 510); // Each sampling window is 510*SYSCLKOUT
    GPIO_setInterruptType(GPIO_INT_XINT3, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
    GPIO_setInterruptPin(SW_ON_Pin, GPIO_INT_XINT3);
    GPIO_enableInterrupt(GPIO_INT_XINT3); //Interrupt_enable(INT_XINT3);

    //Configuracion SW_OFF_Pin
    GPIO_setPadConfig(SW_OFF_Pin, GPIO_PIN_TYPE_PULLUP);
    //GPIO_setQualificationMode(SW_OFF_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
    GPIO_setQualificationMode(SW_OFF_Pin, GPIO_QUAL_6SAMPLE);
    GPIO_setPinConfig(GPIO_65_GPIO65);
    GPIO_setDirectionMode(SW_OFF_Pin, GPIO_DIR_MODE_IN);
    GPIO_setQualificationPeriod(SW_OFF_Pin, 510); // Each sampling window is 510*SYSCLKOUT
    GPIO_setInterruptType(GPIO_INT_XINT4, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
    GPIO_setInterruptPin(SW_OFF_Pin, GPIO_INT_XINT4);
    GPIO_enableInterrupt(GPIO_INT_XINT4); //Interrupt_enable(INT_XINT3);

}
/*****
/*****
void Cfg_2E020I12_A_GPIOs(void)
{
    //Configuracion RST_EN_A_Pin
    GPIO_setPadConfig(RST_EN_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_writePin(RST_EN_A_Pin, 0);
    GPIO_setPinConfig(GPIO_58_GPIO58);
    GPIO_setDirectionMode(RST_EN_A_Pin, GPIO_DIR_MODE_OUT);

    //Configuracion RST_A_Pin
    GPIO_setPadConfig(RST_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_writePin(RST_A_Pin, 0);
    GPIO_setPinConfig(GPIO_61_GPIO61);
    GPIO_setDirectionMode(RST_A_Pin, GPIO_DIR_MODE_OUT);

    //Configuracion RDY_A_Pin
    GPIO_setPadConfig(RDY_A_Pin, GPIO_PIN_TYPE_PULLUP);
    //GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
    GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_6SAMPLE);

```

```

GPIO_setQualificationMode(RDY_A_Pin, GPIO_QUAL_ASYNC);
GPIO_setPinConfig(GPIO_122_GPIO122);
GPIO_setDirectionMode(RDY_A_Pin, GPIO_DIR_MODE_IN);
GPIO_setQualificationPeriod(RDY_A_Pin, 510); // Each sampling window is 510*SYSCLKOUT
GPIO_setInterruptType(GPIO_INT_XINT1, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
GPIO_setInterruptPin(RDY_A_Pin, GPIO_INT_XINT1);
GPIO_enableInterrupt(GPIO_INT_XINT1); //Interrupt_enable(INT_XINT3);

//Configuracion FLT_A_Pin
GPIO_setPadConfig(FLT_A_Pin, GPIO_PIN_TYPE_PULLUP);
//GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_6SAMPLE | GPIO_QUAL_ASYNC);
GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_6SAMPLE);
GPIO_setQualificationMode(FLT_A_Pin, GPIO_QUAL_ASYNC);
GPIO_setPinConfig(GPIO_123_GPIO123);
GPIO_setDirectionMode(FLT_A_Pin, GPIO_DIR_MODE_IN);
GPIO_setQualificationPeriod(FLT_A_Pin, 510); // Each sampling window is 510*SYSCLKOUT
GPIO_setInterruptType(GPIO_INT_XINT2, GPIO_INT_TYPE_FALLING_EDGE); // Falling edge interrupt
GPIO_setInterruptPin(FLT_A_Pin, GPIO_INT_XINT2);
GPIO_enableInterrupt(GPIO_INT_XINT2); //Interrupt_enable(INT_XINT3);

}

/*****
/*****
void Cfg_Addrs_GPIOs(void)
{
    GPIO_SetupPinMux(ADDR_A_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinMux(ADDR_B_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinMux(ADDR_C_PIN, GPIO_MUX_CPU1, 0);
    GPIO_SetupPinOptions(ADDR_A_PIN, GPIO_OUTPUT, GPIO_PULLUP);
    GPIO_SetupPinOptions(ADDR_B_PIN, GPIO_OUTPUT, GPIO_PULLUP);
    GPIO_SetupPinOptions(ADDR_C_PIN, GPIO_OUTPUT, GPIO_PULLUP);

}

/*****
/*****
void Cfg_Relay_GPIO(void)
{
    //Configuracion RELAY_A_Pin
    GPIO_setPadConfig(RELAY_A_Pin, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_59_GPIO59);
    GPIO_setDirectionMode(RELAY_A_Pin, GPIO_DIR_MODE_OUT);

}

/*****
/*****
void Cfg_RSTs_GPIO(void)
{
    //Configuracion RST_A_Pin
    GPIO_setPadConfig(RST_A_Pin, GPIO_PIN_TYPE_PULLUP);

```

```

GPIO_setPinConfig(GPIO_61_GPIO61);
GPIO_setDirectionMode(RST_A_Pin, GPIO_DIR_MODE_OUT);

//Configuracion RST_EN_A_Pin
GPIO_setPadConfig(RST_EN_A_Pin, GPIO_PIN_TYPE_PULLUP);
GPIO_setPinConfig(GPIO_58_GPIO58);
GPIO_setDirectionMode(RST_EN_A_Pin, GPIO_DIR_MODE_OUT);

GPIO_writePin(RST_A_Pin, 0);
GPIO_writePin(RST_EN_A_Pin, 0);

}
/*****
/*****/
void Cfg_Leds_GPIO(void)
{
    Uint16 i;

    //Configuracion LED_ON_OFF
    GPIO_setPadConfig(LED_ON_OFF, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_31_GPIO31);
    GPIO_setDirectionMode(LED_ON_OFF, GPIO_DIR_MODE_OUT);

    //Configuracion LED_ERROR
    GPIO_setPadConfig(LED_ERROR, GPIO_PIN_TYPE_PULLUP);
    GPIO_setPinConfig(GPIO_34_GPIO34);
    GPIO_setDirectionMode(LED_ERROR, GPIO_DIR_MODE_OUT);

    for(i = 0; i < 3; i++)
    {
        DELAY_US(500000);
        GPIO_writePin(LED_ON_OFF, LED_ON);
        GPIO_writePin(LED_ERROR, LED_ON);
        DELAY_US(500000);
        GPIO_writePin(LED_ON_OFF, LED_OFF);
        GPIO_writePin(LED_ERROR, LED_OFF);
    }
}
/*****/
/*****/
void IC2E020I12_RDYs_OK(void) // esta fubcion al correr el programa se usa para forzar los PWMs en b
{
    Uint16 Addr_RDY, Branch_RDY;

    Set_Addr_Bus(0);

    if(GPIO_readPin(RDY_A_Pin) == 0)
    {
        for(Addr_RDY = 1; Addr_RDY < 7; Addr_RDY++)

```

```

    {
        Set_Addr_Bus (Addr_RDY);
        if (GPIO_ReadPin (RDY_A_Pin) == 0)
        {
            if (Addr_RDY == 1 || Addr_RDY == 4)
            {
                EPWM_setActionQualifierContSWForceAction (EPWM_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_O
                setupEPWMActiveHigh (EPWM_BASE);
            }

            if (Addr_RDY < 4)
            {
                Branch_RDY = Addr_RDY;
            }
            else
            {
                Branch_RDY = Addr_RDY-3;
            }
        }
    }
    Set_Addr_Bus (0);
    DELAY_US (20);
}

/*****
/*****
void Turn_Slow_Charge_OFF (void)
{
    //Change State Indicators
    GPIO_writePin (RELAY_A_Pin, RELAY_ON);
}

/*****
/*****
void Set_Addr_Bus (Uint16 addr)
{
    //Uint32 tmp = 0x0, Old_Data = GpioDataRegs.GPADAT.all;
    Uint32 tmp = 0x0, Old_Data = GPIO_readPortData (GPIO_PORT_A);

    Uint16 Addr_TMP = addr;
    //Array address pines
    const int Address_Pin [ADDR_LENGTH] = {ADDR_A_PIN, ADDR_B_PIN, ADDR_C_PIN}; // LSB - MSB
    //Move data to address directions
    int idx;
    for ( idx = 0; idx < ADDR_LENGTH; idx++)
    {
        //Clean address register values
        Old_Data &= ~( (Uint32) 1 << Address_Pin[idx]);
        // Move data to the address bit
        tmp |= (Uint32) ((Addr_TMP >> idx) & 1) << Address_Pin[idx];
    }
}

```

```

    }
    //Send to GPIO's Data
    //GpioDataRegs.GPADAT.all = Old_Data | tmp;
    GPIO_writePortData(GPIO_PORT_A, Old_Data | tmp);
    DELAY_US(2);
}
/*****
/*****/
void Reset_All(void)
{
    GPIO_writePin(RST_A_Pin, 0);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 0);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 1);
    DELAY_US(2);
    GPIO_writePin(RST_EN_A_Pin, 0);
}
/*****
/*****/
void IC2E020I12_FLT_RST(void)
{
    Uint16 Addr_FLT, Branch_FLT;

    Set_Addr_Bus(0);

    if(GPIO_readPin(FLT_A_Pin) == 0)
    {
        for(Addr_FLT = 1; Addr_FLT < 7; Addr_FLT++)
        {
            Set_Addr_Bus(Addr_FLT);
            if(GPIO_readPin(FLT_A_Pin) == 0)
            {
                if(Addr_FLT < 4)
                {
                    Branch_FLT = Addr_FLT;
                }
                else
                {
                    Branch_FLT = Addr_FLT-3;
                }

                GPIO_writePin(RST_A_Pin, 1);
                DELAY_US(2);
                GPIO_writePin(RST_A_Pin, 0);
                DELAY_US(1);

