UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Implementación y análisis de algoritmos de sincronización de fase, para fuentes de energía renovables en sistemas trifásicos

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Autores:

Andrés Antonio Segovia Vásquez
C.I: 110449514-6
andres4058@gmail.com

Director:

Ing. Luis Gerardo González Morales, PhD
C.I: 010390703-6

Cuenca – Ecuador
26 de julio de 2021

Resumen

Con la finalidad de lograr que dos o más fuentes externas estén sincronizadas entre sí y evitar daños en dispositivos finales; se han desarrollado algoritmos de sincronización que trabajen en conjunto con sistemas de electrónica de potencia y sistemas de control, que permitan sensar la señal de las Renewable Energy Sources (RES), para establecer de forma correcta los parámetros necesarios para inyectar energía a la red principal. Con este preámbulo; el presente trabajo de titulación muestra la implementación, análisis y comparación de los algoritmos de sincronización que más utilizados en este campo; tomando como objeto de estudio aquellos que son lazo de bloque de fase:

- con marco de referencia síncrono (dqPLL)
- con marco de referencia estacionario (PLL)
- con doble desacople en el marco de referencia síncrono (DDSRF-PLL)
- con doble integrador generalizado de segundo orden (DSOGI-PLL).

Para el cumplimiento de objetivos y desarrollo de la investigación, fue necesaria la combinación de herramientas computacionales como PSIM® y el entorno de desarrollo integrado (Code Composer Studio (CCS)), además de un elemento de hardware como un Procesador Digital de Señales (Digital Signal Processor (DSP)), este último permite recolectar los datos por medio de los puertos del módulo convertidor A/D, posteriormente realiza el procesamiento de la señal ejecutando los algoritmos programados. Para la adquisición de datos se requiere de la implementación de un sistema de potencia que permita acoplar la señal de la red con los parámetros de entrada de los puertos que soportan el DSP.

La obtención de resultados se resume en una primera prueba en la que en modo de simulación se insertó una distorsión en frecuencia a la RES; y en una segunda prueba para resultados prácticos, con la utilización de paneles solares poli-cristalinos del laboratorio de micro red de Universidad de Cuenca; estas pruebas reflejan la eficiencia de cada algoritmo expuesto a las mismas fallas. Dichos antecedentes a su vez fueron resultado de la visualización de su comportamiento y funcionamiento (uso del osciloscopio del software PSIM, comunicación serial entre el DSP y la PC, puertos PWM del dispositivo).

Finalmente, mediante un proceso de conexión y desconexión de la RES, se observó el comportamiento y estabilidad de cada algoritmo al someterse a este tipo de perturbaciones, y se realizó una comparativa de las repuestas obtenidas, teniendo como parámetro principal el tiempo de establecimiento.

Palabras clave : RES. PSIM. DSP. CCS. Electrónica de potencia. Sistemas de Control. algoritmos de sincronización. lenguaje C++. Módulo convertidor A/D.

Abstract

In order to achieve that two or more external electrical energy sources are synchronized with each other and prevent damage to end devices, synchronization algorithms have been developed that work in conjunction with power electronics systems and control systems that allow the detection of the power signal of the Renewable Energy Sources (RES) to correctly establish the necessary parameters to inject energy into the main network. With this preamble, the present degree work shows the implementation, analysis and comparison of the most used synchronization algorithms in this field, taking as object of study those that are phase block loops:

- Synchronous reference Frame (dqPLL)
- Stationary reference Frame ($\alpha\beta$-PLL)
- Double decoupling synchronous reference frame (DDSRF-PLL)
- Dual second-order generalized integrator (DSOGI-PLL).

In order to fulfill the objectives and development of the research, it was necessary to combine computational tools such as PSIM® and a integrated development environment)Code Composer Studio (CCS)) and a hardware element such as a Digital Signal Processor (DSP). The latter allows data to be collected through the ports of the A/D converter module, and to carry out signal processing by executing the programmed algorithms. Data acquisition requires the implementation of a power system that allows coupling the network signal with the input parameters of the ports that support the DSP.

The obtaining of results is summarized in a first test in which in simulation mode a frequency distortion was inserted at the RES; and in a second test for practical results, with the use of polycrystalline solar panels from the micro-grid laboratory of the University of Cuenca. These tests reflect the efficiency of each algorithm exposed to the same faults and disturbances. These antecedents in turn were the result of the visualization of its behavior and operation (use of the PSIM software oscilloscope, serial communication between the DSP and the PC, PWM ports of the device).
Finally, through a process of connection and disconnection of the RES, the behavior and stability of each algorithm when subjected to this type of disturbance was observed, and a comparison of the responses obtained was made with the establishment time as the main parameter.
Keywords: RES. PSIM. DSP. CCS. Power electronics. Control Systems. Synchronization Algorithms. C++ language. A/D converter module.

Índice general

Resumen 1

Abstract 2

Índice general 3

Índice de figuras 7

Índice de tablas 11

Cláusula de Propiedad Intelectual 12

Cláusula de licencia y autorización para publicación en el Repositorio Institucional 13

Certifico 14

Dedicatoria 15

Agradecimientos 16

Abreviaciones y acrónimos 17

1. Introducción 18
- 1.1. Introducción . . . 18
- 1.2. Antecedentes . . . 19
- 1.3. Alcance . . . 19
- 1.4. Objetivos . . . 20
  - 1.4.1. Objetivo general . . . 20
  - 1.4.2. Objetivos específicos . . . 20
- 1.5. Estructura del documento . . . 20

2. Marco Teórico 22
- 2.1. Fuentes de Generación de Energía Eléctrica Renovables y No Renovables . . . 22
  - 2.1.1. Fuentes de Generación de Energía no renovables . . . 23
    - 2.1.1.1. Petróleo . . . 23
    - 2.1.1.2. Carbón . . . 23
    - 2.1.1.3. Gas Natural . . . 23
    - 2.1.1.4. Nuclear . . . 24

2.1.2. Fuentes de Generación de Energía Renovables 24
2.1.2.1. Hidráulica 24
2.1.2.2. Eólica 25
2.1.2.3. Solar 25
2.1.2.4. Biomasa 25
2.2. Estructura de los Sistemas de Transmisión y Distribución Eléctricos 26
2.3. Aspectos Regulatorios para la Transmisión y Distribución de Energía Eléctrica 28
2.3.1. Nivel de voltaje 29
2.3.2. Perturbaciones rápidas de voltaje (Flicker) 29
2.3.3. Distorsión armónica de voltaje 29
2.3.4. Desequilibrio de voltaje 30
2.3.5. Frecuencia 30
2.3.6. Sincronización 30
2.3.7. Reconexión al sistema de potencia eléctrico 30
2.4. Algoritmos de Sincronización de Red 31
2.4.1. Detección de cruce por cero 31
2.4.2. Algoritmos de lazo de bloqueo de fase (PLL) 31
2.4.2.1. Lazo de bloque de fase con marco de referencia síncrono (dqPLL) . . 35
2.4.2.2. Lazo de bloque de fase con marco de referencia estacionario ($\alpha\beta$PLL) 36
2.4.2.3. Lazo de bloqueo de fase con doble desacople en el marco de referencia síncrono (DDSRF-PLL) 36
2.4.2.4. Lazo de bloqueo de fase con desacople en el marco de referencia estacionario (DSRF-PLL) 39
2.4.2.5. Lazo de bloqueo de fase con doble integrador generalizado de segundo orden (DSOGI-PLL) 39
2.5. Comparación de desempeño mediante herramienta computacional PSIM 40
2.6. Procesador digital de señales (DSP) TMDSCNCD28335 41
2.7. Estudio creador de código (CCS) 41

3. Diseño y Desarrollo Experimental 42
3.1. Adquisición de datos por medio del DSP 42
3.1.1. Acondicionamiento de la señal para lectura del DSP 43
3.1.1.1. Diseño e implementación de la fuente simétrica 44
3.1.1.2. Diseño e implementación del circuito de acoplamiento 45
3.1.2. Configuración de PSIM 47
3.1.3. Generación de código C/C++ 49
3.1.4. Configuración de Code Composer Studio (CCS) 50
3.2. Configuración del puerto serial para la visualización de datos por medio de la interfaz Serial Communication Interface (SCI) 50
3.2.1. Diseño, Implementación y Configuración de la comunicación serial entre PC y DSP 50
3.2.2. Implementación de la conexión para la comunicación serial en PSIM 52
3.3. Visualización de datos 53
3.3.1. Utilizando el Osciloscopio de PSIM 53
3.3.2. Utilizando el osciloscopio digital físico 54

3.4. Implementación de algoritmos de sincronización de fase 55
3.4.1. Normalización de las señales de entrada 56
3.4.2. Establecimiento de parámetros del Proporcional-Integral (PI) 56
3.4.3. Distorsión de frecuencia en la señal de entrada 58
3.4.4. Distorsión armónica en la señal de entrada 59
3.4.5. Diseño e implementación del algoritmo dqPLL 60
3.4.6. Diseño e implementación del algoritmo $\alpha\beta$PLL 60
3.4.7. Diseño e implementación del algoritmo DDSRF-PLL 61
3.4.8. Diseño e implementación del algoritmo DSOGI-PLL 63

4. Resultados, Conclusiones y Recomendaciones 65
4.1. Esquema para la adquisición de datos 65
4.2. Señales de entrada, tomadas a la salida del circuito de acondicionamiento 68
4.3. Normalización de la señal para la entrada de los algoritmos 70
4.4. Resultados de los algoritmos propuestos 71
4.4.1. Distorsión en frecuencia 71
4.4.1.1. dq-PLL 71
4.4.1.2. $\alpha\beta$-PLL 71
4.4.1.3. DDSRF-PLL 72
4.4.1.4. DSOGI-PLL 72
4.4.2. Distorsión armónica 73
4.4.2.1. dq-PLL 73
4.4.2.2. $\alpha\beta$-PLL 73
4.4.2.3. DDSRF-PLL 74
4.4.2.4. DSOGI-PLL 74
4.4.3. Desconexión y Conexión de la fuente renovable 75
4.4.3.1. dq-PLL 75
4.4.3.2. $\alpha\beta$-PLL 75
4.4.3.3. DDSRF-PLL 76
4.4.3.4. DSOGI-PLL 77
4.5. Análisis y discusión de los resultados 78
4.6. Conclusiones y Recomendaciones 79
4.6.1. Conclusiones 79
4.6.2. Recomendaciones 80
4.7. Trabajos futuros 81

A. Elaboración del proyecto, diseño de componentes para fabricación de la PCB y generación del archivo .gerber en Altium 82
A.1. Elaboración del proyecto en Altium 82
A.2. Diseño esquemático del elemento en Atium 86
A.3. Diseño del footprint del componente en Altium 89
A.4. Elaboración del archivo .Gerber 94

B. Configuración de CCS **104**
B.1. Configuración de CCS para la importación del código generado en PSIM 104
B.2. Grabado del código en la tarjeta de control F28335 107

C. Código para la lectura de datos **110**

D. Código para el cálculo de los parámetros kp y Ti del controlador **116**

Bibliografía **117**

Índice de figuras

2.1. Topología básica de conexión de fuentes renovables con la red principal [17] 27
2.2. Esquema de una sistema de Generación, Transmisión y Distribución de la red Eléctrica. [20] 27
2.3. Estructura general de un Sistema Eléctrico [20] 28
2.4. Detección de cruce por cero de una señal senoidal ruidosa [35] 31
2.5. Diagrama de Bloques básico de un PLL 32
2.6. Topología básica de los PLL ([17]) 32
2.7. Diagrama de Bloques de un dqPLL [33] 35
2.8. Diagrama de Bloques de un $\alpha\beta$PLL [17] 36
2.9. Diagrama de Bloques de un Decoupled Double Synchronous Reference Frame-Phase Locked Loop (DDSRF-PLL) [17] 38
2.10. Red de desacoplamiento para secuencias positivas y negativas, utilizada en DSRF-PLL [33] 38
2.11. Diagrama de Bloques de un DSRF-PLL [44] 39
2.12. Estructura de bloques del algoritmo SOGI [42] 40
2.13. Estructura de bloques del algoritmo DSOGI-PLL [42] 40
2.14. Placa (DSP) TMDSCNCD28335 41

3.1. Estructura básica del funcionamiento del sistema de sincronización de fase 43
3.2. Circuito esquemático de la fuente de alimentación 44
3.3. Circuito de acondicionamiento de señal 45
3.4. Diseño esquemático del circuito de acoplamiento 47
3.6. Bloque Convertidor Analógico/Digital de PSIM 48
3.5. Librerías de PSIM para la tarjeta F28335 48
3.7. Configuración del control de simulación 49
3.8. Lectura de las señales de ingreso, para su análisis 49
3.9. Configuración de CCS para la compilación y la exportación a la tarjeta 50
3.10. Conexión de los puertos de la PC y DSP 51
3.11. Diagrama de conexión entra la PC y el DSP 51
3.12. Conexión física para la comunicación serial 51
3.13. Configuración del puerto de comunicación 52
3.14. Configuración General para el módulo SCI 52
3.15. Configuración de los módulos I/0 para la interfaz SCI 53
3.16. Interfaz de usuario del Osciloscopio 54
3.17. Configuración de la tarjeta para el uso de los puertos digitales como PWM 54

3.18. Configuración de los parámetros del puerto PWM 55
3.19. Circuito integrador como filtro paso bajo 55
3.20. Paso de un sistema de coordenadas de 3 fases a un sistema estacionario ortogonal por medio de la transformada de Clarke 56
3.21. Normalización de la señal trifásica, utilizando las componentes $\alpha\beta$ 56
3.22. Respuesta al impulso del sistema con un tiempo de establecimiento de 3ms 57
3.23. Raíces y Polos del sistema utilizados para análisis de estabilidad 57
3.24. Diagrama de bode para análisis de estabilidad del sistema 58
3.25. Sub-circuito para la distorsión de frecuencia en la simulación 58
3.26. Respuesta de la distorsión de frecuencia para simulación 59
3.27. Distorsión armónica para las tres fases de entrada 59
3.28. Señales trifásicas con distorsión armónica 60
3.29. Esquema del algoritmo dq-PLL en PSIM 60
3.30. Esquema del algoritmo $\alpha\beta$PLL en PSIM 61
3.31. Esquema del algoritmo DDSRFPLL en PSIM 61
3.32. Transformada $\alpha\beta$ a dq 62
3.33. Esquema de desacople para la secuencia positiva 62
3.34. Esquema de desacople para la secuencia negativa 63
3.35. Esquema del algoritmo DSOGIPLL en PSIM 63
3.36. Esquema del algoritmo SOGI-PLL en PSIM para la fase α 64
3.37. Esquema del algoritmo SOGI-PLL en PSIM para la fase β 64

4.1. Tablero inversor para tensión de PANELES SOLARES POLI-CRISTALINOS 66
4.2. Voltajes de salida del inversor para la entrada del circuito de acoplamiento 66
4.3. Ángulos de desfases de las señales de salida del inversor para la entrada del circuito de acoplamiento 67
4.4. Conexión al breaker principal de los transformadores 67
4.5. Conexión al circuito de acondicionamiento para el Digital Signal Processor (DSP) . . . 68
4.6. Conexión del Osciloscopio para las pruebas de la señal de entrada 68
4.7. Señales de entrada al DSP tomadas a la salida del circuito de acondicionamiento . . . 69
4.8. Señal trifásica de entrada en los puertos analógicos del DSP 69
4.9. Componentes $\alpha\beta$ de la transformada de Clarke 70
4.10. Normalización de las componentes $\alpha\beta$ de la transformada de Clarke 70
4.11. Respuesta del algoritmo dqPLL con distorsión de fase 71
4.12. Respuesta del algoritmo $\alpha\beta$PLL con distorsión de fase 72
4.13. Respuesta del algoritmo DDSRFPLL con distorsión de fase 72
4.14. Respuesta del algoritmo DSOGI-PLL con distorsión de fase 73
4.15. Respuesta del algoritmo dq-PLL con presencia de armónicos 73
4.16. Respuesta del algoritmo $\alpha\beta$-PLL con presencia de armónicos 74
4.17. Respuesta del algoritmo DDSRF-PLL con presencia de armónicos 74
4.18. Respuesta del algoritmo DSOGI-PLL con presencia de armónicos 74
4.19. Esquema de conexión para la captura de datos del algoritmo dq-pll 75
4.20. Respuesta del algoritmo dq-PLL con los paneles solares 75
4.21. Esquema de conexión para la captura de datos del algoritmo $\alpha\beta$-PLL 76

4.22. Respuesta del algoritmo $\alpha\beta$-PLL con los paneles solares 76
4.23. Esquema de conexión para la captura de datos del algoritmo DDSRF-PLL 76
4.24. Respuesta del algoritmo DDSRF-PLL con los paneles solares 77
4.25. Esquema de conexión para la captura de datos del algoritmo DSOGI-PLL 77
4.26. Respuesta del algoritmo DSOGI-PLL con los paneles solares 78

A.1. Dirección para la creación de un nuevo proyecto 83
A.2. Configuración de la dirección y nombre del proyecto 83
A.3. Añadir archivo esquemático en el proyecto 84
A.4. Añadir archivo PCB en el proyecto 84
A.5. Añadir archivo esquemático para librería en el proyecto 85
A.6. Añadir archivo PCB para librería en el proyecto 85
A.7. Archivos añadidos al proyecto 86
A.8. Propiedades para los archivos esquemáticos de las librerías 86
A.9. Añadir y configurar nombre del componente 87
A.10. Crear superficie del componente 87
A.11. Propiedades de diseño del componente 87
A.12. Añadir pines en el componente 88
A.13. Añadir pines en el componente 88
A.14. Diseño final del componente 89
A.15. Propiedades para los archivos PCB de las librerias 89
A.16. Añadir componente para diseño PCB 90
A.17. Añadir pin al componente 90
A.18. Propiedades de los pines para el componente 91
A.19. Diseño del contorno del componente para la PCB 91
A.20. Aplicación para obtener modelos en 3D 92
A.21. Modelo 3D del puente rectificador 92
A.22. Agregar modelo 3D al footprint del componente 92
A.23. Propiedades del modelo 3D 93
A.24. Modelo 3D añadido al componente 93
A.25. Modelo 3D del componente 93
A.26. Alinear modelo 3D con la placa 94
A.27. Crear archivo de salida en el proyecto 95
A.28. Crear un archivo de salida PCB 95
A.29. Alinear modelo 3D con la placa 96
A.30. Configuración de propiedades del archivo de salida 96
A.31. Configuración de las capas a imprimir 97
A.32. Configuración para la generación del archivo PDF 97
A.33. Crear archivo para la salida .gerber 98
A.34. Configuración del archivo de salida .gerber 98
A.35. Creación del archivo para los agujeros 98
A.36. Configuración de las propiedades para los agujeros 99
A.37. Generar archivos en el contenedor 99
A.38. Archivos generados 99

A.39.Diseño del PCB de la fuente de alimentación 100
A.40.Modelo 3D de la fuente de alimentación 100
A.41.Archivo final para impresión de la fuente de alimentación 100
A.42.Fuente simétrica de 12V y regulable de 0-5VDC 101
A.43.Diseño de la PCB del circuito de acoplamiento 101
A.44.Modelado 3D del circuito de acoplamiento 102
A.45.Circuito de acoplamiento de la señal, para lectura de los puertos analógicos del DSP . 102
A.46.Circuito de acoplamiento de la señal, para lectura de los puertos analógicos del DSP . 103

B.1. Interfaz de usuario de CCS 105
B.2. Procedimiento para importar proyecto 105
B.3. Ventana que permite seleccionar el archivo del proyecto a importar 106
B.4. Ventana que permite seleccionar el compilador para el proyecto 107
B.5. Archivo importado correctamente 107
B.6. Construcción del proyecto 108
B.7. Depuración del proyecto para la grabación del código en la tarjeta 108
B.8. Programa en tiempo real 109

Índice de tablas

2.1. Producción total de energía por fuentes Bioenergéticas 26
2.2. Niveles de Voltajes 29
2.3. Limites Máximos de Armónicos de Voltaje 30
2.4. Tiempo de respuesta para frecuencias anormales 30
2.5. Correspondencias utilizadas para la transformación al dominio z 34

3.1. Materiales para el diseño e implementación de la fuente simétrica 45
3.2. Valores nominales del partidor de tensión 46
3.3. Materiales para el diseño del circuito de acoplamiento 47
3.4. Parámetros para el controlador PI utilizados en el sistema 57

4.1. Parámetros y resultados de los algoritmos analizados 79

Cláusula de Propiedad Intelectual

Yo, Andrés Antonio Segovia Vásquez, autor del trabajo de titulación "Implementación y análisis de algoritmos de sincronización de fase, para fuentes de energía renovables en sistemas trifásicos", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 26 de julio de 2021

Andrés Antonio Segovia Vásquez
110449514-6

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Andrés Antonio Segovia Vásquez en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Implementación y análisis de algoritmos de sincronización de fase, para fuentes de energía renovables en sistemas trifásicos", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 26 de julio de 2021

Andrés Antonio Segovia Vásquez
110449514-6

Certifico

Que el presente proyecto de tesis: Implementación y análisis de algoritmos de sincronización de fase, para fuentes de energía renovables en sistemas trifásicos, fue dirigido y revisado por mi persona.

Ing. Luis Gerardo González Morales, PhD
Director

Dedicatoria

A mi familia y amigos

Este trabajo de titulación, dedico en especial a mis padres que me han apoyado a lo largo de mi formación académica, por su incondicional amor, a mis hermanas y mi sobrina que han estado conmigo, a mis amigos que sin duda me han apoyado.

Andrés Segovia

Agradecimientos

En primer lugar extiendo mi agradecimiento al Ingeniero Luis González que me ha brindado sus conocimientos y su apoyo como guía a lo largo de este trabajo de titulación, así como tu total confianza para alcanzar los objetivos del presente trabajo.

También extiendo un agradecimiento a Pablo Delgado y al Ing. Edisson Villa encargados del Laboratorio del Micro Red, que me han dado un espacio de trabajo amigable y lleno de conocimientos, a pesar de las restricciones establecidas me han dado su confianza para poder realizar las pruebas en el Laboratorio, sin su ayuda no hubiera sido posible este trabajo.

Andrés Segovia

Abreviaciones y Acrónimos

ARCONEL Agencia de Regulación y Control de Electricidad. 29
AT alta Tensión. 26, 27

BT baja Tensión. 28

CCS Code Composer Studio. 1, 2, 4, 6, 7, 10, 20, 22, 41, 43, 49, 50, 80, 104, 105

DDSRF-PLL Decoupled Double Synchronous Reference Frame-Phase Locked Loop. 7, 19, 36, 38, 39, 78–80
DR recursos distribuidos. 30
DSOGI-PLL Dual Second-Order Generalized Integrator-Phase Locked Loop. 19, 78–80
DSP Digital Signal Processor. 1, 2, 8, 10, 20, 22, 26, 41–43, 48–50, 53, 68, 69, 75, 79, 80, 102, 103
DSRF-PLL Decoupled Stationary Reference Frame-Phase Locked Loop. 19

GNL Gas Natural Licuado. 18, 19
GSC Grid Size Converters. 26

HD Total Harmonic Distortion. 29

IDE Integrated Development Environment. 41
IRENA International Renewable Energy Agency. 24

MT media Tensión. 26, 27

PCB Printed Circuit Board. 44, 82–84, 87, 90, 91
PCC punto de acoplamiento en común. 19, 26, 28–31
PI Proporcional-Integral. 5, 32, 35, 36, 56
PLL Phase-Locked Loop. 19, 31–33
PSE Positive Sequence Extractor. 40

RES Renewable Energy Sources. 2, 18, 19, 22, 26, 29, 32, 71, 76, 78, 81

SCI Serial Communication Interface. 4, 7, 42, 47, 49, 50, 52, 53, 56, 69, 110
SISO Single Input Single Output. 33
SNT Sistema Nacional de Transmisión. 22
SRF-PLL Synchronous Reference Frame-Phase Locked Loop. 19, 39

TI Texas Instruments. 41

VCO Voltage Controlled Oscillator. 19, 31, 32, 34

Capítulo 1

Introducción

En este capítulo se contextualiza la problemática de la interconexión entre las Renewable Energy Sources (RES) y la red eléctrica pública, sus limitantes y ventajas. Es así, que en la sección 1.1 se presenta la introducción, en la que se habla de la problemática y se establecen pautas de la solución de este trabajo, en la sección 1.2 se presentan los antecedentes, a continuación, se presentan los alcances del proyecto en la sección 1.3, posteriormente se establecen los objetivos en la sección 1.4 y finalmente se presenta la estructura del documento en la sección 1.5.

1.1. Introducción

En la actualidad, el desarrollo tecnológico ha permitido que poco a poco el Gas Natural Licuado (GNL), utilizado como combustible para la generación de energía eléctrica, sea reemplazado por fuentes de energía renovables alternativas (RES), que sean confiables y seguras, estas fuentes de energía limpias, tales como, solar, eólica o hidráulica; tienen la posibilidad de inyectar su energía eléctrica generada a la red pública, en el nivel de transmisión o en el de distribución, esto depende de la capacidad de generación de cada fuente.

Para lograr inyectar la energía generada, es necesario que las redes cumplan ciertos parámetros técnicos que permiten que la red sea eficiente y no cause daños en los equipos terminales o en los inversores de las centrales de la red principal. Es por ello que en cada región, se establecen códigos que permiten evitar estos problemas, entre ellos se establece parámetros como redes sincronizadas en fase, frecuencia de la red, la regulación del voltaje, el número de armónicos, reconexión ante fallas, etc.

Uno de los parámetros más importantes a considerar es la sincronización entre generadores, para lograr este objetivo se han desarrollado diferentes metodologías, tales como, algoritmos matemáticos que van de la mano con sistemas de electrónica de potencia y sistemas de control, que permiten lograr este objetivo. Entre ellos se encuentran la detección de cruce por cero, este procedimiento se basa en detectar cuando la amplitud de la señal cambia de positivo a negativo o viceversa, es decir cuando la señal cruza por el eje del tiempo (eje x).

Este procedimiento resulta ser el más sencillo, pero a su vez ineficiente, esto debido a que la señal

no es completamente pura, es decir no es perfectamente senoidal, por lo que conlleva a que exista armónicos en la señal de entrada. Estas distorsiones en la señal se pueden interpretar como un cruce por cero de la señal original, impidiendo la correcta sincronización del sistema.

Por el contrario, existen otros métodos de control que son los **Lazo de bloqueo de fase** (Phase-Locked Loop (PLL)), su funcionamiento se centra en detectar la fase de la señal de entrada, luego ingresa al filtro de primer orden o al controlador PI, que permite obtener la fase, a partir de una frecuencia de referencia, finalmente esta fase, pasa al **Oscilador Controlado por Voltaje** (Voltage Controlled Oscillator (VCO)), para generar la señal de salida.

Con estos antecedentes, a lo largo del tiempo se han ido desarrollando algoritmos de acuerdo con la calidad de la red de cada región, esto ha permitido mejorar los PLLs, para lograr mayor eficiencia en la sincronización, permitiendo así una mejor distribución de las RES, logrando un amplio desarrollo energético a lo largo de la región.

1.2. Antecedentes

Debido a la fuerte demanda energética a nivel mundial y al proceso de innovación en la generación de energía limpia, con la finalidad de reemplazar los generadores de energía con GNL, en el Ecuador se han experimentado nuevas fuentes de energías renovables, como son la hidráulica (COCA CODO SINCLAIR) y eólica (Villonaco-Loja), estos proyectos permiten un gran avance en generación de energía limpia, produciendo grandes cantidades de energía al mes.

A su vez, nuevos proyectos se empiezan a desarrollar, como la construcción de parques solares, que producen energía mediante paneles solares, con ello a lo largo del país existen empresas que han optado por tener su propio parque solar instalado en sus edificios, esto a gran escala. Por otro lado, debido a que su precio ha disminuido en los últimos años, actualmente se pueden adquirir para su uso doméstico. Estas fuentes, se deben interconectar a la red pública, en un punto de acoplamiento en común (PCC).

Al tener dos fuentes de energía distinta, existe problemas de sincronización, causando daños en la línea principal, consecuentemente se dan penalizaciones para los usuarios que instalan estos generadores. Para evitar esto, se utilizan sistemas de control como los PLLs y sus variantes, dichos algoritmos han ido evolucionando dependiendo a la calidad de energía de la red principal, por lo que se han implementado variantes como **PLL con marco de referencia síncrono** (Synchronous Reference Frame-Phase Locked Loop (SRF-PLL)), **PLL con desacople en el marco de referencia estacionario-PLL** (Decoupled Stationary Reference Frame-Phase Locked Loop (DSRF-PLL)), **PLL con doble integrador generalizado de segundo orden** (Dual Second-Order Generalized Integrator-Phase Locked Loop (DSOGI-PLL)), **PLL con doble desacople en el marco de referencia síncrono-PLL** (DDSRF-PLL), etc.

Estos algoritmos de control, tienen como idea principal, entregar una buena calidad de energía a la red, además, poder contribuir con el medio ambiente ya que provienen de RES, con ello se puede asegurar que los equipos conectados a su red, puedan trabajar sin problemas.

1.3. Alcance

En el desarrollo de este trabajo, se implementan los algoritmos de sincronización como: SRF-PLL, DSRF-PLL, DSOGI-PLL, DDSRF-PLL; utilizando herramientas computacionales como el PSIM® y

CCS, además se utiliza un DSP TMDSCNCD28335 para poder almacenar dichos algoritmos. El DSP se encargará de realizar la adquisición de datos, es decir, este dispositivo por medio de las entradas analógicas, captura la señal de entrada, luego por medio del modulo A/D, permite digitalizar la señal, para finalmente realizar el procesamiento de la misma, con los algoritmos mencionados.

Por lo tanto, para la adquisición de datos, se necesita que la señal de entrada, esté acorde a las especificaciones técnicas de las entradas analógicas, por lo que se implementará un circuito de acondicionamiento de la señal que permite el correcto acoplamiento.

Para visualizar el comportamiento de cada algoritmo, se utiliza el osciloscopio del software PSIM, para ello se realiza la comunicación serial entre el DSP y la PC por medio de un convertidor USB-Serial.

1.4. Objetivos

1.4.1. Objetivo general

Implementación y estudio de algoritmos de sincronización para sistemas trifásicos, en la red del laboratorio de Micro Red de la Universidad de Cuenca.

1.4.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Estudio del estado del arte para comprender el funcionamiento de cada algoritmo.
- Implementación de los algoritmos estudiados en las plataformas de PSIM y CCS
- Diseño y construcción de los circuitos de acondicionamiento entre la red y el DSP
- Implementación del proceso de comunicación entre la PC y DSP
- Análisis del comportamiento obtenido con cada algoritmo.

1.5. Estructura del documento

En este capítulo se ha priorizado la importancia de los algoritmos de sincronización de fase para el uso de fuentes de energías limpias como reemplazo de los derivados de petróleo, además que permite que los ciudadanos puedan ser generadores de su propia energía y aportar a la red principal sin causar daños en la misma, posteriormente el documento se encuentra estructurado de la siguiente manera:

- **Capítulo 2:** presenta una revisión de los principales generadores eléctricos renovables y no renovables, luego se da a conocer la estructura y los aspectos regulatorios, de los sistemas de transmisión y distribución eléctricos. También se da a conocer los fundamentos teóricos de los elementos que integran el sistema de control de cada algoritmo de sincronización, en conjunto con su estructura. Finalmente se presenta la descripción técnica y teórica de los componentes de software y hardware que se utilizan en el desarrollo del trabajo de titulación.
- **Capítulo 3:** presenta de forma esquemática el diseño y la implementación de los algoritmos de sincronización utilizados para el estudio, además se detalla los sistemas de control y electrónica de potencia que se han utilizado para el funcionamiento del sistema.

- **Capítulo 4:** se presenta los datos técnicos de los escenarios en los que los distintos algoritmos fueron probados, se expone los resultados obtenido en la puesta en marcha del sistema, realizando un análisis comparativo del rendimiento de cada algoritmo. Finalmente se presenta las contribuciones y experiencias tratadas en el trabajo de titulación, introduciendo trabajos futuros en base a la temática tratada.

Capítulo 2

Marco Teórico

En este capítulo se presenta el sustento teórico en la que se fundamenta la propuesta de este trabajo de titulación. Este capítulo inicialmente presenta una reseña global de los distintos generadores de fuentes de energías en la sección 2.1, luego se realiza un enfoque de la estructura técnica de los sistemas de transmisión y distribución eléctricos en la sección 2.2. Además se introduce una explicación de los aspectos regulatorios del Sistema Nacional de Transmisión (SNT) en la sección 2.3. Posteriormente en la sección 2.4 se realiza una explicación de los algoritmos existentes en el campo de la sincronización. Asimismo en la sección 2.5 se detalla los aspectos del software PSIM, luego se explica sobre los aspectos técnicos del DSP en la sección 2.6 y finalmente en la sección 2.7 se explica sobre el software CCS.

2.1. Fuentes de Generación de Energía Eléctrica Renovables y No Renovables

Las fuentes de energía a nivel mundial, son recursos de primera necesidad, debido a su vital importancia, ya que permiten la generación de energía eléctrica, para el consumo humano.

Con esto se presentan ciertos problemas, ya que se requiere que su fuente de energía vaya acorde a las posibilidades de cada región, por ejemplo en Ecuador, el caudal de las fuentes hídricas son significativas, por lo que su principal fuente de energía es la hidráulica, permitiendo así, que este sea exportador de energía.

Sin embargo, en otras regiones su principal fuente de energía proviene del petróleo o sus derivados, lo que conlleva a buscar nuevas formas de energía alternativa, aun así teniendo en cuenta que cada año el desarrollo de las RES va incrementando, las fuentes no renovables tiene un porcentaje alto en generación de energía, llegando a tener un 73 % de participación en el mercado, frente al 27 % que proviene de fuentes renovables como hidráulica, solar y eólica. [1]

2.1.1. Fuentes de Generación de Energía no renovables

Los generadores de energía no renovables, son aquellos que han sido de uso común, entre ellas están aquellas que funcionan con derivados de petróleo, estas fuentes resultan ser más prácticas ya que se tiene la posibilidad de transportar la materia prima hasta las estaciones principales, para posteriormente someterlas a un proceso para generar energía. Al ser no renovables, estas fuentes causan un gran impacto al medio ambiente, por lo que se han desarrollado estudios en varias regiones para poder disminuir el uso de estas fuentes o a su vez desarrollar métodos para no causar graves impactos ambientales.[2]

2.1.1.1. Petróleo

El petróleo es la principal fuente de energía a nivel mundial, abarcando aproximadamente el 31 % de la producción total de energía [1]. En el año 2019 se generó aproximadamente **8000TWh**. Este producto se obtiene de restos fósiles, formado de varios gases, principalmente de azufre [3]. El crudo, como se lo conoce habitualmente, en los últimos años ha tenido una caída exponencial en el precio del barril, esto debido a la saturación del mercado y la falta de demanda [4].

Este problema ha llevado a que la economía mundial esté en riesgo ya que es uno de los principales productos de comercio a nivel mundial y su gran impacto en el tema energético resulta ser de gran relevancia, por ser su principal fuente de energía [5], además que sus derivados resultan ser esenciales para el consumo en la vida cotidiana, como la gasolina, el diésel, el gas, kerosene, etc.

2.1.1.2. Carbón

EL carbón es un mineral, considerado como la primera fuente de energía, este mineral abarca aproximadamente un 26 % de producción de energía a nivel mundial [1]. Al inicio de la revolución industrial, el carbón fue la principal fuente de generación de energía, [3][6], por consiguiente, era el principal elemento para poner en marcha a los motores de los principales medios de transporte e incluso motores de las industrias en donde se trabajaba con hierro y acero.
En el año 2019, a nivel mundial se generó aproximadamente **7000TWh**. Este mineral es un recurso abundante sobre la tierra por lo que es la principal fuente de energía a nivel industrial, países como Estados Unidos y China, son los principales consumidores de este mineral [3].

2.1.1.3. Gas Natural

El gas natural se encuentran en los yacimientos fósiles, estos se pueden encontrar con o sin fuentes de petróleo y son procesados para su uso comercial. En la industria, el gas natural permite generar energía térmica, es decir son más utilizados en la industria metalúrgica [7]. Este producto también es de uso doméstico, por lo que, conlleva grandes ventajas con respecto a otros derivados de petróleo que se trataron anteriormente.
Entre las ventajas más importantes, no emite grandes cantidades de C02, por lo que en ciertas regiones de Europa como en España, se ha impulsado el reemplazo del carbón por el gas natural como fuente de respaldo. [8] A nivel mundial, el gas natural abarca el 23 % de la generación de energía eléctrica, produciendo aproximadamente **6000TWh** en el año 2019.

2.1.1.4. Nuclear

La energía nuclear es aquella energía contenida en el núcleo de los átomos, que al ser estimulados, liberan su energía contenida, esto produce que se genere energía calorífica, lo que a su vez produce vapor, esto entra a las turbina que son las encargadas de producir energía eléctrica. En China [5], la energía nuclear es la principal fuente encargada de reemplazar las fuentes de carbón, ya que emite CO2 en pequeñas cantidades.

Este tipo de energía se puede obtener de dos procesos diferentes que son: la fisión, que consiste en partir el núcleo del átomo y la fusión que une el núcleo del átomo; estos procesos por lo general se logran usando Uranio o Plutonio [9], por su inestabilidad, no se encuentran fácilmente en la naturaleza, estos pueden generar un ambiente radiactivo permanente por años.