                Turn_Sys_OFF();
            }
        }
    }
}

```

```

    }
    Set_Addr_Bus(0);
    DELAY_US(20);
}
/*****
/*****
void Turn_Sys_ON(void)
{
    State_ON = 1;
    //Change State Interrupts
    GPIO_disableInterrupt(GPIO_INT_XINT3); //Interrupt_enable(INT_XINT3);
    //XintRegs.XINT3CR.bit.ENABLE = 0;           // Disable ON Interrupt
    GPIO_enableInterrupt(GPIO_INT_XINT4);
    //XintRegs.XINT4CR.bit.ENABLE = 1;           // Enable OFF Interrupt
    //Change State Indicators
    GPIO_writePin(RST_EN_A_Pin, 1);
    GPIO_writePin(LED_ON_OFF, LED_ON);
    GPIO_writePin(LED_ERROR, LED_OFF);
    //Change State PWM's
    setupEPWMActiveHighComplementary(EPWM_BASE);
    EPWM_setActionQualifierContSWForceAction(EPWM_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_DISABLED);
    EPWM_setActionQualifierContSWForceAction(EPWM2_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    EPWM_setActionQualifierContSWForceAction(EPWM3_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    setupEPWMActiveHigh(EPWM2_BASE);
    setupEPWMActiveHigh(EPWM3_BASE);
}
/*****
/*****
void Turn_Sys_OFF(void)
{
    State_ON = 0;
    //Change State Interrupts
    GPIO_disableInterrupt(GPIO_INT_XINT4);
    GPIO_enableInterrupt(GPIO_INT_XINT3);           // Enable ON Interrupt
    //Reset All Drivers
    Reset_All();
    //Control Reset DQ
    Init_PI_Outs(&PI_i_d, 0);
    Init_PI_Outs(&PI_i_q, 0);
    //Init_PI_Outs(&PI_v_dc, _IQ(0.0));
    //Change State Indicators
    GPIO_writePin(LED_ON_OFF, LED_OFF);
    //Change State PWM's
    EPWM_setActionQualifierContSWForceAction(EPWM_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    EPWM_setActionQualifierContSWForceAction(EPWM2_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    EPWM_setActionQualifierContSWForceAction(EPWM3_BASE, EPWM_AQ_OUTPUT_A, EPWM_AQ_SW_OUTPUT_LOW);
    setupEPWMActiveHigh(EPWM_BASE);
}
/*****
/*****

```

```

void setupEPWMActiveHighComplementary(uint32_t base)
{
    //
    // Use EPWMA as the input for both RED and FED
    //
    EPWM_setRisingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);
    EPWM_setFallingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);

    //
    // Set the RED and FED values
    //
    EPWM_setFallingEdgeDelayCount(base, 200);
    EPWM_setRisingEdgeDelayCount(base, 200);

    //
    // Invert only the Falling Edge delayed output (AHC)
    //
    EPWM_setDeadBandDelayPolarity(base, EPWM_DB_RED, EPWM_DB_POLARITY_ACTIVE_HIGH);
    EPWM_setDeadBandDelayPolarity(base, EPWM_DB_FED, EPWM_DB_POLARITY_ACTIVE_LOW);

    //
    // Use the delayed signals instead of the original signals
    //
    EPWM_setDeadBandDelayMode(base, EPWM_DB_RED, true);
    EPWM_setDeadBandDelayMode(base, EPWM_DB_FED, true);

    //
    // DO NOT Switch Output A with Output B
    //
    EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_A, false);
    //EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_B, false);

}

/*****
/*****
void setupEPWMActiveHigh(uint32_t base)
{
    //
    // Use EPWMA as the input for both RED and FED
    //
    EPWM_setRisingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);
    EPWM_setFallingEdgeDeadBandDelayInput(base, EPWM_DB_INPUT_EPWMA);

    //
    // Set the RED and FED values
    //
    EPWM_setFallingEdgeDelayCount(base, 200);
    EPWM_setRisingEdgeDelayCount(base, 200);

    //

```

```

// Do not invert the delayed outputs (AH)
//
EPWM_setDeadBandDelayPolarity(base, EPWM_DB_RED, EPWM_DB_POLARITY_ACTIVE_HIGH);
EPWM_setDeadBandDelayPolarity(base, EPWM_DB_FED, EPWM_DB_POLARITY_ACTIVE_HIGH);

//
// Use the delayed signals instead of the original signals
//
EPWM_setDeadBandDelayMode(base, EPWM_DB_RED, true);
EPWM_setDeadBandDelayMode(base, EPWM_DB_FED, true);

//
// DO NOT Switch Output A with Output B
//
EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_A, false);
EPWM_setDeadBandOutputSwapMode(base, EPWM_DB_OUTPUT_B, false);
}
/*****
/*****
void Init_PI_Outs(PI_REGS *p, float ref)
{
    p->Ref = ref;
    p->Out = 0;
    p->Ui = 0;
    p->Up = 0;
    p->U6 = 1;
}
/*****
/*****
void DCL_PIp(PI_REGS *v)
{
    v->Err = (v->Ref - v->Fdb)*((float32)State_ON); // v1
    v->Up = v->Kp * v->Err; // v2
    v->Ui = v->Ui + ((v->Ki * v->Up) * v->U6); // v4
    v->OutPreSat = v->Up + v->Ui; // v5

    if(v->OutPreSat > v->OutMax)
    {
        v->Out = v->OutMax;
        v->U6 = 0.0; // sat
    }
    else if(v->OutPreSat < v->OutMin)
    {
        v->Out = v->OutMin;
        v->U6 = 0.0; // sat
    }
    else
    {
        v->Out = v->OutPreSat;
        v->U6 = 1.0; // No sat
    }
}

```

```
    }  
}  
  
//  
// Configure DAC - Setup the reference voltage and output value for the DAC  
//  
void configureDAC(void)  
{  
    //  
    // Set VDAC as the DAC reference voltage.  
    // Edit here to use ADC VREF as the reference voltage.  
    //  
    DAC_setReferenceVoltage(DACB_BASE, DAC_REF_ADC_VREFHI);  
  
    //  
    // Enable the DAC output  
    //  
    DAC_enableOutput(DACB_BASE);  
  
    //  
    // Set the DAC shadow output to 0  
    //  
    DAC_setShadowValue(DACB_BASE, 0);  
  