En el 2018, a nivel mundial, se tuvo 450 reactores operativos, por lo que la capacidad energética producida a nivel mundial fue de **2563TWh** [10] [11]. En el 2019 la capacidad obtenida fue de **2657TWh** aproximadamente.[12]

2.1.2. Fuentes de Generación de Energía Renovables

Las fuentes de energía renovables, son aquellas que se obtienen de fuentes limpias e inagotables, además no emiten CO2. Debido al impacto ambiental actual, los gobiernos se han visto obligados a desarrollar a gran rapidez estas fuentes de energía.
Existen muchos países que se encuentran migrando sus fuentes de energía contaminantes a energías limpias, por lo que en el 2009, se creó International Renewable Energy Agency (IRENA), una institución que permite la cooperación internacional de políticas, recursos, tecnología, etc, permitiendo así que más países se unan a estos proyectos. IRENA actualmente cuenta con 180 países adscritos a este proyecto de transformación.

2.1.2.1. Hidráulica

La fuente de energía hidráulica produce energía en base a la fuerza y el movimiento del agua, por lo general se utilizan ríos o vertientes de la montañas, que permiten mover las turbinas.
Una de las grandes ventajas de estas fuentes es que posee una gran nivel de respuesta inmediata ante fluctuaciones, lo que permite cubrir picos de demanda de manera confiable. En el 2019 se generó cerca del 16.2 % de energía eléctrica a nivel mundial [1]. Estas estadísticas apuntan a que el crecimiento de producción de energía en las hidroeléctricas vaya en aumento.

El crecimiento hasta el 2018 ha sido cerca de **4000TWh**, por lo que se pretende que cada año vaya incrementando. [13]. Entre los países que cuentan con las hidroeléctricas más grandes del mundo están: China, Brasil, Estados Unidos y Rusia. Estos países, en conjunto con más de 160 países [14], están llevando a las hidroeléctricas a ser las primeras fuentes de energía en el mundo, ya que se ha observado que cada año la capacidad de producción por país es mayor.

2.1.2.2. Eólica

Las fuentes de energía eólica son aquellas que utilizan la velocidad del viento para mover la turbina que permite generar energía, es decir, utiliza la energía cinética creada por el movimiento del aire. En los últimos años, la presencia de este tipo de fuente ha ido creciendo, teniendo un 6 % de energía generada en relación a las fuentes de energía renovables. [1]

En el año 2018 las fuentes eólicas generaron alrededor de **1200TWh** a nivel mundial, la capacidad de generación ha crecido en los últimos años. En el 2019 la energía hidráulica experimentó el crecimiento del 1.1 % [1], al punto que, la generación de energía con derivados de petróleo o carbón, ha caído, por lo que se pretende incrementar el número de turbinas instaladas en el mar, para llegar en los próximos años, reemplazar las fuentes de energía no renovables.
La energía eólica es la segunda fuente de energía renovable con más capacidad de generación en el mundo, después de la hidráulica. [3]

2.1.2.3. Solar

La energía solar es la energía más abundante en la tierra que se puede obtener de dos formas: fotovoltaica y térmica. Se estima que la energía que proviene del sol está entre los **1.5x10^{11} y 1.8x10^{11} MW**. [3] En el 2019 esta fuente generó aproximadamente el 3 % de lo generado por todas las energías renovables en el mundo. [1]

Sin embargo, la producción se divide en dos, siendo la fotovoltaica la principal fuente de generación eléctrica solar. Las fuentes fotovoltaica se basan en el efecto fotoeléctrico, estas fuentes utilizan un material semiconductor que puede ser de silicio mono-cristalino, poli-cristalino o amorfo, estos materiales absorben los fotones y liberan electrones, generando flujos de corriente en las celdas [13]. Este proceso generó aproximadamente **550TWh** alrededor del mundo en el año 2018.

Por otro lado las fuentes concentradas, absorbe los rayos de sol utilizando espejos, lo que permite calentar el fluido generando vapor y moviendo las turbinas; en relación con las fotovoltaicas se puede seguir generando energía aun sin la presencia del sol, pues almacena vapor que puede ser reutilizado [13], este proceso generó **12.200GWh** de energía en el año 2018 a nivel mundial.

2.1.2.4. Biomasa

La biomasa permite aprovechar los desechos orgánicos: sólidos (leña, astillas, carbón vegetal, etc.), líquidos (alcoholes, hidrocarburos, aceites vegetales, etc.) o gaseosos (gas de gasógeno, biogas, hidrógeno). Este tipo de generación energético se ha desarrollado con gran rapidez en los últimos años, permitiendo a países como España, Estados Unidos, China o Brasil, reemplazar los combustibles fósiles a escala industrial, debido a que en algunas fábricas las instalaciones de calderas gigantes es un hecho, esto les permite quemar materiales de sus propios desechos de materia prima o de material de fabricación. Entre los principales materiales que se utilizan para la generación de energía se tiene a la madera, que a lo largo de los tiempos el hombre ha utilizado para generar calor. [15]

En contexto energético, las tres cuartas partes del porcentaje total de la generación de energía con fuentes renovables, abarca la bioenergía [16], considerando que más de la mitad consiste en el uso de materiales tradicionales de biomasa. La bioenergía representó aproximadamente el 10 % del consumo

total de energía final y el 1,9 % de la generación de energía mundial en 2015. [13]

En el 2019 la producción de bioenergía a nivel mundial, abarcó el 9 % de a energía total [1], en la tabla 2.1 se presentan las principales fuentes de generación con su respectiva capacidad energética producida, con ello que se prevé que en los próximos años, estas fuentes de energía reemplacen a nivel industrial a los combustibles fósiles. Con estos resultados, se ha visto reflejado el compromiso de ciertos organismos estatales por el desarrollo energético y el medio ambiente.

Tabla 2.1: Producción total de energía por fuentes Bioenergéticas

Fuente Bioenergéticas	**Energía producida (GWh)**
Biocombustible íquido	7.344
Desechos municipales	62.148
Biogas	88.378
Biocombustible sólido	365.034

2.2. Estructura de los Sistemas de Transmisión y Distribución Eléctricos

En el apartado anterior se habló de las fuentes de energía renovables y no renovables, que resultan ser la parte principal de un sistema eléctrico. Estas fuentes se encuentran principalmente en las centrales eléctricas.

Considerando que, para la transmisión y distribución, las fuentes de energía externas se interconectan a la red principal, en un PCC, en este punto se realizan controles de acuerdo a los códigos de cada región, además que se realizan métodos de sincronización para poder realizar esta operación.

Es por ello que cada RES necesita su convertidor basado en electrónica de potencia (Grid Size Converters (GSC)), estos elementos deben funcionar en condiciones normales y anormales, es decir deben ser robustos, capaces de actuar ante fallas críticas de la red.[17][18][19]

Como se observa en la figura 2.1 se tiene un sistema de control de sincronización de fase, utilizando DSP que permite realizar el procesamiento de la señal, para conectar una red principal con una fuente de energía externa.

A más de las fuentes generadoras, como se puede ver en la figura 2.2, un sistema eléctrico consta también de elementos, materiales e instalaciones que permiten transmitir y distribuir energía eléctrica desde las plantas principales pasando por subestaciones y terminando en los usuarios finales ya sean domésticos o industriales.

La **red de transporte** es la encargada de conectar las estaciones elevadoras con las subestaciones reductoras, esta se puede dividir en dos partes que son: redes de transporte de alta Tensión (AT) y redes de transporte de media Tensión (MT). La función principal de las redes de AT es transportar energía a grandes distancias, por lo que es importante elevar la tensión en las centrales elevadoras para que exista la menor cantidad de pérdidas en la línea (por lo general se eleva la tensión a voltajes aproximados a 138kV, 230kV, 500kV o mayores [21]). Es importante saber que también pueden existir

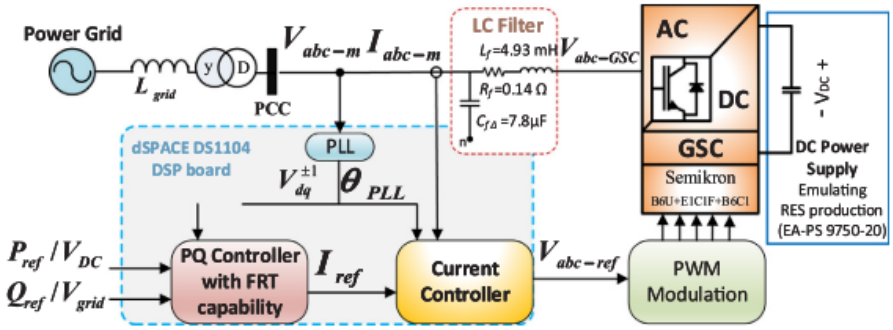


Figura 2.1: Topología básica de conexión de fuentes renovables con la red principal [17]

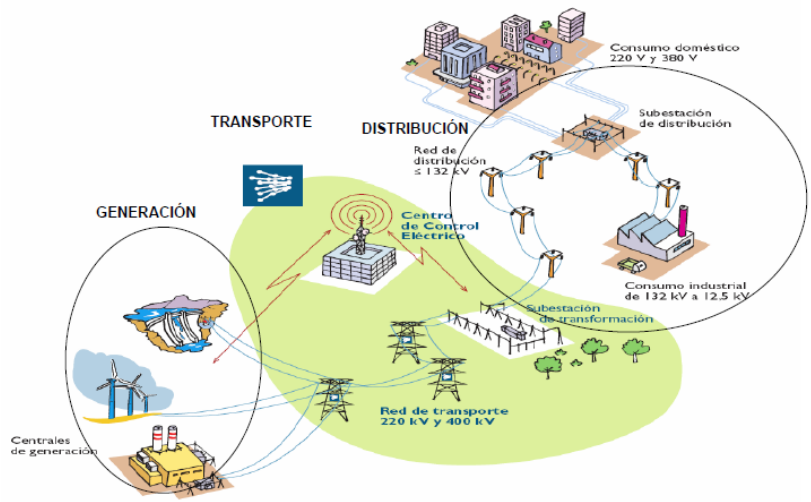


Figura 2.2: Esquema de una sistema de Generación, Transmisión y Distribución de la red Eléctrica. [20]

redes de distribución de AT y MT.[22]

Los principales elementos que conforman las redes de transmisión son:

- Torres de transmisión: es una estructura que permite colocar las lineas de transmisión. Comúnmente son torres de 25 o 45m de altura, dependiendo de la cantidad de circuitos que vayan sobre la torre, debiendo considerar la franja de servidumbre. [23]
- Líneas de transmisión: son los conductos por donde se va a transportar la energía, estos conductos se seleccionan debido a su capacidad a transportar, caída de tensión y pérdidas de potencia, etc.
- Aisladores de línea: Son los encargados de sujetar mecánicamente a los conductores, estos se encuentran aislados por lo que no permite conducción eléctrica en las sujeciones de la torre.
- Subestaciones: Las subestaciones, son pequeños sectores en donde se establece el voltaje necesario para su transmisión, estas son encargadas de elevar o disminuir la tensión según lo requiera la zona a transportar.
- Compensación de potencia reactiva: permite regular los voltajes de las barras de transmisión, estos se encuentran en las subestaciones, en donde hallamos bancos de inductores o capacitores. [24]
- Transformadores: Estos elementos son los encargados de elevar o disminuir la tensión en las

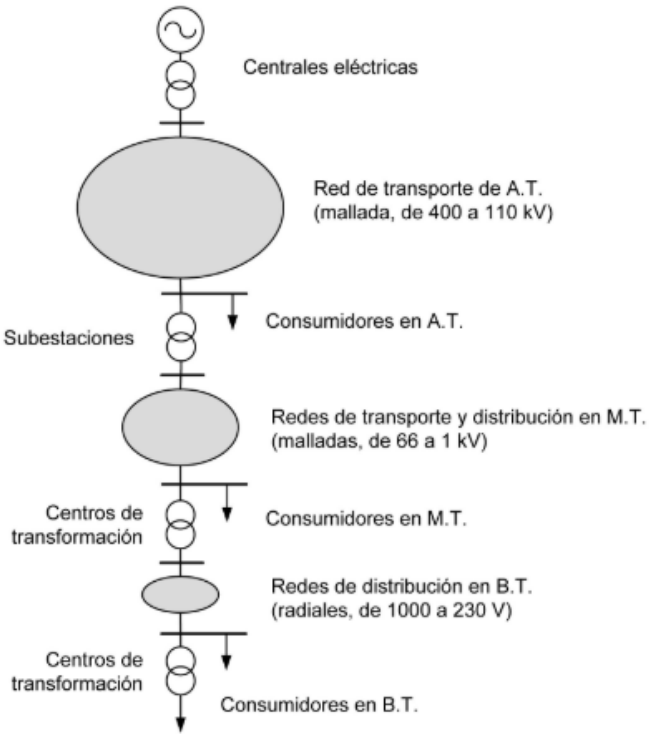


Figura 2.3: Estructura general de un Sistema Eléctrico [20]

subestaciones o en los centros de generación, estos elementos deben ser fijados de acuerdo a la cargabilidad que soportará cada uno según las normativas de cada región. [24]

En relación a la **red de distribución**, son redes de baja Tensión (BT) que son utilizadas para la comercialización, tanto doméstica como industrial. Estas redes pueden estar de forma aérea o subterránea. Uno de los componentes importantes son los centros de transformación, que son los encargados de regular el voltaje que llega de las subestaciones para el consumo doméstico o industrial. De forma general en la figura 2.3 se presenta un esquema general del sistema eléctrico (como se observa, tiene relación con lo mencionado anteriormente).

2.3. Aspectos Regulatorios para la Transmisión y Distribución de Energía Eléctrica

Uno de los aspectos más importantes para la conexión entre distintas fuentes de energía son las regulaciones técnicas que se establecen en cada región. Esto permite que una fuente externa no cause perturbaciones en la red eléctrica principal, con el fin de evitar problemas en los equipos de consumo o en las plantas generadoras principales.

Varias fuentes de energía renovables y no renovables, pueden estar conectadas en un PCC, esto permite que las fuentes, dependiendo de su capacidad generadora, pueden conectarse a una red de transmisión o distribución. Con ello, debido al desarrollo tecnológico, se ha logrado diseñar e implementar fuentes de energía de uso comercial como paneles fotovoltaicos, que pueden instalarse de forma

rápida y correcta en los domicilios.

Estas fuentes comerciales tienen la capacidad de disminuir el consumo eléctrico de la red principal o a su ves entregar a la red carga residual que no es aprovechada por el consumidor, esto permite al usuario vender parte de su energía generada al distribuidor de energía la región.

A continuación, se describe los aspectos técnicos más relevantes que se consideran para la interconexión de fuentes externas con la red principal, para esto se ha tomando en consideración el estándar IEEE Interconnecting Distributed Resources with Electric Power Systems [25], que permite conectar diferentes RES, en un PCC, también se han tomando en cuenta las recomendaciones de la serie 1547 Para la Interconexión Recursos energéticos distribuidos [26], [27], [28]. Además el estándar IEEE Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces [29], otorga ciertas normas que permite interconectar paneles fotovoltaicos a la red. Debido a que cada región tiene sus propios códigos se ha considerado las resoluciones de la Agencia de Regulación y Control de Electricidad (ARCONEL) REGULACIÓN No. 004/15 [30], 005/18 [31], relacionado con los aspectos técnicos y calidad de servicio de la distribución y comercialización de la red eléctrica y la resolución REGULACIÓN Nro. ARCONEL – 003/18 para la Generación fotovoltaica para autoabastecimiento [32].

2.3.1. Nivel de voltaje

Los niveles de voltajes son valores que clasifican las redes de transmisión o distribución, estos valores pueden tener ciertas variaciones que son admitidas por el ente regulador. En la tabla 2.2 se puede observar las características de los niveles de tensión regulados por la ARCONEL.

Tabla 2.2: Niveles de Voltajes

Clasificación	Nivel de voltaje	Variación permitido
Bajo Voltaje	menor igual a 0,6 kV	+- 8.0 %
Medio Voltaje	mayor a 0,6 y menor igual a 40 kV	+- 6.0 %
Alto Voltaje Grupo 1	mayor a 40 y menor igual a 138 kV	+- 5.0 %
Alto Voltaje Grupo 2	mayor a 138 kV	+- 5.0 %

2.3.2. Perturbaciones rápidas de voltaje (Flicker)

Estas perturbaciones moderadas de voltaje, son fluctuaciones en la luminosidad, que pueden ser percibidas por el ojo humano.

2.3.3. Distorsión armónica de voltaje

Son perturbaciones causadas por las cargas no lineales conectadas a la red, esto causa que la forma de onda de voltaje o corriente se altere, causando daños en los equipos finales o en las plantas distribuidoras. En la tabla 2.3 se presenta los límites permitidos de armónico individual y Total

Harmonic Distortion (HD), establecidos por el ente regulador, en la regulación citada en [31] se presenta las fórmulas para su cálculo.

Tabla 2.3: Limites Máximos de Armónicos de Voltaje

Nivel de voltaje	Armónico individual	THD
Bajo Voltaje	5 %	8.0 %
medio voltaje	3 %	5.0 %
alto voltaje Grupo 1	1.5 %	2.5 %
alto voltaje Grupo 2	1 %	1.5 %

2.3.4. Desequilibrio de voltaje

El desequilibrio de voltaje se produce cuando cargas monofásicas o bifásicas se conectan de manera desbalanceada a la red trifásica, lo que ocasiona que los equipos resulten afectados.

2.3.5. Frecuencia

Cuando por distorsiones de voltaje sobrepasa los niveles de la tabla 2.4, los recursos distribuidos (DR) se desconectan de la red principal por un instante de tiempo, llamado tiempo de limpieza, en este tiempo los DR ajustan sus parámetros a los establecidos por la red principal. Algunos valores pueden ser ajustados en campo.

Tabla 2.4: Tiempo de respuesta para frecuencias anormales

Rango DR	Rango de frecuencia (Hz)	Tiempo de limpieza (s)
$\leq 30kV$	>60.5	0.16
	<50.3	0.16
>30kV	>60.5	0.16
	<59.8-57.0(ajustable)	0.16-300 (ajustable)
	<57.0	0.16

2.3.6. Sincronización

Los DR deben estar en paralelo con los sistemas de potencia del sector, conectados a un PCC sin causar interferencias en la línea mayor a +-5 % del nivel de voltaje establecido en el sistema de potencia.

2.3.7. Reconexión al sistema de potencia eléctrico

Cuando existe una falla en el área de sistema de potencia, el DR es desconectado, hasta que el voltaje del sistema se encuentre en los rangos establecidos por cada región, además que cada DR tendrá que tener un retardo ajustable después de la interconexión del sistema de potencia, a la red.

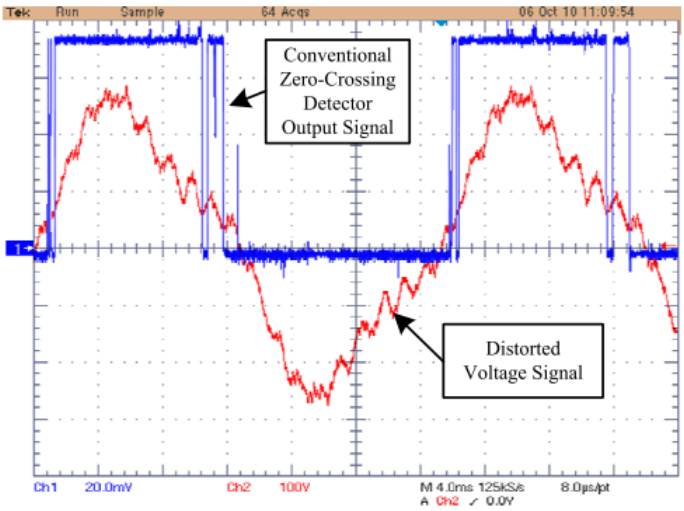


Figura 2.4: Detección de cruce por cero de una señal senoidal ruidosa [35]

2.4. Algoritmos de Sincronización de Red

Debido a las perturbaciones que se presentan en la interconexión de fuentes en un PCC, es necesario implementar métodos, capaces de permitir que los parámetros de las fuentes se asemejen entre sí para evitar daños, para ello los algoritmos de sincronización permiten sensar los parámetros de la red principal y por medio de procesamientos de señales adaptar las fuentes externas a los mismo.

2.4.1. Detección de cruce por cero

La detección de cruce por cero es uno de los métodos más sencillos, que permite capturar el estado transitorio de la onda senoidal cuando pasa de 0° a 180° en AC. Este tipo de circuito, resultan ser eficientes cuando la señal es más pura, es decir cuando no hay presencia de armónicos o perturbaciones en la señal. En circuitos de corriente alterna, la señal no es completamente pura, ya sea por elementos no lineales conectados a la misma o por el simple hecho de la distribución, por lo que se tiene fluctuaciones en los cambios de amplitud, permitiendo que el algoritmo, realice una detección falsa de cruce por el eje. [33] Debido a estos problemas, se han desarrollado algoritmos con filtros que permitan tener la señal más pura, para tener una mejor percepción de la señal original y hacer que el algoritmo sea más eficiente, como nos indica [34] en donde busca la optimización configurando sistemas analógicos que permitan un correcto sensado.

En la figura 2.4 se puede observar que la señal no es pura por lo que, incluso puede tener armónicos no deseados, provocando que la onda tanto en amplitud como en frecuencia se distorsione y que la detección de cruce no sea fiable.

2.4.2. Algoritmos de lazo de bloqueo de fase (PLL)

Otros métodos desarrollados para sincronización de red, son los PLL, estos algoritmos permiten sincronizar la señal, tomando en consideración una señal de entrada (frecuencia y fase). En la figura 2.5 se presenta el diagrama de bloque del funcionamiento general del sistema, se observa que básicamente un PLL esta conformado por un detector de fase, un filtro de lazo y un VCO. [36] [37] [38] [39]

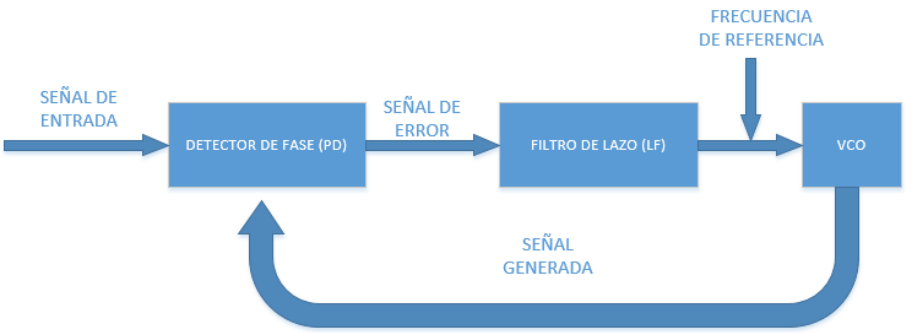


Figura 2.5: Diagrama de Bloques básico de un PLL

Debido a que en cada región la calidad de la red eléctrica es variable, se han desarrollado diferentes cambios en la estructura principal de los PLL con ello se ha logrado tener mejor rendimiento en la sincronización de fase.

De forma analítica los PLL se pueden conformar como se presenta en la figura 2.6, observando que es necesario trabajar con la fase de la RES, para poder sincronizar con la red principal. En este contexto inicialmente se realiza una detección de fase, esto se logra comparando la fase de la **señal de entrada** (θ_{grid}) con la **señal generada** (θ_{PLL}) por el VCO, obteniendo una **señal de error** (θ_{error}) con distorsiones de frecuencias altas, por lo que luego, se introduce un **filtro de lazo** para eliminarlas, a esta señal filtrada se introduce una **frecuencia de referencia** ($\mathbf{W}_{ref}$), obteniendo una señal $V1_f$, que finalmente ingresa al VCO para obtener la fase de la señal θ_{PLL}, para ello se tiene en cuenta la ecuación 2.1.

$$\theta = \int \omega(t)dt \tag{2.1}$$

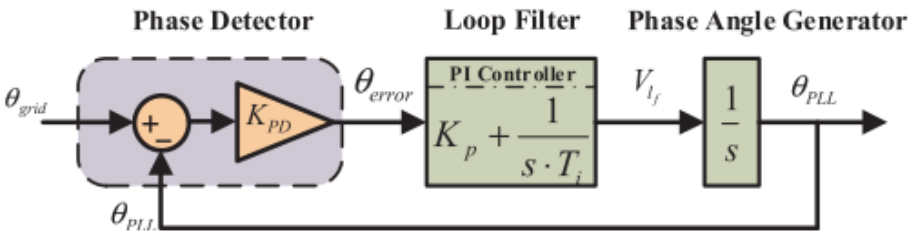


Figura 2.6: Topología básica de los PLL ([17])

El aspecto más importante en los esquemas **PLL** es el controlador PI, por lo que se debe calcular los parámetros para sintonizar el mismo. Sabiendo que las funciones de transferencia deben estar en el dominio discreto, se puede encontrar la función de transferencia en el dominio continuo y utilizando correspondencias entre la transformada de Laplace y transformada Z, se realiza la transformación para la modelización en el dominio discreto.

Utilizando el esquema mencionado en la figura 2.6 de la sección 2.4.2, se tiene la función de transferencia de **lazo abierto** ($F_{OL}(s)$) como se observa en la ecuación 2.2, para ello se considera una ganancia para el detector de fase $k_{pd} = 1$, ya que las distorsiones de frecuencia baja pueden aumentar

su amplitud y causar problemas en el bucle de control.

Siendo **PDs** la función de transferencia del **detector de fase**, **LFs** la función de transferencia del **Filtro de Lazo**, **VCOs** la función de transferencia del **VCO**. En donde K_p es la **constante proporcional** y T_i representa el **tiempo integral**.

$$\begin{aligned} F_{OL}(s) &= PD(s) * LF(s) * VCO(s) \\ &= k_{pd} * (k_p * (1 + \frac{1}{s * T_i})) * \frac{1}{s} \\ &= \frac{k_p s + \frac{k_p}{T_i}}{s^2} \end{aligned} \tag{2.2}$$

Para la función de transferencia de lazo cerrado, se tiene la ecuación 2.3. En donde θ_{PLL} corresponde al ángulo de fase resultante obtenido del PLL y θ_{grid} en ángulo de fase de la red inyectada al algoritmo. Esta ecuación se tiene, ya que el sistema es Single Input Single Output (SISO).

$$\begin{aligned} \frac{\theta_{PLL}}{\theta_{grid}} &= \frac{LF_(s)}{s + LF(s)} \\ &= \frac{k_p * (1 + \frac{1}{s*T_i})}{s + (k_p * (1 + \frac{1}{s*T_i}))} \\ &= \frac{k_p s + \frac{k_p}{T_i}}{s^2 + k_p s + \frac{k_p}{T_i}} \end{aligned} \tag{2.3}$$

Para normalizar las ecuaciones de segundo orden obtenidas de las funciones de transferencia, se considera las expresiones 2.4 y 2.5, considerando que ω_n es la frecuencia natural de oscilación, que es la frecuencia en donde se da el sobreimpulso y ξ es el factor de amortiguamiento del sistema.

$$\omega_n = \sqrt{\frac{k_p}{T_i}} \tag{2.4}$$

$$\xi = \frac{\sqrt{k_p * T_i}}{2} \tag{2.5}$$

Con ello tenemos la ecuación 2.6 ya normalizada.

$$\frac{\theta_{PLL}}{\theta_{grid}} = \frac{2 * \xi * \omega_n * s + \omega_n^2}{s^2 + 2 * \xi * \omega_n * s + \omega_n^2} \tag{2.6}$$

Hay que tener en cuenta que las variables proporcional k_p e integral T_i vienen representadas por las ecuaciones 2.7 y 2.8 en función del tiempo de establecimiento t_s definida por la ecuación 2.9, esto nos permite sintonizar el controlador.

Una notación importante es que la constante proporcional k_p mueve los polos de lazo cerrado para garantizar estabilidad, por lo que hay que tener cuidado al momento de la sintonización, de variar este

Tabla 2.5: Correspondencias utilizadas para la transformación al dominio z

Dominio de Laplace	Dominio Z
$\frac{1}{s}$	$\frac{z}{z-1}$
$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$

parámetros de forma brusca, y la variable integral T_i mueve los polos cerca del origen.

$$k_p = \frac{9{,}2}{t_s} \tag{2.7}$$

$$T_i = \frac{t_s * \xi^2}{2{,}3} \tag{2.8}$$

$$t_s = \frac{4{,}6}{\xi * \omega_n} \tag{2.9}$$

Para poder pasar de tiempo continuo a tiempo discreto, es necesario conocer las correspondencias entra la transformada de Laplace y la transformada Z, en la tabla 2.5 se presenta las correspondencias utilizadas para el diseño.

Con estas correspondencias y considerando las funciones de transferencia 2.2 y 2.3, se realiza la transformación en el dominio discreto. Al momento de realizar los cálculos hay que tener en cuenta el periodo de muestreo, ya que esto influye mucho en el diseño de los algoritmos. En la ecuación 2.12 se tiene la función de transferencia de lazo cerrado del controlador discreto.

$$\frac{\theta_{PLL}(z)}{\theta_{grid}(z)} = \frac{H_1(z)}{1 + H_1(z)} \tag{2.10}$$

Con ello se considera la ecuación 2.11, que representan el filtro de lazo y el VCO en el dominio discreto respectivamente, lo denotaremos como una función $H_1(Z)$. En donde **T** es el periodo de muestreo.

$$\begin{aligned} H_1(z) &= k_p * \frac{z}{z-1} + (\frac{kp}{T_i} * \frac{Tz}{(z-1)^2}) \\ &= k_p * \frac{z * (z - 1 + \frac{T}{T_i})}{(z-1)^2} \end{aligned} \tag{2.11}$$

La ecuación 2.11 podemos reemplazar en la ecuación 2.12, y obtenemos la función de transferencia en el dominio discreto. Para este caso se debe tener en cuenta que los polos de la función de transferencia se deben encontrar dentro del circulo unitario para garantizar estabilidad. Para simplificar el desarrollo vamos a definir la expresión $\alpha = 1 - \frac{T}{Ti}$.

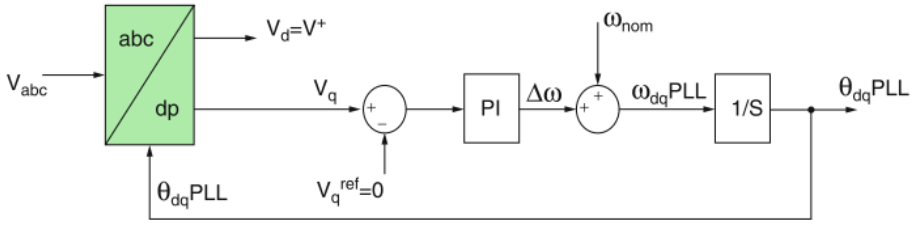


Figura 2.7: Diagrama de Bloques de un dqPLL [33]

$$
\begin{aligned}
\frac{\theta_{PLL}(z)}{\theta_{grid}(z)} &= \frac{H_1(z)}{1+H_1(z)} \\
&= \frac{k_p z(z-\alpha)}{(z-1)^2 + k_p z(z-\alpha)} \\
&= \frac{k_p z(z-\alpha)}{z^2 - 2z + 1 + k_p z^2 - k_p z\alpha} \\
&= \frac{k_p z(z-\alpha)}{z^2(1+k_p) - z(2+k_p\alpha) + 1} \\
&= \frac{k_p}{1+k_p} * \frac{z(z-\alpha)}{z^2 - z(\frac{2+k_p\alpha}{1+k_p}) + \frac{1}{1+k_p}}
\end{aligned}
\tag{2.12}
$$

2.4.2.1. Lazo de bloque de fase con marco de referencia síncrono (dqPLL)

Este algoritmo utiliza la transformada de Park debido a que transforma los valores de voltaje **abc** a un sistema de referencia de marco síncrono **dq**, es decir transforma valores senoidales en el tiempo a valores constantes en régimen permanente, permitiendo así el desacople entre secuencias con el fin de simplificar los cálculos, en [40] [41] se presenta el desarrollo matemático de las expresiones. La ecuación 2.13, indica el sistema de matrices para la transformada de Park, que se utiliza para el cambio de marco de referencia. [33] [42]. En [17] se presenta el estudio del algoritmo sometido a fallos brusco utilizando una fuente programable trifásica (California Instrument 2253iX), este algoritmo presenta una respuesta dinámica rápida ya que no requiere una amplia carga computacional, además, cuando en la red se tiene presencia de ruido y armónicos, este algoritmo permite ajustar el ancho de banda para lograr mejor rendimiento [42], sin embargo, cuando hay un desbalanceo de cargas este algoritmo resulta ser ineficiente ya que existe un efecto oscilatorio entre secuencias.

$$
\begin{bmatrix} V_d \\ V_q \end{bmatrix} = \frac{2}{3} \begin{bmatrix} cos(\theta) & cos(\theta-120) & cos(\theta+120) \\ -sen(\theta) & -sen(\theta-120) & -sen(\theta+120) \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}
\tag{2.13}
$$

Como se mencionó inicialmente, en la figura 2.7 se tiene al ingreso un sistema de coordenadas de 3 fases por lo que usando las ecuaciones 2.13 se obtienen dos componentes de ejes **directo ($\mathbf{V}_d$)** y **cuadratura ($\mathbf{V}_q$)**, teniendo en cuenta que la componente de eje en cuadratura debe mantenerse en cero utilizando un controlador **PI**, para que la componente de eje directo mantenga el voltaje de la fase A.[37] El lazo de control realiza la aproximación de la fase de la señal en cuadratura para realizar la retroalimentación ante cualquier fluctuación.

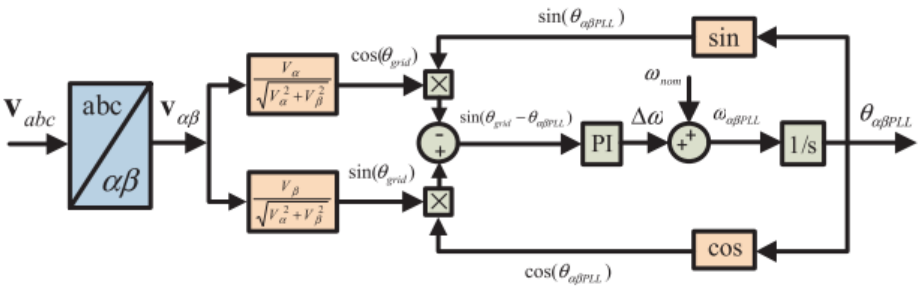


Figura 2.8: Diagrama de Bloques de un $\alpha\beta$PLL [17]

2.4.2.2. Lazo de bloque de fase con marco de referencia estacionario ($\alpha\beta$PLL)

Para este algoritmo se utiliza el diagrama de bloques presentado en la figura 2.8. En este caso, se tiene que transformar a un sistema de referencia $\alpha\beta$, utilizando la transformada de Clarke, transformando de un marco de referencia rotatorio a un marco de referencia estacionario ortogonal $\alpha\beta$. Considerando que al igual que en **dqPLL** se tiene el mismo problema en condiciones desbalanceadas, ya que tiende a errar cuando hay fallas de voltaje como presenta en [43].

Este algoritmo permite eliminar el sobre impulso que se genera al estimar el ángulo de fase en dq-PLL, se puede decir que este algoritmo es una mejor del anterior mencionado.

Para lograr esto, el algoritmo en cuestión, considera el mismo sistema de ecuaciones de 2.13, como se indica en [42], haciendo que el ángulo θ sea igual a cero, en adición esto permite estimar el ángulo de fase de $\alpha\beta$PLL en condiciones desbalanceadas. En consecuencia se tiene la ecuación 2.14 y utilizando identidades trigonométricas se tiene la ecuación 2.15.

El control de lazo (controlador PI), lo que hace en este algoritmo es tratar que por medio del bucle de control la diferencia entre el ángulo de fase del voltaje de la red (θgr)) y el ángulo de fase resultante ($\theta_(\alpha\beta)$) sea igual a cero. [17]

Los resultados mostrados en [44] y [42], presentan el ajuste de parámetros para los algoritmos **dq-PLL** y **$\alpha\beta$-PLL**, los resultados de los mismo concluyen que estos algoritmos no trabajan adecuadamente en condiciones desbalanceadas, además que su respuesta dinámica no requiere mucha carga computacional.

$$\Delta\theta \approx \theta_{gr} - \theta_{\alpha\beta PLL} \approx sin(\theta_{gr} - \theta_{\alpha\beta PLL}) \tag{2.14}$$

$$\Delta\theta \approx sin(\theta_{gr}) * cos(\theta_{\alpha\beta PLL}) - cos(\theta_{gr}) * sin(\theta_{\alpha\beta PLL}) \tag{2.15}$$

2.4.2.3. Lazo de bloqueo de fase con doble desacople en el marco de referencia síncrono (DDSRF-PLL)

El algoritmo DDSRF-PLL, aparece como una mejora de los algoritmos **dq-PLL** y **$\alpha\beta$-PLL**, debido a que estos algoritmos trabajan de forma incorrecta en condiciones de red desbalanceada. Esto se debe a que integra una red de desacoplamiento, que trabaja sobre las secuencias positivas y negativas de voltaje. Esta red, convierte ambas secuencias de voltaje, en marcos de referencia síncronos. [45] [46] [17]

La secuencia positiva dq^{+1} y la secuencia negativa dq^{-1}, rotan con velocidad angular positiva $+\omega$ y

negativa $-\omega$ y ángulo de fase θ^+ y θ^-, respectivamente. Con estos antecedentes, red de desacoplamiento cancela el efecto oscilatorio que existe entre ambas secuencias. Además se utiliza el algoritmo **dq-PLL**, para estimar el ángulo de fase de la red de voltaje.

$$v_{dq}^{+1} = \begin{bmatrix} V_d^{+1} \\ V_q^{+1} \end{bmatrix} = \begin{bmatrix} T_{dq}^{+1} \end{bmatrix} * v_{\alpha\beta} = V^{+1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + V^{-1} \begin{bmatrix} cos(-2 * t\omega) \\ sin(-2 * t\omega) \end{bmatrix} \tag{2.16}$$

$$v_{dq}^{-1} = \begin{bmatrix} V_d^{-1} \\ V_q^{-1} \end{bmatrix} = \begin{bmatrix} T_{dq}^{-1} \end{bmatrix} * v_{\alpha\beta} = V^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + V^{+1} \begin{bmatrix} cos(2 * t\omega) \\ sin(2 * t\omega) \end{bmatrix} \tag{2.17}$$

En donde:

$$\begin{bmatrix} T_{dq}^{+1} \end{bmatrix} = \begin{bmatrix} T_{dq}^{-1} \end{bmatrix} = \begin{bmatrix} cos(\omega) & sin(\omega) \\ -sin(\omega) & cos(\omega) \end{bmatrix} \tag{2.18}$$

En las ecuaciones 2.16 y 2.17 se puede observar que se tiene dos componentes, una **DC** y otra componente oscilatorio.[33] La componente oscilatorio es el efecto que existe entre las secuencias y la componente **DC**, permite estimar la magnitud del voltaje de la red. Para obtener la componente **DC**, se utilizan filtros paso bajo con frecuencia de corte $2 * \pi * \frac{f}{0{,}707}$ **rad/s**, con el fin de eliminar el efecto oscilatorio que existe entre las dos secuencias dq^{+1} y dq^{-1}.

En la figura 2.9 se presenta la estructura de bloques del algoritmo, como se puede observar las componentes **abc**, pueden ser representadas como una red desbalanceada utilizando el marco de referencia $\alpha\beta$, con esto, inicialmente se obtiene las componentes de voltaje de referencia **dq** positivas y negativas V_d^{+1}, V_q^{+1}, V_d^{-1}, V_d^{-1}, utilizando las expresiones de la ecuación 2.18.

Estas componentes, ingresan a la red de desacoplamiento presentada en la figura 2.10, para luego ingresar a un filtro paso bajo que elimina las componentes oscilatorias, que anulan los efectos entre componentes. Con ello entra al detector de fase y se realiza el proceso de generación del ángulo de fase. Este algoritmo no permite reducir el ancho de banda del bucle de control, logrando así, detectar con exactitud la amplitud de voltaje de la secuencia positiva,

En [47], [48] y [33], los autores presenta la implementación del algoritmo, además los resultados que presentan, muestran que es un algoritmo robusto cuando existen fallas en condiciones desbalanceadas y que su respuesta dinámica es rápida ante fallos.

La red de desacoplamiento se presenta en la figura 2.10, como se puede observar, se considera θ=ω**t**, con ello se logra disminuir el ancho de banda para el filtro de lazo y se logra una mejor sintonización del control.

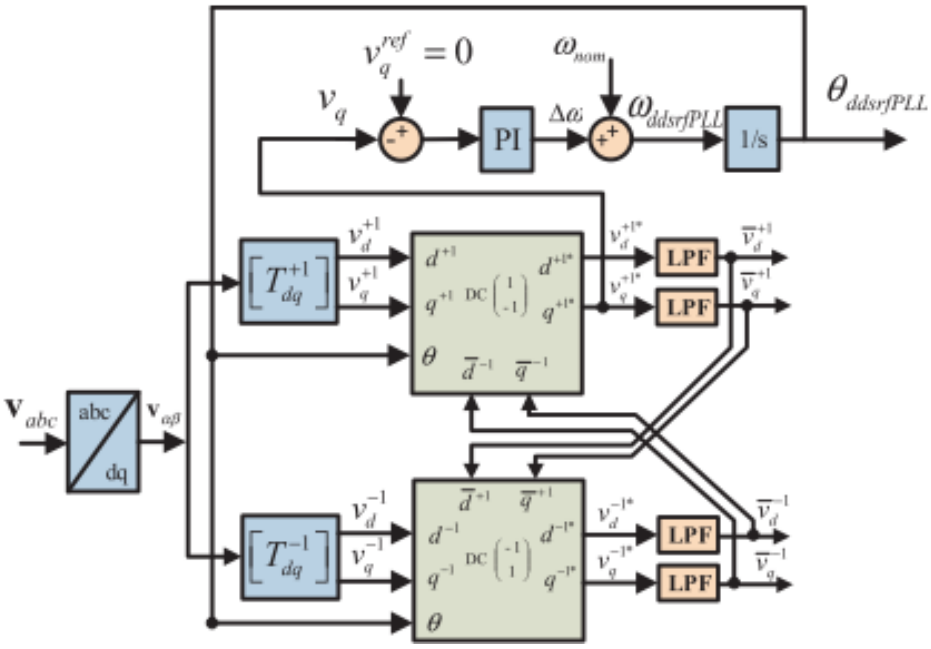


Figura 2.9: Diagrama de Bloques de un DDSRF-PLL [17]

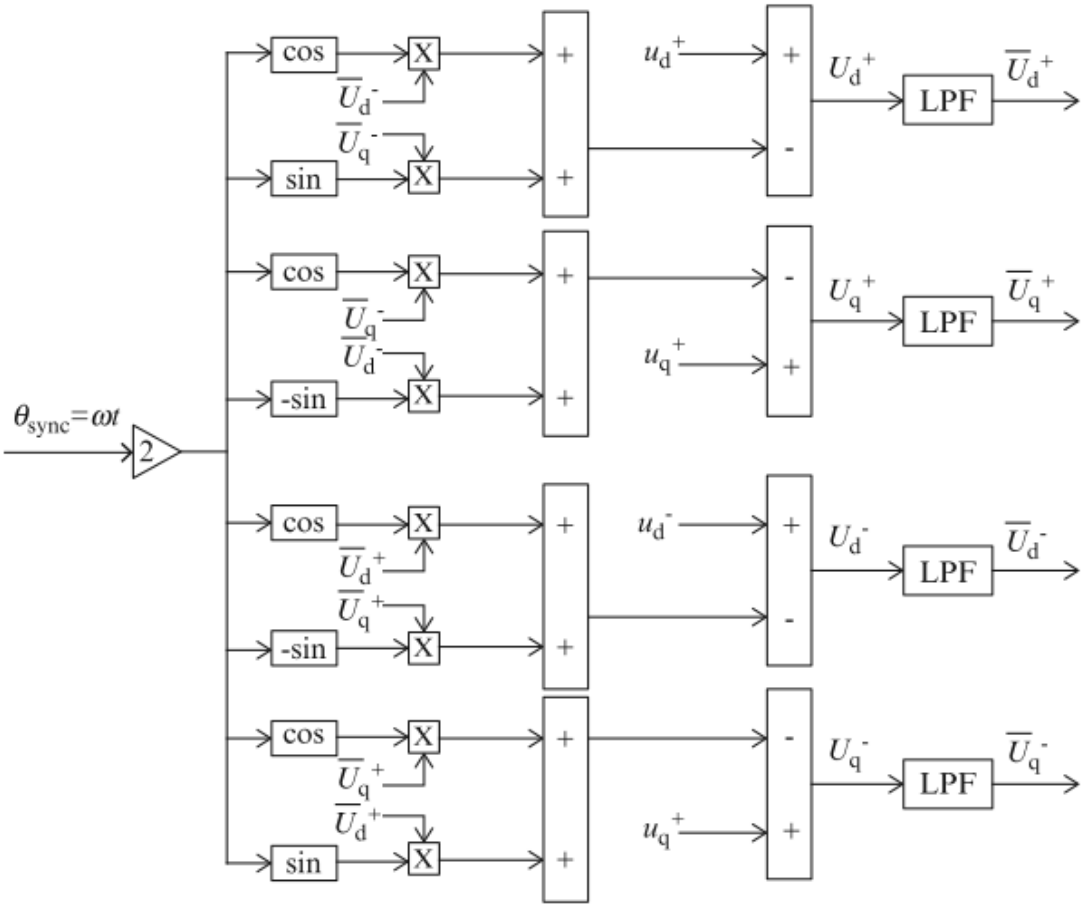


Figura 2.10: Red de desacoplamiento para secuencias positivas y negativas, utilizada en DSRF-PLL [33]

2.4.2.4. Lazo de bloqueo de fase con desacople en el marco de referencia estacionario (DSRF-PLL)

Este algoritmo es una mejora de los algoritmos **dqPLL**, DDSRF-PLL, ya que utiliza la red de desacoplamiento del DDSRF-PLL y la estimación de ángulo de fase del $\alpha\beta$**-PLL**, con ello se logra disminuir el ancho de banda del bucle control y reduce el sobre impulso que se genera en **dqPLL**. En este algoritmo, se utiliza las componentes tanto positivas como negativas de los vectores de voltaje de la red, con el fin de eliminar los errores de detección de los algoritmos SRF-PLL, en [49], [44] y [33] nos presentan la respuesta del algoritmo ante fallas desbalanceadas de la red trifásica, lo que conlleva a que este algoritmo responde de forma rápida y que requiere una amplia carga computacional.

En la figura 2.11 se puede observar el diagrama de bloques del algoritmo en mención.

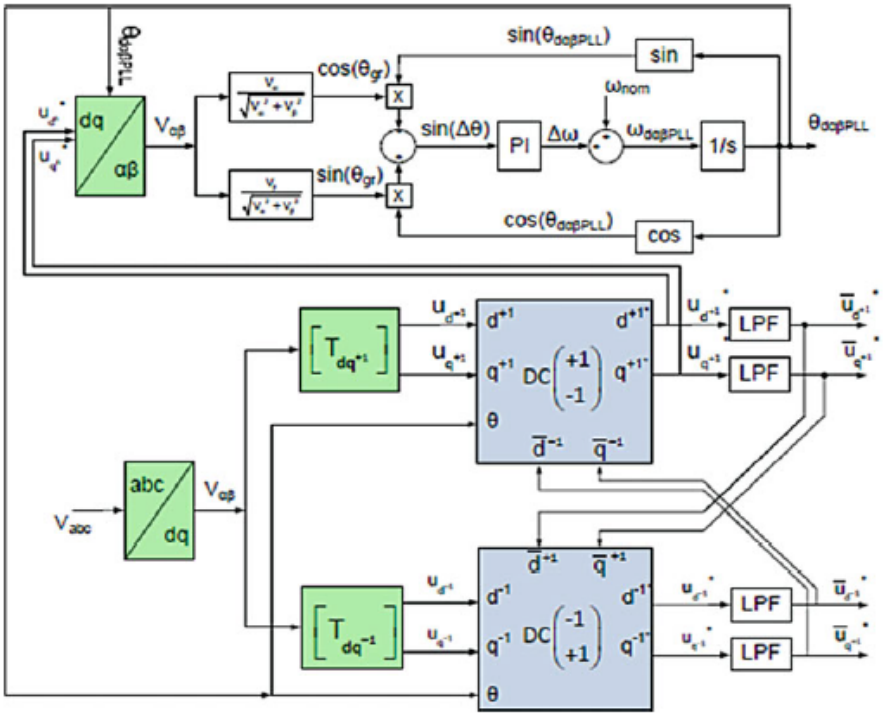


Figura 2.11: Diagrama de Bloques de un DSRF-PLL [44]

2.4.2.5. Lazo de bloqueo de fase con doble integrador generalizado de segundo orden (DSOGI-PLL)

Este algoritmo utiliza la transforma de Clarke para trabajar con el marco de referencia $\alpha\beta$ como se presenta en la estructura de la figura 2.13. Además trabaja con dos algoritmos integrados generalizados de segundo orden (**SOGI**), generando componentes en cuadratura que actúa como un filtro adaptativo paso banda, en [42] [50] los autores presentan la implementación del algoritmo, dando como resultado una respuesta robusta ante fallos desbalanceados y presencia de armónicos. En la figura 2.12 se presenta el esquema del algoritmo **SOGI**.

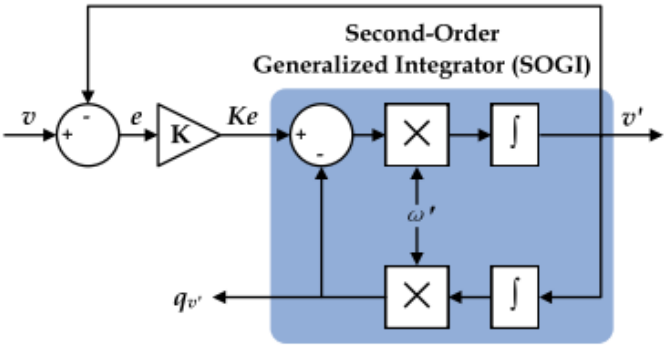


Figura 2.12: Estructura de bloques del algoritmo SOGI [42]

Una vez que las componentes pasan por un Positive Sequence Extractor (PSE), ya permite obtener la secuencia positiva, para la estimación de la frecuencia y la fase de la señal. Se utiliza el esquema del dq-PLL para la estimación del ángulo de fase. Por lo tanto, este algoritmo resulta ser más eficiente en casos de redes no balanceadas, pero resulta tener una respuesta dinámica más lenta y grande sobre impulsos de frecuencia, como nos presentan los resultados de [51], [52] y [46].

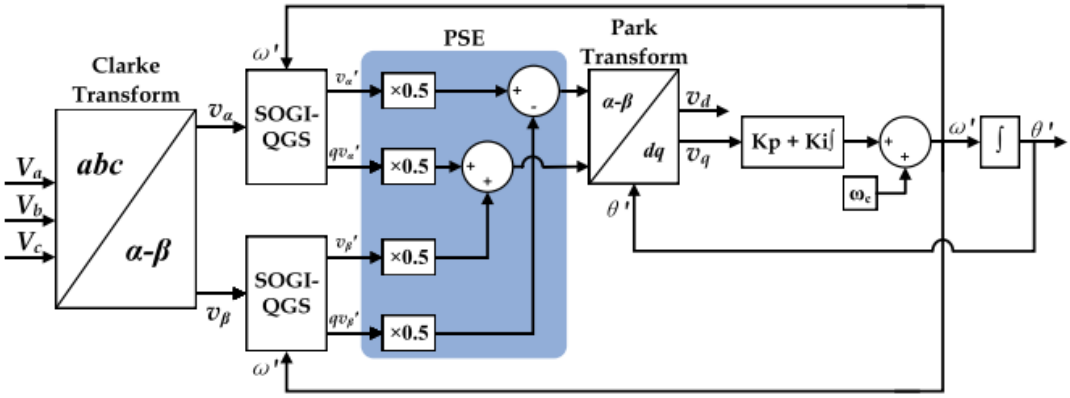


Figura 2.13: Estructura de bloques del algoritmo DSOGI-PLL [42]

2.5. Comparación de desempeño mediante herramienta computacional PSIM

A partir de los algoritmos presentados en la sección 2.4, existen varias herramientas computacionales que permiten desarrollar los mismos, uno de ellos es el software **PSIM**. Este software permite el diseño de circuitos electrónicos de potencia.

Este software permite realizar simulaciones a alta velocidad, además que arroja resultados de gran confiabilidad debido a sus propiedades de configuración. De las funciones integradas en el software, se tienen las transformadas de **Park** y de **Clarke**, estas funciones son muy relevantes en la implementación de los algoritmos ya mencionados, también posee filtros que son variables en sus parámetros y como eje principal permite crear funciones que no se encuentran en la lista de funciones propias de **PSIM**, considerando que cada función trabaja en diferente marco de referencia que puede ser continuo o discreto. [53] [54] [55]

Otra de las ventajas importantes, debido a sus características, es que nos permite capturar datos en tiempo real, para el presente trabajo de titulación, se utilizará el osciloscopio, utilizando una comunicación serial con el hardware utilizado que se describe en la sección 2.6. [56]

Además, **PSIM** permite generar automáticamente el código, en lenguaje **C/C++**, considerando que, para cargar los códigos se debe utilizar librerías que estén diseñadas para el hardware a utilizar. [57]

2.6. Procesador digital de señales (DSP) TMDSCNCD28335

Para la adquisición de datos se utilizará el DSP TMDSCNCD28335 (2.14), esta placa tiene adjunta una librería en **PSIM** que permite configurar parámetros de entrada, salida y simulación. El DSP contiene entradas analógicas y digitales, en [58] se puede observar las características técnicas de dichas entradas, debido a que en el presente trabajo de titulación se ocupará redes trifásicas, se debe realizar circuitos de acoplamiento para insertar señales por las entradas analógicas.

Figura 2.14: Placa (DSP) TMDSCNCD28335

Este dispositivo se comunica con una PC de forma serial, lo que permite la captura y visualización en tiempo real, de los puertos de entrada del hardware. [59] [38] [60]

Entre las características más importantes del hardware se puede destacar:

- Tensión de alimentación 5VDC
- Tensión de entrada y salida lógicas 0-3,3V
- 12 Salidas PWM
- 6 entradas digitales de 32 bits o PWM auxiliares
- Convertidor Analógico-Digital de 12 bits, de 2 bloques de entradas con 16 canales de conversión
- Interfaz de comunicación serial SCI
- 88 entradas y salidas digitales
- Frecuencia de reloj hasta 150MHz

2.7. Estudio creador de código (CCS)

CCS es un Integrated Development Environment (IDE) desarrollado por Texas Instruments (TI), que permite la programación de procesos integrados y microcontroladores. Integra, entre muchas características, un compilador C/C++ optimizador, editor de código fuente, entorno de construcción de proyectos, depurador, generador de perfiles.

Capítulo 3

Diseño y Desarrollo Experimental

En este capítulo se explica el desarrollo experimental del presente trabajo de titulación, se detalla el procedimiento que se utilizó para la obtención de los resultados. Inicialmente en la sección 3.1 se presenta con detalle el proceso para la adquisición de datos, posteriormente se presenta la configuración para la visualización de datos por medio del puerto serial SCI de PSIM en la sección 3.2, finalmente en la sección 3.4 se presenta la implementación de los algoritmos utilizando PSIM, se detalla las características de configuración de cada uno.

3.1. Adquisición de datos por medio del DSP

Entre las características del DSP se tiene los 16 canales del módulo **ADC**, por lo que, los principales objetivos del hardware son:

- Lectura de la señal analógica por medio de los canales del convertidor A/D
- Procesamiento de la señal (cambio del sistema de referencias de 3 fases a uno de dos fases ($\alpha\beta$))
- Normalización de las señales del sistema de referencia $\alpha\beta$.
- Aplicación del algoritmo para sincronización de fase
- Visualización por medio del puerto serial SCI
- Envío de señal de salida por medio del puerto digital PWM

En la figura 3.1 se presenta la estructura básica del funcionamiento del sistema de sincronización de fase. Como se menciona al inicio de esta sección, el proceso de acondicionamiento, permite que los parámetros de la señal de entrada a los puertos del DSP, estén acorde a las características técnicas del mismo.

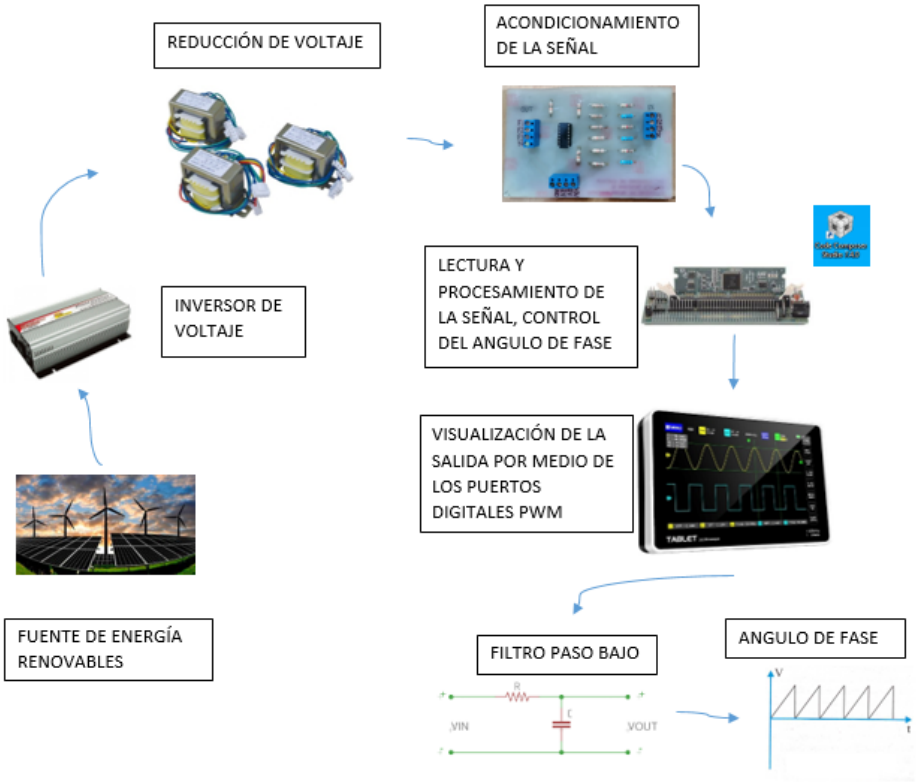


Figura 3.1: Estructura básica del funcionamiento del sistema de sincronización de fase

Para la lectura de datos, es importante seguir los siguientes pasos, ya que con ello aseguramos protección para los canales del convertidor A/D, a continuación se detalla el proceso de lectura.

- Acondicionamiento de la señal de entrada
- Configuración del convertidor A/D, declaración de puertos utilizados
- Generación del código para lectura de datos en lenguaje C/C++
- Importación de código a CCS y configuración del mismo para cargar el código en DSP

3.1.1. Acondicionamiento de la señal para lectura del DSP

El acondicionamiento de la señal resulta ser el punto de partida del trabajo de titulación, para ello inicialmente se realiza la reducción de tensión en cada fase, en este punto se realiza la medición de la señal antes y después del transformador, con la finalidad de asegurar un desfase de la señal de **120 grados**.

El circuito de acondicionamiento, consta de un amplificador operacional en configuración sumador (figura 3.3), con ello se obtiene un offset de la señal de entrada, es decir un desplazamiento de voltaje DC (eje y), esto permite acoplar el voltaje de entrada a los parámetros técnicos de los puertos de entrada del DSP. En nuestro caso el voltaje permitido por los puertos es **de 0 a 3VDC**, por lo que inicialmente se coloca un voltaje de referencia de 1.8VDC, para realizar el offset de la señal.

Para lograr el acoplamiento, se ha dividido en 2 etapas:

1. Diseño e implementación de la fuente asimétrica: Esta fuente permite alimentar el amplificador operacional y entregar el voltaje **DC** que ingresa el circuito sumador, para el desplazamiento de

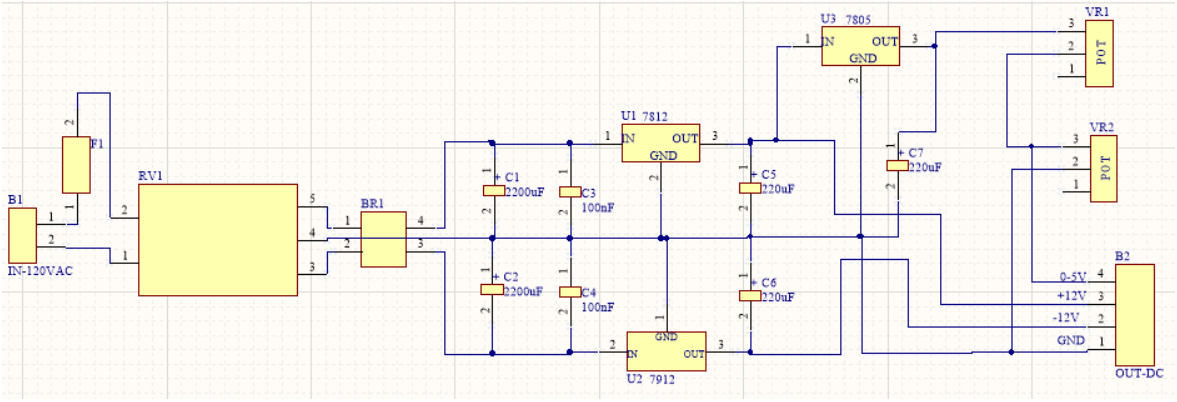


Figura 3.2: Circuito esquemático de la fuente de alimentación

la señal.

2. Diseño e implementación del circuito sumador: Este circuito permite desplazar la señal dependiendo del voltaje **DC** ingresado.

3.1.1.1. Diseño e implementación de la fuente simétrica

Para el presente trabajo de titulación se ha diseñado una fuente simétrica de +/-12VDC y una fuente variable de 0-5VDC, para ello se utiliza el software **Altium** para el diseño de la Printed Circuit Board (PCB).

Esta herramienta computacional, permite diseñar de forma precisa nuestra PCB configurando parámetros, como ancho de pistas, número de caras impresas, etc. A su vez, al no poseer ciertos componentes, nos da la posibilidad de diseñarlos teniendo en cuenta los parámetros técnicos del mismo. En el Anexo A se presenta el desarrollo paso a paso de la creación de proyectos y configuración del ambiente, además se indica el proceso para crear archivos (esquemático y el footprint) de los componentes y se detalla el proceso para generar los archivos **.gerber**, que son los necesarios para implementar la PCB.

Con este preámbulo, el diagrama de la fuente se presenta en la figura 3.2. A su vez, en la tabla 3.1 se presenta los materiales utilizados en el esquema y su respectiva nominación.

Tabla 3.1: Materiales para el diseño e implementación de la fuente simétrica

Elemento	Descripción	Cantidad
Bornera (B1)	Para PCB 2 vias	1
Bornera (B2)	Para PCB 4 vias	1
Porta fusible (F1)	Para PCB 5mmx20mm	1
Transformador (RV1)	VAC 12V-0V-12V	1
Puente rectificador (BR1)	DB107	1
Capacitor Electrolítico (C1,C2)	2200μF/25V	2
Capacitor Electrolítico (C3,C4,C7)	220μF/25V	2
Capacitor Cerámico (C5,C6)	100ηF	2
Regulador de Voltaje (U1)	7812	1
Regulador de Voltaje (U2)	7912	1
Regulador de Voltaje (U3)	7805	1
Potenciómetro (VR1)	10k	1
Potenciómetro (VR2)	100k	1

3.1.1.2. Diseño e implementación del circuito de acoplamiento

Una vez ya diseñada la fuente, se diseña el circuito de acoplamiento, el esquema presentado en la figura 3.3, es un circuito sumador, usando un amplificador operacional y una señal en DC que permite desplazar la señal con respecto al eje horizontal. Para ello se ha utilizado PSIM para realizar la simulación y poder validar el funcionamiento del circuito.

Cabe señalar que este circuito sirve para el acoplamiento de una fase, en nuestro caso se trabajó en un sistema trifásico, por lo que es necesario replicar este esquema para el acoplamiento.

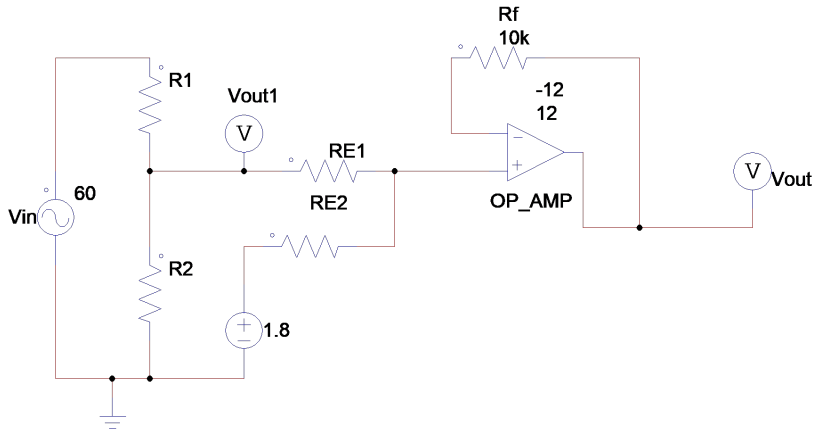


Figura 3.3: Circuito de acondicionamiento de señal

El circuito de acoplamiento se desarrolla en 2 fases. La primera fase obtener un voltaje de salida $\mathbf{V}_{out1}$, teniendo en cuenta que es la señal que vamos a desplazar, por lo que es necesario definir el

voltaje pico de la señal. Con ello podemos realizar un partidor de tensión por lo que se necesita obtener los valores de **R1** y **R2**.
Con estos datos se utiliza la ecuación 3.1 para obtener los datos mencionados.

$$V_{OUT} = V_{IN} * \frac{R_2}{R_1 + R_2} \tag{3.1}$$

Como se tiene el voltaje de salida deseado y el voltaje de entrada, se plantea un valor de resistencia y se calcula el valor de la variable restante, los datos considerados se presenta en la tabla 3.2.

Tabla 3.2: Valores nominales del partidor de tensión

Variable	**Valores nominales**
V_{IN}	12VAC %
V_{OUT1}	1.1VAC
R1	1kOhm

Resolviendo la ecuación se obtiene el valor de **R2**.

$$R_2 = \frac{1{,}1 * 1000}{12 - 1{,}1} = 100{,}91$$

La segunda fase es el diseño del circuito sumador no inversor con ganancia **G=1**, para lograr el desplazamiento. Con esto la señal de voltaje $\mathbf{V}_{out1}$ se desplaza, considerando el voltaje **DC**. De forma general para calcular la ganancia del amplificador en cualquier entrada se considera la ecuación 3.2, en donde R_f es la resistencia de feedback y RE_N la resistencia de la entrada siendo **N** en número de la entrada.

$$G_{inN} = \frac{R_f}{RE_N} \tag{3.2}$$

Con estos datos se realiza el diseño en **Altium** para su impresión, en la figura 3.4 se presenta el diagrama esquemático del circuito, como se mencionó, en este esquema esta replicado tres veces el circuito de la figura 3.3. De la misma forma los esquemas constructivos de la placa se presentan en el Anexo A.

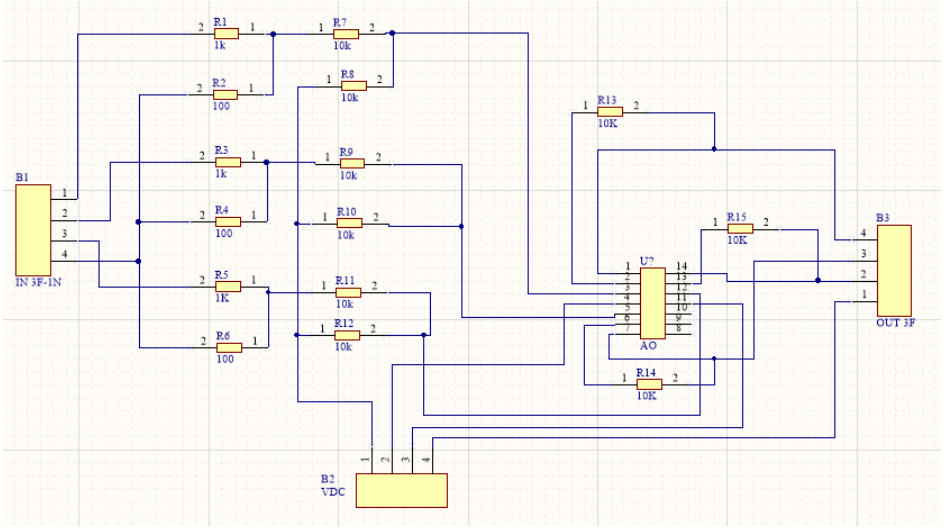


Figura 3.4: Diseño esquemático del circuito de acoplamiento

Los materiales utilizados en el diagrama de la figura 3.4 se detallan en la tabla 3.3.

Tabla 3.3: Materiales para el diseño del circuito de acoplamiento

Elemento	Descripción	Cantidad
Bornera (B1, B3)	Para PCB 4 vias	3
Resistencia (R7-R15)	10kOhm a 1/2W	9
Resistencia (R1-R3-R5)	1kOhm a 1/2W	3
Resistencia (R2-R4-R6)	100Ohm a 1/2W	3
Amplificador Operacional (U1)	LM324N	1

3.1.2. Configuración de PSIM

La configuración del software **PSIM** resulta ser importante para obtener el código y grabarlo en la placa de control, con esto, se utilizan las librerías específicas de la tarjeta de control **TI 28335**, entre sus funcionalidades se encuentra el módulo convertidor A/D, el módulo de comunicación SCI, las interfaces **PWM**, entre otras.

En la figura 3.5 se presentan una parte de las librerías usadas por la tarjeta F28335, hay que considerar que en la sección de herramientas las opciones marcadas con **TI** o **CG**, son propicias para la configuración y generación del código.

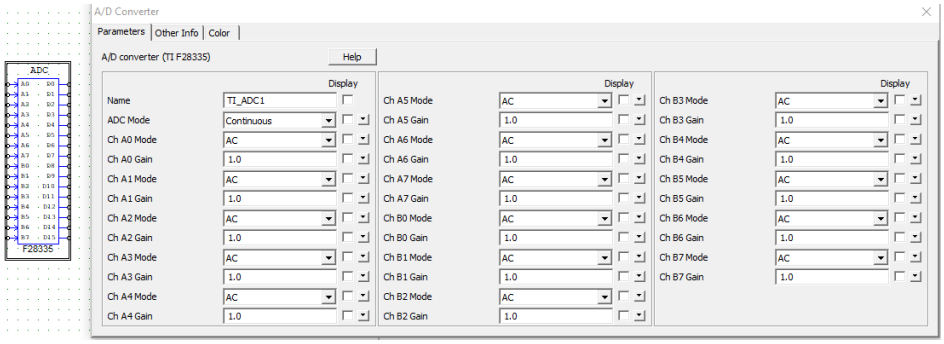


Figura 3.6: Bloque Convertidor Analógico/Digital de PSIM

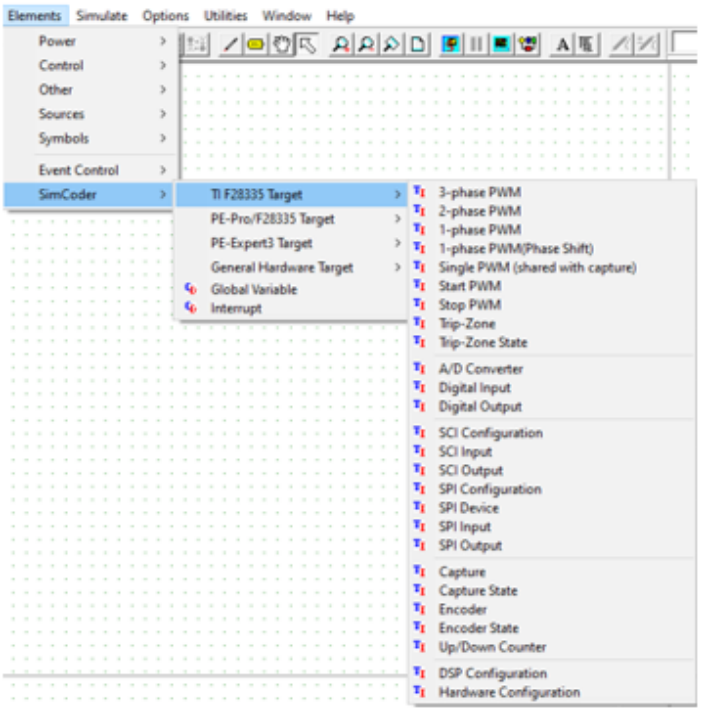


Figura 3.5: Librerías de PSIM para la tarjeta F28335

Para la configuración de los puertos analógicos, se utiliza el módulo convertidor A/D interno del DSP, para esto en la dirección **Elements-SimCoder-TIf28335 Target** tenemos el **A/D Converter**, en la figura 3.6 se presenta el diagrama del módulo y las características de configuración, en ellas permite declarar el tipo de lectura y la ganancia del puerto.

PSIM permite realizar simulaciones sin necesidad de conectar la tarjeta de control, por lo que se debe configurar el modo de simulación, para ello en la dirección **Simulate-Simulation Control**, se tiene el controlador de simulación. Este permite realizar las simulaciones acorde a la capacidad de PC que se utiliza, variando el tiempo de simulación y pasos para la toma de datos, además, en la parte inferior, nos permite configurar la tarjeta de hardware a usar, con esta herramienta se realiza una primera compilación de los elementos utilizados, ya que si uno de ellos no pertenece a la librería de la tarjeta, nos presenta error en la simulación. En la figura 3.7 se puedo observar la ventana de configuración del control.

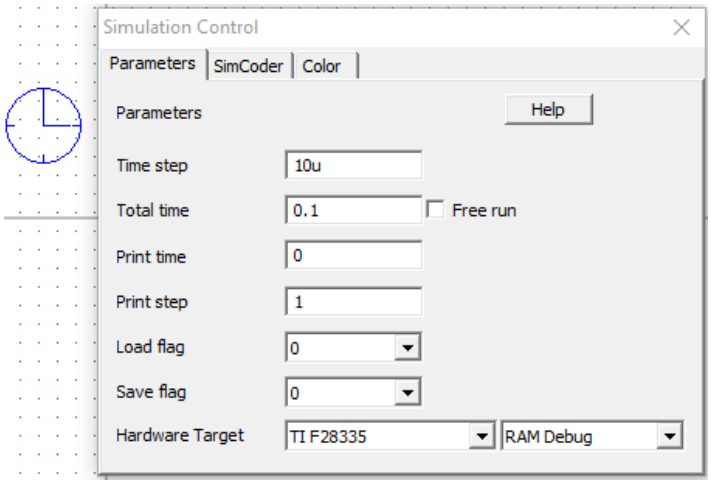


Figura 3.7: Configuración del control de simulación

Con las configuraciones establecidas, se leen los datos de entrada, en la figura 3.8 se presenta el esquema utilizado para el mismo, en el esquema presentado, además se utiliza los módulos de salida de la interfaz SCI con ello se puede visualizar las señales de entrada utilizando **PSIM**.