    //  
    // Delay for buffered DAC to power up  
    //  
    DEVICE_DELAY_US(10);  
}
```

10.6. Planos

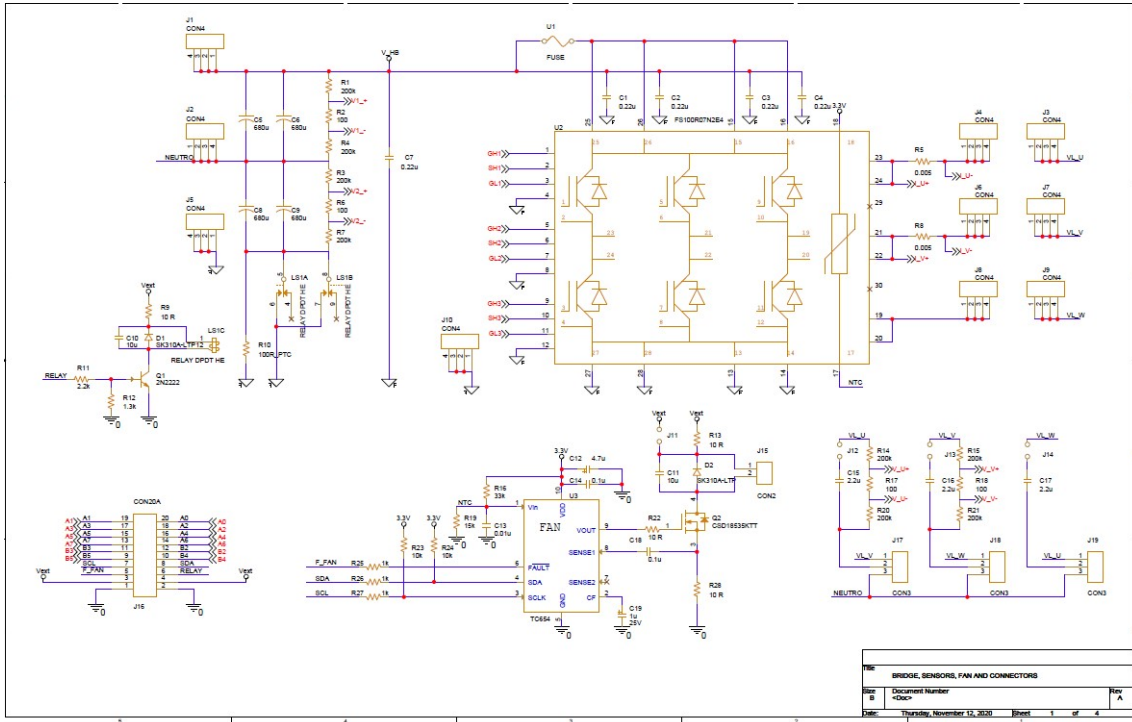


FIGURA 89: Etapa Puente H Fuente: Autor

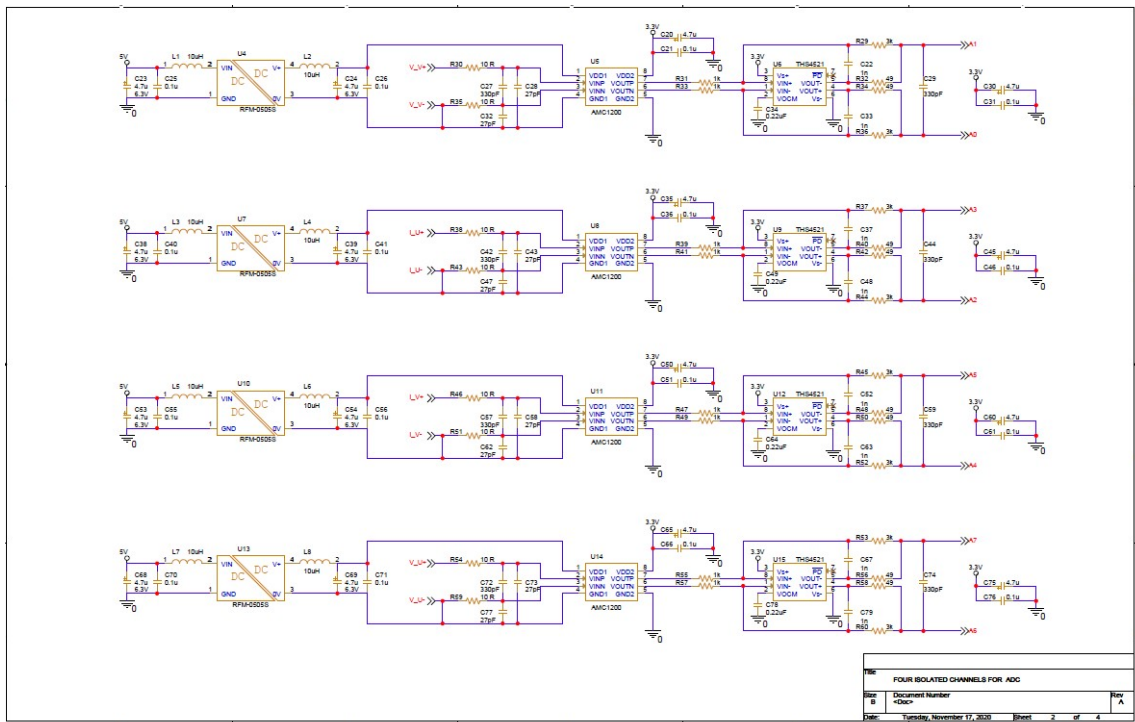


FIGURA 90: Etapa sensores Corriente y voltaje Fuente: Autor

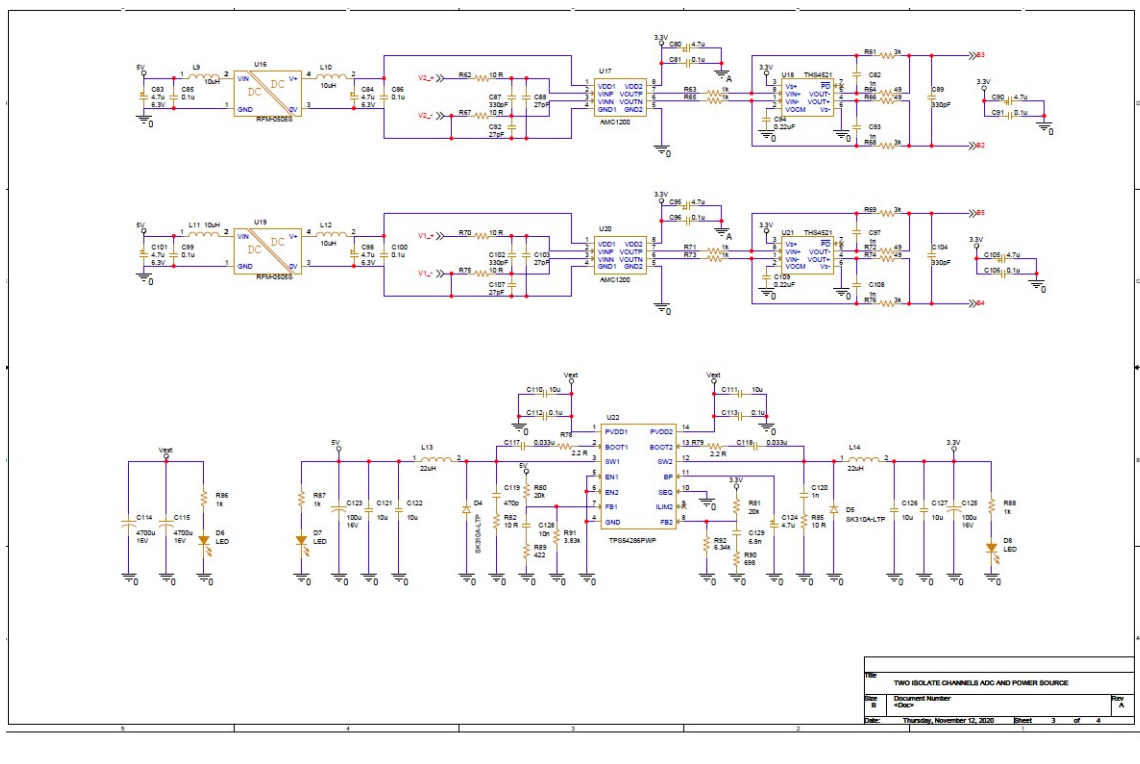


FIGURA 91: Etapa sensores de voltaje Fuente: Autor

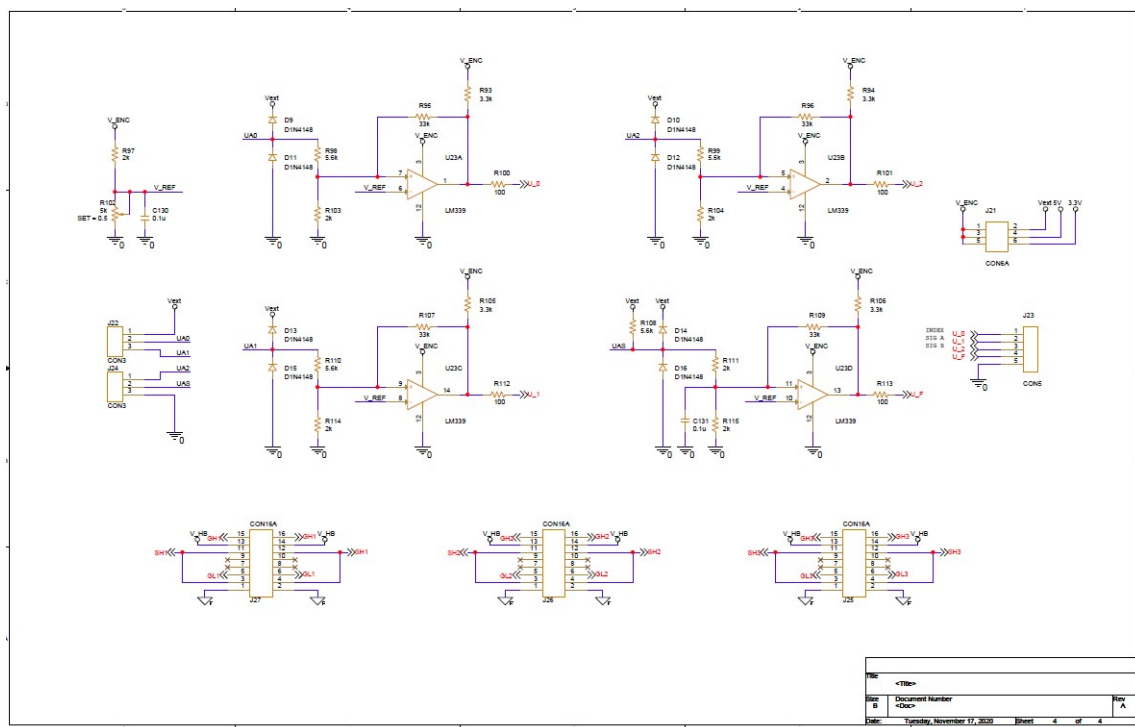


FIGURA 92: Etapa amplificadores **Fuente: Autor**

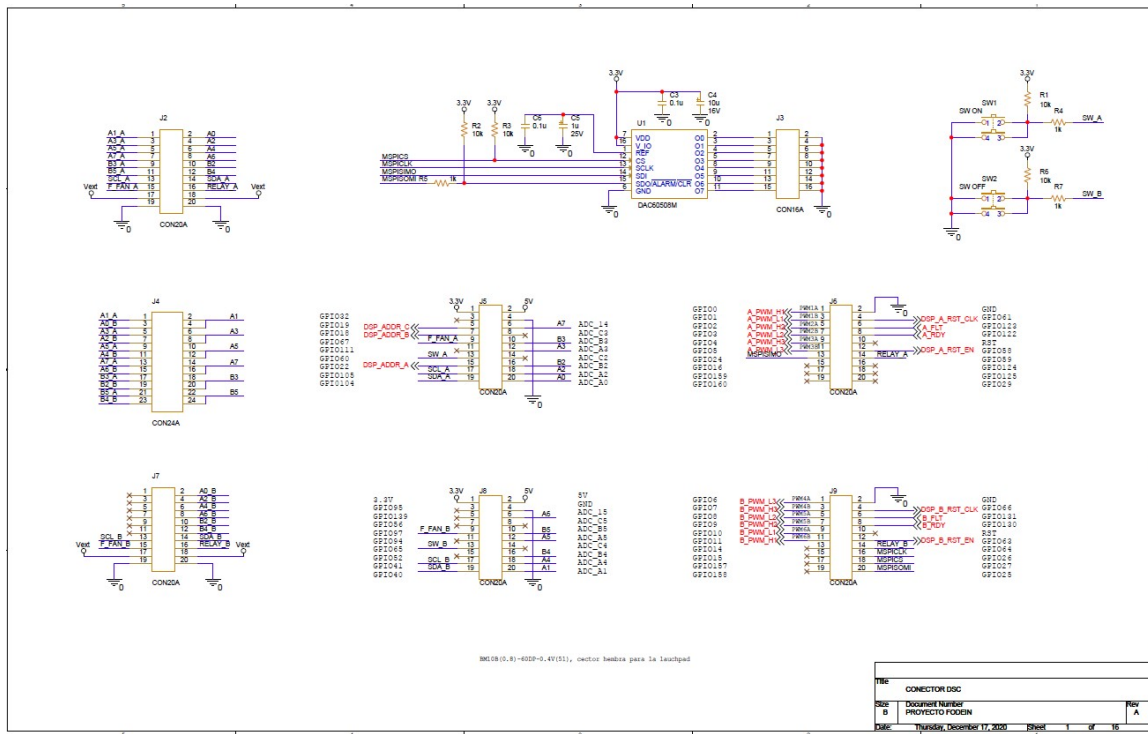


FIGURA 93: Conector DSP Fuente: Autor

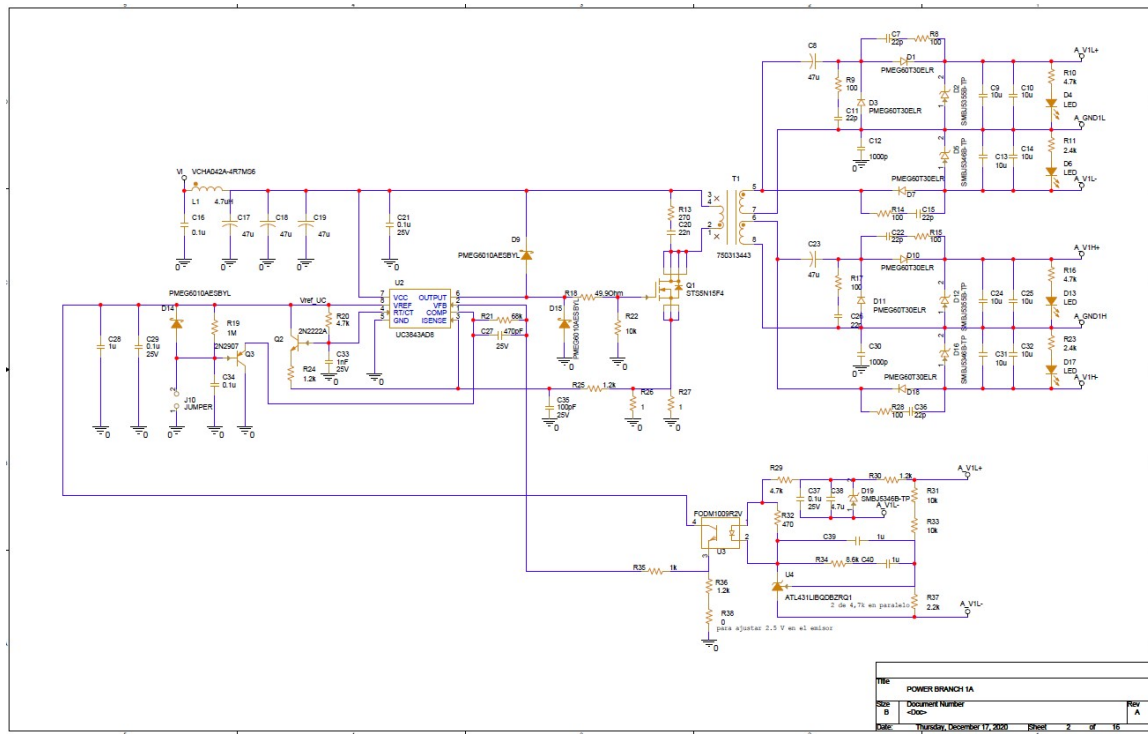


FIGURA 94: Power Branch 1A Fuente: Autor

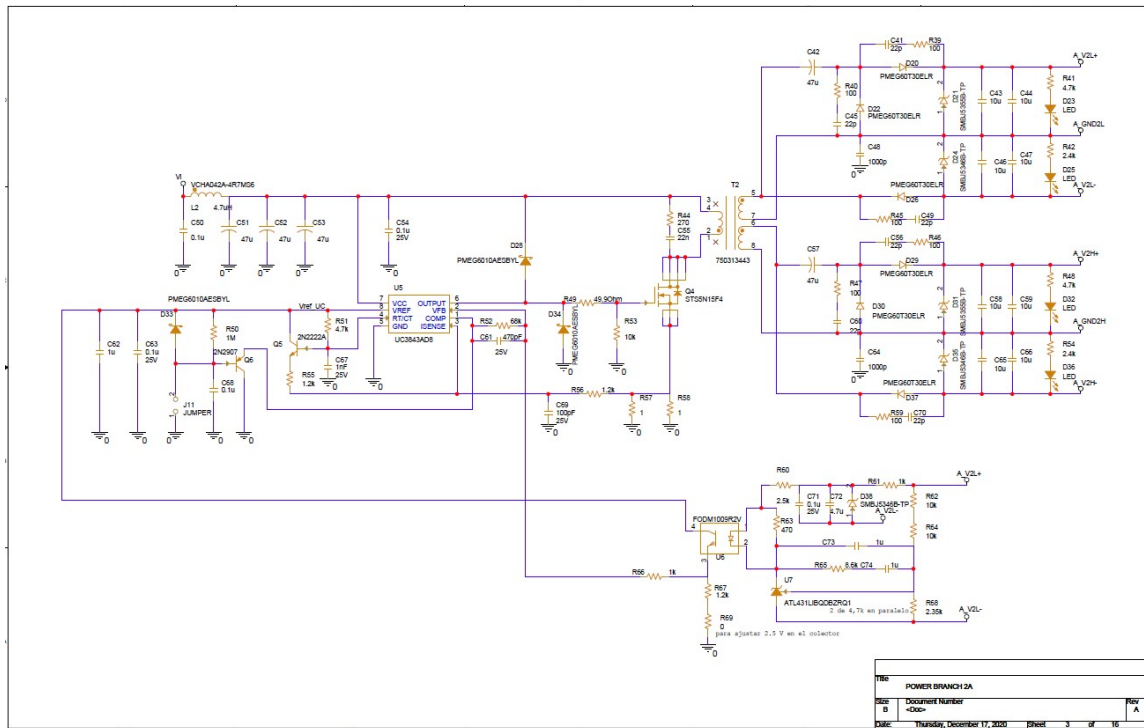


FIGURA 95: Power Branch 2A Fuente: Autor

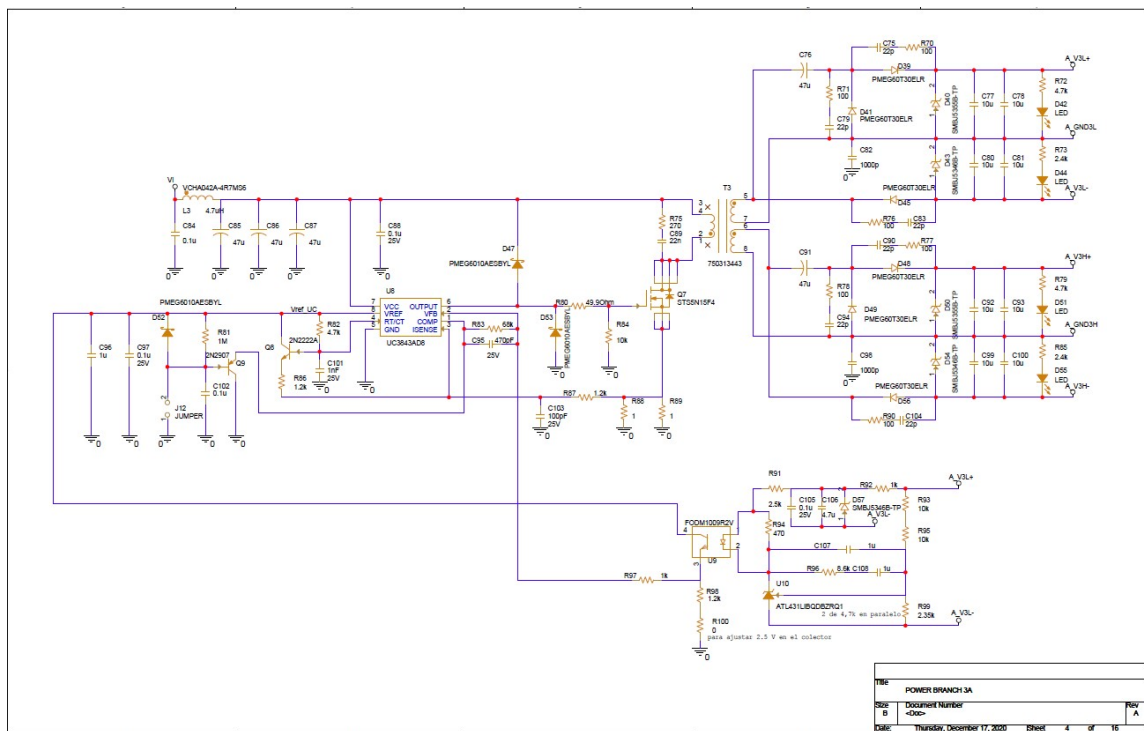


FIGURA 96: Power Branch 3A Fuente: Autor

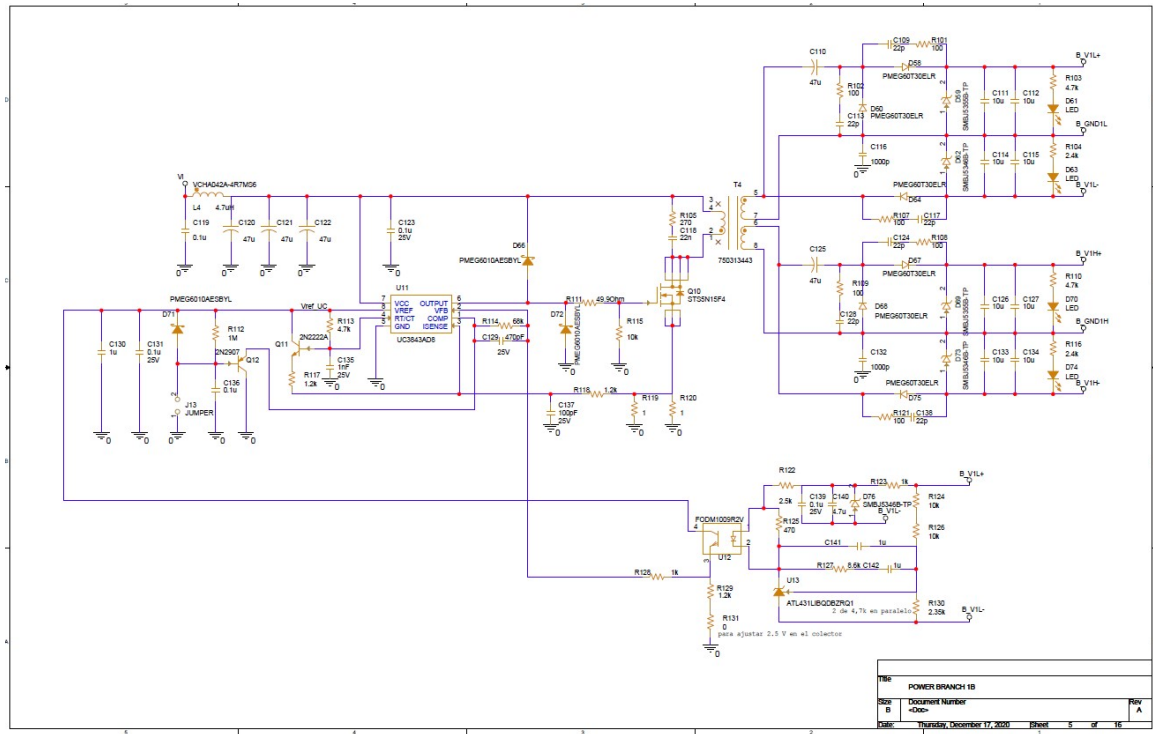


FIGURA 97: Power Branch 1B Fuente: Autor

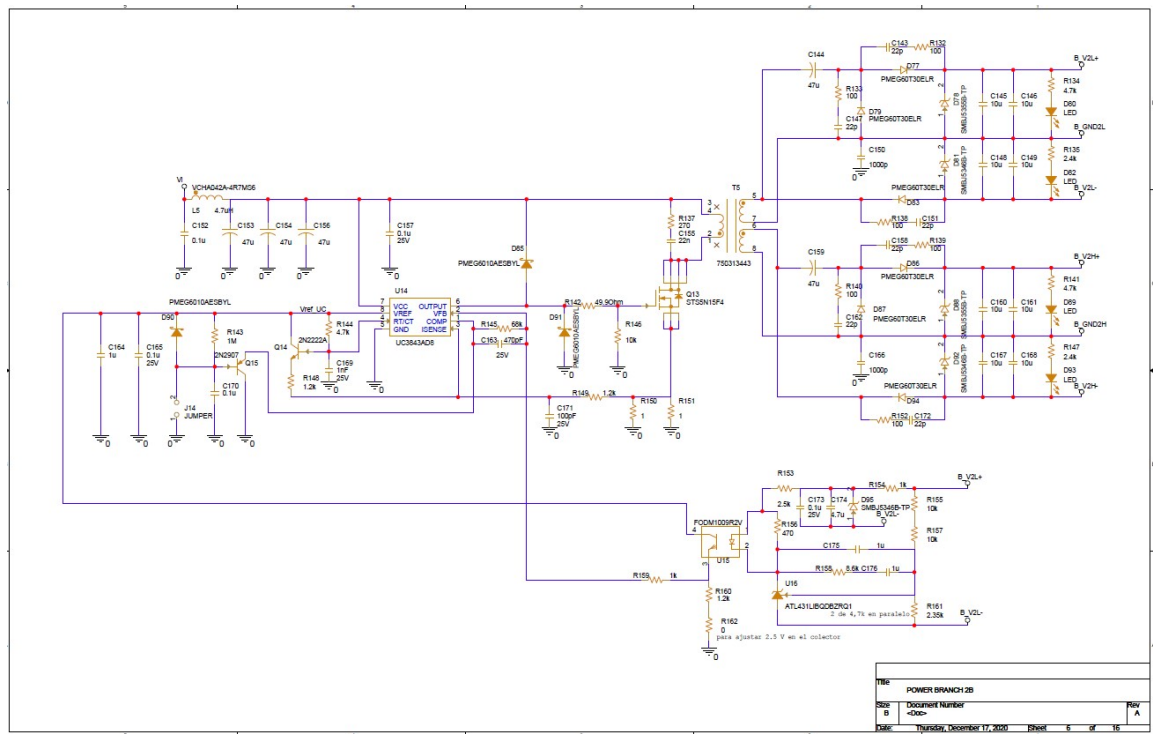


FIGURA 98: Power Branch 2B Fuente: Autor

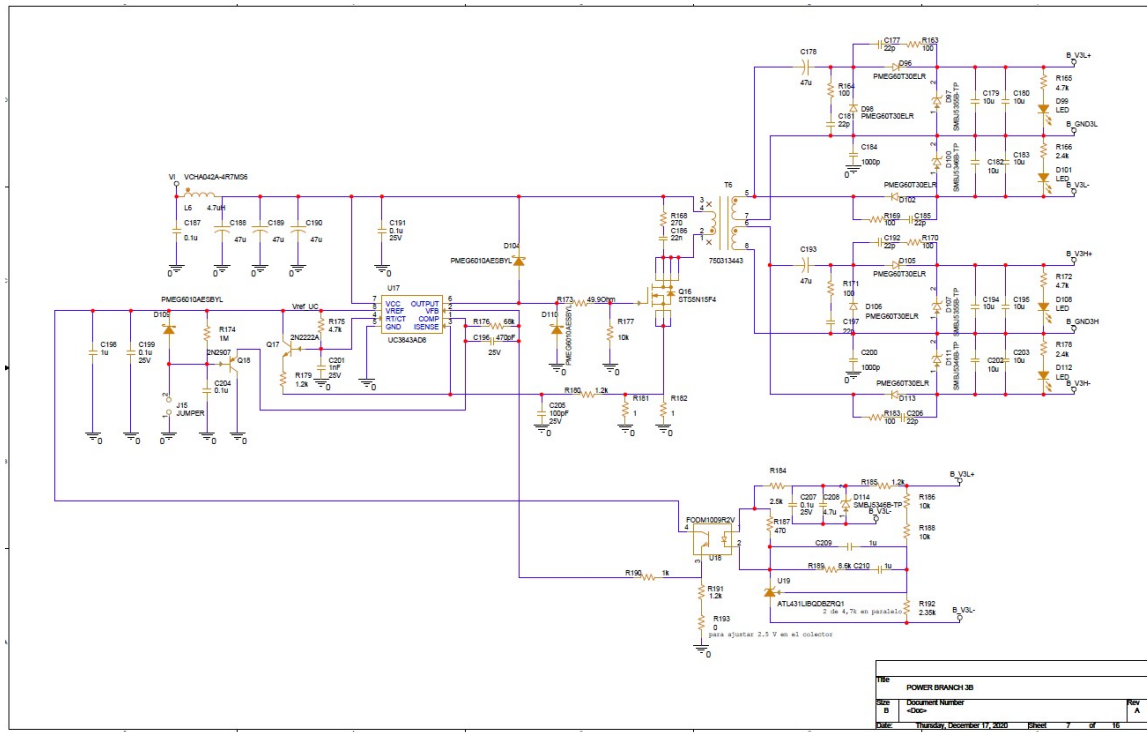


FIGURA 99: Power Branch 3B Fuente: Autor

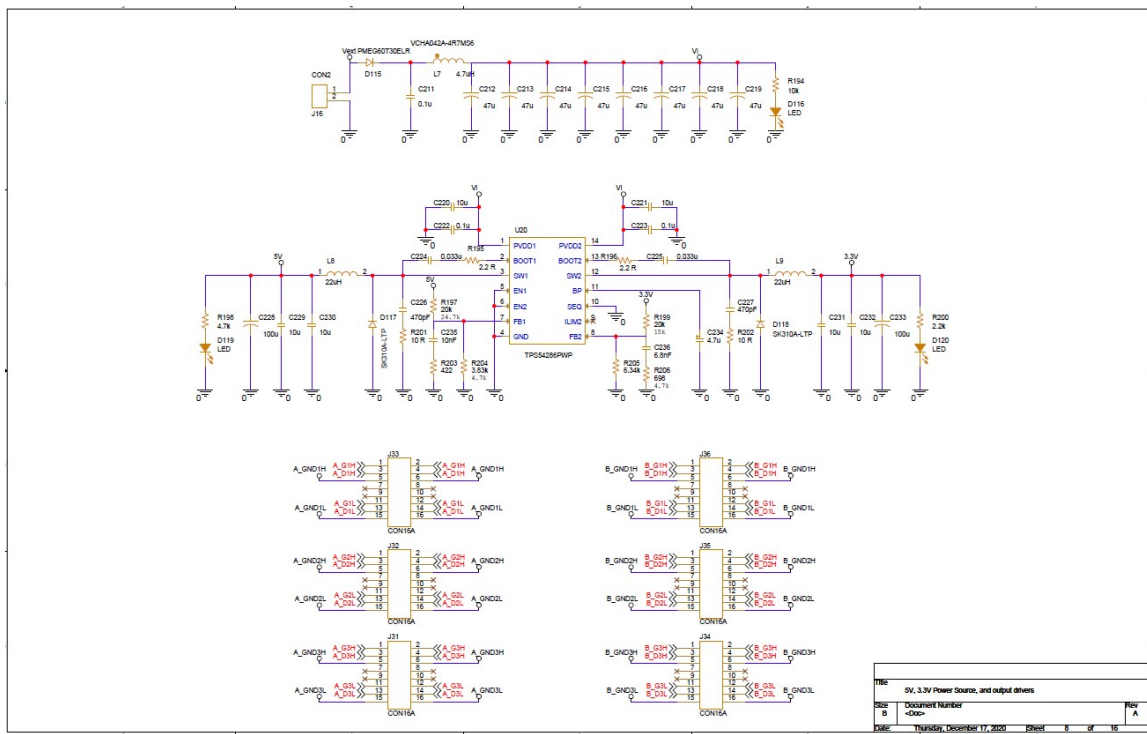


FIGURA 100: Power Source, and outputs drivers **Fuente: Autor**

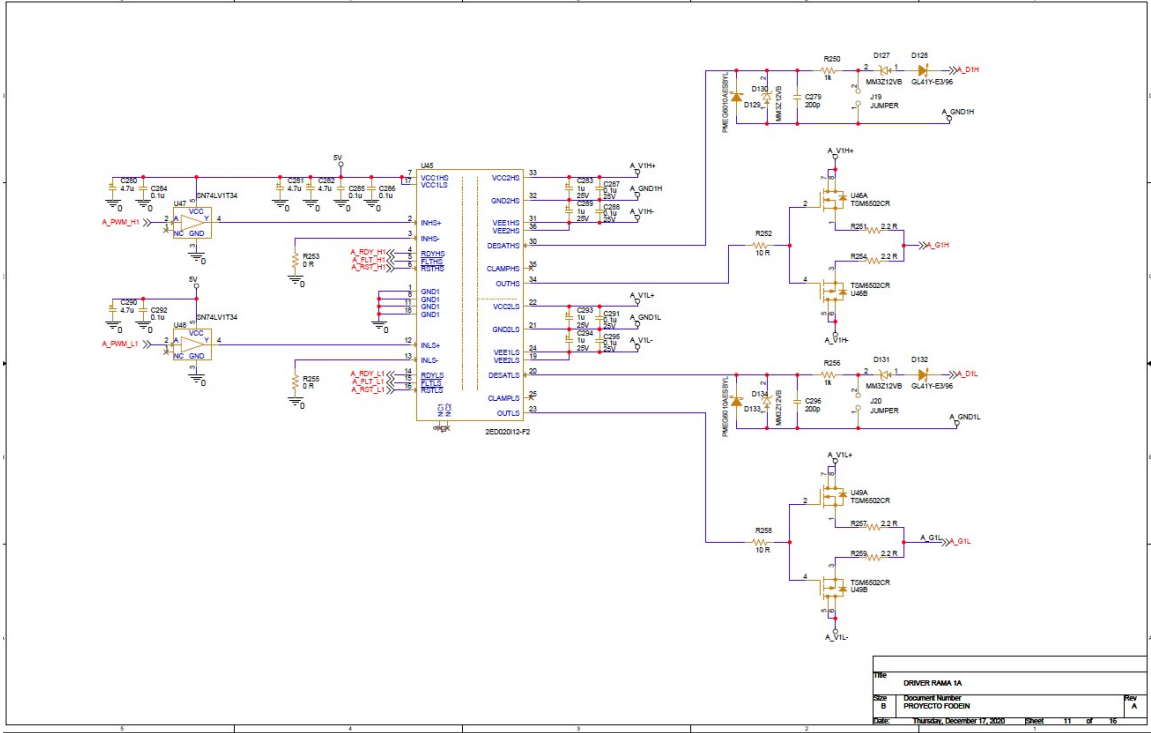


FIGURA 103: Driver 1 Fuente: Autor

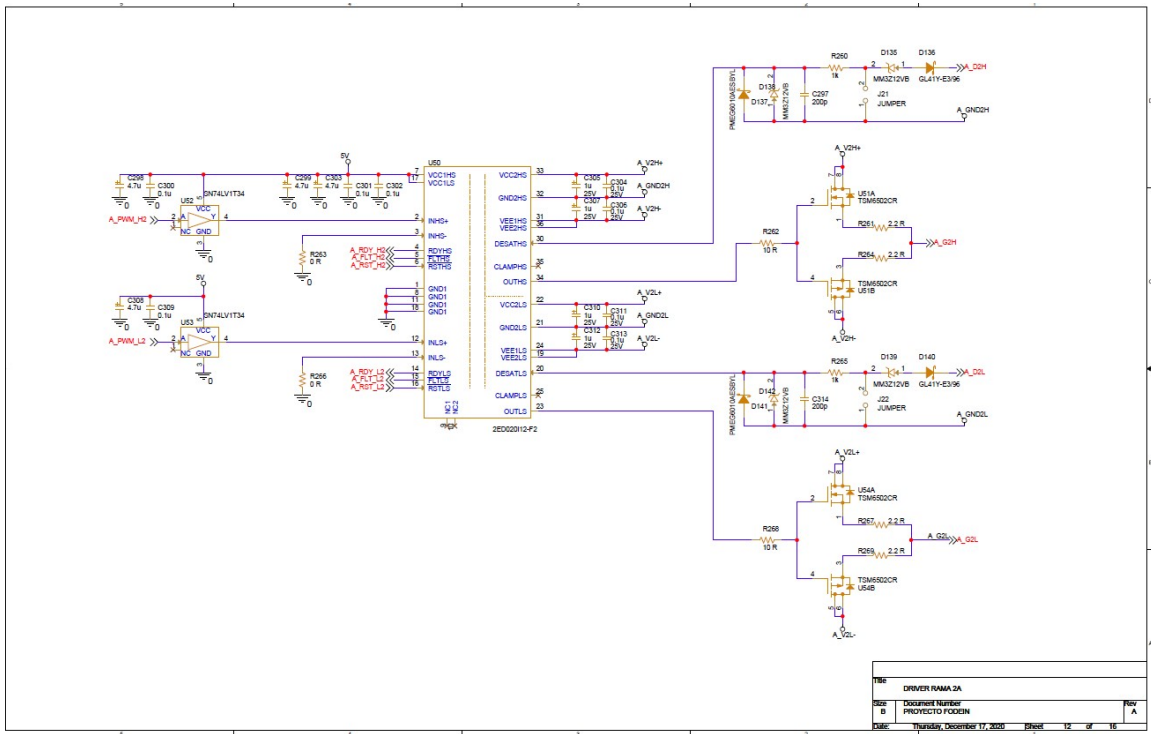


FIGURA 104: Driver 2 Fuente: Autor

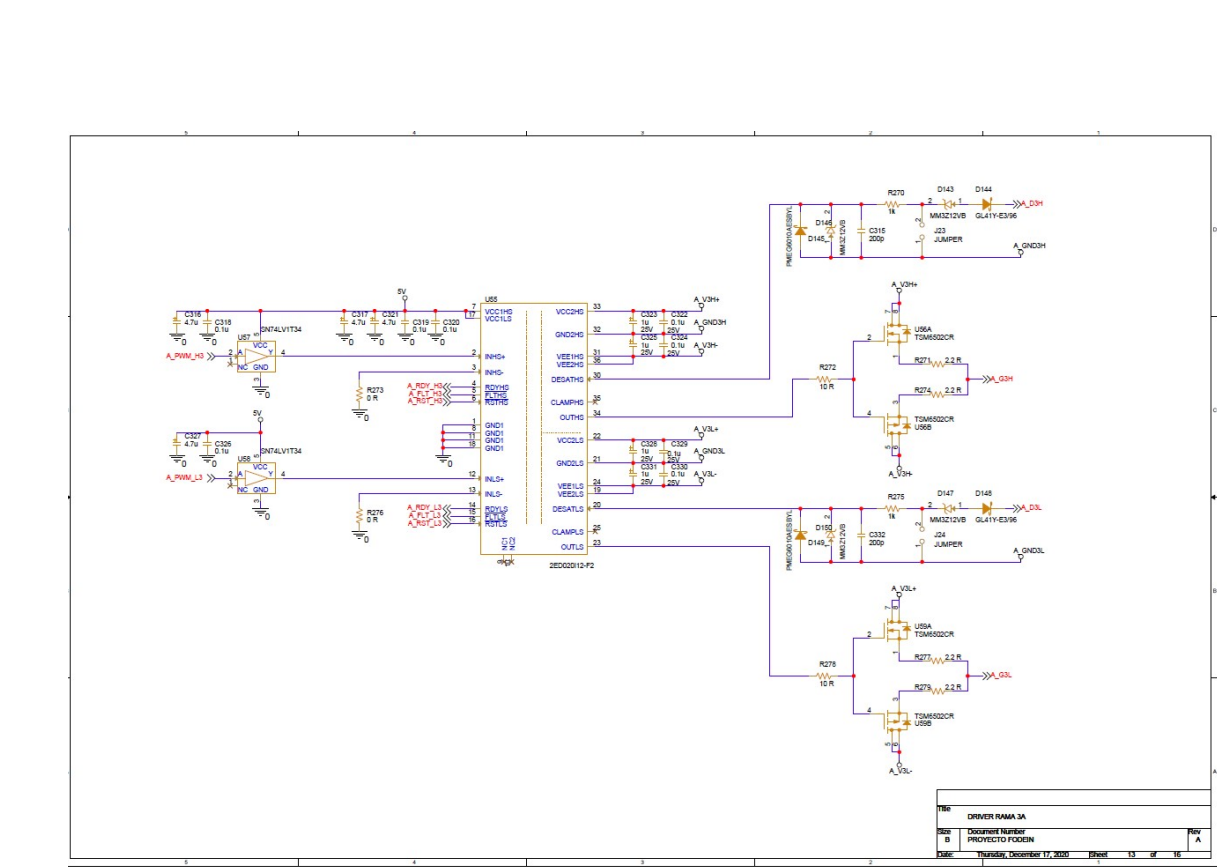


FIGURA 105: Driver 3 Fuente: Autor

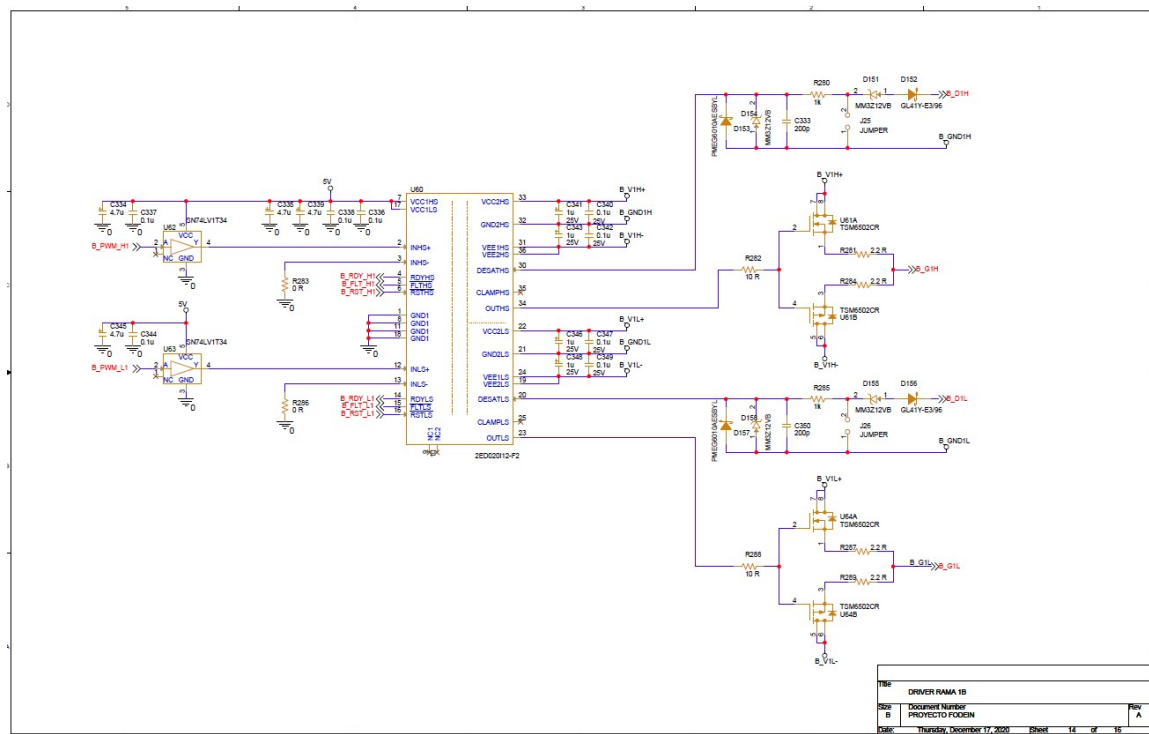


FIGURA 106: Driver 4 Fuente: Autor

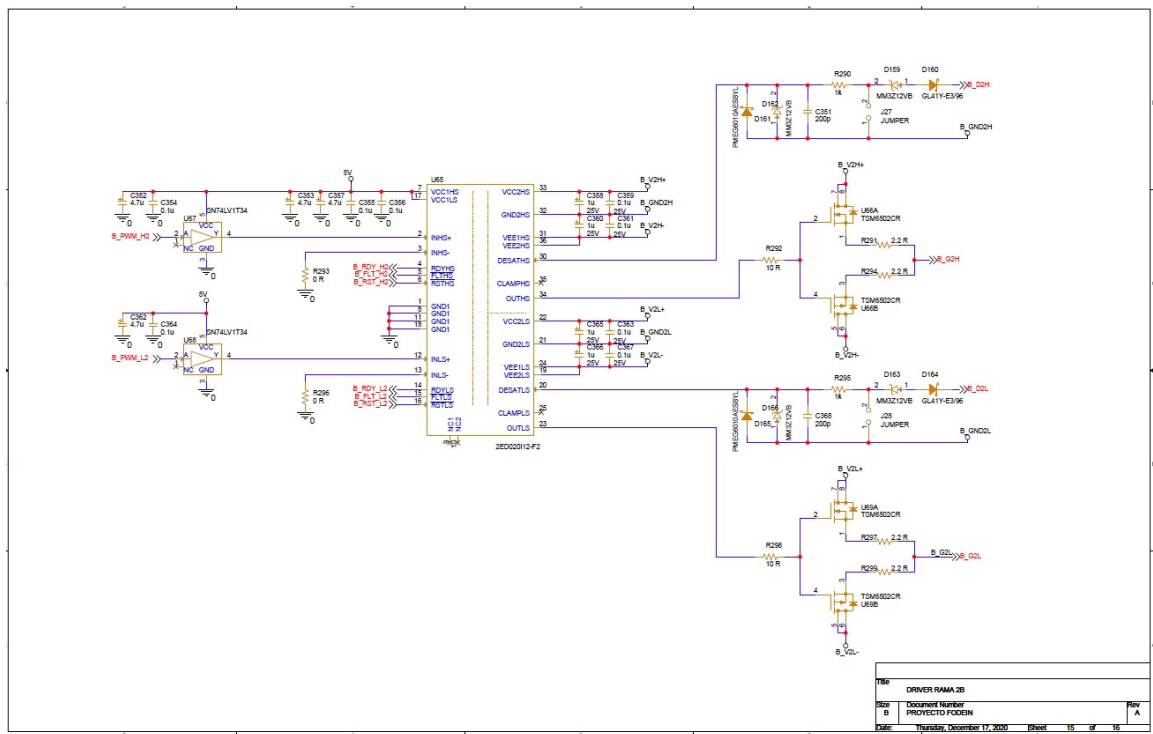


FIGURA 107: Driver 5 Fuente: Autor

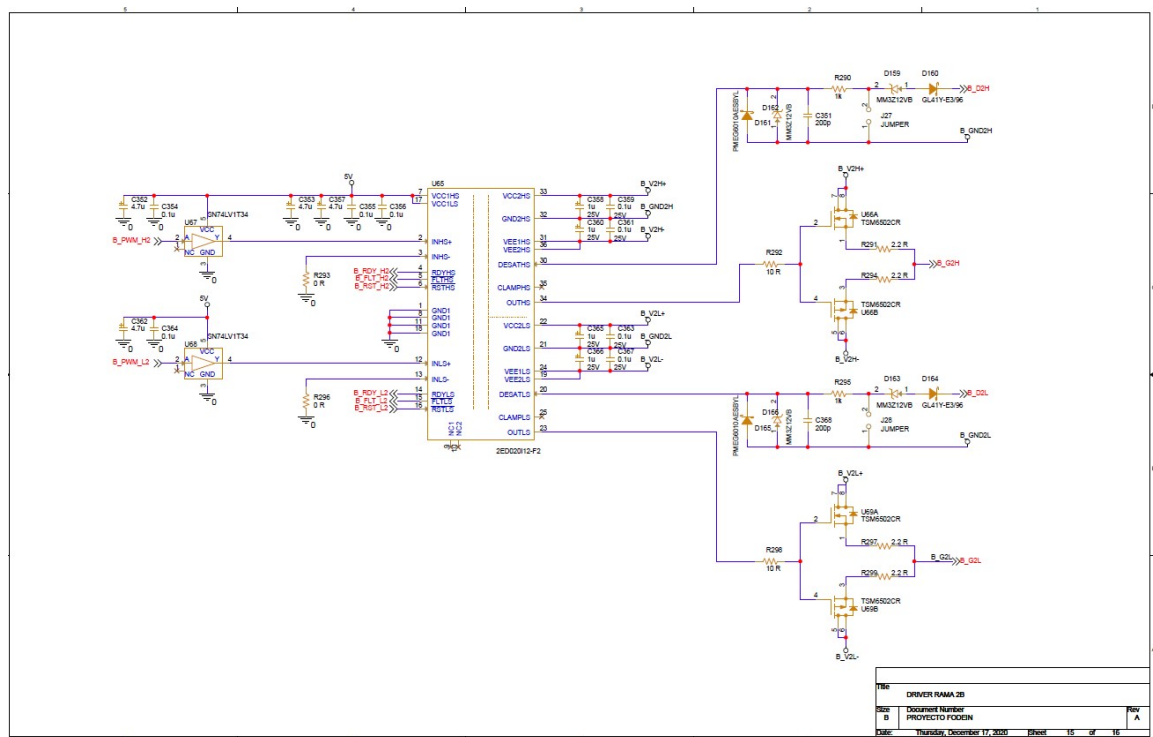


FIGURA 108: Driver 6 Fuente: Autor

Bibliografía

- [1] E. M. B. al-Iliktrūnīyāt (Cairo, I. of Electrical, E. E. E. Section, I. of Electrical, E. Engineers y E. I. C. on New Paradigms in Electronics Information Technology (5th : 2019 : Ghardah, «ACCS/PEIT 2019 : 2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) 2019 5th International Conference on New Paradigms in Electronics information Technology (PEIT) : Hurghada, Egypt 17-20 November 2019,»
- [2] P. CHÉVEZ, *Energías renovables y eficiencia energética. Análisis de medidas orientadas al sector residencial*, 2018.^a ed. 2018, págs. 0-190.
- [3] I. Rafael y T. Morales, *Tecnológico Nacional de México Tesis de Maestría Análisis Comparativo de las Estrategias de Control OCC y VCCR para la CFP Empleando un Convertidor Sepic*.
- [4] D. D. S. Guayaquil, S. Arguello y C. Javier, *UNIVERSIDAD CATÓLICA*.
- [5] V. V. Saldaña, *Citación recomendada*. dirección: https://ciencia.lasalle.edu.co/ing_electronica.
- [6] M. Tlapa, J. Upaep, E. Peralta et al., *SIMULACIÓN "HARDWARE IN THE LOOP" DE UN INVERSOR TRIFÁSICO CONECTADO A LA RED ELÉCTRICA*, 2017. dirección: <http://itcelaya.edu.mx/ojs/index.php/pistas>.
- [7] M. A. C. G. Asesor, M. A. R. Mesías, P. M. D. O. D. Jesus y O. G. Bellmunt, *Estabilidad de pequeña señal de un sistema de potencia con VSCs de tipos grid-following y grid-forming considerando los algoritmos VSM, droop control y VOC*.
- [8] Y. I. G. Landera, L. I. L. Viltre, G. I. Q. de Basterra y D. I. L. León, «Review of current controllers for LVRT conditions in a grid-connected photovoltaic system,» vol. 2022, pág. 1005,
- [9] Z. Emaeh, «Factors Promoting Renewable Energy Applications,» *Renewable Energy System Design*, págs. 1-32, 2014. DOI: 10.1016/B978-0-12-374991-8.00001-5.
- [10] *El problema energético mundial. Energías renovables y cambio climático – Revista Energy Management*. dirección: <https://e-management.mx/2019/11/16/el-problema-energetico-mundial-energias-renovables-y-cambio-climatico/>.
- [11] T. T. Chow, «Combined photovoltaic/thermal technology for building applications,» *Advances in Solar Heating and Cooling*, págs. 525-546, jun. de 2016. DOI: 10.1016/B978-0-08-100301-5.00019-9.
- [12] K. Zeb, I. Khan, W. Uddin et al., *A review on recent advances and future trends of transformerless inverter structures for single-phase grid-connected photovoltaic systems*, 2018.
- [13] C. Zhang, X. Zhao, X. Wang, X. Chai, Z. Zhang y X. Guo, «A Grid Synchronization PLL Method Based on Mixed Second- and Third-Order Generalized Integrator for DC Offset Elimination and Frequency Adaptability,» *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 6, págs. 1517-1526, 3 sep. de 2018, ISSN: 21686785. DOI: 10.1109/JESTPE.2018.2810499.