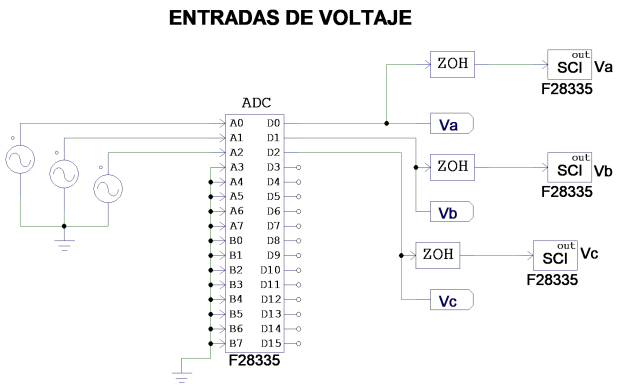


Figura 3.8: Lectura de las señales de ingreso, para su análisis

3.1.3. Generación de código C/C++

El DSP utiliza el lenguaje C/C++ para su funcionamiento, **PSIM** permite generar el código en lenguaje C/C++, del esquema configurado en la hoja de trabajo, es por ello, que los elementos utilizados en el esquema deben ser válidos para la tarjeta de control. Así, generando el código utilizamos el software CCS que permite compilar el código generado, para posteriormente grabar en la memoria de la tarjeta de control.

Para generar el código a partir de **PSIM**, nos vamos a la dirección **Simulate-Generate Code**, y se obtiene el código presentado en el apéndice C, el mismo, pertenece al esquema de la figura 3.8.

Es importante guardar el esquema, ya que al momento de generar el código, se crea un archivo que contiene los parámetros para transferir el código a CCS. La carpeta que se genera, lleva el mismo nombre del archivo del esquema con la terminación **C code**.

3.1.4. Configuración de CCS

CCS es un software de desarrollo integrado, desarrollado por Texas Instruments, esta herramienta computacional permite cargar el código y sus configuraciones, compilar en busca de errores y grabar en la memoria de la tarjeta de control. En el apartado anterior se mencionó que **PSIM** permite la compilación del esquema y genera los archivos necesarios para importar en CCS.

En el Anexo B, se presentan los pasos para importar el proyecto en CCS, al finalizar el proceso, se configura la tarjeta que se va a utilizar, para ello en la dirección **Project-Properties**, se nos habilita una ventana, como se presenta en la figura 3.9. En ella podemos escoger la tarjeta que vamos a utilizar, el compilador para la tarjeta.

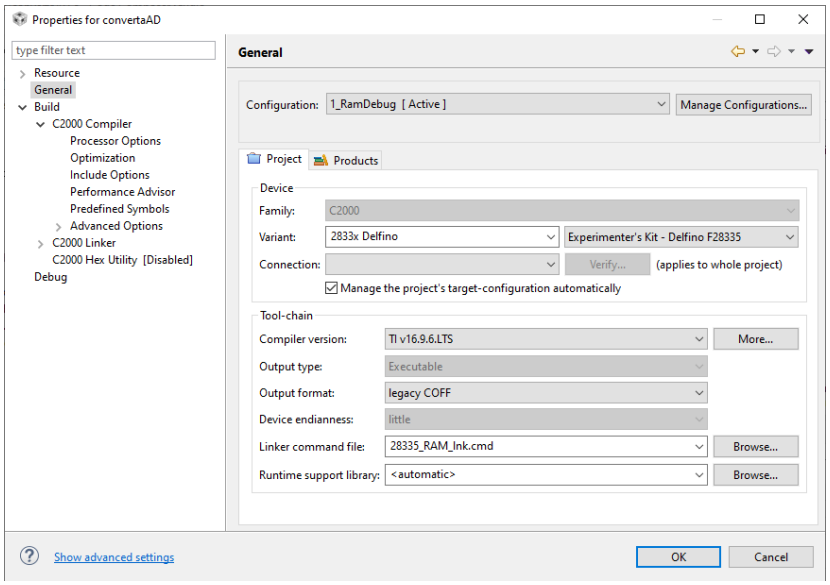


Figura 3.9: Configuración de CCS para la compilación y la exportación a la tarjeta

3.2. Configuración del puerto serial para la visualización de datos por medio de la interfaz SCI

El puerto serial es una interfaz de comunicación de datos digitales, el DSP incorpora este tipo de comunicación, a partir de su configuración, se puede realizar la lectura de datos en tiempo real ya sea en CCS o en **PSIM**.

3.2.1. Diseño, Implementación y Configuración de la comunicación serial entre PC y DSP

La comunicación entra la PC y el DSP se da por medio del puerto **RS-232**, para ello se utiliza un convertidor **USB-Serial**. La conexión entre el puerto de comunicación y la tarjeta de control es cruzada como se presenta en la figura 3.10.

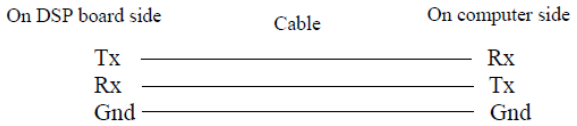


Figura 3.10: Conexión de los puertos de la PC y DSP

El esquema mostrado en la figura 3.11, representa la conexión física que se debe implementar para la comunicación serial.

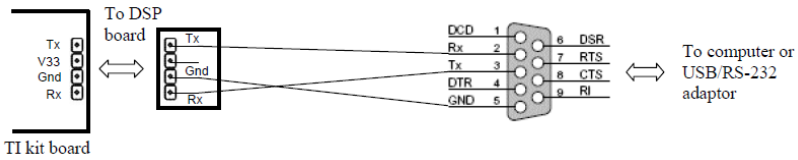


Figura 3.11: Diagrama de conexión entra la PC y el DSP

Es así que en la figura 3.12, se presenta el resultado del proceso mencionado.

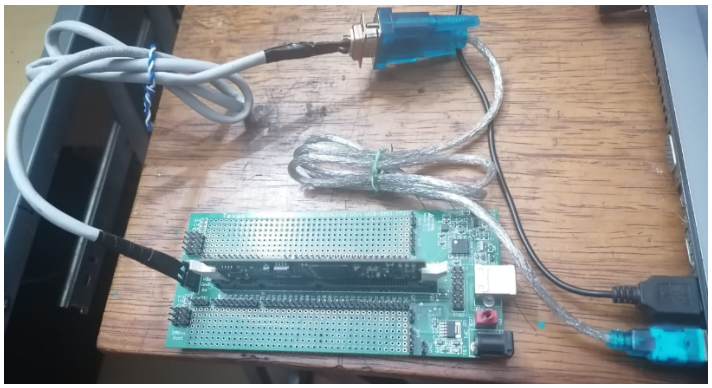


Figura 3.12: Conexión física para la comunicación serial

Además, es importante instalar el controlador del convertidor, con ello se puede configurar los datos de comunicación, en la sección de configuración de dispositivos de Windows, existen parámetros, tales como, el nombre del puerto, la tasa de datos, la longitud de datos, la paridad, etc. (figura 3.13)

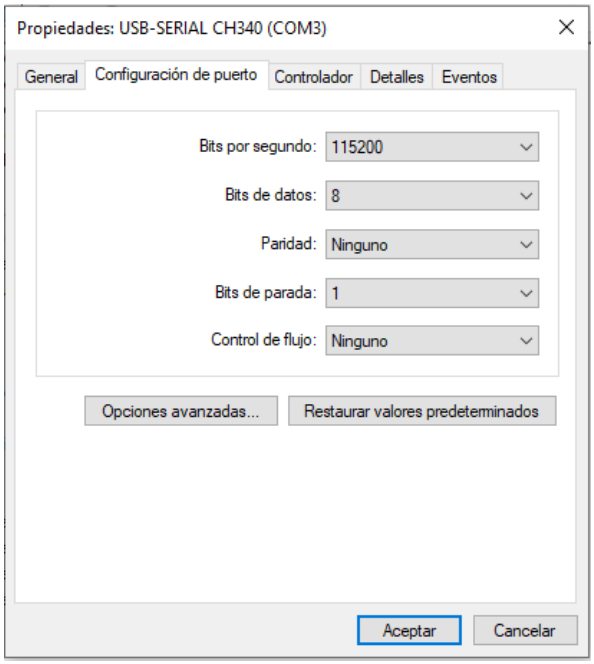


Figura 3.13: Configuración del puerto de comunicación

3.2.2. Implementación de la conexión para la comunicación serial en PSIM

Para establecer la comunicación entre la tarjeta de control de **PSIM** se utiliza el módulo **SCI**, esta interfaz permite realizar una comunicación bidireccional con el entorno de visualización y la tarjeta de control. Para utilizar esa interfaz se debe declarar y configurar los parámetros que se presenta en la figura 3.14

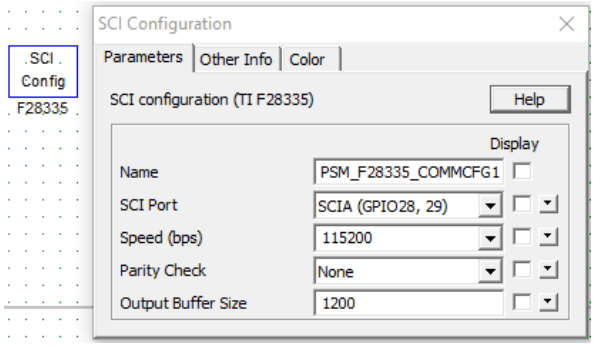
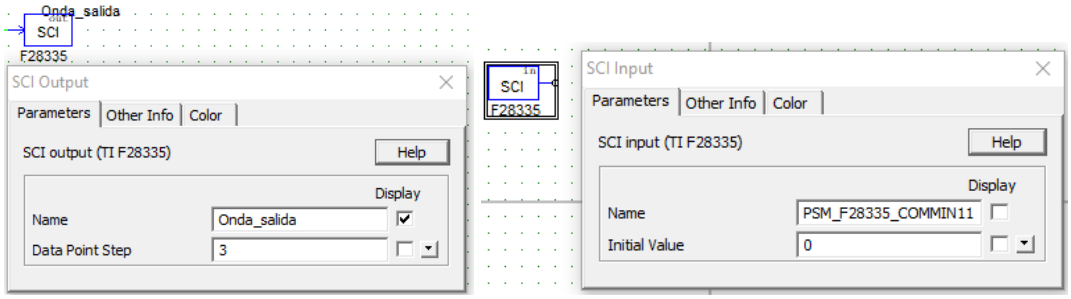


Figura 3.14: Configuración General para el módulo SCI

La interfaz integra módulos de entrada (figura 3.15a) y salida de datos (figura 3.15a). Estos permiten cambiar los valores de las variables establecidos en el esquema con el código en ejecución.



(a) Módulo para salida de datos (b) Módulo para entrada de datos

Figura 3.15: Configuración de los módulos I/0 para la interfaz SCI

3.3. Visualización de datos

Para la visualización de datos, se realiza tanto en software como en hardware, a continuación se detalla el proceso para la visualización.

3.3.1. Utilizando el Osciloscopio de PSIM

En **PSIM** se puede visualizar los datos en tiempo real, capturados por el DSP, a partir de la comunicación establecida por la interfaz SCI. Esta herramienta se encuentra en la barra de herramientas en la dirección **Utilities-DSP Oscilloscope**.

La interfaz de usuario se presenta en la figura 3.16, en ella se puede escoger el puerto de comunicación, la tasa de datos, la paridad, además que permite hacer un test de comunicación.

En esta interfaz, cuando se establece la comunicación, nos presenta las variables de entrada o salida configurada en los módulos mencionados.

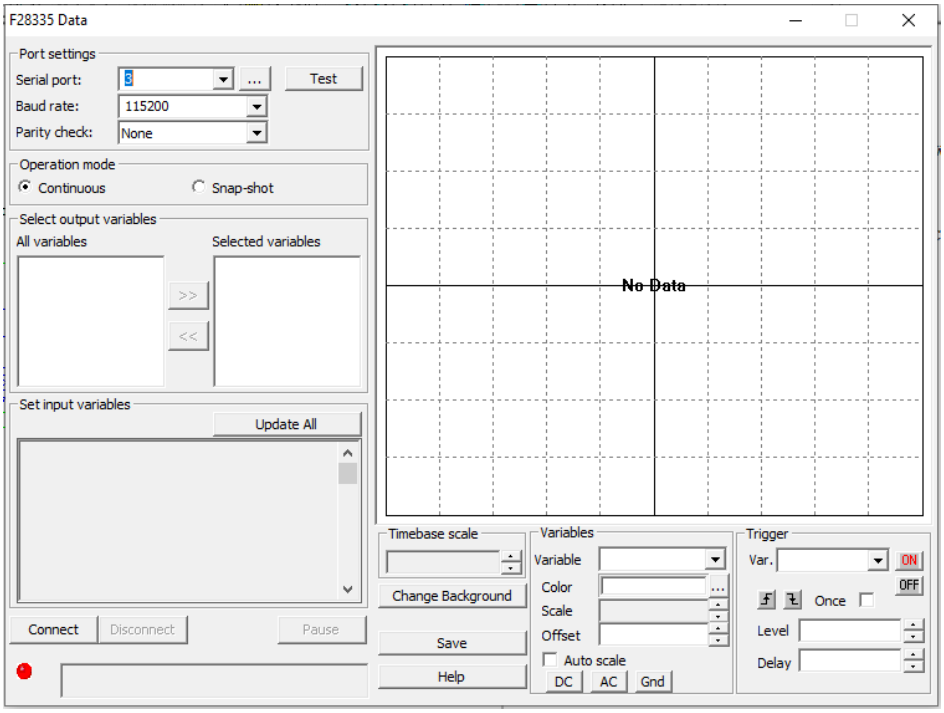


Figura 3.16: Interfaz de usuario del Osciloscopio

3.3.2. Utilizando el osciloscopio digital físico

Para la visualización de datos por medio del osciloscopio digital se utiliza el módulo **PWM** de las salidas digitales, en la figura 3.17 se presenta el bloque para la configuración de los puertos **PWM** para la tarjeta de control, en esta configuración es necesario bloquear la configuración, con el fin de que estos puertos no se puedan utilizar para entrada de datos.

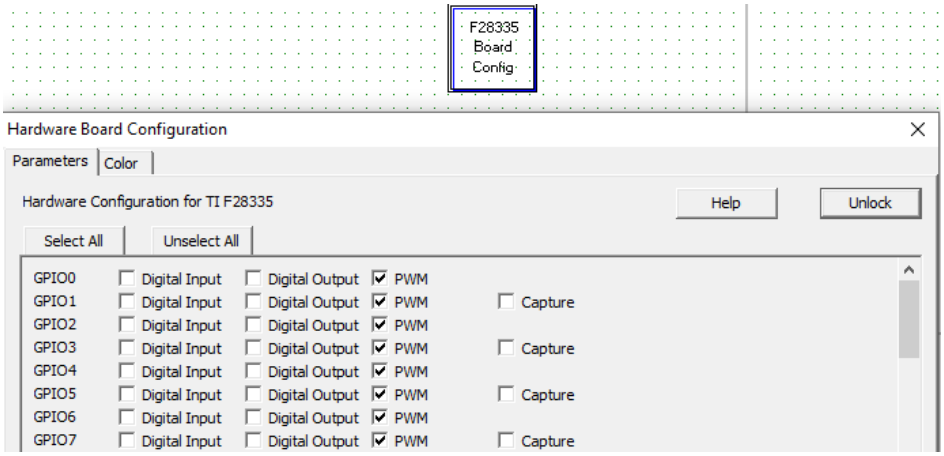


Figura 3.17: Configuración de la tarjeta para el uso de los puertos digitales como PWM

A partir de la configuración de la tarjeta se debe configurar los parámetros para el **PWM** en la figura 3.18 se presenta los parámetros que nos permite configurar.

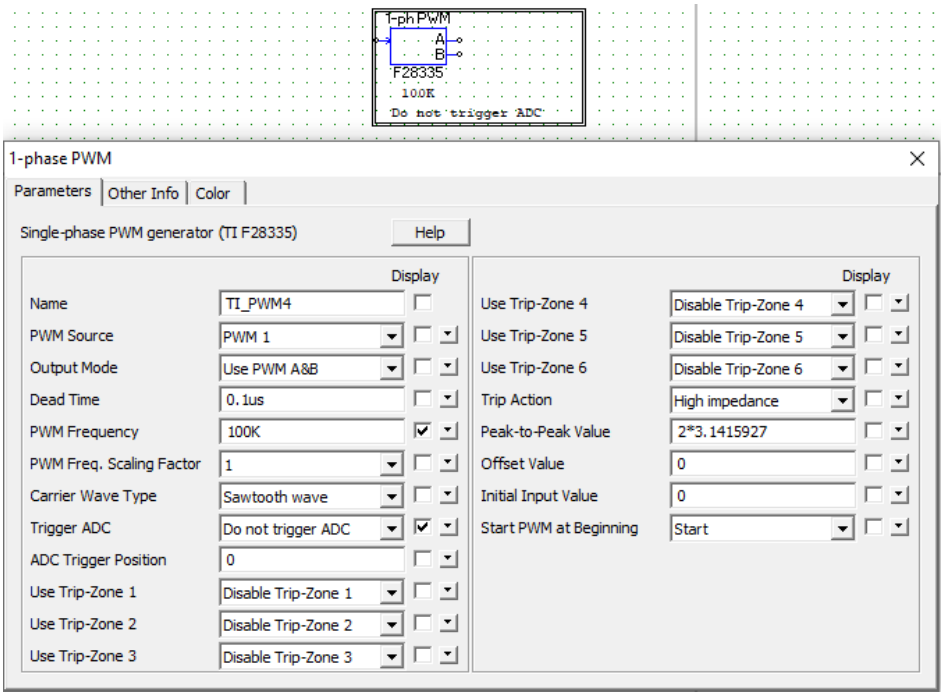


Figura 3.18: Configuración de los parámetros del puerto PWM

El proceso presentado en el apartado anterior, nos entrega una onda cuadrada, para visualizar una onda **dientes de sierra**, se diseña un circuito integrador como un filtro paso bajo, ya que el condensador se carga cuando la entrada es de frecuencia alta y se descarga cuando la frecuencia es baja, la onda varía con respecto al ciclo de trabajo de la onda cuadrada, en la figura 3.19 se presenta el diagrama del mismo, para esto se utiliza un **capacitor=10uF** y una resistencia variable para ajustar la señal.

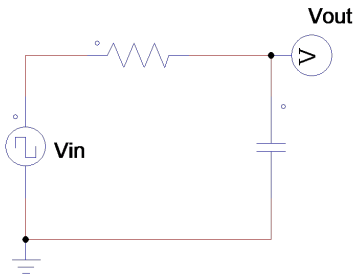


Figura 3.19: Circuito integrador como filtro paso bajo

3.4. Implementación de algoritmos de sincronización de fase

Cuando se obtiene la señal trifásica del circuito de acoplamiento, se realiza una conversión de un sistema de coordenadas de tres fases a un sistema de dos componentes de un marco estacionario ortogonal $\alpha\beta$, estas componentes se visualizan como dos señales sinusoidales desplazadas 90° entre si. (figura 3.20) [61]

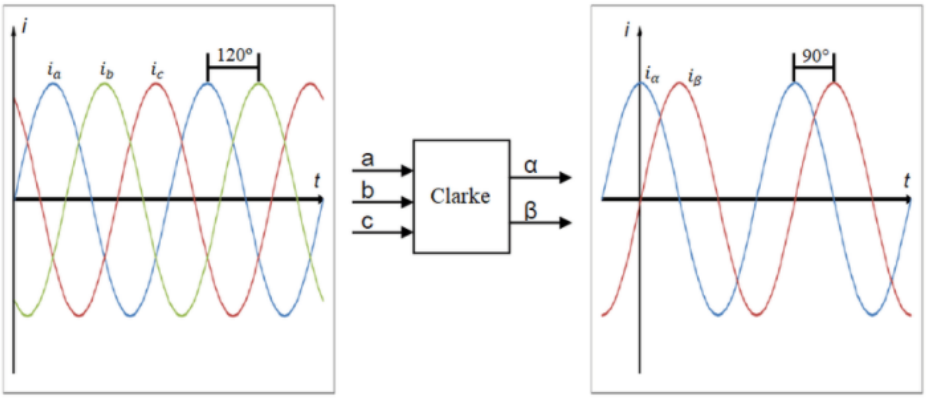


Figura 3.20: Paso de un sistema de coordenadas de 3 fases a un sistema estacionario ortogonal por medio de la transformada de Clarke

3.4.1. Normalización de las señales de entrada

La normalización de la señal, es importante para que no exista variaciones de voltaje debido a las distorsiones de la señal de entrada, esto permite tener las referencias $\alpha\beta$ en un estado constante, para ello se utiliza la ecuación 3.3 para obtener la normalización.

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \frac{\hat{e}_\alpha}{\sqrt{\hat{e}_\alpha^2+\hat{e}_\beta^2}} \\ \frac{\hat{e}_\beta}{\sqrt{\hat{e}_\alpha^2+\hat{e}_\beta^2}} \end{bmatrix} \approx \begin{bmatrix} cos\theta(k) \\ sin\theta(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \tag{3.3}$$

Con estas premisas, en la figura 3.8 se presenta el esquema para la lectura de datos, se utilizan tres canales analógicos correspondientes al convertidor A/D, se utiliza además la interfaz de comunicación SCI para la visualización de datos.

A partir de estas señales, se realiza la conversión a los componentes $\alpha\beta$, con ello, se realiza la normalización de la señal con las fórmulas de la ecuación 3.3, en la figura 3.21 se presenta el esquema de esta operación.

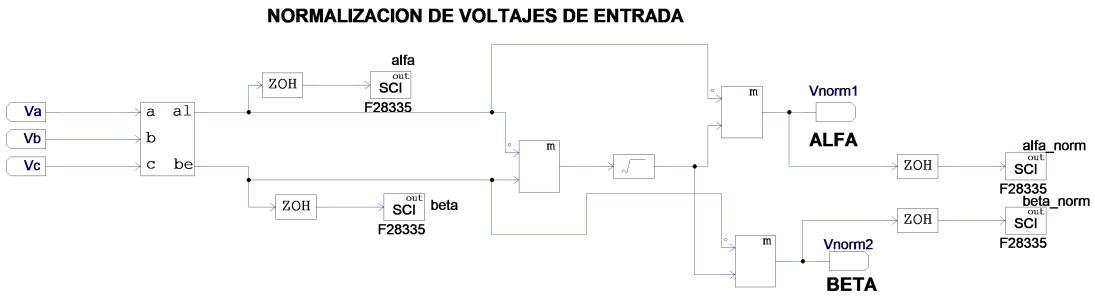


Figura 3.21: Normalización de la señal trifásica, utilizando las componentes $\alpha\beta$

3.4.2. Establecimiento de parámetros del PI

Para establecer los parámetros del controlador, se establece las funciones de transferencias y las variables mencionadas en la sección 2.4.2. En la tabla 3.4 se presenta los datos usados para el controlador.

En el anexo D se presenta el código utilizado para el cálculo de los mismo, se ha utilizado el software de **MATLAB**.

Tabla 3.4: Parámetros para el controlador PI utilizados en el sistema

Respuesta	**Ts**	**Kp**	**Ti**
rápida	1ms	9200	0.00027
media	3ms	3066	0.00065
lenta	6ms	1533	0.0013

Con estos valores, se tiene una respuesta al impulso como se presenta en la figura 3.22, se tiene una oscilación del sistema debido a la componente integral del mismo. En la figura 3.23, se tiene las raíces y polos del sistema, debido a que el sistema es de segundo orden se tiene dos raíces y un polo, en la gráfica se puede observar que el sistema es estable considerando los parámetros de la tabla 3.4.

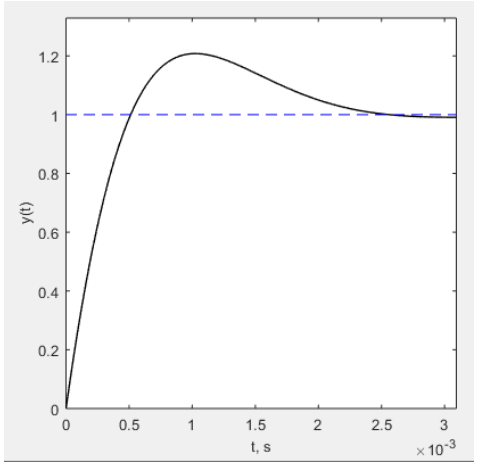


Figura 3.22: Respuesta al impulso del sistema con un tiempo de establecimiento de 3ms

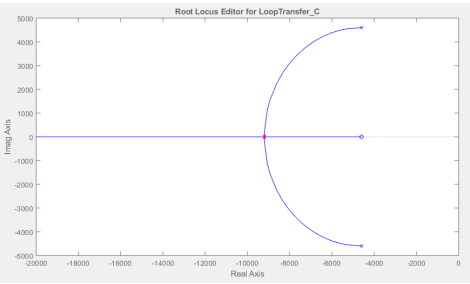


Figura 3.23: Raíces y Polos del sistema utilizados para análisis de estabilidad

Finalmente la estabilidad del sistema, se puede corroborar con el diagrama de bode, como se puede observar en la figura 3.24, el sistema tiene un margen de ganancia positivo, por lo que es coincidente con el análisis de polos y ceros. En el diagrama de fase, se marca la frecuencia de corte, que indica el pico del sobre impulso de la respuesta del sistema.

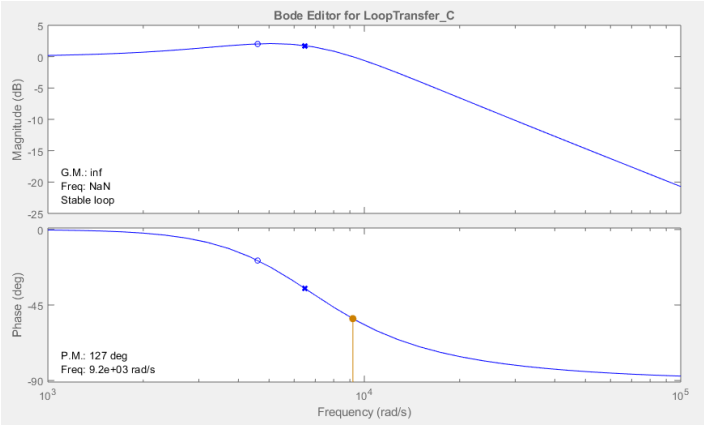


Figura 3.24: Diagrama de bode para análisis de estabilidad del sistema

3.4.3. Distorsión de frecuencia en la señal de entrada

Para concepto de simulación y prueba de los algoritmos, se realiza un esquema que permite variar la frecuencia de la señal, en cualquier punto de la simulación. En la figura 3.25 se presenta el sub-circuito que permite la distorsión de frecuencia, en este esquema, inicialmente se genera una señal continua de amplitud **60V**, luego, se integra un pulso que permite elevar la amplitud en un tiempo determinado, en nuestro caso elevamos la amplitud a **80V** en **0.5 segundos** de la simulación, este proceso es para obtener la frecuencia de la señal. Además, en el bloque proporcional, multiplicamos la señal por **2***π, que permite obtener la **velocidad angular en radianes**, con esta señal, se realiza la operación integral de la misma, obteniendo una **onda diente de sierra**.

Con la onda resultante, se utiliza los bloques de funciones **senoidales en radianes** proporcionados por **PSIM**, para obtener la forma de onda senoidal. Para los voltajes **Vb y Vc**, se suma a la onda diente de sierra, el desfase en radianes ($\frac{2*\pi}{3}$ **y** $\frac{4*\pi}{3}$ **respectivamente**). Las etiquetas **V1, V2 y V3** permiten utilizar las ondas resultantes en toda la hoja de trabajo.

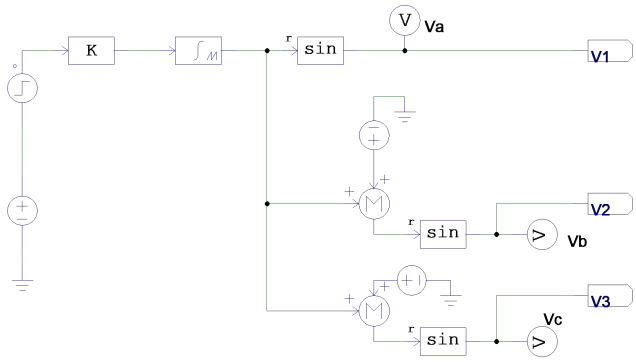


Figura 3.25: Sub-circuito para la distorsión de frecuencia en la simulación

Con este esquema, la respuesta se presenta en la figura 3.26, como se observa en 0.5s se realiza el cambio de frecuencia de la señal de entrada. Con esto se puede corroborar de mejor forma la respuesta de los algoritmos.

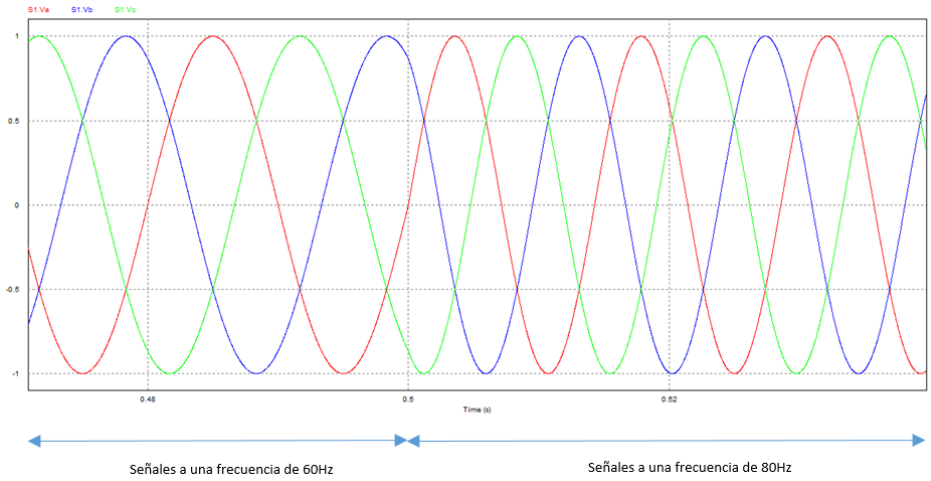


Figura 3.26: Respuesta de la distorsión de frecuencia para simulación

3.4.4. Distorsión armónica en la señal de entrada

Por otro lado, la distorsión armónica también es un parámetro importante que nos permite establecer el correcto funcionamiento de los algoritmos, como concepto de simulación se ha introducido a la señal de entrada el **3er y 5to armónico**, en la figura 3.27, se presenta el esquema utilizado para introducir armónicos a las señales trifásicas.

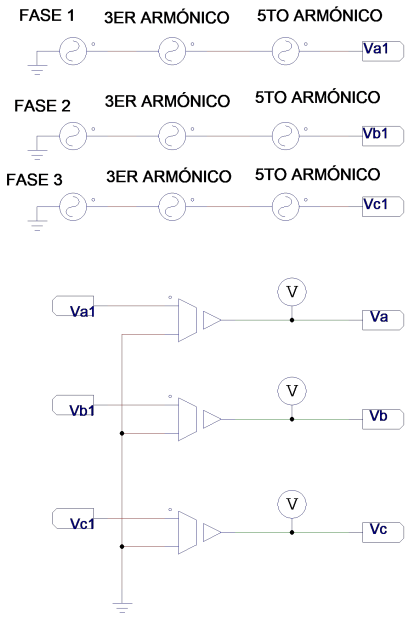


Figura 3.27: Distorsión armónica para las tres fases de entrada

Con los esquemas presentados, en la figura 3.28 se puede apreciar, que en las señales de entrada, se tiene la presencia de armónicos, es decir no se tiene una señal completamente pura, este caso permite idealizar una red con fallas permitiendo corroborar la respuesta de los algoritmos a estas perturbaciones.

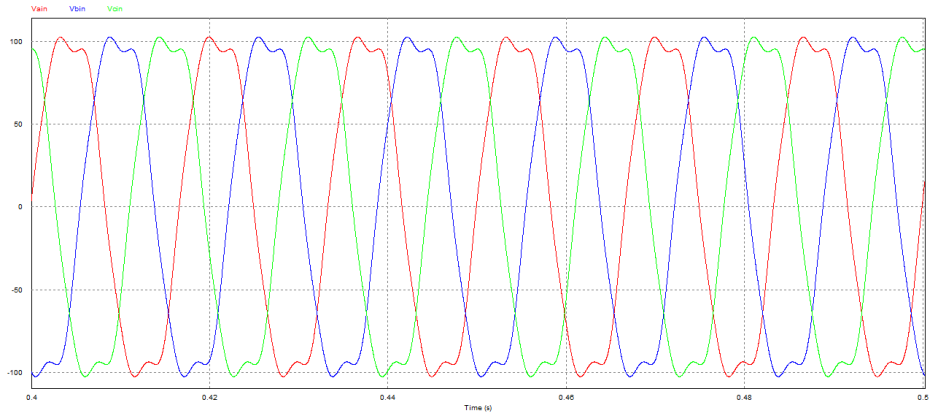


Figura 3.28: Señales trifásicas con distorsión armónica

3.4.5. Diseño e implementación del algoritmo dqPLL

En la sección 2.4.2.1, se explica el desarrollo del algoritmo **dq-PLL**, por lo que, basados en el diagrama, se diseña el esquema en PSIM. El esquema presentado en la figura 3.21 esta diseñando considerando una frecuencia de muestreo de **10kHz**.

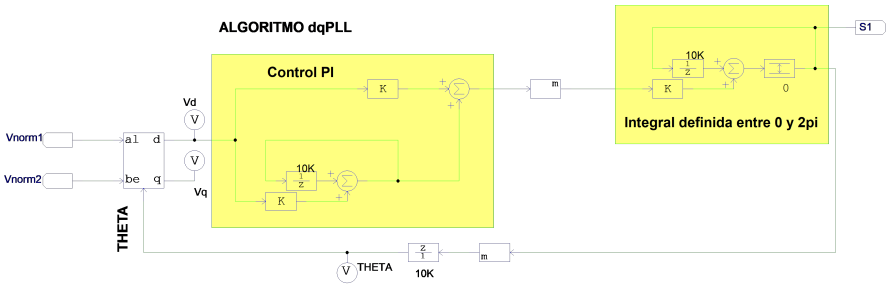


Figura 3.29: Esquema del algoritmo dq-PLL en PSIM

3.4.6. Diseño e implementación del algoritmo $\alpha\beta$PLL

En este algoritmo se ha utilizado bloques para funciones matemáticas ya que como nos presenta el diagrama en la sección 2.4.2.2, este algoritmo utiliza la función seno y coseno para lograr estimar el ángulo de fase.

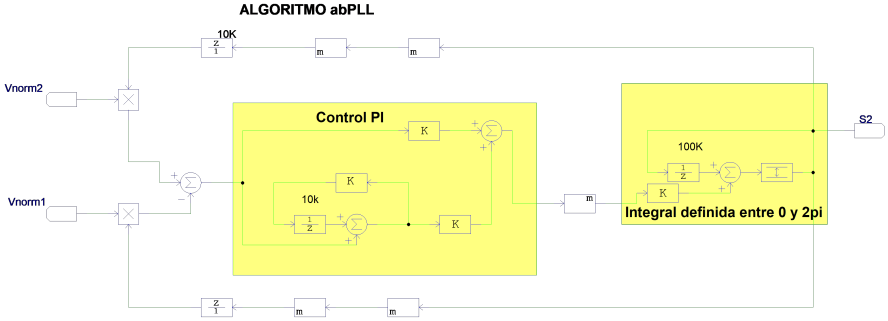


Figura 3.30: Esquema del algoritmo $\alpha\beta$PLL en PSIM

3.4.7. Diseño e implementación del algoritmo DDSRF-PLL

Para el diseño del algoritmo, se realizaron 3 sub-circuitos, el diagrama general se presenta en la figura 3.31, El primer sub-circuito implementado corresponde a la transformada **dq** por lo que se necesita obtener las componentes de secuencia positiva y negativa de la señal de entrada, esto se presenta en la figura 3.32.

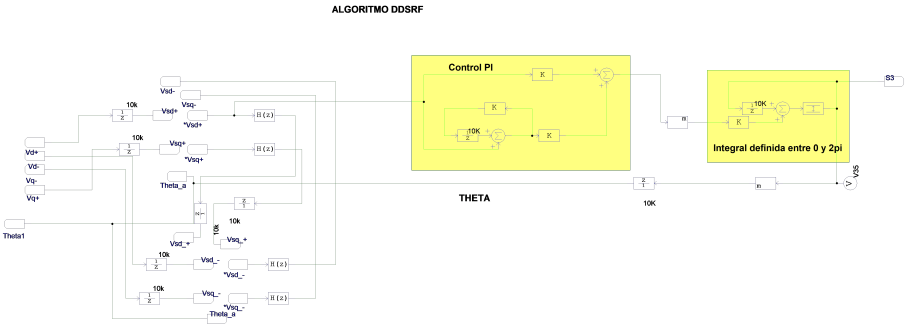


Figura 3.31: Esquema del algoritmo DDSRFPLL en PSIM

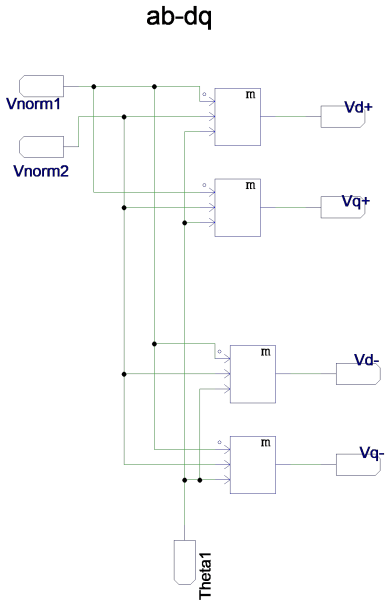


Figura 3.32: Transformada $\alpha\beta$ a dq

Cuando se obtienen las dos secuencias, pasan por los esquemas de desacople, esto permite calcular la magnitud del voltaje de la red, a partir de la componente DC, luego de eliminar el efecto oscilatorio que se produce entre ambas secuencias. En las figuras 3.33 y 3.34 se presenta los esquemas para dicho proceso.

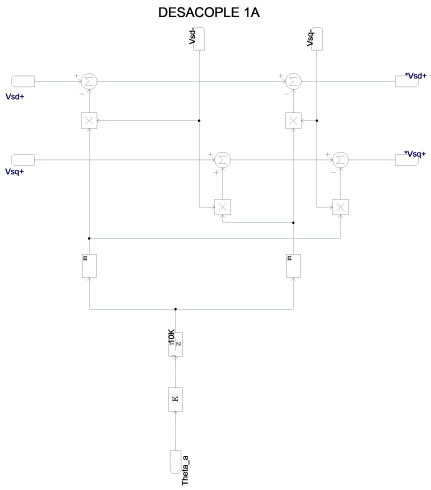


Figura 3.33: Esquema de desacople para la secuencia positiva

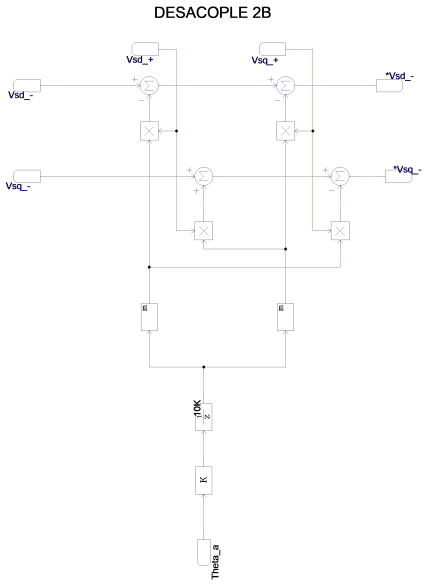


Figura 3.34: Esquema de desacople para la secuencia negativa

3.4.8. Diseño e implementación del algoritmo DSOGI-PLL

Para el diseño de DSOGI, teniendo en cuenta el esquema de la sección 2.4.2.5, por lo tanto el esquema del algoritmo se presenta en la figura 3.35.

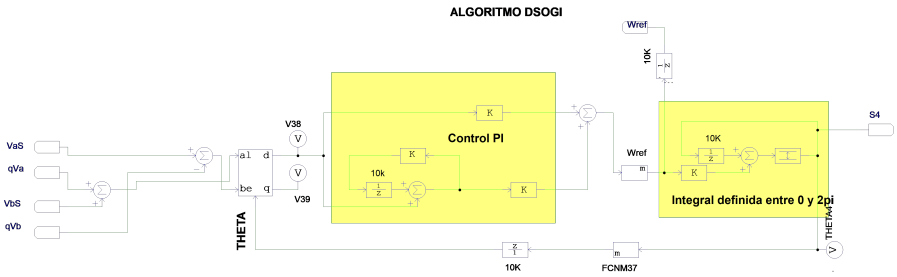


Figura 3.35: Esquema del algoritmo DSOGIPLL en PSIM

Los esquemas diseñados para el **SOGI** se presentan en las figuras 3.37 y 3.36, este es el mismo esquema para cada componente.

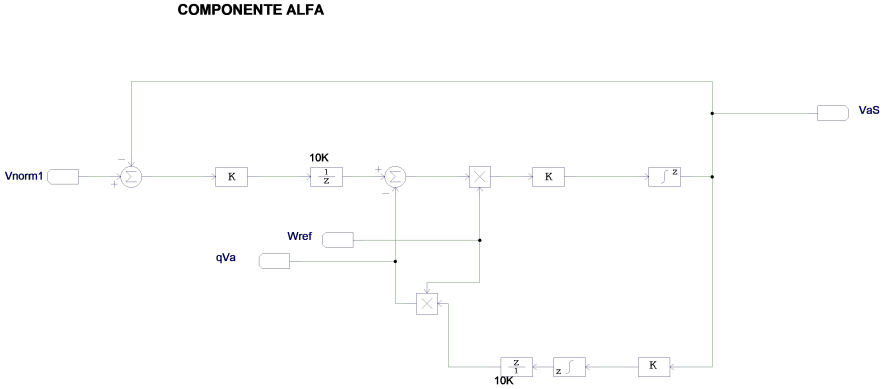


Figura 3.36: Esquema del algoritmo SOGI-PLL en PSIM para la fase α

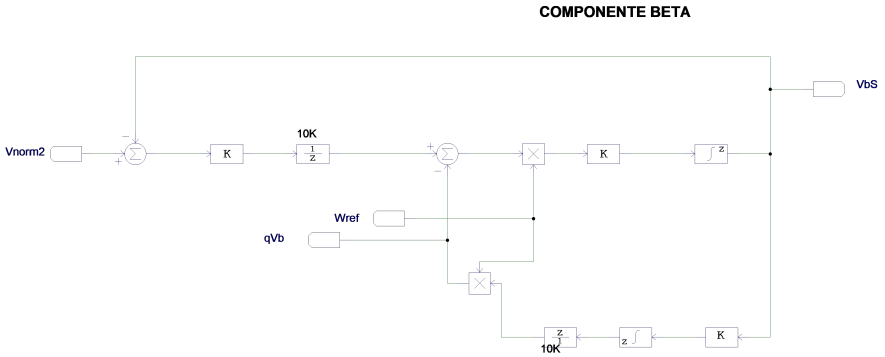


Figura 3.37: Esquema del algoritmo SOGI-PLL en PSIM para la fase β

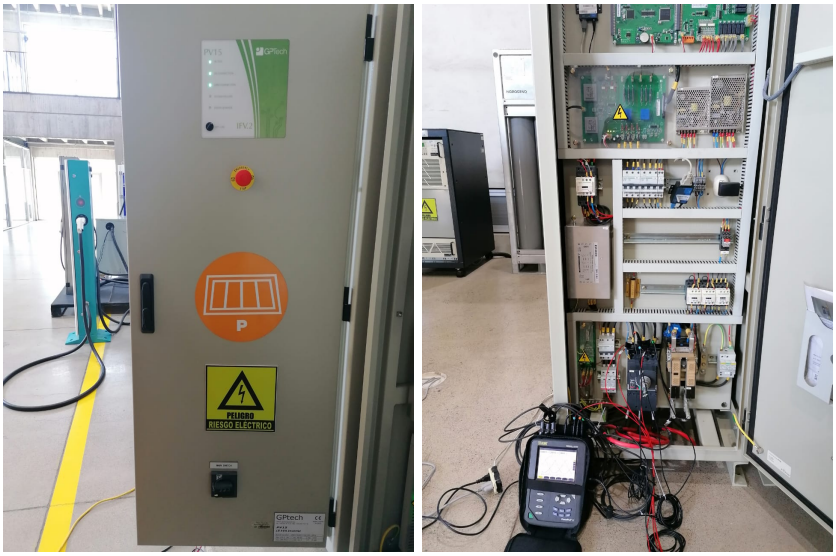
Capítulo 4

Resultados, Conclusiones y Recomendaciones

En este capítulo se detallan los resultados obtenidos en las pruebas realizadas con cada algoritmo de sincronización de fase. Inicialmente en la sección 4.1 se detalla las características de la fuente renovable utilizada para el presente trabajo de titulación, además que se presentan las conexiones realizadas para el acondicionamiento de señales, posteriormente se presentan los resultados de las señales de entrada en la sección 4.2. Luego en la sección 4.3 se presentan los resultados de la normalización de las mismas, posterior a ello, se presenta la respuesta de los algoritmos diseñados en la sección 4.4, luego en la sección 4.5 se realiza un breve análisis a los datos obtenidos en la sección anterior, luego se presentan las conclusiones y recomendaciones en 4.6 y finalmente se proponen trabajos futuros en 4.7 basados en el presente trabajo de titulación.

4.1. Esquema para la adquisición de datos

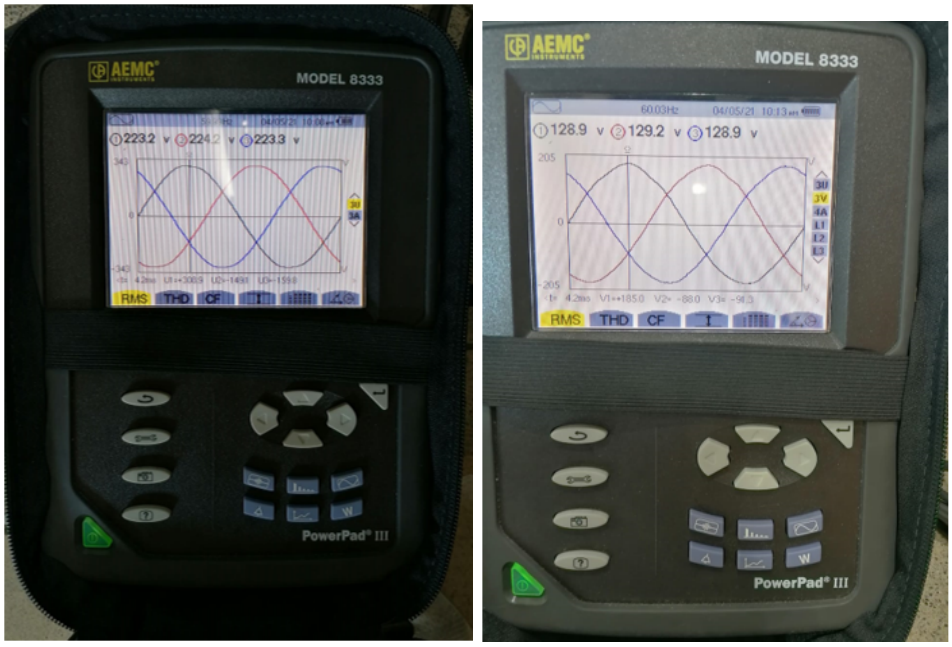
Para la adquisición de datos se utilizó la señal de voltaje de los **paneles solares poli-cristalinos del laboratorio de microred de la Universidad de Cuenca**, la conexión se realizó en el breaker de salida del tablero de control para los paneles **IFV.2**, después del inversor. (figura 4.1)



(a) Vista externa (b) Vista interna

Figura 4.1: Tablero inversor para tensión de PANELES SOLARES POLI-CRISTALINOS

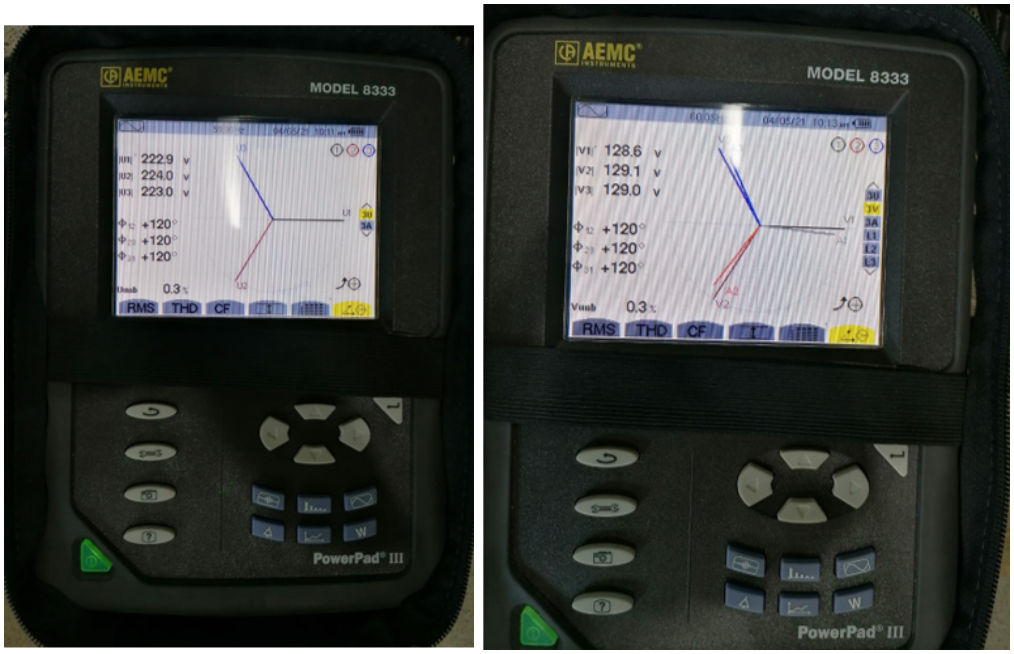
Además se utilizó el analizador de red **AEMC modelo 8333** para visualizar voltajes de fase, voltajes de línea y fases de cada señal, como se observa en la figura 4.2a los voltajes de fase, aproximadamente **223VAC** y en la figura 4.2b los voltajes de linea aproximadamente **128VAC**



(a) Voltajes entre fases (b) Voltajes entre lineas

Figura 4.2: Voltajes de salida del inversor para la entrada del circuito de acoplamiento

También se visualizan los ángulos de fase entre fases (figura 4.3a) y entre líneas (figura 4.3b), como se observa, existe un desfase de **120°** entre líneas, por lo que se corrobora que después del inversor la señales están correctas y que se pueden introducir al circuito de acoplamiento, para el procesamiento.



(a) Ángulos de fase entre fases de las señales (b) Ángulos de fase de lineas

Figura 4.3: Ángulos de desfases de las señales de salida del inversor para la entrada del circuito de acoplamiento

A partir de ello, se realizó la conexión al breaker del tablero del transformador, en la figura 4.4 se presenta la conexión realizada al breaker.

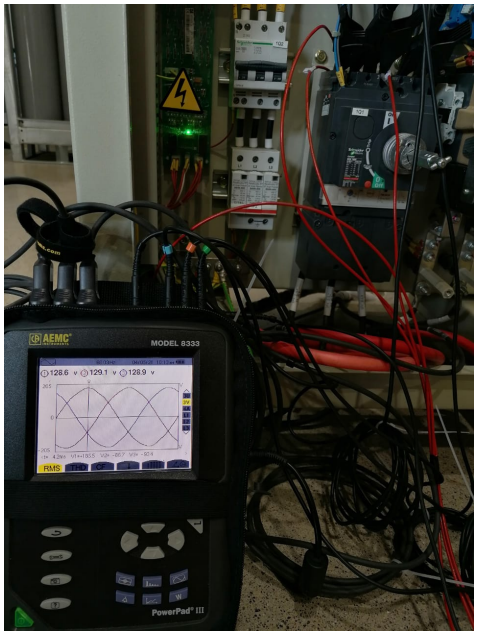


Figura 4.4: Conexión al breaker principal de los transformadores

En el esquema presentado en la figura 4.5 se visualiza la conexión realizada desde la reducción de voltaje pasando por el circuito de acondicionamiento.

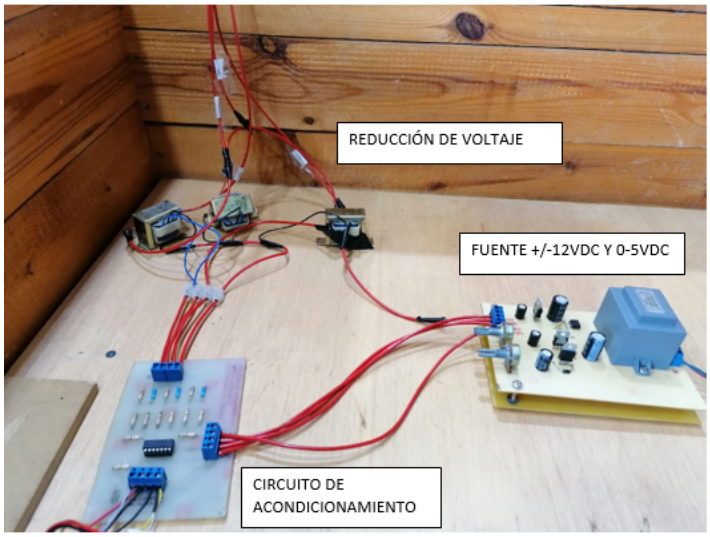


Figura 4.5: Conexión al circuito de acondicionamiento para el DSP

4.2. Señales de entrada, tomadas a la salida del circuito de acondicionamiento

Una vez realizada la conexión de las fases y los circuitos, es necesario corroborar que las señales estén dentro del rango permitido por las entradas analógicas del DSP, por lo que para ello se utiliza el osciloscopio manual que permite realizar la conexión de la figura 4.6.

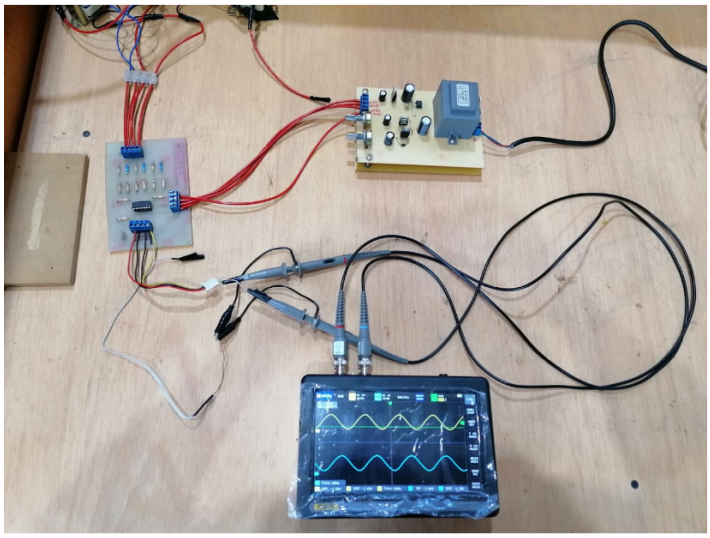
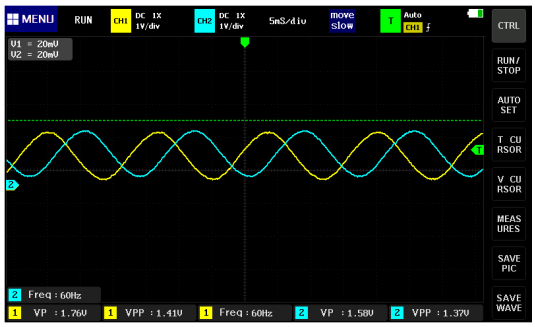


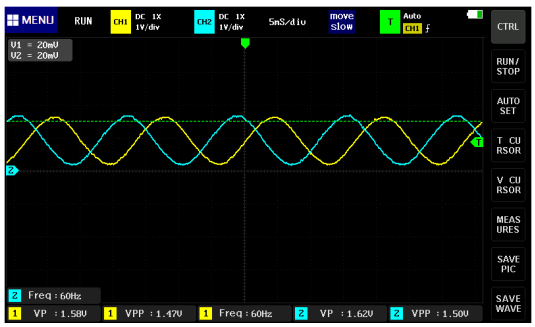
Figura 4.6: Conexión del Osciloscopio para las pruebas de la señal de entrada

Debido a que el mismo tiene solo dos canales de lectura, se realizó tres lecturas a la salida del circuito de acondicionamiento, obteniendo los resultados de la figura 4.7. Para estos resultados se utilizaron **transformadores de 120V-9V**, con **voltaje DC en el circuito sumador de 1.6V**, se puede observar en cada medición, que el voltaje se encuentra dentro de las características, por lo que

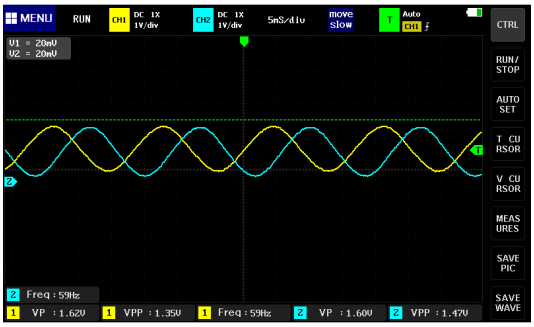
se puede ingresar estos voltajes al DSP.



(a) Señales de entrada **F1-F2**



(b) Señales de entrada **F1-F3**



(c) Señales de entrada **F2-F3**

Figura 4.7: Señales de entrada al DSP tomadas a la salida del circuito de acondicionamiento

Una vez corroborado estas señales, se visualizan utilizando la interfaz SCI, para uso del osciloscopio de **PSIM** obteniendo las señales de la figura 4.8, con estos resultados se puede tener la seguridad que los parámetros de entrada están acorde a las características del DSP y que las señales de entrada para el procesamiento están correctamente establecidas, lo que permite asegurar que los circuitos de acoplamiento y de alimentación están correctamente diseñados.

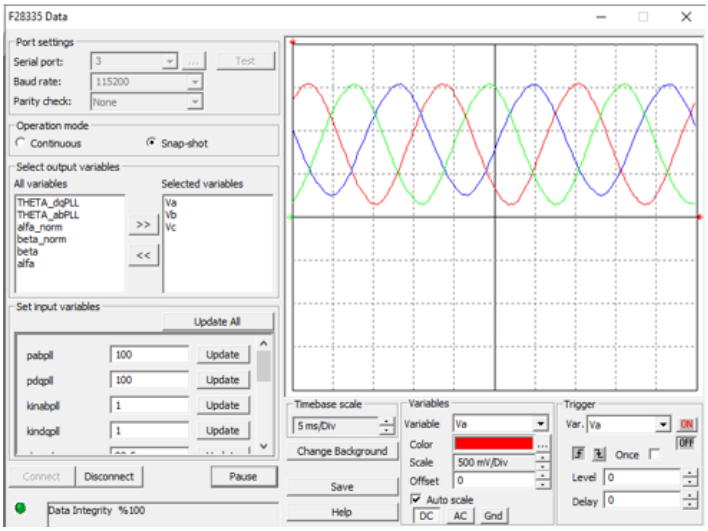


Figura 4.8: Señal trifásica de entrada en los puertos analógicos del DSP

4.3. Normalización de la señal para la entrada de los algoritmos

Una vez que se tiene las señales de entrada, se procede al procesamiento de la señal, por lo que se corroboró que la transformada de Clarke, nos presenta dos componentes **desfasadas 90 grados** entre si, en la figura 4.9 se presenta los resultados de la transformada.

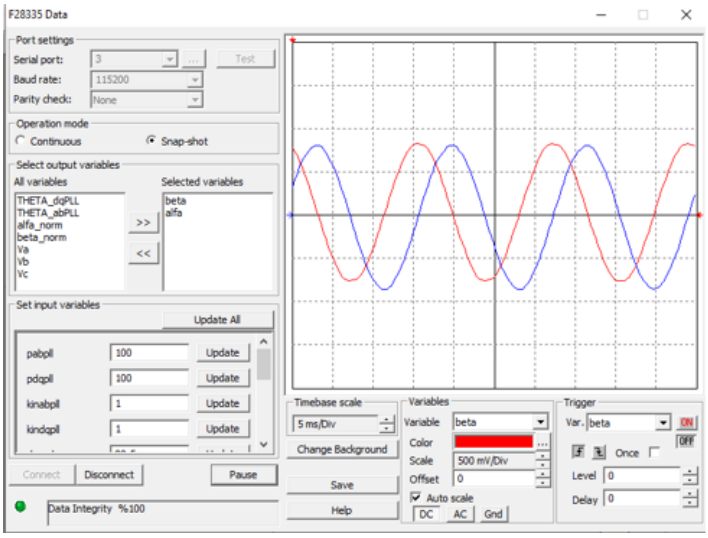


Figura 4.9: Componentes $\alpha\beta$ de la transformada de Clarke

Con los resultados presentados, observamos que debido a las alteraciones en las señales de entrada, existen también variaciones de tensión en las componentes $\alpha\beta$, por lo que para ello se realiza la normalización de las componentes, obteniendo los resultados de la figura 4.10. Observamos que la amplitud de la señal se mantiene en **1V** con un **voltaje pico-pico de 2V**.

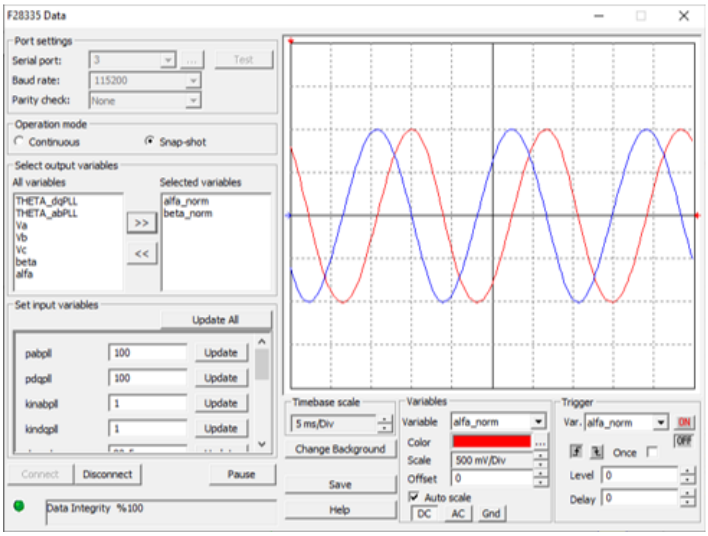


Figura 4.10: Normalización de las componentes $\alpha\beta$ de la transformada de Clarke

4.4. Resultados de los algoritmos propuestos

Con las componentes normalizadas de forma correcta, se realiza el análisis de cada algoritmo, los resultados que se presentan a continuación, se detallan en 3 secciones, en primera instancia los resultados con distorsión en frecuencia y distorsión armónica se presentan de forma simulada y los resultados prácticos, son tomados en la desconexión y conexión de la RES.

4.4.1. Distorsión en frecuencia

Para estas pruebas, se utilizó el diagrama de la figura 3.25, para ello se tiene en cuenta que a **0.5s** se la simulación se realiza el cambio de frecuencia, por lo que los siguientes resultados presentan las respuestas dinámicas de cada algoritmo ante esta falla.

4.4.1.1. dq-PLL

En la figura 4.11, se tiene una buena respuesta dinámica a la perturbación de frecuencia, se observa en esta figura, que en 0.5 segundos existe un cambio de frecuencia en la señal de entrada, por lo que la forma de onda del ángulo de fase (diente de sierra), presenta una respuesta dinámica rápida, ya que no se presentan cambios bruscos en la forma de la rampa de la señal. Los parámetros utilizados para este esquema fueron **kp=3066** y un **Ti=0.00065**.

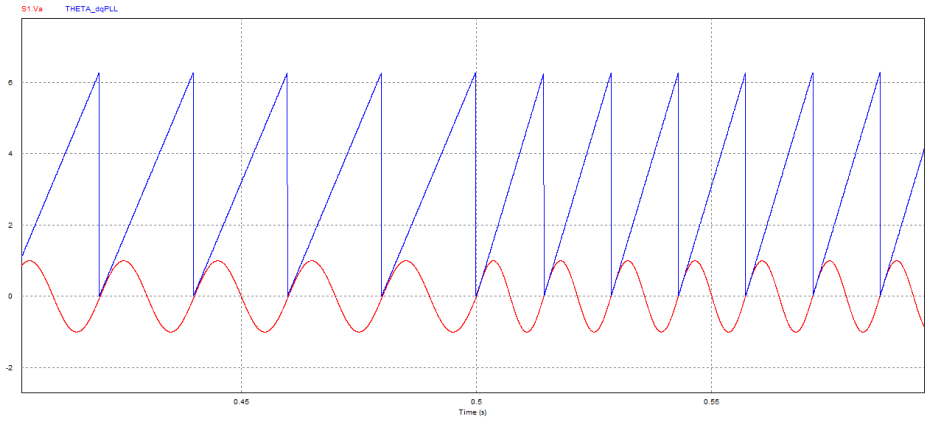


Figura 4.11: Respuesta del algoritmo dqPLL con distorsión de fase

4.4.1.2. $\alpha\beta$-PLL

Para esta respuesta, se ajustan los parámetros del ancho de banda del bucle de control, obteniendo como parámetros **kp=100 y Ti=0.02**, estos parámetros se ajustan de forma brusca ya que el algoritmo es capaz de eliminar los errores del algoritmo debido a condiciones de desequilibrio.

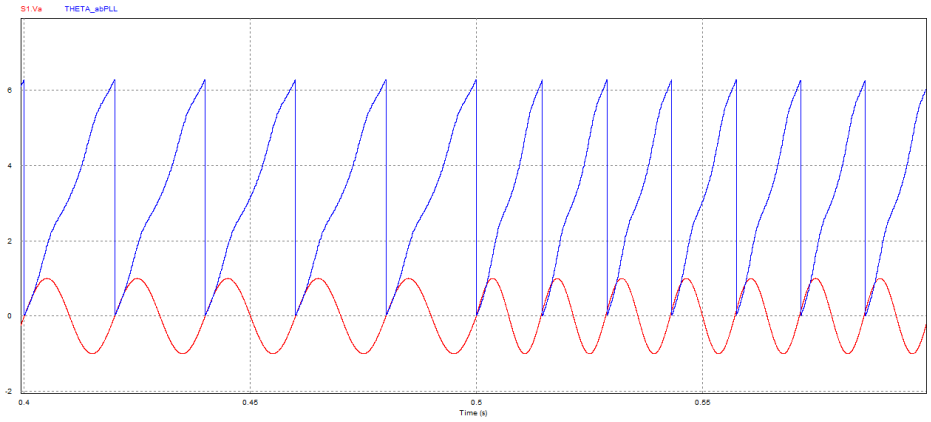


Figura 4.12: Respuesta del algoritmo $\alpha\beta$PLL con distorsión de fase

4.4.1.3. DDSRF-PLL

En respuesta al diseño presentado, se tiene los siguientes resultados de la simulación, con la distorsión de frecuencia. En la figura 4.13 se tiene el comportamiento del algoritmo, se observa que bajo las mismas condiciones para el lazo de control utilizadas para el algoritmo **dq-PLL**, la sincronización se mantiene, teniendo **kp=3066** y **Ti=0.00065**. Además su correcto funcionamiento se debe a su red de desacople.

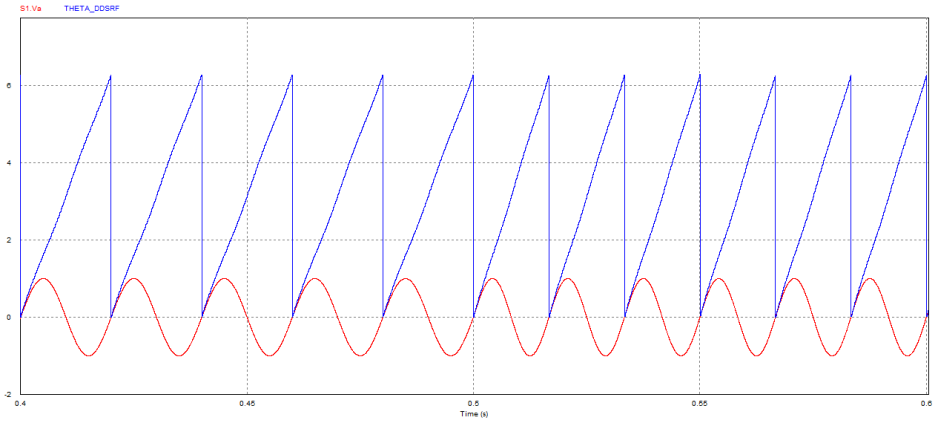


Figura 4.13: Respuesta del algoritmo DDSRFPLL con distorsión de fase

4.4.1.4. DSOGI-PLL

En respuesta a la simulación con los parámetros establecidos, se tiene los resultados de la figura 4.14, al igual que las respuestas, se realiza un cambio de frecuencia en 0.5s por lo que se obtiene el sincronismo correcto. Debido a la capacidad computacional requerida por el algoritmo en cuestión, los parámetros se ajustaron de forma brusca, considerando **kp=15** y **Ti=0.0015**.

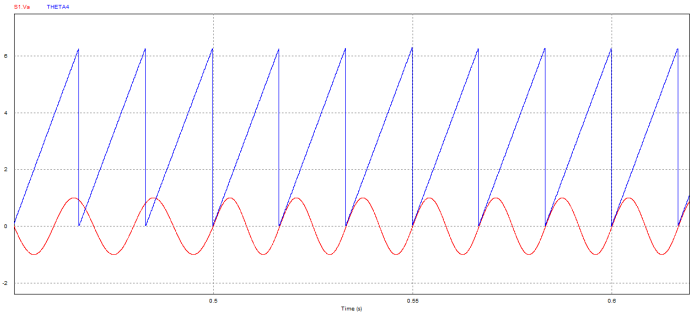


Figura 4.14: Respuesta del algoritmo DSOGI-PLL con distorsión de fase

4.4.2. Distorsión armónica

Para estas pruebas, se utilizó los diagramas de las figuras 3.27 y **??**, para los siguientes resultados, se consideran los mismo parámetros se sintonización explicados en el apartado anterior, ya que en la práctica los mismo parámetros deben ser capaces de responder de forma correcta a diferentes perturbaciones.

4.4.2.1. dq-PLL

El resultado del algoritmo se presenta en la figura 4.15, en esta figura se puede notar que se tiene armónicos en la señal de entrada, por lo que la respuesta del algoritmo es correcta, se puede observar que no existe perturbaciones en la señal del ángulo de fase. Este algoritmo tiene una respuesta estable ante la presencia de armónicos.

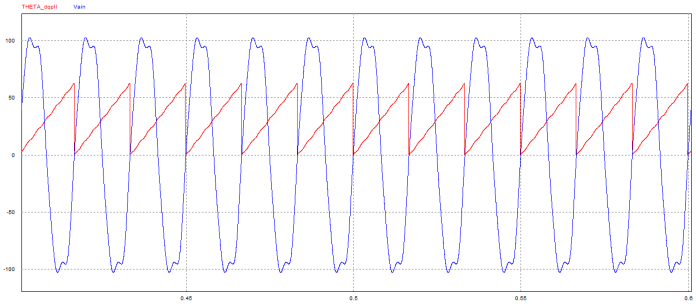


Figura 4.15: Respuesta del algoritmo dq-PLL con presencia de armónicos

4.4.2.2. $\alpha\beta$-PLL

En la figura 4.16 se presenta la respuesta del algoritmo ante la presencia de armónicos en la señal de entrada, en este caso se puede observar que la respuesta de este algoritmo con presencia de armónicos es mejor que la respuesta con distorsión de frecuencia. Recordando que este algoritmo trabaja mejor cuando desbalanceo de cargas.

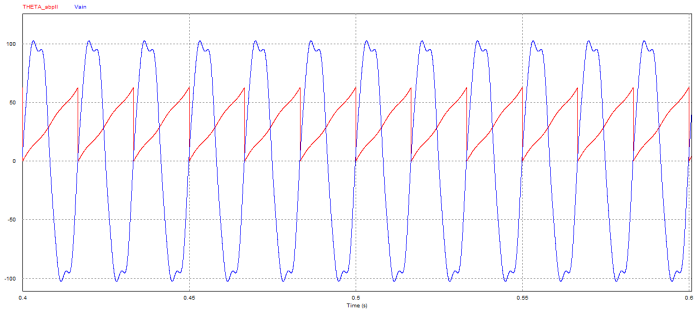


Figura 4.16: Respuesta del algoritmo $\alpha\beta$-PLL con presencia de armónicos

4.4.2.3. DDSRF-PLL

En la figura 4.17, se presenta la respuesta del algoritmo a la señal ruidosa de entrada, se observa que el sistema se mantiene estable en toda la ejecución de la simulación, haciendo que el algoritmo sea robusto cuando se inyectan estas perturbaciones a la red.

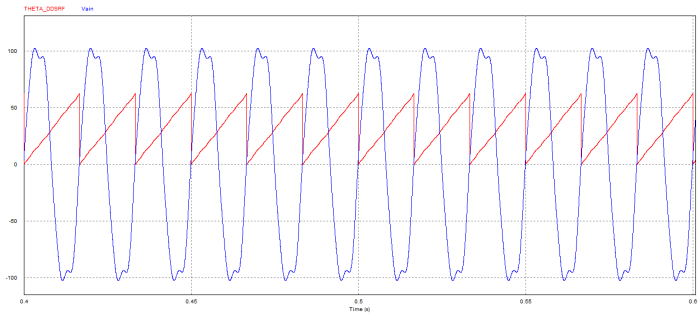


Figura 4.17: Respuesta del algoritmo DDSRF-PLL con presencia de armónicos

4.4.2.4. DSOGI-PLL

La respuesta del algoritmo en cuestión, se presenta en la figura 4.18, se observa que la respuesta del mismo, se mantiene estable, considerando la presencia de los armónicos, teniendo en cuenta que este algoritmo presenta una respuesta dinámica lenta.