- [14] K. Geetha y B. V. Sreenivasappa, *H5-ZVR Single-Phase Grid-Tied Inverters for Photovoltaic Application*, 2021.

-
- [15] M. Waqas, T. Ahmed, R. M. Elavarasan et al., «DQ Transformation Based Control of Single-Phase Grid-Tied Inverter,» 2021.
 - [16] L. Carlos y D. E. Araujo, *CURSO DE ENGENHARIA ELÉTRICA INVERSOR MONOFÁSICO PARA GERAÇÃO FOTOVOLTAICA LOCAL COM BAIXA TAXA DE DISTORÇÃO HARMÔNICA (THD)*.
 - [17] I. Staff, *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*. IEEE, 2019, ISBN: 9781538678503.
 - [18] J. Andrés, L. Muñoz, M. Carlos y J. M. Casallas, *Clasificacion Reduccion Armonicos*.
 - [19] A. Sharma, A. Kashyap, A. Saxena, A. Govindharaj y A. Ambikapathy, «Active Power Flow Control in Inverter using Sliding Mode Controller,» Institute of Electrical y Electronics Engineers Inc., ago. de 2021, ISBN: 9781728185835. DOI: 10.1109/ASIANCON51346.2021.9544527.
 - [20] M. R. Montezuma, E. F. Colet y F. Angulo, «Sliding control applied to a single-phase current source inverter,» Institute of Electrical y Electronics Engineers Inc., 2021, págs. 122-126, ISBN: 9781665418836. DOI: 10.1109/CCAC51819.2021.9633286.
 - [21] N. U. of Computer, E. S. (Pakistan), I. of Electrical, E. E. I. Section., I. of Electrical y E. Engineers, *15th International Conference on Emerging Technologies (ICET) : National University of Computer and Emerging Sciences, Peshawar, Pakistan, 2-3 December, 2019*. ISBN: 9781728154046.
 - [22] G. Xu, Q. Teng, X. Zheng y X. Ma, «An Adaptive Backstepping Sliding Mode Control Strategy for Single-phase Grid-Connected Inverter System under Weak Grid,» Institute of Electrical y Electronics Engineers Inc., 2021, págs. 5684-5689, ISBN: 9781665426473. DOI: 10.1109/CAC53003.2021.9727360.
 - [23] «DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE INVERSOR MONOFÁSICO PARA PRÁCTICAS DE ELECTRÓNICA DE POTENCIA,» ISSN: 2021-2022.
 - [24] H. Lindblom, *Design and construction of a voltage source converter for a laboratory setup Masterprogram i förnybar elgenerering Master Programme in Renewable Electricity Production*. dirección: <http://www.teknat.uu.se/student>.
 - [25] I. Serrano, *Modelling and Control of VSC with a Photovoltaic Source Report Author: Escola Tècnica Superior d'Enginyeria Industrial de Barcelona*.
 - [26] J. D. Bastidas-Rodríguez, C. A. Ramos-Paja, J. D. Bastidas-Rodríguez y C. Ramos-Paja, *Este artículo puede compartirse bajo la licencia CC BY-ND 4.0 y se referencia usando el siguiente formato: "Types of inverters and topologies for microgrid applications" Tipos de inversores y topologías para aplicaciones de microrredes*, 2017.
 - [27] «Convertidores CC-CA: inversores de onda cuadrada,»
 - [28] *EXTENSIONISMO, INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA-CLAVES PARA EL DESARROLLO-VOLUMEN 2*.
 - [29] Pág. «Convertidores CC/CA para la conexión directa a la red de sistemas fotovoltaicos ANEXO B. Convertidores CC/CA Multinivel,»
 - [30] *TEMA 11 Inversores Índice*.
 - [31] *Solar Integration: Inverters and Grid Services Basics | Department of Energy*. dirección: <https://www.energy.gov/eere/solar/solar-integration-inverters-and-grid-services-basics>.
 - [32] A. S. R. P. A. F. Z. Rivera, *INVERSOR MULTINIVEL CON REDUCCIÓN DE PERDIDAS POR CONMUTACIÓN*, 2014.
 - [33] *Sintonización de un controlador PID mediante algoritmos BIO-Inspirados para la corrección del factor de potencia de un convertidor elevador*. dirección: <https://repository.usta.edu.co/handle/11634/38807>.

- [34] E. Segura, R. Morales y J. A. Somolinos, «Modelo dinámico y control no lineal para un convertidor de energía de las corrientes marinas de primera generación y dos grados de libertad,» Universidade da Coruna, ago. de 2019, págs. 42-47. DOI: 10.17979/spudc.9788497497169.042.
- [35] M. Apurva, R. Ulhe, D. Aissms, C. Pune, I. M. Vikas y V. Kulkarni, *PID Control based Boost Converter for Renewable Power Source*. dirección: www.ijert.org.
- [36] F. Illich y K. Gainza, «ARTÍCULO PROFESIONAL DE ALTO NIVEL MIGUEL ANGEL MARCHAN VACA CONTROL NO LINEAL DE CONVERTIDORES PARA MEJORAR LA ESTABILIDAD Y CONFIABILIDAD DE UNA MICRORRED CC CONECTADA A LA RED DE DISTRIBUCIÓN,» 2019.