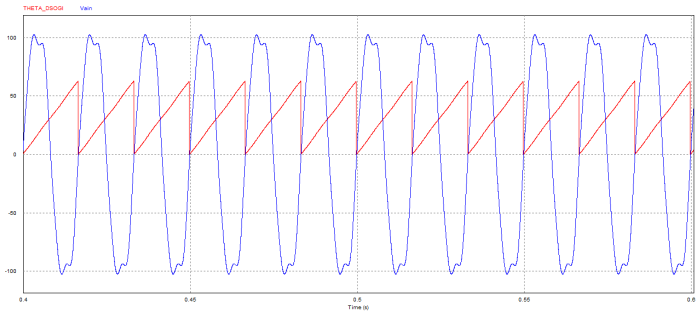


Figura 4.18: Respuesta del algoritmo DSOGI-PLL con presencia de armónicos

4.4.3. Desconexión y Conexión de la fuente renovable

En este apartado se tiene los resultados de las pruebas prácticas realizadas, cabe señalar que para estos resultados de la misma forma, se utilizaron los parámetros **kp** y **Ti** de las sub secciones anteriores.

4.4.3.1. dq-PLL

En la figura 4.19 se presenta la conexión para visualizar la respuesta del primer algoritmo, cabe señalar que se tomo la **fase 1** para la comparativa de la respuesta, también se utilizó el puerto **GPIO 0** del PWM y el filtro paso bajo, para visualizar la onda resultante.

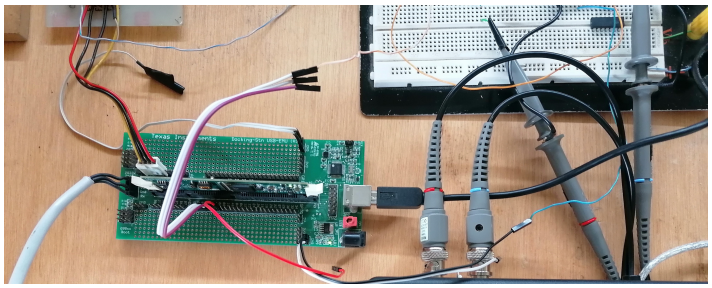


Figura 4.19: Esquema de conexión para la captura de datos del algoritmo dq-pll

En las figuras 4.20 se presenta el resultado del algoritmo **dq-PLL**, como se puede observar en la gráfica, el algoritmo responde correctamente a la sincronización con la señal de entrada. La señal diente de sierra (señal azul), presenta el ángulo de fase resultante, además, se tienen pequeñas perturbaciones en la señal de entrada (señal amarilla), por lo que el algoritmo funciona de forma correcta, realizando un control de lazo adecuado para este caso.

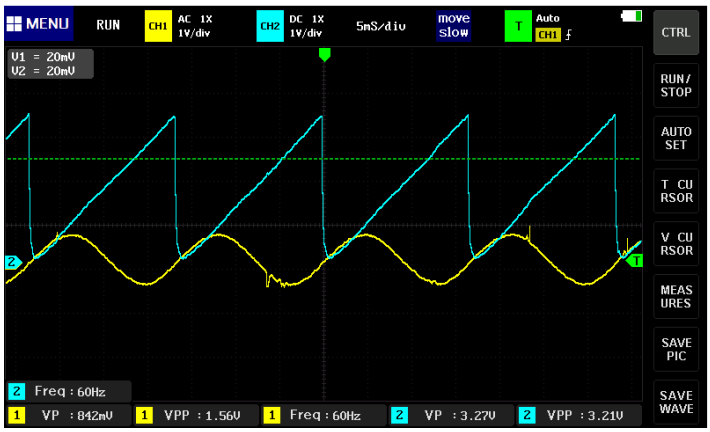


Figura 4.20: Respuesta del algoritmo dq-PLL con los paneles solares

4.4.3.2. $\alpha\beta$-PLL

En la figura 4.21 se puede observar las pruebas realizadas para el algoritmo en cuestión, la conexión se realizó desde el puerto **GPIO 2** del DSP, a la entrada del osciloscopio. En este caso se observa que la rampa no tiene perturbaciones como en el caso de la figura 4.12.

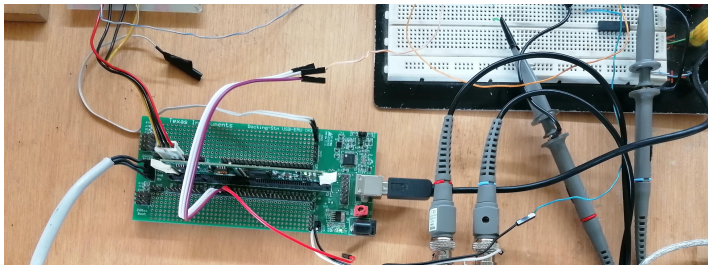


Figura 4.21: Esquema de conexión para la captura de datos del algoritmo $\alpha\beta$-PLL

Hay que tomar en cuenta que para estos resultados la RES este conectada a las barras del tablero principal, por lo que esta entregando tensión al laboratorio. Con esto se observa que el algoritmo responde correctamente, teniendo en cuenta que esta trabajando con las mismas condiciones de operación que el algoritmo **dq-PLL**. En la figura 4.22 se puede observar, que en la señal de entrada existen pequeñas perturbaciones, que no afectan el funcionamiento del algoritmo en ejecución.

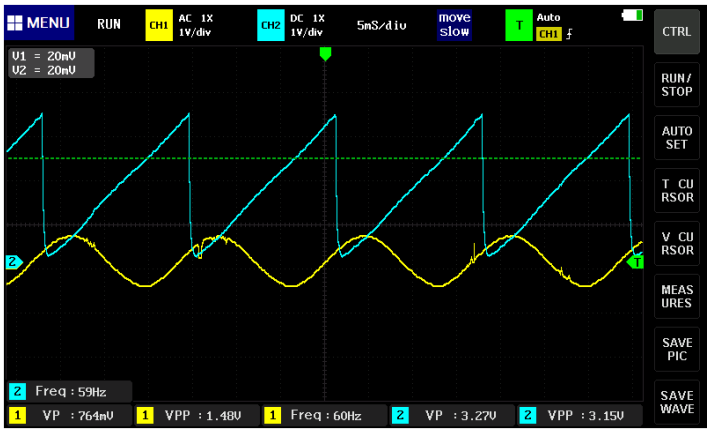


Figura 4.22: Respuesta del algoritmo $\alpha\beta$-PLL con los paneles solares

4.4.3.3. DDSRF-PLL

En la figura 4.23 se presenta la conexión realizada para la captura de datos, utilizando el puerto **GPIO 4** del PWM. También se puede ver pequeñas perturbaciones en la señal, lo que permite observar que el algoritmo esta trabajando de forma correcta.

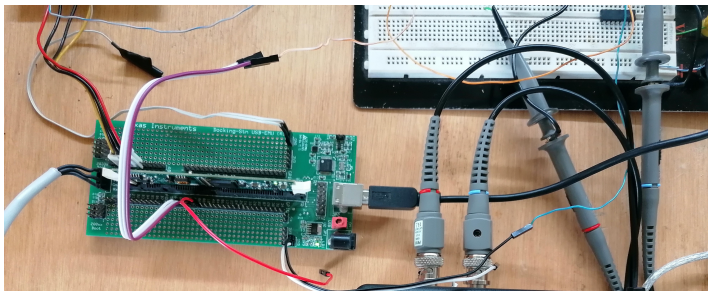


Figura 4.23: Esquema de conexión para la captura de datos del algoritmo DDSRF-PLL

Con esto, los resultados obtenidos, presenta una excelente sincronización, considerando los mismo

criterios de operación de los algoritmos anteriores, el algoritmo responde de forma rápida a la distorsión de voltaje. Esta respuesta se presenta en la figura 4.24.

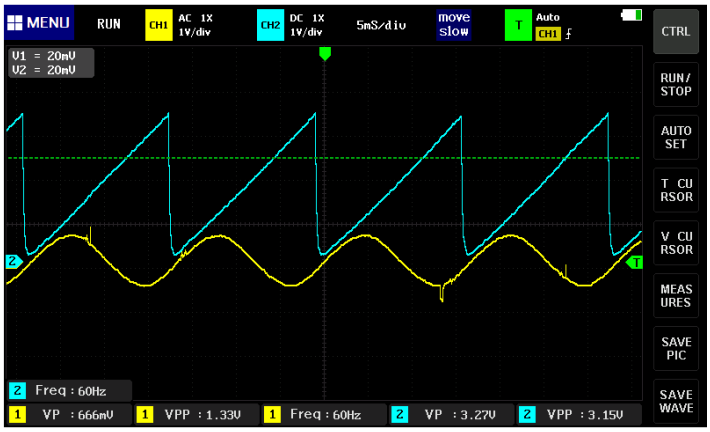


Figura 4.24: Respuesta del algoritmo DDSRF-PLL con los paneles solares

4.4.3.4. DSOGI-PLL

En la figura 4.25 se presenta la conexión para la captura de datos con el osciloscopio digital utilizando el puerto **GPIO 6** del **PWM**

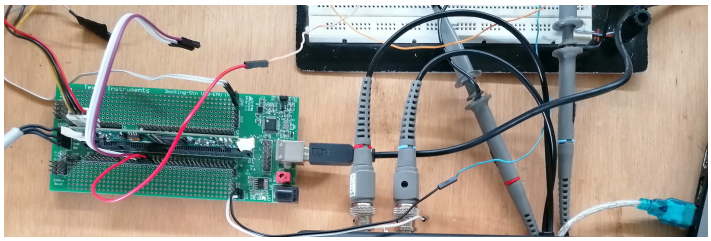


Figura 4.25: Esquema de conexión para la captura de datos del algoritmo DSOGI-PLL

En estas pruebas, se puede observar que el algoritmo responde de forma correcta, a la distorsión que presenta la señal de entrada. Pero en respuesta al tiempo de establecimiento, el algoritmo resulta tener una respuesta más lenta, en comparación con los algoritmos anteriores. En la figura 4.26, se observa el comportamiento del algoritmo, con ello aunque el tiempo de establecimiento es lento, cuando existen perturbaciones en la señal, el sistema de mantiene estable.

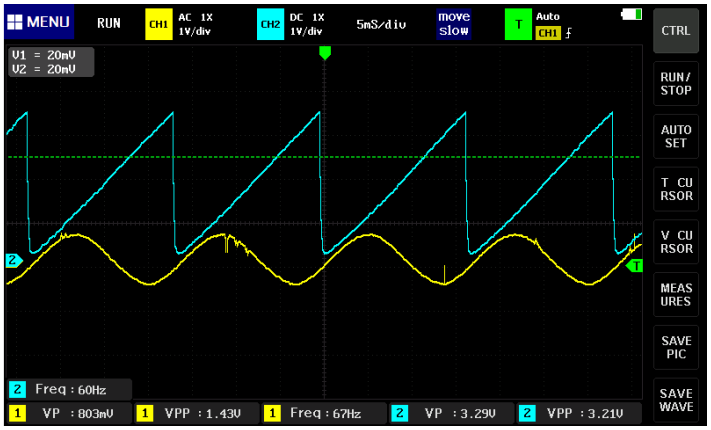


Figura 4.26: Respuesta del algoritmo DSOGI-PLL con los paneles solares

4.5. Análisis y discusión de los resultados

Con los resultados de las simulaciones se pudo constatar que los algoritmos **dq-PLL** y DDSRF-PLL presentan mayor rendimiento en el ajuste de parámetros del **PI**, debido a que ambos trabajan con la transformada de Park, sus resultados se asemejan entre sí. Los parámetros utilizados para estos algoritmos permiten un tiempo de establecimiento de **3ms**.

A su vez, dado que DSOGI-PLL trabaja con el mismo sistema de coordenadas (**dq**), se utilizaron los mismo parámetros, dando como resultado ineficiencia para hacer el control del ángulo de fase, esto, debido a que la capacidad computacional no es apta para su funcionamiento, es por ello que se ajustan los parámetros y se observa que su dinámica presenta resultados favorables cuando existe una distorsión de frecuencia en la señal.

En cuanto a $\alpha\beta$**-PLL** se necesita ajustar los parámetros del control de lazo para obtener una respuesta más precisa, esto se logra visualizar cuando la pendiente de la señal diente de sierra no tiene muchas perturbaciones, aun así se puede observar que la onda de salida presenta distorsiones en la pendiente, esto se debe a la presencia de armónicos de alto orden.

Con respecto a las simulaciones con distorsiones de frecuencia y armónicas, se pudo observar que los algoritmos **dq-PLL** y DDSRF-PLL, tuvieron un mejor rendimiento, por la forma de onda de la señal de salida. A pesar de lo que algoritmos DSOGI-PLL y $\alpha\beta$**-PLL**, presentaron buenas respuestas, el ajuste de parámetros fue un factor importante a considerar.

De forma práctica, se realiza el proceso de desconexión y conexión del inversor de la RES, ya que al instante de su puesta en marcha, este no presenta la frecuencia exacta, por lo que se requiere un cierto tiempo para que el inversor logre estabilizar las señales. Con este procedimiento, se pudo observar que los algoritmos **DDSRF-PLL, DSOGI-PLL** responden de forma más rápida cuando hay distorsión en la señal de entrada comparados con el **dq-PLL,** $\alpha\beta$**-PLL**.

El algoritmo **dq-PLL** es más lento en su tiempo de respuesta, en comparativa, el algoritmo no trabaja bien cuando se presentas distorsiones, a su vez el algoritmo $\alpha\beta$**-PLL** es una mejora, pero cuando existe una distorsión más alta, su tiempo de establecimiento aumenta e incluso el control de bucle

necesariamente necesita ser ajustado variando el ancho de banda, lo que no permite considerarlo como un algoritmo eficiente. Estos algoritmos presentaron **tiempos de establecimientos** aproximados **de 4 a 5ms.**

Desde otra perspectiva, realizando el apagado del DSP se observó que el tiempo de establecimiento fue menor cuando se trata de los algoritmos DDSRF-PLL y DSOGI-PLL, dando robustez a la señal de control en términos de tiempo de establecimiento. En comparativa entre estos dos algoritmos **DDSRF-PLL** tuvo un tiempo de establecimiento aproximado correspondiente a 3ms, por otro lado, **DSOGI-PLL** su tiempo de establecimiento se aproxima a un valor entre 3ms a 4ms.

Hay que tener en cuenta que debido a las distorsiones presentadas, los tiempos de establecimiento son bajos (milisegundos), pero esto permite que el sistema sea robusto, ya que si el algoritmo reacciona de forma inmediata a distorsiones permitirá la protección del sistema eléctrico y equipos en general.

En resumen, en la tabla 4.1, se presentan los parámetros utilizados para cada algoritmo y su tiempo de establecimiento, cabe señalar que debido a que son tiempos muy bajos, estos valores son aproximados.

Tabla 4.1: Parámetros y resultados de los algoritmos analizados

Algoritmo	**Ts**	**Kp**	**Ti**
dq-PLL	3-4ms	3066	0.00065
$\alpha\beta$-PLL	4-5ms	100	0.02
DDSRF-PLL	3ms	3066	0.00065
DSOGI-PLL	5ms	15	0.0015

4.6. Conclusiones y Recomendaciones

4.6.1. Conclusiones

Con este trabajo de titulación se evidenció que el funcionamiento de cada algoritmo está relacionado a la calidad de la red, puesto que, al someter los cuatro algoritmos a las mismas condiciones se observó que para cada uno de ellos los parámetros de control deben ser ajustados para lograr su correcto funcionamiento.

Al ajustar los parámetros del controlador se logró obtener una respuesta rápida y eficiente de cada uno, esto se pudo evidenciar en el tiempo de establecimiento y las formas de onda (diente de sierra) presentadas en los resultados. De esta forma los algoritmos implementados proporcionan la certeza de una red segura ante posibles daños en la interconexión con fuentes externas. El uso de un DSP abre una línea de investigación a la calidad energética de nuestra región.

En la implementación de los algoritmos DSOGI-PLL y DDSRF-PLL, la capacidad computacional fue un aspecto determinante, pero en DSOGI-PLL los parámetros se tuvieron que ajustar de manera brusca, este factor es relevante para la implementación en el DSP a utilizar. En nuestro caso, el DSP utilizado tiene una frecuencia de trabajo máxima de 150MHz, por lo que el funcionamiento de los

algoritmos no tuvo efecto negativo en la parte práctica.

Otro aspecto relevante es la configuración del simulador, como se mencionó, la carga computacional de ciertos algoritmos no permite variar de manera extensa los parámetros en el simulador, por lo que los ajustes que desembocaron en una mejor respuesta fueron: paso de tiempo = 10us, tiempo total = 1s, paso de impresión = 1.

Con estos antecedentes y con las pruebas realizadas se puede considerar que los algoritmos DDSRF-PLL y DSOGI-PLL prueban ser eficientes cuando se tiene distorsiones considerables, puesto que, como se evidencia en el diseño del controlador, su respuesta al escalon unitario se mantiene en el orden de los mili-segundos (ms).

Cabe mencionar, que con DDSRF-PLL no se varían los parámetros bruscamente como se realiza con DSOGI-PLL. Hay que tener en cuenta que el tiempo de establecimiento es crucial al momento de inyectar energía a la red, pues si los algoritmos no actúan de forma inmediata, las protecciones pueden actuar cortando el suministro de energía eléctrica, produciendo fallos en los equipos en general.

Los parámetros del controlador establecidos para cada algoritmo se mantuvieron para cada caso de prueba. Se observó que los parámetros fijados permiten que cada algoritmo responda de forma correcta ante las perturbaciones a las que fueron sometidos.

En la simulación se observa la presencia de perturbaciones en la rampa de la onda diente de sierra, esto a causa de la presencia de armónicos de alto orden en la señal. Este comportamiento se puede evitar ajustando el ancho de banda del bucle de control.

Por otro lado, los resultados prácticos no presentaron dichas perturbaciones, constatando que no existen armónicos de alto orden en la señal de entrada. Con estos resultados, se evidencia que al usar los paneles solares poli-cristalinos del laboratorio de micro-red, el algoritmo DDSRF-PLL resulta ser el más adecuado para lograr inyectar energía a la red pública sin causar daños en los equipos.

Con este trabajo también se constató que se puede diseñar un sistema de sincronización de fase con un presupuesto bajo, dado que, el kit de la placa DSP ronda los 150 dólares, las herramientas computacionales utilizadas son de libre distribución (PSIM y CCS), y la arquitectura de acondicionamiento de señales tiene un valor aproximado de 40 dolares.

4.6.2. Recomendaciones

- Se debe corroborar que las señales estén desfasadas 120° entre si.
- El diseño del circuito de acoplamiento debe ir acorde a los parámetros del DSP, ya que si no se cumple con las características técnicas del hardware se puede dañar los puertos o dar datos incoherentes.
- Debido a la capacidad computacional se debe establecer los parámetros ideales, considerando además que la frecuencia máxima de trabajo del DSP es de 150MHz.
- Para realizar las pruebas es necesario parar la ejecución del DSP desde la compilación del CCS, ya que al desconectar de forma directa el hardware, se puede dañar la memoria.

- El uso de resistencias de valor muy grande, para reducir la tensión del sensado, produce que la corriente sea muy baja, por lo que el sistema sería susceptible al ruido conducido del medio en donde se esta trabajando.

4.7. Trabajos futuros

- Probar la eficiencia de las variantes de los códigos presentados en una RES
- Probar la eficiencia de los algoritmos presentados con fuentes programables AC, sometiendo su funcionamiento a perturbaciones más exageradas.
- Diseñar e implementar un banco de pruebas de algoritmos de sincronización de fase, en conjunto con un sistema SCADA que permita recolectar datos de forma continua de las RES

Anexo A

Elaboración del proyecto, diseño de componentes para fabricación de la PCB y generación del archivo .gerber en Altium

En este anexo se presenta la creación de la PCB, se indica paso a paso el desarrollo de los esquemas y la elaboración del footprint que permite obtener la impresión para elaborar la placa, en A.1 se presenta los pasos para crear un proyecto, además se indica la elaboración de las librerías que vamos a utilizar en el diseño, posteriormente se presenta el diseño de los componentes que no posee Altium en sus librerías, para ello, inicialmente en A.2 se presenta el diseño esquemático del componente, luego en A.3 se presenta el diseño del componente para la PCB. Finalmente en A.4 se presenta la generación del archivo **.gerber** para la elaboración final de la placa.

A.1. Elaboración del proyecto en Altium

Para la elaboración de la PCB se utiliza **Altium Designer 18.0** ya que es un plataforma que provee varias funcionalidades para la creación de la misma.

Para crear un proyecto en Altium Designer nos vamos a la barra de herramientas File-New-Project-Project como se presenta en la figura A.1.

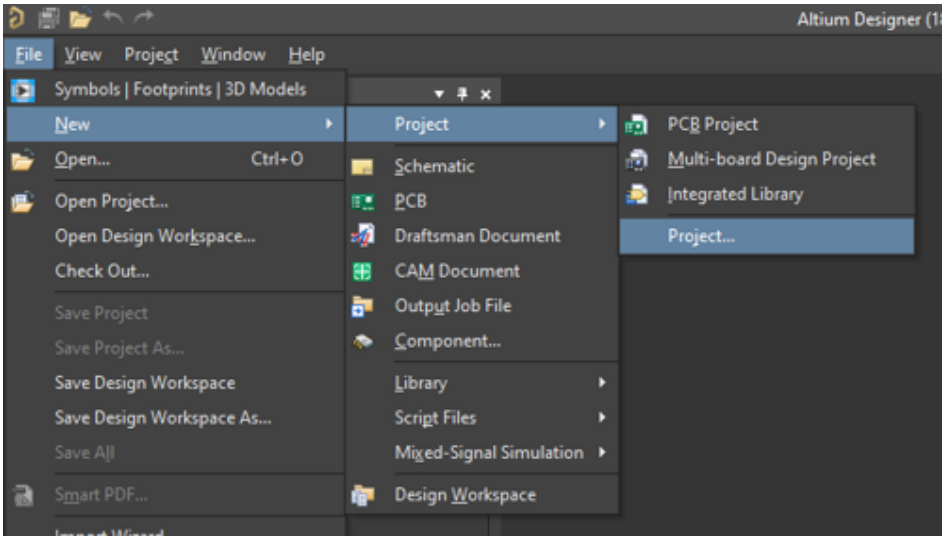


Figura A.1: Dirección para la creación de un nuevo proyecto

Una vez seleccionado el nuevo proyecto, nos aparece una ventana como se presenta en la figura A.2, esto nos permite cambiar el nombre y la dirección, además que nos da la posibilidad de usar plantillas establecidas para un diseño más rápido, en nuestro caso escogeremos la plantilla por defecto.

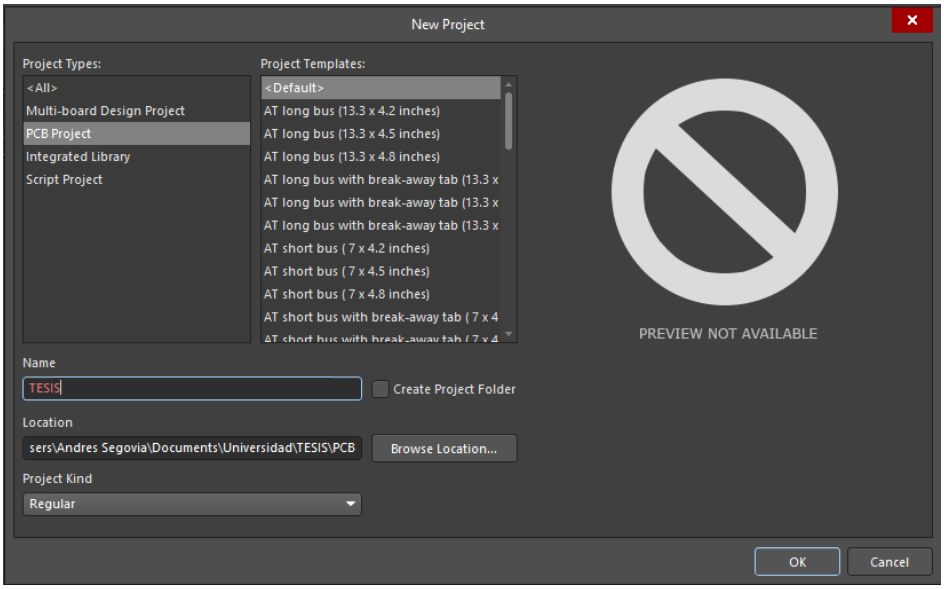


Figura A.2: Configuración de la dirección y nombre del proyecto

Una vez creado el proyecto, vamos a añadir los archivos que permiten el diseño, inicialmente añadimos el archivo esquemático en donde se alojara el diagrama del circuito, esto se presenta en la figura A.3, además se añade el archivo PCB en donde se presenta el diseño final de la placa, como se ve en la figura A.4.

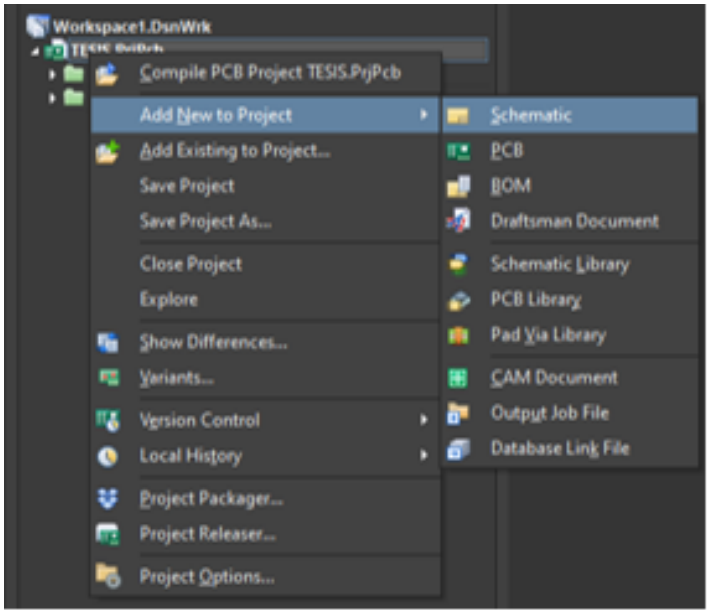


Figura A.3: Añadir archivo esquemático en el proyecto

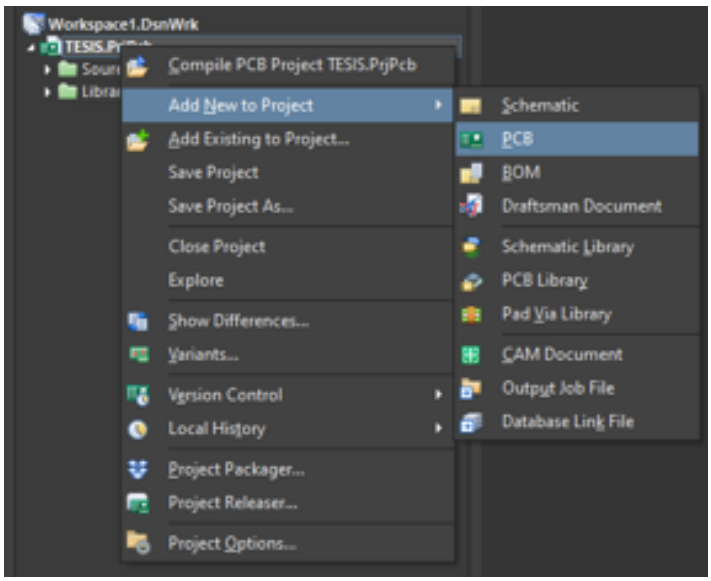


Figura A.4: Añadir archivo PCB en el proyecto

Una vez añadido los archivos principales, es necesario crear nuevas librerías ya que hay componentes que no se tienen diseñados en Altium, entonces es necesario crearlos manualmente. Para ello añadimos los archivos esquemáticos y PCB para las librerías como se presenta en las figuras A.5 y A.6.

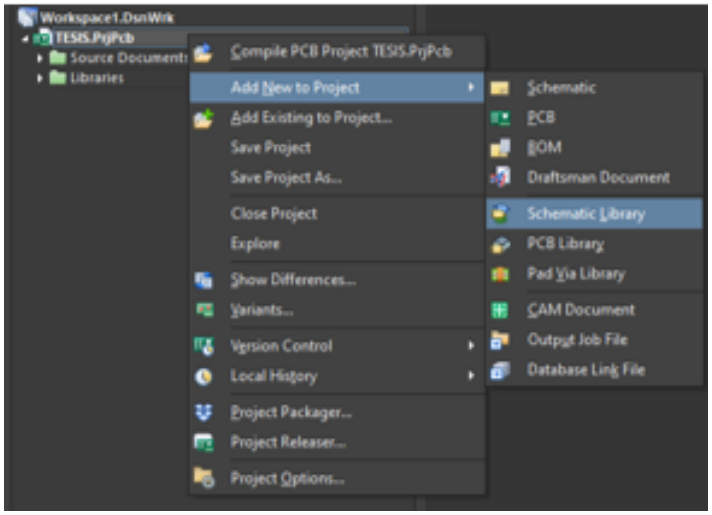


Figura A.5: Añadir archivo esquemático para librería en el proyecto

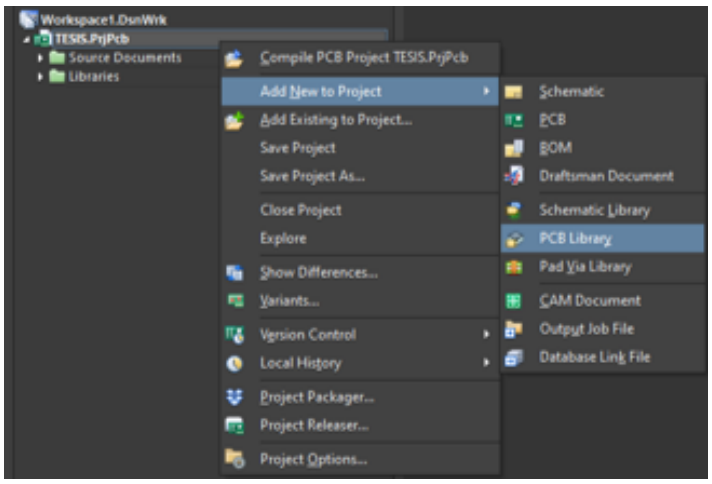


Figura A.6: Añadir archivo PCB para librería en el proyecto

En los archivos para librerías vamos a crear los elementos faltantes, para ello es necesario tener las hojas de datos de cada componente, ya que se necesita las dimensiones entre pines y las dimensiones del componente. Finalmente con los archivos creados nos vamos a **File-Save All** y nos permite cambiar los nombres de los archivos como se ve en la figura A.7.

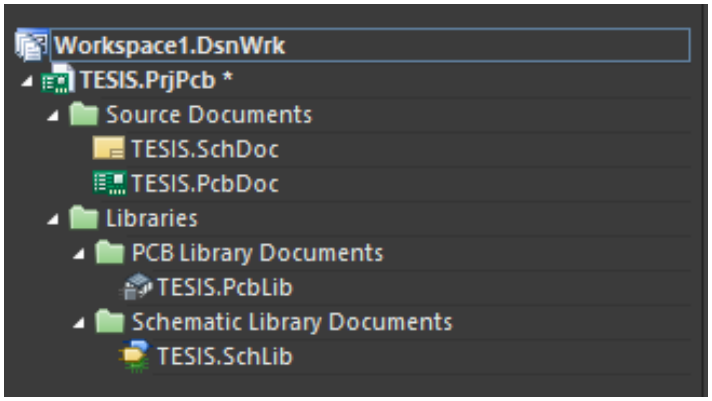


Figura A.7: Archivos añadidos al proyecto

A.2. Diseño esquemático del elemento en Atium

Para empezar con el diseño, hemos seleccionado el puente rectificador, inicialmente entramos el archivo esquemático para librería con doble clik sobre el mismo. Esto habilitará las opciones en la parte inferior como se presenta en la figura A.8 en la pestaña **SCH Library**.

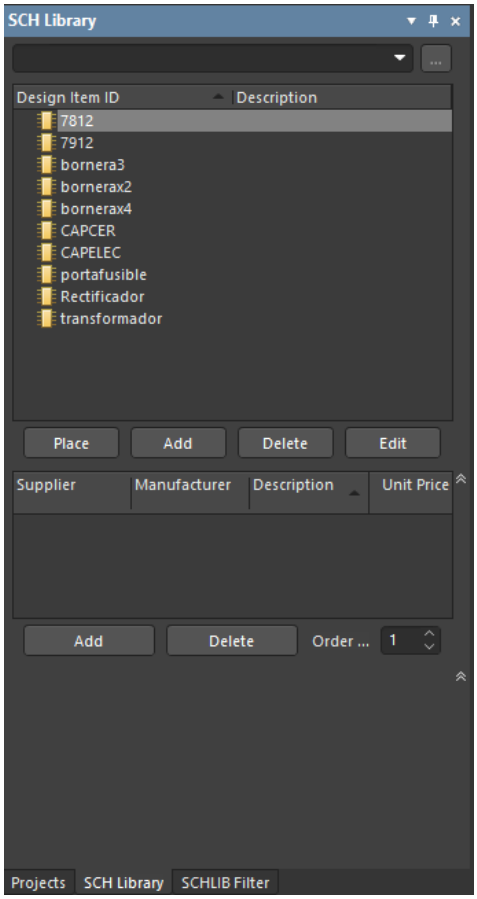


Figura A.8: Propiedades para los archivos esquemáticos de las librerías

Esto permite añadir nuevos esquemáticos en el caso que se requiera crear varios componentes, entonces se selecciona la pestaña **Add** y nos aparece una ventana, como se ve en la figura A.9, en donde se puede configurar el nombre del mismo.

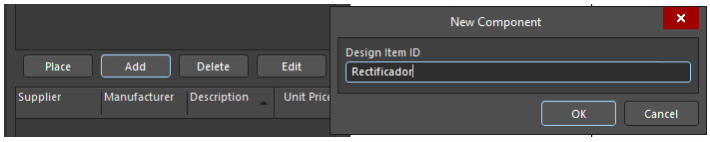


Figura A.9: Añadir y configurar nombre del componente

En este esquemático vamos a diseñar el componente, primero vamos a la barra de herramientas, como se ve en la figura A.10, para añadir el contorno del componente, seleccionamos la forma de la superficie y en A.11 se observa que podemos configuramos el tamaño del mismo, para acceder a las propiedades podemos dar doble click sobre el componente, cabe recalcar que el tamaño en el esquemático no influye en el tamaño del PCB.

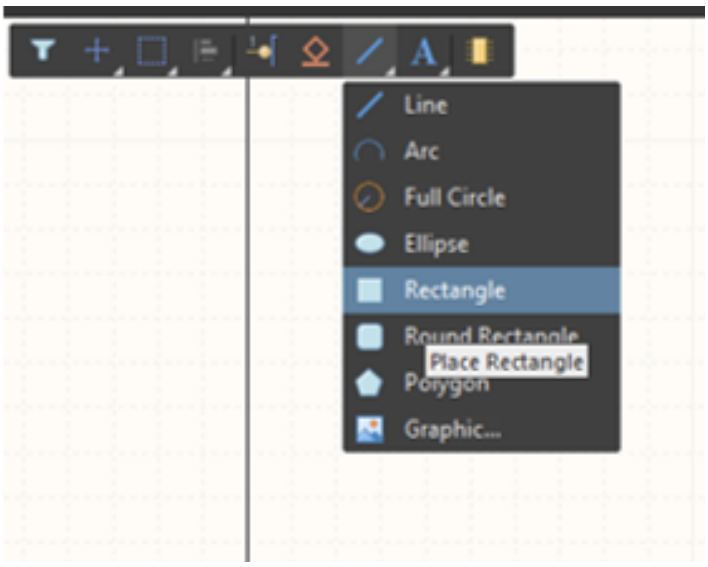


Figura A.10: Crear superficie del componente

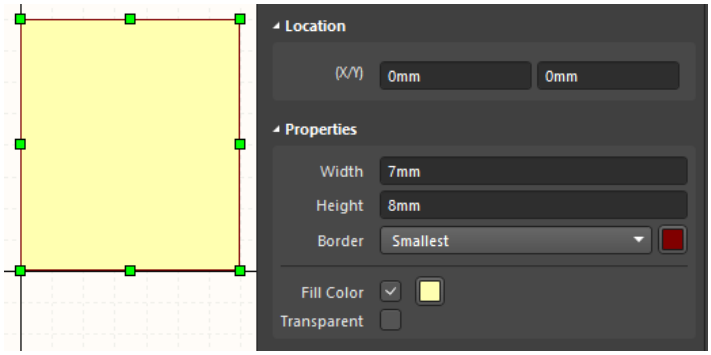


Figura A.11: Propiedades de diseño del componente

Ahora vamos a añadir los pines, como se presenta en la figura A.12, hay te tener en cuenta que para

rotar algún componente se puede utilizar la tecla **espacio**, en las propiedades de los pines podemos configurar la ubicación, el tamaño, el nombre, la descripción, el tipo de pin, ya que este puede ser de entrada, I/O, salida, etc.