- [37] «Tecnológico Nacional de México Tesis de Maestría,»
- [38] «Dialnet-AnalisisDeDiferentesTecnicasDeModulacionPWM-6828021,»
- [39] J. Yoe, R. Morales, E. S. Bustos, E. C. Calderón y M. Calixto-Rodriguez, *SEXTO SEMINARIO DE INGENIERIAS ENTRE CUERPOS ACADEMICOS (SEGUNDA EDICIÓN)*, 2023. dirección: <https://www.researchgate.net/publication/369472205>.
- [40] V. T. Le y H. H. Lee, «An Enhanced Model-Free Predictive Control to Eliminate Stagnant Current Variation Update for PWM Rectifiers,» *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, págs. 6804-6816, 6 dic. de 2021, ISSN: 21686785. DOI: 10.1109/JESTPE.2021.3058737.
- [41] T. D. T. Previo, A. L. Obtención, D. T. De, D. Alejandro y S. Freire, *ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA DISEÑO E IMPLEMENTACIÓN DE UN INVERSOR MULTINIVEL TRIFÁSICO TOPOLOGÍA DIODE-CLAMPED DE BAJA POTENCIA CON MODULACIÓN PD-PWM*.
- [42] D. Q. Muñoz y C. B. Stuardo, *UNIVERSIDAD DEL BÍO-BÍO*.
- [43] «Edición 18 / ENERO-JUNIO 2022 Sello editorial UTCJ // ISSN: 2448-7007,»
- [44] R. Bascuñana, D. Tutor, M. Román y J. Andrés, «UNIVERSITAT POLITÈCNICA DE VALÈNCIA Escuela Técnica Superior de Ingeniería Industrial,» ISSN: 2021/2022.
- [45] «4E.0458.IM,»
- [46] I. Staff, *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, 2019, ISBN: 9781728131856.
- [47] E. P. Superior y J. A. Ruiz, *UNIVERSIDAD DE ALCALÁ*.
- [48] E. David y E. Caicedo, «Control MPC distribuido para el despacho óptimo de potencia activa y reactiva en microrredes,»
- [49] «MPC,»
- [50] H. A. Pipino, E. Bernardi, F. Regional et al., *Formulación de un LPV-MPC Adaptativo para Procesos Industriales No Lineales*.
- [51] L. V. Paredes, *DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL PREDICTIVO PARA UNA PLANTA DE FLUJO UTILIZANDO UN CONTROLADOR DE AUTOMATIZACIÓN PROGRAMABLE PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE EXTENSIÓN LATACUNGA*.
- [52] *Sliding Mode Control of Switching Power Converters: Techniques and ...* - Siew-Chong Tan, Yuk-Ming Lai, Chi-Kong Tse - Google Libros. dirección: https://books.google.es/books?hl=es&lr=&id=5lvRBQAAQBAJ&oi=fnd&pg=PP1&dq=Sliding+Mode+Control+of+Switching+Power+Converters&ots=zmcDDy9_C9&sig=gARL0RyxLosseKCyUGvGb8s5lR0#v=onepage&q=Sliding%20Mode%20Control%20of%20Switching%20Power%20Converters&f=false.

-
- [53] H. Komurcugil, S. Biricik, S. Bayhan y Z. Zhang, «Sliding Mode Control: Overview of Its Applications in Power Converters,» *IEEE Industrial Electronics Magazine*, vol. 15, págs. 40-49, 1 mar. de 2021, ISSN: 19410115. DOI: 10.1109/MIE.2020.2986165.
 - [54] C. Ocho, *EL PROBLEMA DEL CHATTERING*.
 - [55] -Transformada de Park [3]. | Download Scientific Diagram. dirección: https://www.researchgate.net/figure/Figura-6-Transformada-de-Park-3_fig3_340809565.
 - [56] A. Roshan, *A DQ ROTATING FRAME CONTROLLER FOR SINGLE PHASE FULL-BRIDGE INVERTERS USED IN SMALL DISTRIBUTED GENERATION SYSTEMS*.
 - [57] «GrupoMEM,Potencia,»
 - [58] «GrupoMeM,Digital,»
 - [59] «DRUVER,»
 - [60] S. Tokat, M. S. Fadali y O. Eray, «A classification and overview of sliding mode controller sliding surface design methods,» en Springer International Publishing, 2015, vol. 24, págs. 417-439. DOI: 10.1007/978-3-319-18290-2_20.
 - [61] J. Zenteno-Torres, J. Cieslak, J. Dávila y D. Henry, «Sliding Mode Control with Application to Fault-Tolerant Control: Assessment and Open Problems,» *Automation*, vol. 2, págs. 1-30, 1 feb. de 2021. DOI: 10.3390/automation2010001.
 - [62] J. Liu, W. Luo, Y. Gao, Y. Yin y G. Sun, «Sliding mode control of grid-connected power converters for microgrid applications,» *Distributed Control Methods and Cyber Security Issues in Microgrids*, págs. 3-27, ene. de 2020. DOI: 10.1016/B978-0-12-816946-9.00001-3.
 - [63] L. Dong y S. K. Nguang, «Sliding mode control for multiagent systems with continuously switching topologies based on polytopic model,» *Consensus Tracking of Multi-Agent Systems with Switching Topologies*, págs. 87-105, ene. de 2020. DOI: 10.1016/B978-0-12-818365-6.00005-7.
 - [64] R. Zhang, M. Cardinal, P. Szczesny y M. Dame, *A Grid Simulator with Control of Single-phase Power Converters in D-Q Rotating Frame*.

title=Bibliografía,heading=bibintoc