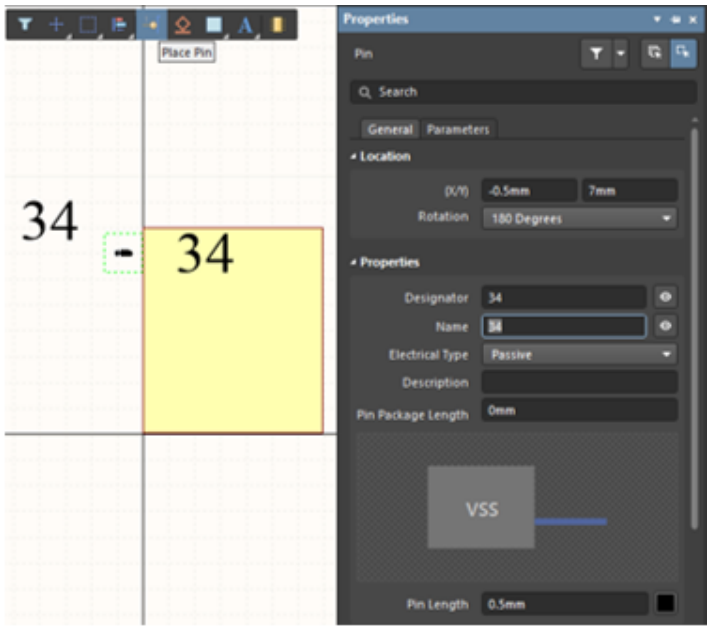


Figura A.12: Añadir pines en el componente

Finalmente, se tiene que colocar la descripción y el designador del componente, para ello accedemos a las propiedades del mismo, en la figura A.13 se presenta las configuraciones y en A.14 se presenta el componente final.

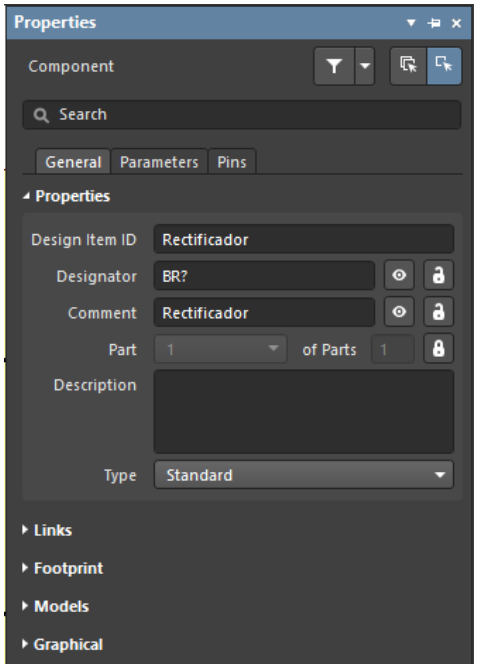


Figura A.13: Añadir pines en el componente

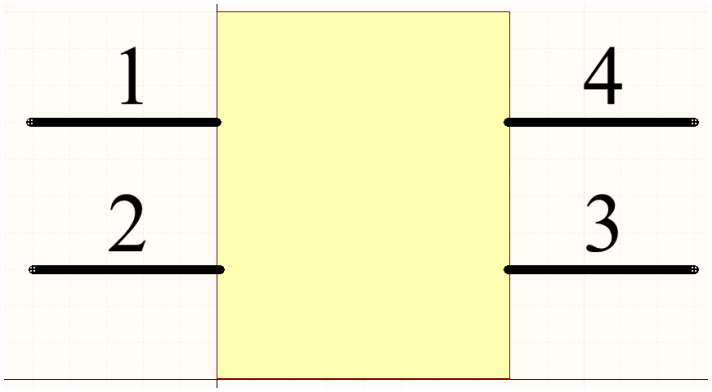


Figura A.14: Diseño final del componente

A.3. Diseño del footprint del componente en Altium

Una vez creado el diseño esquemático, vamos a diseñar el footprint del elemento para el PCB por lo que vamos al archivo librerías PCB que creamos inicialmente en el proyecto, y accedemos a la hoja de diseño como se ve en la figura A.15.

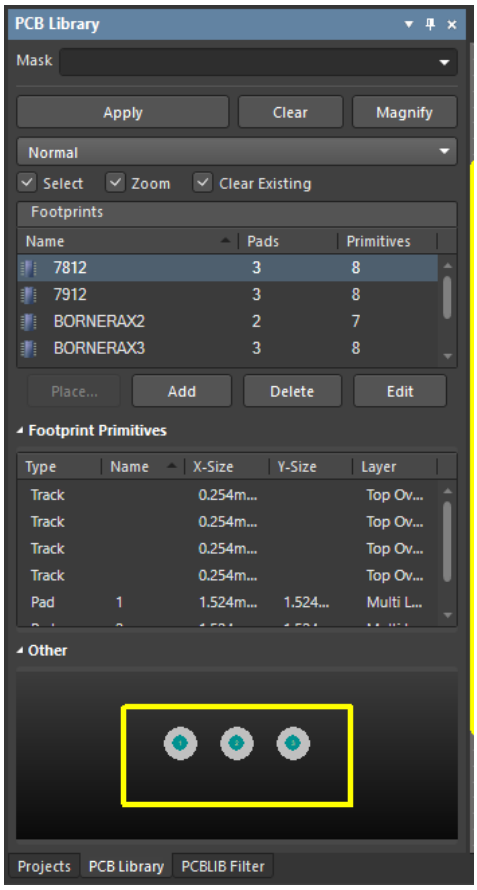


Figura A.15: Propiedades para los archivos PCB de las librerias

Ahora vamos a añadir un nuevo componente, para ello vamos a la pestaña **Add**, como se ve en la

figura A.16, además se puede configurar el nombre y el tipo.

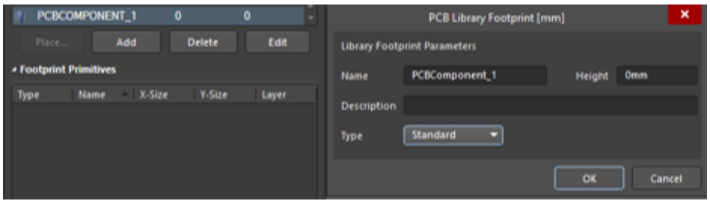


Figura A.16: Añadir componente para diseño PCB

Ya configurado el lugar de trabajo, vamos a añadir los pines del componente, en este caso añadimos 4 pines, en la figura A.17, se observa en la barra de herramientas el icono para añadir.

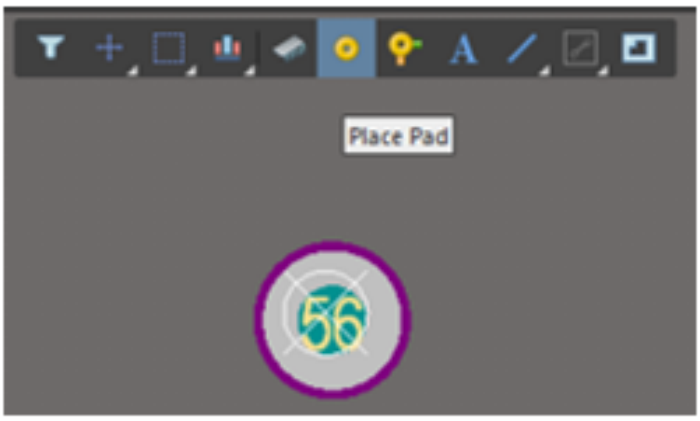


Figura A.17: Añadir pin al componente

Además en la figura A.18 se observa las propiedades de mismo, en ellas podemos configurar el tamaño del pin, esto permite saber las dimensiones de las perforaciones que tendrán en la PCB, también la descripción del mismo y la ubicación. En este punto es importante considerar las dimensiones de las hojas de datos como se mencionó al inicio, ya que la ubicación de cada pin en el footprint es el que se asignara en el diseño final de la PCB.

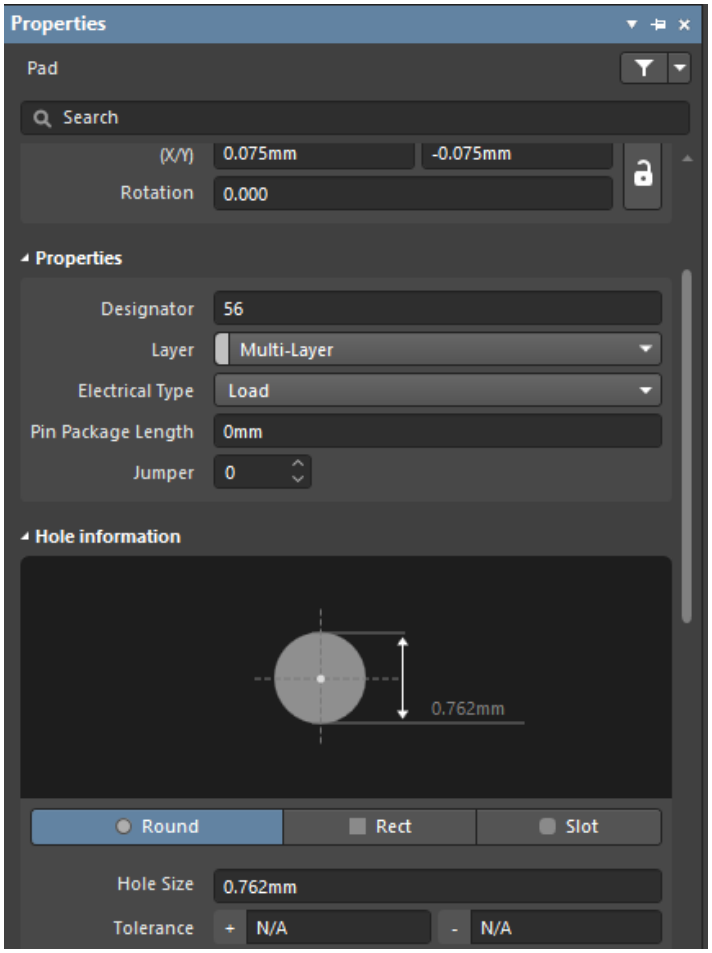


Figura A.18: Propiedades de los pines para el componente

Ahora en la parte inferior, vamos a la capa **Top Overlay**, en la barra de herramientas seleccionamos **line** y realizamos el contorno del elemento, esto permite tener una apreciación de las distancias de los componentes en la PCB. En la figura A.19 se observa el contorno del punte rectificador.

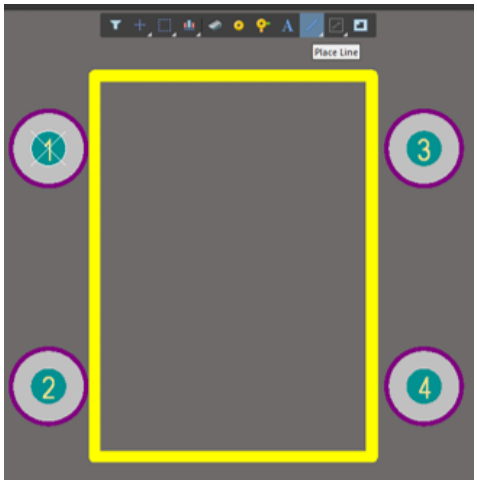
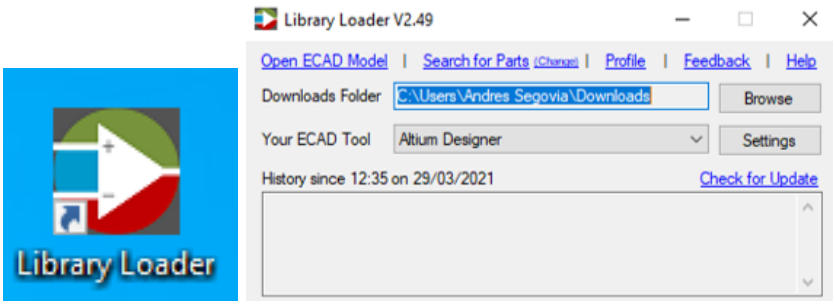


Figura A.19: Diseño del contorno del componente para la PCB

Pues bien ya se tiene el elemento diseñado, una de las características de Altium es que permite realizar un modelado 3D de la placa, para ello es necesario tener los modelos 3D de los elementos, si bien no se puede tener el mismo elemento se puede obtener uno de las dimensiones similares.
Una aplicación que permite descargar modelos 3D es **Library Loader** [62], en la figura A.20 se observa el icono y la ventana de navegación, en donde se puede configurar la dirección de descarga, se puede abrir modelos ya descargados, o en la pestaña **Search for parts** que nos permite redirigir a la página de **Electronic Component SEARCH ENGINE**, en donde podemos buscar el componente a usar.



(a) Icono de escritorio (b) Ventana de navegación de la App

Figura A.20: Aplicación para obtener modelos en 3D

En la figura A.21 se puede observar el componente, y se procede a descargar el modelo 3D, cabe recalcar que se descarga con la extensión **.stp** que permite importar a Altium.

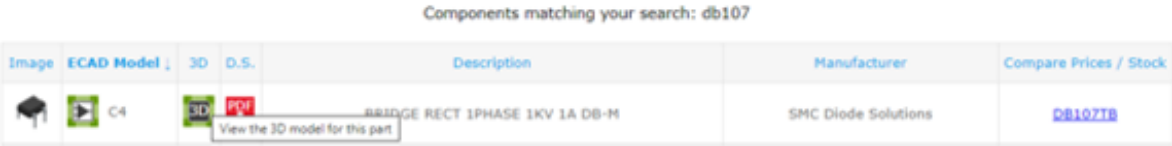


Figura A.21: Modelo 3D del puente rectificador

Una vez obtenido el modelo, en la barra de herramientas, se selecciona la herramienta **Place 3D body** (figura A.22) y con la tecla **Tab** nos permite acceder a las propiedades del modelo 3D.

Figura A.22: Agregar modelo 3D al footprint del componente

En las propiedades, en la pestaña **3D Model Type**, en la sección **Generic**, nos permite seleccionar el modelo 3D, esto se visualiza en la figura A.23.

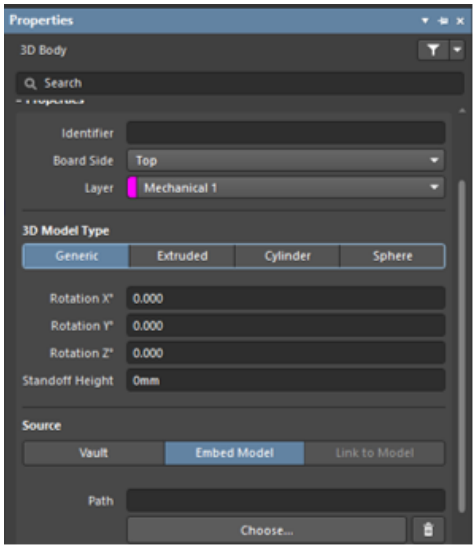


Figura A.23: Propiedades del modelo 3D

Hay que tener en consideración, que se esta trabajando en modelado 2D por lo que al añadir el modelo 3D, el componente se visualiza como en la figura A.24, para ver el diseño en modelo 3D, se puede presionar la tecla numero 3 y se obtiene la vista del la figura A.25

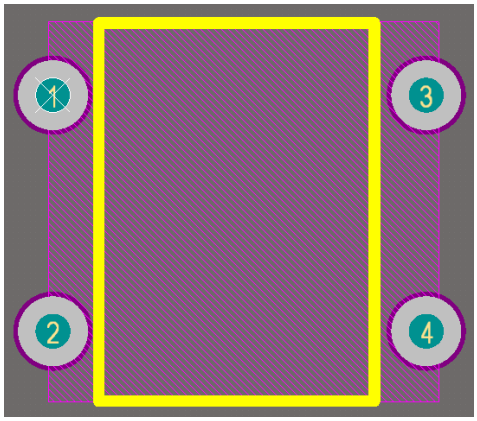


Figura A.24: Modelo 3D añadido al componente

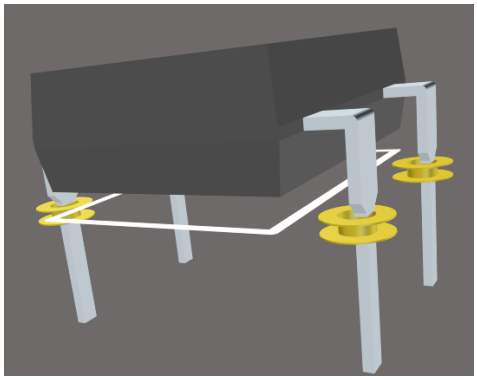


Figura A.25: Modelo 3D del componente

En el caso que el modelo en 3D no este alineado con la base del footprint, nos vamos a la barra de herramientas en la pestaña **Tools-3D Body Placement-Align Face With Board** (figura A.29) y nos pide escoger la cara del modelo que queremos alinear.

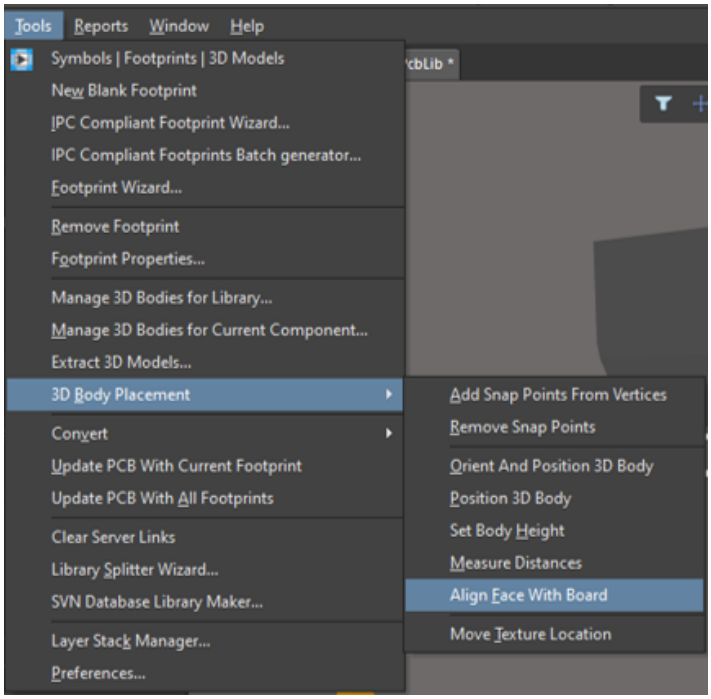


Figura A.26: Alinear modelo 3D con la placa

A.4. Elaboración del archivo .Gerber

En este apartado se desarrolla el proceso para la generación del archivo **.gerber**, este archivo contiene toda la información necesaria para la creación de la placa por lo que es de vital importancia su generación.

Inicialmente agregamos un archivo de salida, click derecho en el proyecto y se despliega los archivos para generar, en la figura A.27, se observa el archivo que se añade al proyecto.

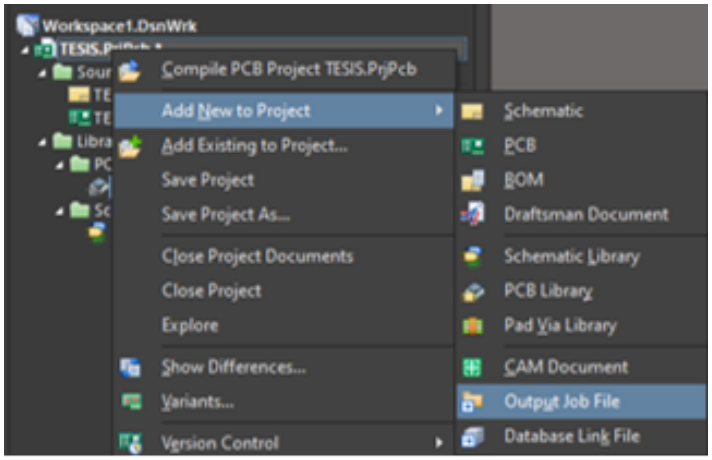


Figura A.27: Crear archivo de salida en el proyecto

Pues bien en la ventana que se despliega, nos vamos a la opción **Documentation Out**, y seleccionamos un archivo PCB de salida como en la figura A.28.

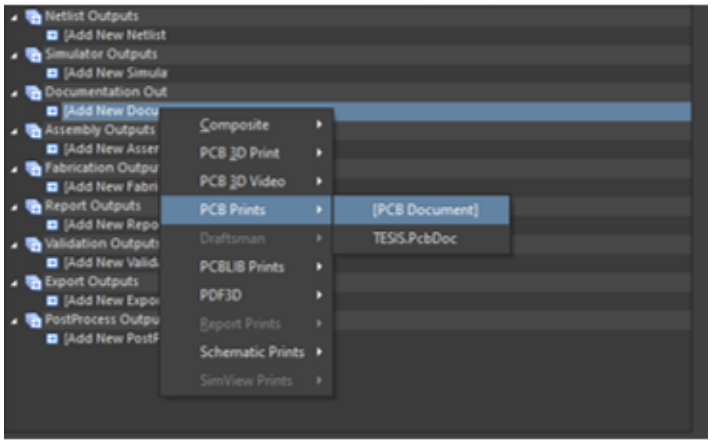


Figura A.28: Crear un archivo de salida PCB

Con esto, podemos configurar el archivo que ser va a crear, le damos click derecho en el archivo PCB generado y escogemos la opción **Page Setup**, con esto se despliega una ventana que permite configurar las dimensiones del papel de impresión, es necesario que la escala este en 1 y este marcada la opción **Mono**.

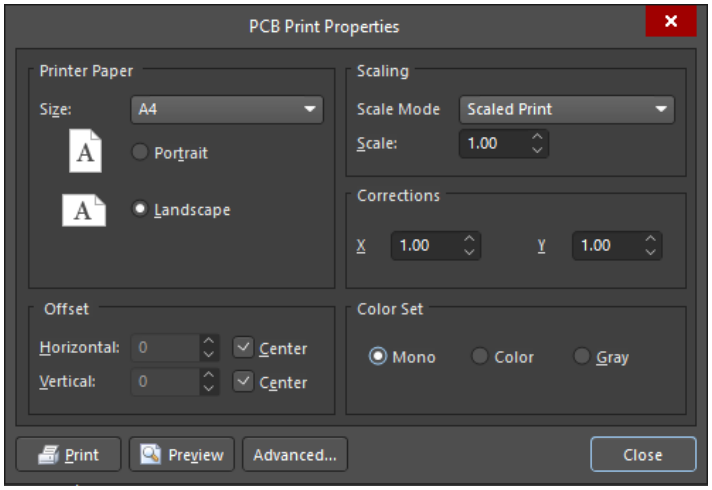


Figura A.29: Alinear modelo 3D con la placa

Luego se configura el archivo de salida, para ello se da click derecho y se escoge la opción **Configure**, se despliega una ventana mostrada en la figura A.30.

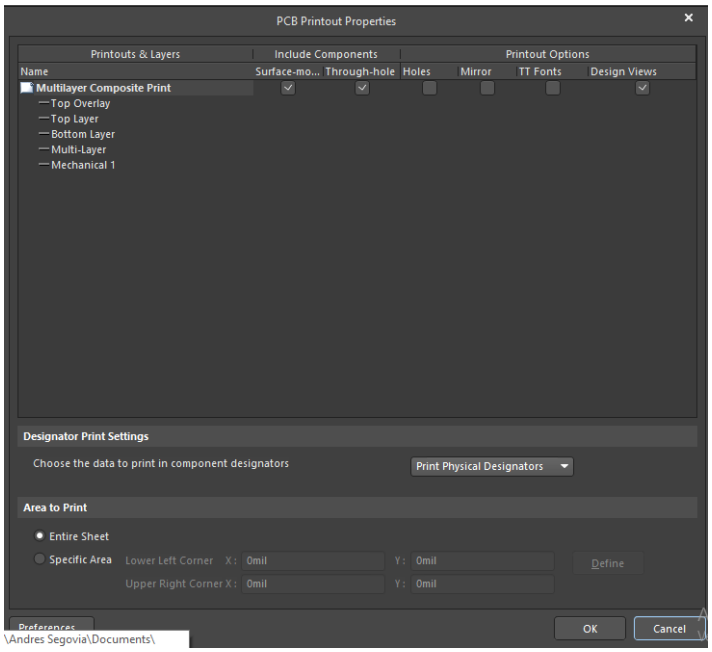


Figura A.30: Configuración de propiedades del archivo de salida

En esta venta, damos doble click en Multilayer Composite Print y accedemos a las capas de impresión, en ella borramos las capas que no queremos que se imprima, además es importante marcar el cuadro **Mirror Layer**, para que se cree un espejo del archivo impreso. Una vez configurado, cerramos la ventana y damos click en **Ok**.

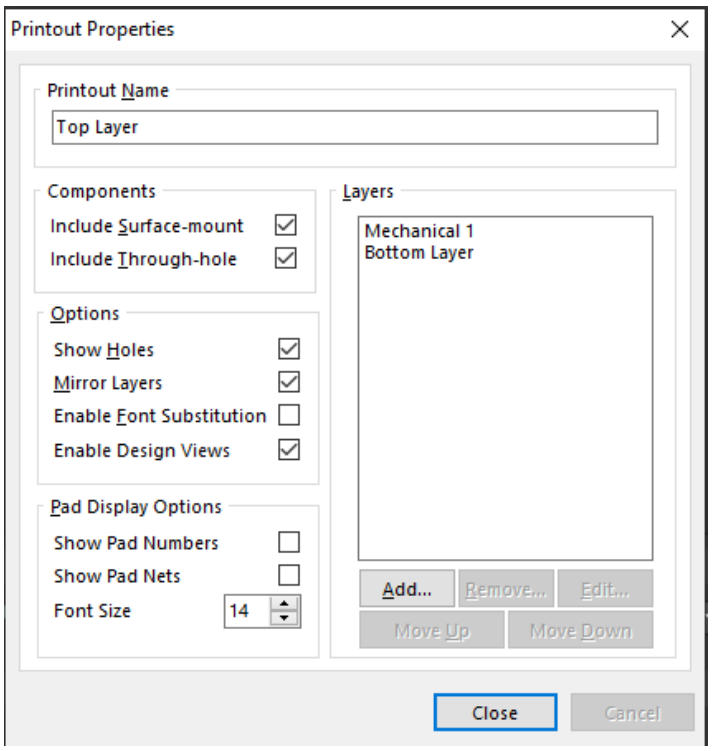


Figura A.31: Configuración de las capas a imprimir

Para generar el PDF en la parte derecha, en la pestaña **Output Containers** permite escojer el modo de salida, seleccionamos PDF y habilitamos el archivo, creado anteriormente, en la figura A.32 se observa este proceso, finalmente seleccionamos **Generate Content** y se general el archivo para impresión.

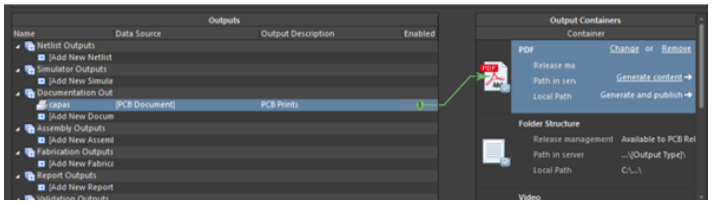


Figura A.32: Configuración para la generación del archivo PDF

Para generar el archivo .gerber en la ventana A.28, escogemos la opción para generar el archivo de salida .gerber como se ve en la figura A.34.

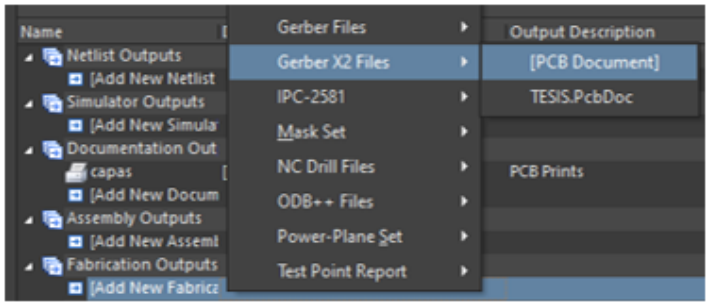


Figura A.33: Crear archivo para la salida .gerber

Una vez creado damos click derecho sobre el archivo y aparece la ventana de configuración, en la figura A.34 se observa esta ventana que permite configurar las unidades, las capas, la escala, etc.

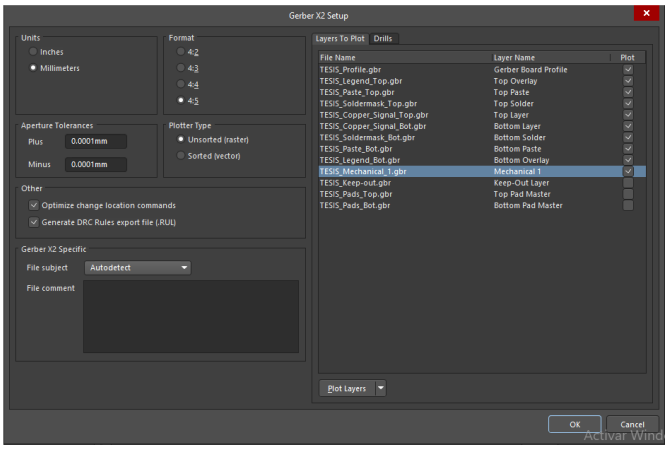


Figura A.34: Configuración del archivo de salida .gerber

Luego se procede a crear los agujeros, para ello creamos otro archivo como en la figura A.35.

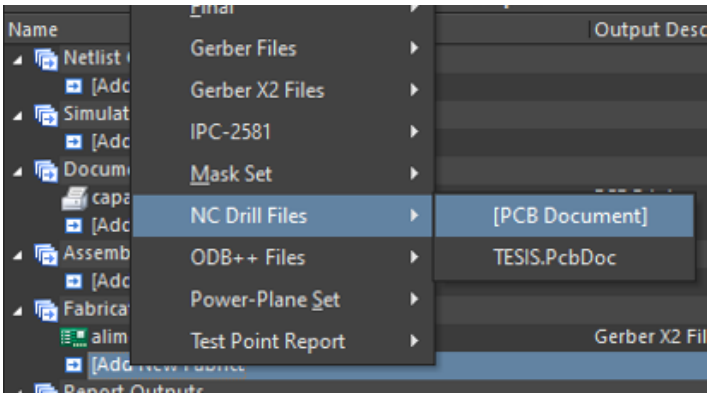


Figura A.35: Creación del archivo para los agujeros

Este archivo permite configurar las unidades, la escala, etc, relacionado con los agujeros (figura A.36).

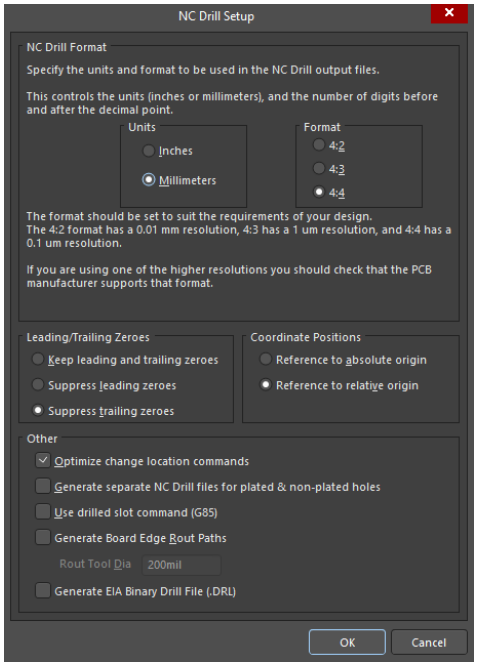


Figura A.36: Configuración de las propiedades para los agujeros

Con las configuraciones establecidas, procedemos a seleccionar el contenedor de salida, escogemos y habilitamos como se presenta en la figura A.37, finalmente generamos el contenido, en la dirección en donde se almacena el proyecto se puede encontrar los archivos de salida (figura A.38).

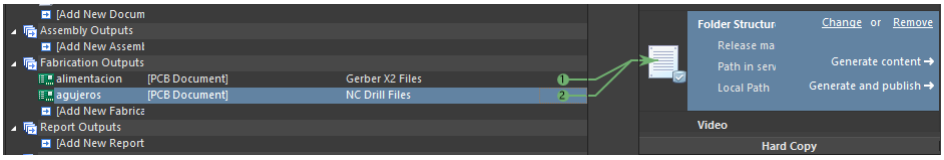


Figura A.37: Generar archivos en el contenedor

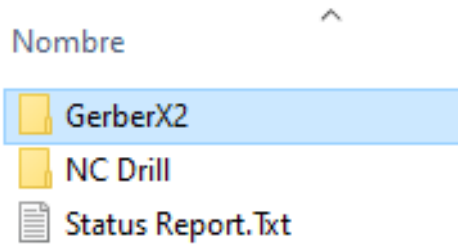


Figura A.38: Archivos generados

El resultado de realizar este proceso, en la figura A.39 se presenta el diseño final en PCB de la placa, en la figura A.40 se presenta el diseño en 3D del modelo final, en la figura A.41 se presenta el modelo para la impresión de la placa y finalmente en la figura A.42 se presenta física elaborada.

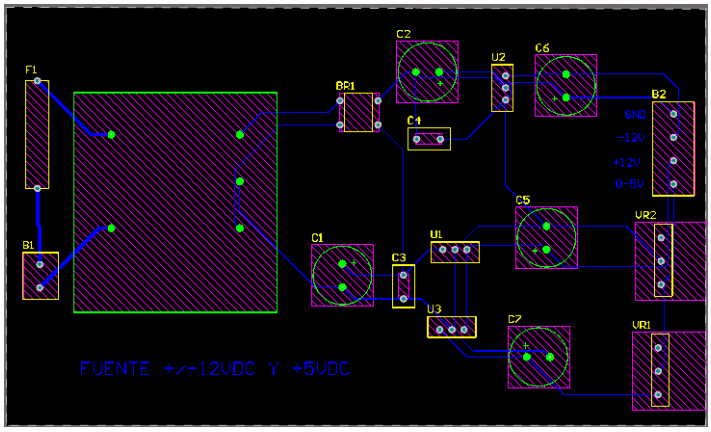


Figura A.39: Diseño del PCB de la fuente de alimentación

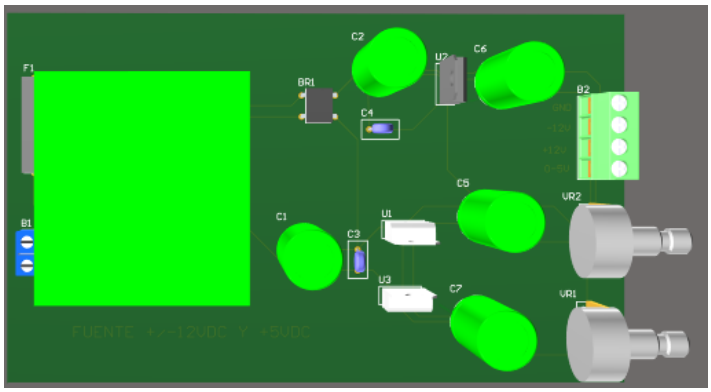


Figura A.40: Modelo 3D de la fuente de alimentación

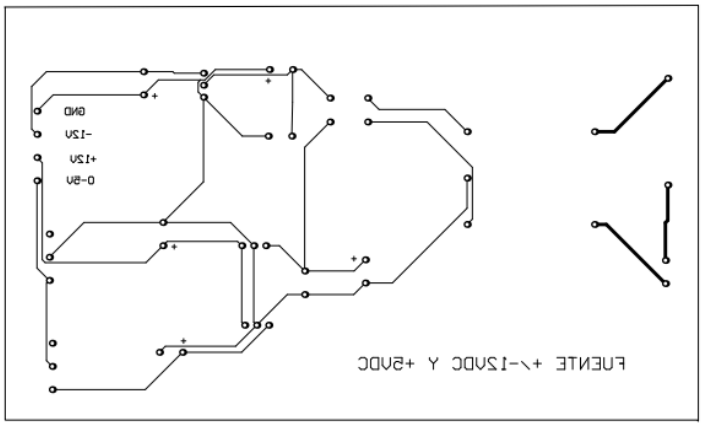


Figura A.41: Archivo final para impresión de la fuente de alimentación

Figura A.42: Fuente simétrica de 12V y regulable de 0-5VDC

Los diagramas de fabricación para el circuito de acondicionamiento se presentan a continuación, en la figura A.43 se presenta la impresión de la **PCB**, en la figura A.44 el modelo en 3D del diseño y finalmente en la figura A.45 al archivo para la impresión.

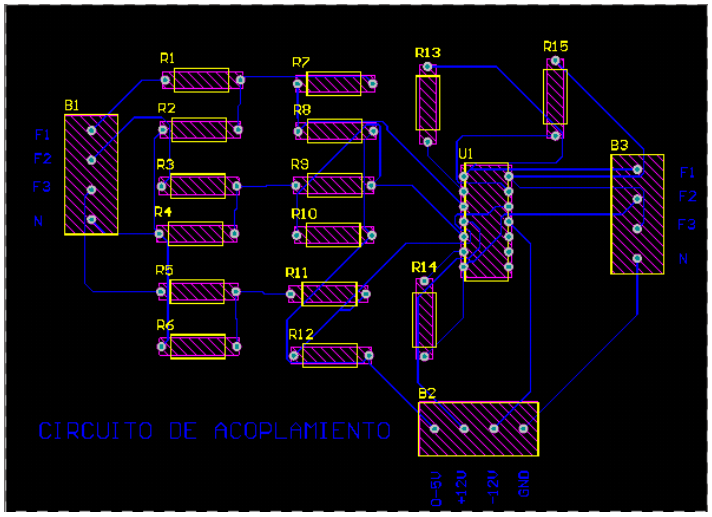


Figura A.43: Diseño de la PCB del circuito de acoplamiento

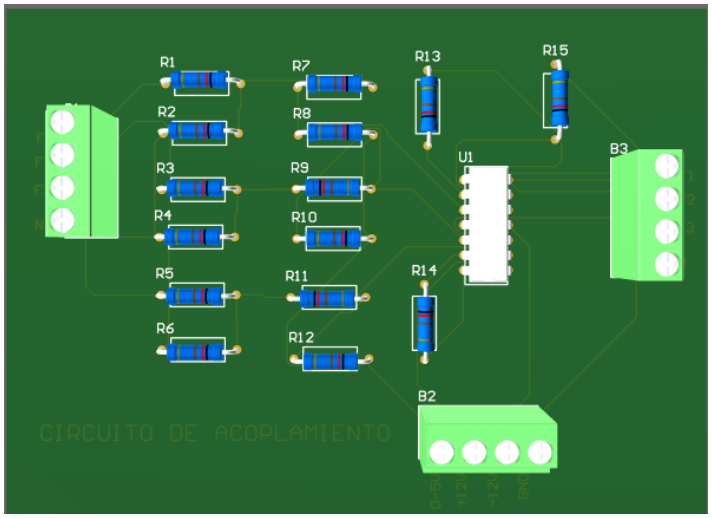


Figura A.44: Modelado 3D del circuito de acoplamiento

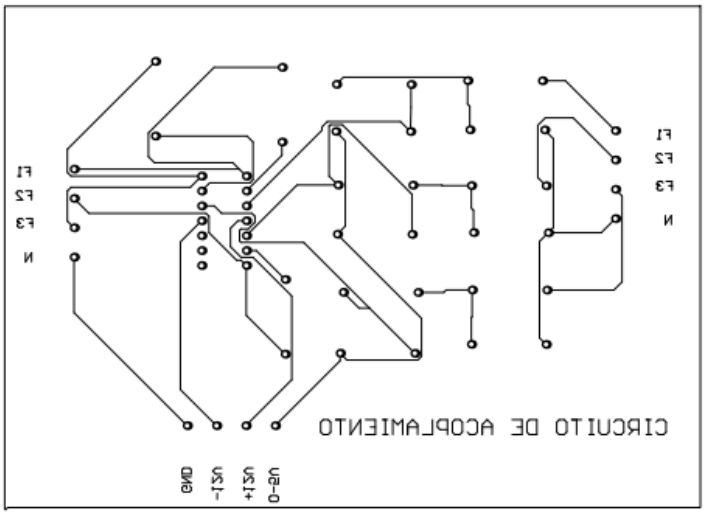


Figura A.45: Circuito de acoplamiento de la señal, para lectura de los puertos analógicos del DSP

En la figura A.46 se presenta la placa final del circuito.

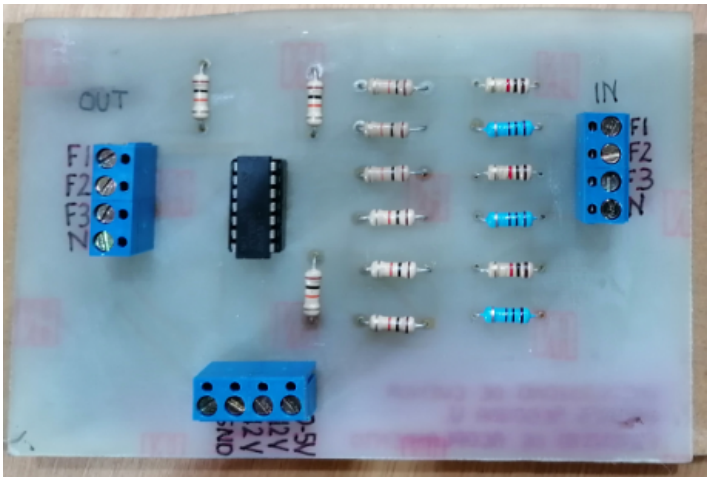


Figura A.46: Circuito de acoplamiento de la señal, para lectura de los puertos analógicos del DSP

Anexo B

Configuración de CCS

En este anexo se presenta los pasos para importar el código generado en **PSIM** a CCS y posteriormente los pasos para grabar el código en la tarjeta de control. Es importante establecer correctamente la configuración de la tarjeta de control en **PSIM**, ya que la configuración de CCS depende se la misma. En B.1 se detalla el proceso para importar el archivo del proyecto y finalmente en B.2 se presenta el proceso para grabar el código en la tarjeta de control.

B.1. Configuración de CCS para la importación del código generado en PSIM

Para importar los archivos **PSIM**, es importante verificar que al momento de generar el código, el software genere un archivo con extensión **.pjt**, este archivo contiene todo el proyecto (código, configuraciones, librerías, etc.). La versión de software utilizada para importar el código es **CCS v/7**, debido a que esta versión es compatible con la tarjeta y con el sistema operativo del PC.

En la figura B.1 se presenta la interfaz de usuario del software.

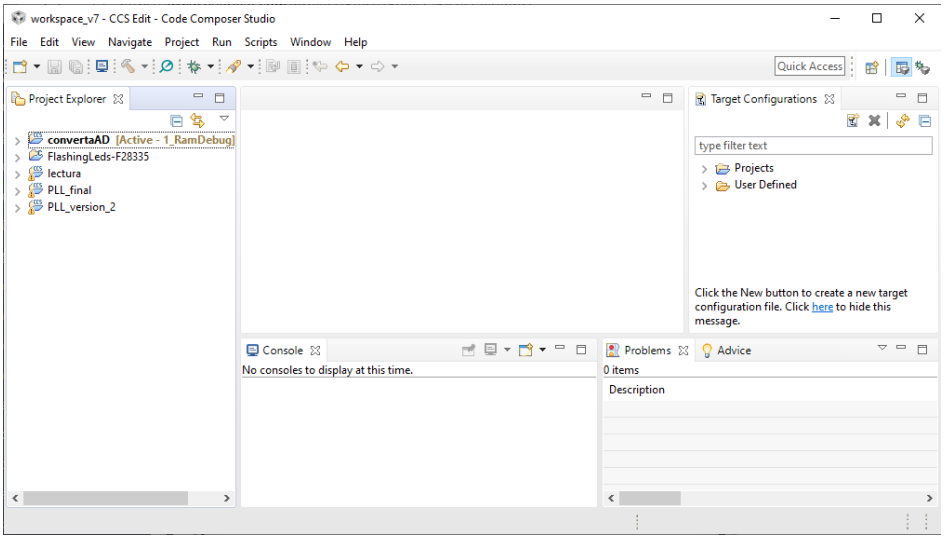


Figura B.1: Interfaz de usuario de CCS

Para importar el proyecto, nos vamos a la barra a la dirección **Project-Import Legacy CCSv3.3 Projects..**, como se presenta en la figura B.2.

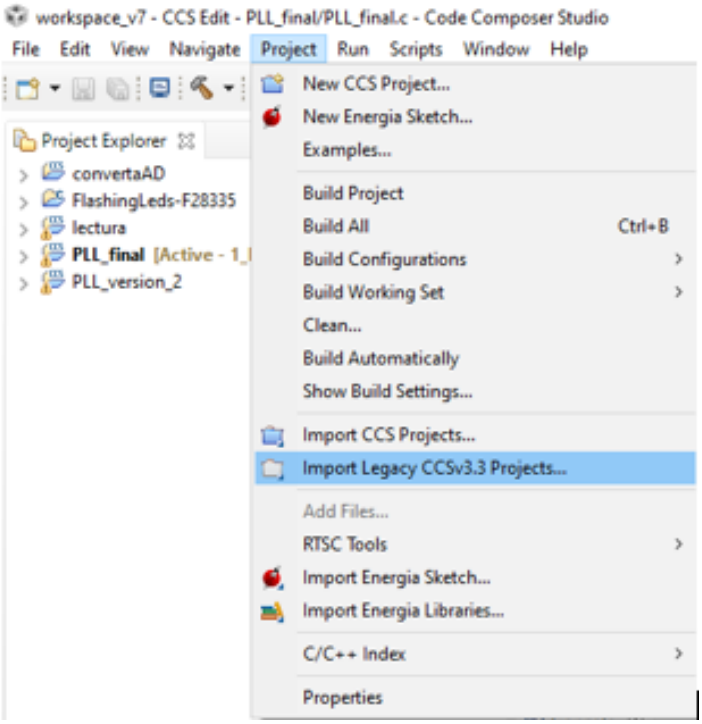


Figura B.2: Procedimiento para importar proyecto

A partir de esta acción, se abre una ventana que permite escoger el archivo, figura B.3, para ello vamos a la pestaña **Browse..**, y escogemos el proyecto a importar, este archivo se debe encontrar en la misma dirección en donde se guardo el esquema diseñado en **PSIM**.

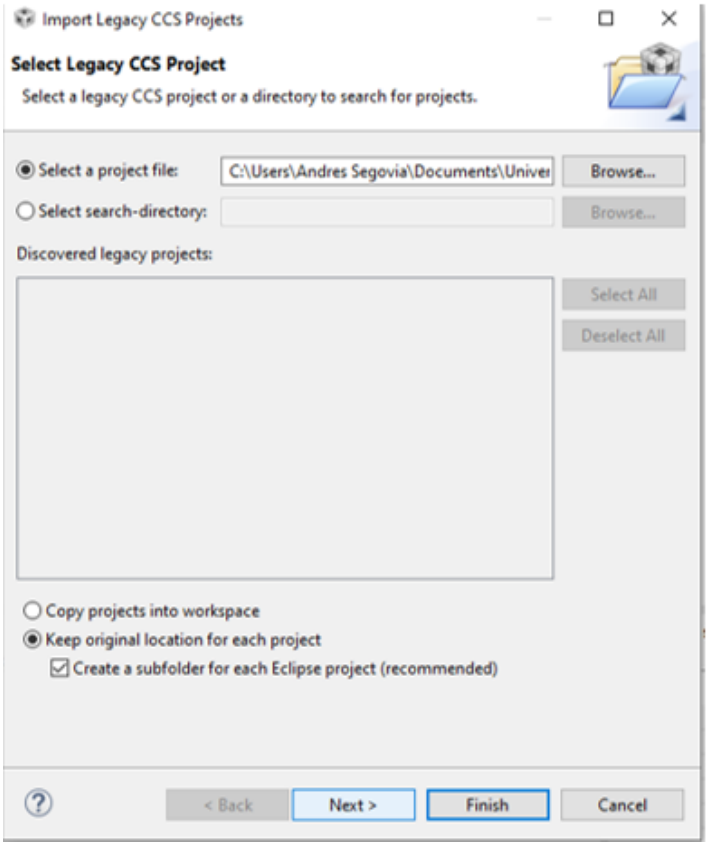


Figura B.3: Ventana que permite seleccionar el archivo del proyecto a importar

Para finalizar el proceso, es importante escoger el compilador que se va a utilizar, para ello en la figura B.4, se presenta la ventana que permite esta acción. Una vez seleccionado el compilador se finaliza la importación.

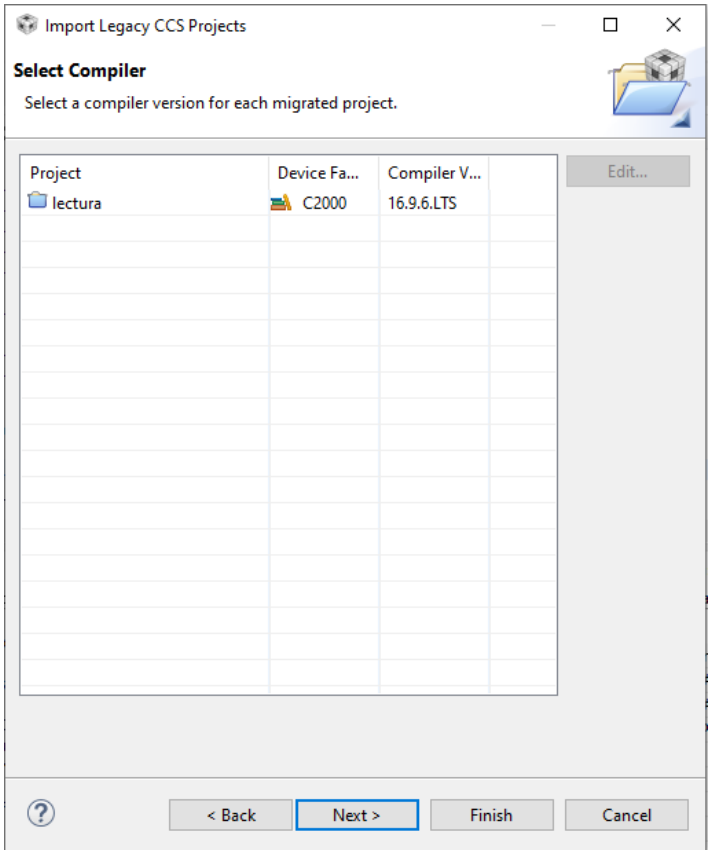


Figura B.4: Ventana que permite seleccionar el compilador para el proyecto

El archivo importado se presenta en la figura B.5, se observa que se han cargado las librerías y el archivo **.c** que es el que contiene el código.

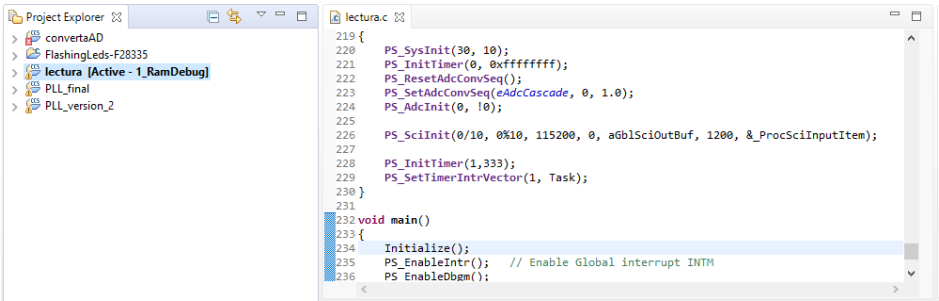


Figura B.5: Archivo importado correctamente

B.2. Grabado del código en la tarjeta de control F28335

Para realizar la exportación del código a la tarjeta, debemos considerar realizar la configuración de forma correcta del compilador. Para iniciar la grabación primero construimos el proyecto, para esto damos click derecho sobre el proyecto y se despliega las opciones del mismo seleccionamos **Build Project** como se ve en la figura B.6.

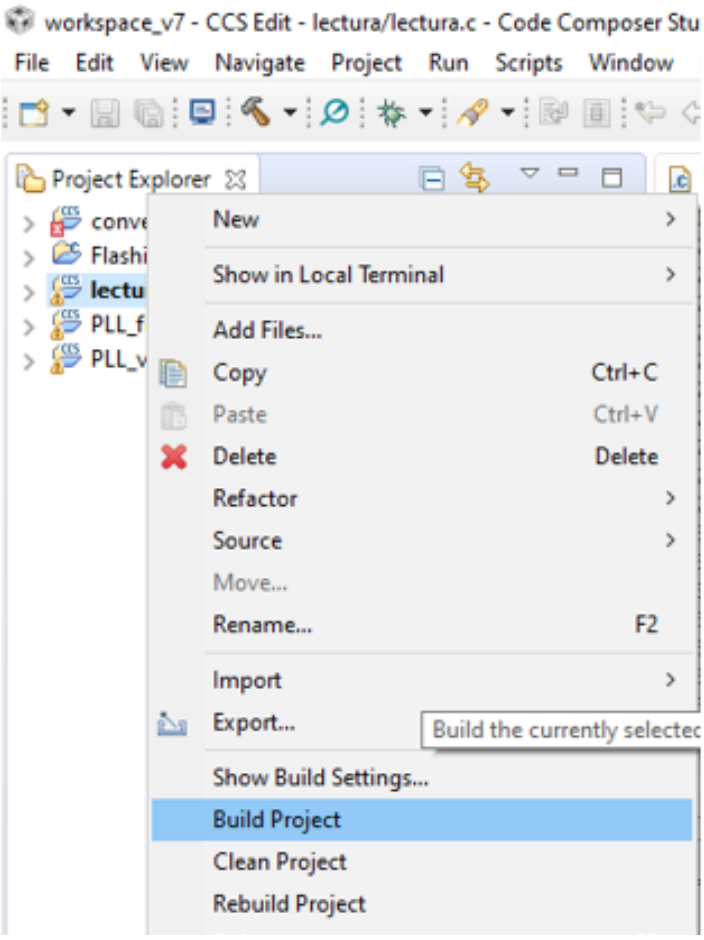


Figura B.6: Construcción del proyecto

Una vez compilado el proyecto, realizamos una depuración, al realizar esta acción, se realiza la exportación del código a la tarjeta de control, por lo que en la figura B.7 se presenta la acción a realizar.

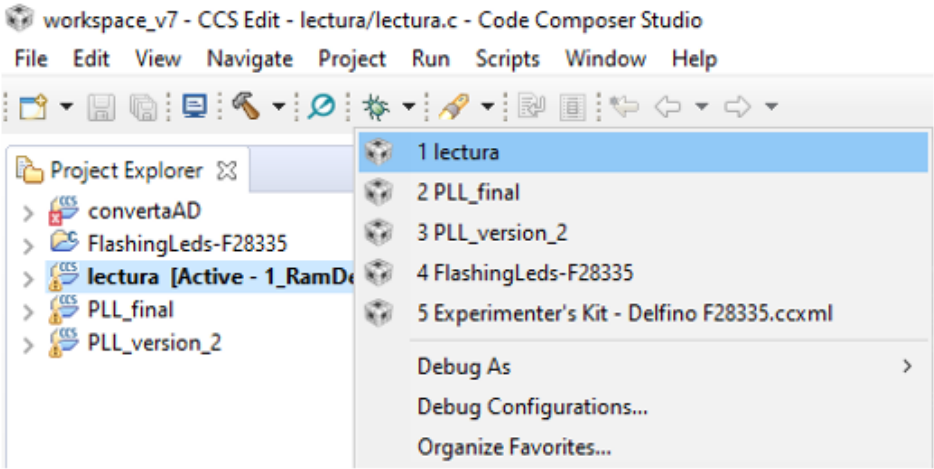


Figura B.7: Depuración del proyecto para la grabación del código en la tarjeta

Al momento de realizar la depuración, el programa comienza a correr, por lo que se visualiza los eventos en tiempo real en la ventana de interfaz, esto se ve en la figura B.8.

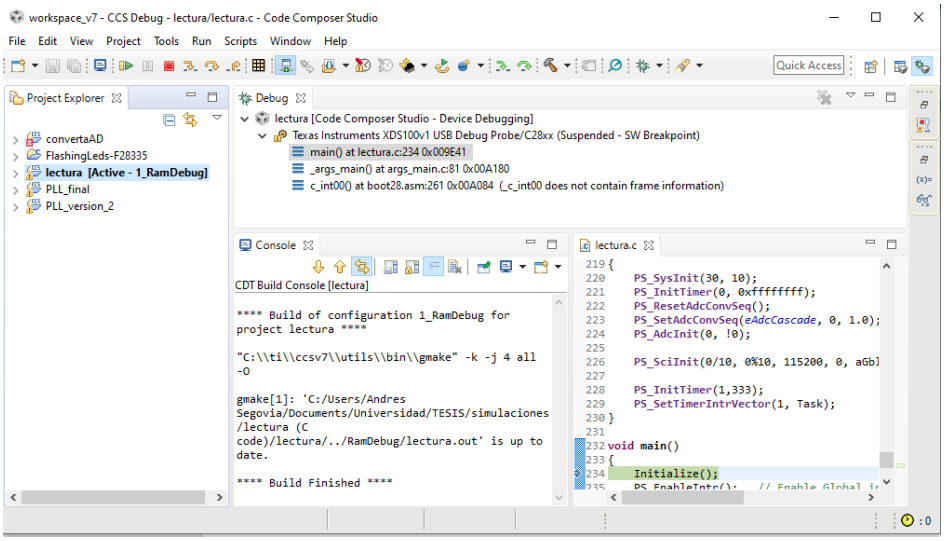


Figura B.8: Programa en tiempo real

Anexo C

Código para la lectura de datos

En este anexo se presenta el código para la conversión de datos analógicos a digitales, también permite la visualización de datos por medio de la interfaz SCI.

```
#include  <math.h>
#include  "PS_bios.h"
typedef float DefaultType;
#define GetCurTime() PS_GetSysTimer()

interrupt void Task();

PST_BufItem aGblSciOutBuf[1200];
Uint16 aGblSciOutAllow[7] = {0,0,0,0,0,0,0};
Uint16 aGblSciOutCnt[7] = {0,0,0,0,0,0,0};
Uint16 nGblSciState = 0;
Uint16 aGblSciDateSetPt[7] = {3,3,3,3,3,3,3};
char* aGblSciInitStr = "\0016,1:Va=3333.333252,3\003\0016,2:Vb
        =3333.333252,3\003\0016,3:Vc=3333.333252,3\003\0016,4:alfa_norm
        =3333.333252,3\003\0016,5:beta_norm=3333.333252,3\003\0016,6:beta
        =3333.333252,3\003\0016,7:alfa=3333.333252,3\003";
#define PSC_SCI_SENDOUT_FLAG  0x2000
#define PSC_SCI_INITIAL   0
#define PSC_SCI_START   0x5000000
#define PSC_SCI_PAUSE   0x1000000
#define PSC_SCI_RESTART   0x2000000
#define PSC_SCI_CONT_MODE 0x3000000
#define PSC_SCI_SNAP_MODE 0x4000000
#define PSC_SCI_CONT_START  1
#define PSC_SCI_CONT_BEGIN  2
#define PSC_SCI_CONT_SEND 3
#define PSC_SCI_CONT_PAUSE  4
#define PSC_SCI_SNAP_START  100
```

```
26 #define PSC_SCI_SNAP_BEGIN  101
27 #define PSC_SCI_SNAP_SEND 102
28 #define PSC_SCI_SNAP_WAIT 103
29 #define PSC_SCI_SNAP_PSEND  104
30 #define PSC_SCI_SNAP_PWAIT  105
31 #define PSC_SCI_SNAP_PAUSE  106
32 void _ProcSciInputItem(PST_BufItem* pItem)
33 {
34   Uint16 i, nSeqNo = pItem->nSeqNo.bit.nSeqNo;
35   switch (nSeqNo) {
36   case 0:
37     switch (pItem->data.dataInt32) {
38     case PSC_SCI_INITIAL:
39       for (i = 0; i < 7; i++) aGblSciOutAllow[i] = 0;
40       PS_SciClearSendBuf();
41       PS_SciSendInitStr(aGblSciInitStr);
42       break;
43     case PSC_SCI_PAUSE:
44       PS_DisableIntr();
45       switch (nGblSciState) {
46       case PSC_SCI_CONT_START:
47       case PSC_SCI_CONT_SEND:
48         PS_SciClearSendBuf();
49         nGblSciState = PSC_SCI_CONT_PAUSE;
50         break;
51       case PSC_SCI_SNAP_SEND:
52         nGblSciState = PSC_SCI_SNAP_PSEND;
53         break;
54       case PSC_SCI_SNAP_WAIT:
55         nGblSciState = PSC_SCI_SNAP_PWAIT;
56         break;
57       default:
58         break;
59       }
60       PS_EnableIntr();
61       break;
62     case PSC_SCI_RESTART:
63       PS_DisableIntr();
64       switch (nGblSciState) {
65       case PSC_SCI_CONT_PAUSE:
66         nGblSciState = PSC_SCI_CONT_START;
67         break;
68       case PSC_SCI_SNAP_PSEND:
69       case PSC_SCI_SNAP_PWAIT:
70       case PSC_SCI_SNAP_PAUSE:
71         nGblSciState = PSC_SCI_SNAP_START;
72         break;
```

```
73        }
74        PS_EnableIntr();
75        break;
76      case PSC_SCI_CONT_MODE:
77        nGblSciState = PSC_SCI_CONT_START;
78        break;
79      case PSC_SCI_SNAP_MODE:
80        nGblSciState = PSC_SCI_SNAP_START;
81        break;
82      default:
83        if (pItem->nSeqNo.bit.nCount == 0) {
84          for (i = 0; i < 7; i++) aGblSciOutAllow[i] = 0;
85        }
86        for (i = 0; i < 4; i++) {
87          int index = (pItem->data.dataInt32 >> (i * 8)) & 0xff;
88          if ((index > 0) && (index <= 7))
89            aGblSciOutAllow[index - 1] = PSC_SCI_SENDOUT_FLAG;
90        }
91        break;
92      }
93      break;
94    default:
95      break;
96    }
97  }
98  void _ProcSciRestart(void)
99  {
100   int i;
101   PST_BufItem item;
102
103   for (i = 0; i < 7; i++)
104     aGblSciOutAllow[i] &= 0xff00;
105   item.nSeqNo.all = 0;
106   switch (nGblSciState++) {
107   case PSC_SCI_CONT_BEGIN:
108     PS_SciClearSendBuf();
109     item.data.dataInt32 = 0;
110     break;
111   case PSC_SCI_SNAP_BEGIN:
112     item.data.dataInt32 = 1;
113     break;
114   case PSC_SCI_SNAP_PWAIT:
115     nGblSciState = PSC_SCI_SNAP_START;
116   case PSC_SCI_SNAP_WAIT:
117     item.data.dataInt32 = 255;
118     break;
119   }
```

```
120   PS_SciSendItem(&item);
121 }
122
123 void _ProcSciWaitStart(void)
124 {
125   PS_DisableIntr();
126   switch (nGblSciState) {
127   case PSC_SCI_CONT_START:
128     nGblSciState = PSC_SCI_CONT_BEGIN;
129     break;
130   case PSC_SCI_SNAP_START:
131     nGblSciState = PSC_SCI_SNAP_BEGIN;
132     break;
133   default:
134     break;
135   }
136   PS_EnableIntr();
137 }
138
139 void _ProcSciOutput(int index, float fVal)
140 {
141   PST_BufItem item;
142   int ok = ((aGblSciOutAllow[index] & PSC_SCI_SENDOUT_FLAG) &&
143     (++aGblSciOutCnt[index] >= aGblSciDateSetPt[index]));
144   PS_DisableIntr();
145   switch (nGblSciState) {
146   case PSC_SCI_CONT_BEGIN:
147   case PSC_SCI_SNAP_BEGIN:
148     _ProcSciRestart();
149     break;
150   case PSC_SCI_CONT_SEND:
151     if (ok) {
152       aGblSciOutCnt[index] = 0;
153       item.nSeqNo.bit.nCount = aGblSciOutAllow[index];
154       item.nSeqNo.bit.nSeqNo = index + 1;
155       item.data.dataFloat = fVal;
156       PS_SciSendItem(&item);
157       aGblSciOutAllow[index]++;
158       aGblSciOutAllow[index] &= ~0x100;
159     }
160     break;
161   case PSC_SCI_SNAP_SEND:
162   case PSC_SCI_SNAP_PSEND:
163     if (ok) {
164       aGblSciOutCnt[index] = 0;
165       item.nSeqNo.bit.nCount = aGblSciOutAllow[index];
166       item.nSeqNo.bit.nSeqNo = index + 1;
```

```
      item.data.dataFloat = fVal;
      if (!PS_SciSendItem(&item)) {
        nGblSciState++;
      } else {
        aGblSciOutAllow[index]++;
        aGblSciOutAllow[index] &= ~0x100;
      }
    }
    break;
  case PSC_SCI_SNAP_WAIT:
    if (PS_IsTxQueueEmpty()) {
      nGblSciState = PSC_SCI_SNAP_START;
    }
    break;
  case PSC_SCI_SNAP_PWAIT:
    if (PS_IsTxQueueEmpty()) {
      nGblSciState = PSC_SCI_SNAP_PAUSE;
    }
    break;
  default:
    break;
  }
  PS_EnableIntr();
}

interrupt void Task()
{
  DefaultType fTI_ADC2, fZOH6, fTI_ADC2_1, fZOH5, fTI_ADC2_2, fZOH7,
        fABC_AB1, fABC_AB1_1;
  DefaultType fFCNM3, fSQ2, fFCNM4, fZOH10, fFCNM7, fZOH11, fZOH14, fZOH15;
  PS_EnableIntr();

  fTI_ADC2 = PS_GetDcAdc(0);
  fTI_ADC2_1 = PS_GetDcAdc(1);
  fTI_ADC2_2 = PS_GetDcAdc(2);
  fZOH6 = fTI_ADC2;
  _ProcSciOutput(0, fZOH6);
  fZOH5 = fTI_ADC2_1;
  _ProcSciOutput(1, fZOH5);
  fZOH7 = fTI_ADC2_2;
  _ProcSciOutput(2, fZOH7);
  fABC_AB1 = 0.81649658 * (fTI_ADC2 - (fTI_ADC2_1 + fTI_ADC2_2) /2.0);
  fABC_AB1_1 = 0.70710678 * (fTI_ADC2_1 - fTI_ADC2_2); // uvw2ab
  fFCNM3 = (fABC_AB1*fABC_AB1)+(fABC_AB1_1*fABC_AB1_1);
  fSQ2 = sqrt(fFCNM3);
  fFCNM4 = fABC_AB1*1.0/fSQ2;
  fZOH10 = fFCNM4;
```

```
213   _ProcSciOutput(3, fZOH10);
214   fFCNM7 = fABC_AB1_1*1.0/fSQ2;
215   fZOH11 = fFCNM7;
216   _ProcSciOutput(4, fZOH11);
217   fZOH14 = fABC_AB1_1;
218   _ProcSciOutput(5, fZOH14);
219   fZOH15 = fABC_AB1;
220   _ProcSciOutput(6, fZOH15);
221   PS_ExitTimer1Intr();
222 }
223
224
225 void Initialize(void)
226 {
227   PS_SysInit(30, 10);
228   PS_InitTimer(0, 0xffffffff);
229   PS_ResetAdcConvSeq();
230   PS_SetAdcConvSeq(eAdcCascade, 0, 1.0);
231   PS_SetAdcConvSeq(eAdcCascade, 1, 1.0);
232   PS_SetAdcConvSeq(eAdcCascade, 2, 1.0);
233   PS_AdcInit(0, !0);
234
235   PS_SciInit(0/10, 0%10, 115200, 0, aGblSciOutBuf, 1200, &_ProcSciInputItem
        );
236
237   PS_InitTimer(1,100);
238   PS_SetTimerIntrVector(1, Task);
239 }
240
241 void main()
242 {
243   Initialize();
244   PS_EnableIntr();  // Enable Global interrupt INTM
245   PS_EnableDbgm();
246   for (;;) {
247     _ProcSciWaitStart();
248   }
249 }
```

Anexo D

Código para el cálculo de los parámetros kp y Ti del controlador

En este anexo se presenta el código para el cálculo de los parámetros del **PI**, en el se presenta el establecimiento de las funciones de transferencias y las expresiones para el cálculo.

```
% Funcion de transferencia de Planta
Gp = tf([1],[1 0])

Ts=0.001;
Xi=1/sqrt(2);

% Funcion de transferencia de controlador PID
Kp = 9.2/Ts
Tao_I = Ts*(Xi^2)/2.3
Tao_D = 0;

Gc = Kp + tf([Kp],[Tao_I 0]) + tf([Kp*Tao_D 0],[1])

% Funcion de transferencia a Lazo Cerrado
G_CL = Gp*Gc/(1+Gp*Gc)
[y,t] = step(G_CL);

% Grafica t vs y
plot(t,y,'k','LineWidth',1)
xlabel('t, s','FontSize',10);
ylabel('y(t)','FontSize',10);
axis ([0 max(t) 0 1.1*max(y)])
axis square
line([0 max(t)],[1 1],'Color','b','LineStyle','--')
```

Bibliografía

[1] Enerdata, "Anuario estadístico mundial de energía 2020." [En línea]. Disponible: https://datos.enerdata.net/energias-renovables/produccion-electricidad-renovable.html

[2] P. César, N. Rincón, P. César, y N. Rincón, "Fuentes convencionales y no convencionales de energía : estado actual y perspectivas The current state of non- conventional sources of energy and related perspectives," vol. 30, num. 3, pp. 165–173, 2010.

[3] J. Rosero; L. Garza; L. Minchala; D. Pozo; L. Morales, "Fuentes de Generación de Energía Eléctrica Convencional y Renovable a Nivel Mundial," vol. 32, pp. 1–13, 2013. [En línea]. Disponible: https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista{_}politecnica2/article/view/217/pdf

[4] D. Santabárbara *y otros*, "Evolución reciente y perspectivas del mercado de petróleo," *Boletín Económico*, vol. 3, p. 2017, 2017.

[5] P. Zhao, Z. Lu, J. Fang, S. Reddy, y K. Jiang, "Determinants of renewable and non-renewable energy demand in China," *Structural Change and Economic Dynamics*, vol. 54, pp. 202–209, 2020. [En línea]. Disponible: https://doi.org/10.1016/j.strueco.2020.05.002

[6] C. Yáñez y M. Garrido-Lepe, "El tercer ciclo del carbón en chile, de 1973 a 2013: del climaterio al rejuvenecimiento," *América Latina en la historia económica*, vol. 24, num. 3, pp. 224–258, 2017.

[7] V. Litvinenko, "The role of hydrocarbons in the global energy agenda: The focus on liquefied natural gas," *Resources*, vol. 9, num. 5, p. 59, 2020.

[8] TOTAL, "APLICACIONES INDUSTRIALES DEL GAS NATURAL." [En línea]. Disponible: https://www.totalenergia.es/es/pymes/blog/uso-industrial-del-gas-natural

[9] L. F. Azuela y J. L. Talancón, *Contracorriente: historia de la energía nuclear en México, 1945-1995*. Plaza y Valdes, 1999.

[10] F. Nuclear, "Energía nuclear en el mundo." [En línea]. Disponible: https://www.foronuclear.org/descubre-la-energia-nuclear/energia-nuclear-en-el-mundo/

[11] I.-O. I. de Energía Atómica, "Mapa sobre la situación de la energía nucleeléctrica." [En línea]. Disponible: https://www.iaea.org/sites/default/files/19/09/pris-map.pdf

[12] W. N. Association, "Nuclear Power in the World Today." [En línea]. Disponible: https://www.world-nuclear.org/information-library/current-and-future-generation/nuclear-power-in-the-world-today.aspx

[13] Hydropower, "Nuclear Power in the World Today." [En línea]. Disponible: https://www.irena.org/hydropower

[14] M. Bilgili, H. Bilirgen, A. Ozbek, F. Ekinci, y T. Demirdelen, "The role of hydropower installations for sustainable energy development in turkey and the world," *Renewable Energy*, vol. 126, pp. 755–764, 2018.

[15] J. Fernández, "Energía de la biomasa," *Energías renovables para el desarrollo. Thomson-Paraninfo*, 2003.

[16] F. S. Nogués, *Energía de la Biomasa (volumen I).* Universidad de Zaragoza, 2010, vol. 173.

[17] Z. Ali, N. Christofides, L. Hadjidemetriou, E. Kyriakides, Y. Yang, y F. Blaabjerg, "Three-phase phase-locked loop synchronization algorithms for grid-connected renewable energy systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 90, num. March, pp. 434–452, 2018.

[18] Z. Chen, S. Member, J. M. Guerrero, S. Member, y F. Blaabjerg, "A Review of the State of the Art of Power Electronics for Wind Turbines," vol. 24, num. 8, pp. 1859–1875, 2009.

[19] J. Garcia, A. Kirsten, D. Gacio, P. Quintana, U. F. S. Maria, y S. Maria, "Study of a Flyback-Based Stage as Grid Interface Topology for Micro-Generation Applications," pp. 1–6, 2012.

[20] E. y Sociedad, "ASPECTOS BÁSICOS DE LA ELECTRICIDAD." [En línea]. Disponible: http://www.energiaysociedad.es/manenergia/1-1-aspectos-basicos-de-la-electricidad/

[21] CELEC, "Infraestructura del Sistema Nacional de Transmisión (SNT)." [En línea]. Disponible: https://www.celec.gob.ec/transelectric/index.php/informacion-de-gestion

[22] F. BARRERO GONZALEZ, *Sistemas de energía eléctrica.* Editorial Paraninfo, 2004.

[23] N. REGULACIÓN, "Arconel 001/18.(2018). regulación nro. arconel 001/18," *AGENCIA DE REGULACIÓN Y CONTROL DE ELECTRICIDAD*, vol. 70.

[24] E. T. interinstitucional, "Plan Maestro de Electricidad 2016-2025," *Ministerio de Electricidad y energía Renovable.*

[25] I. S. Board, *IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems: 1547-2003.* IEEE, 2003.

[26] I. Standards, C. Committee, D. Generation, y E. Storage, *IEEE Std 1547.2-2008 IEEE Application Guide for IEEE Std 1547, IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems*, 2009, num. April.

[27] T. S. Basso y R. DeBlasio, "Ieee 1547 series of standards: interconnection issues," *IEEE Transactions on Power Electronics*, vol. 19, num. 5, pp. 1159–1162, 2004.

[28] P. Systems, "IEEE Standards," num. July, 2003.

[29] D. G. Photovoltaics y E. Storage, "Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," *IEEE Std*, pp. 1547–2018, 2018.

[30] R. N. ARCONEL, "004/15.(2015)," *Requerimientos Técnicos para la conexión y operación de generadores renovables no convencionales a las redes de transmisión y distribución.*

[31] ——, "005/18.(2018)," *Calidad del servicio de distribución y comercialización de energía eléctrica.*

[32] ——, "003/18.(2018)," *Generación fotovoltaica para autoabastecimiento de consumidores finales de energía eléctrica.*

[33] S. Sumathi, L. Ashok Kumar, y P. Surekha, *Solar PV and Wind Energy Conversion Systems*, 2015. [En línea]. Disponible: http://link.springer.com/10.1007/978-3-319-14941-7

[34] P. Lajevardi, A. P. Chandrakasan, y H. S. Lee, "Zero-crossing detector based reconfigurable analog system," *IEEE Journal of Solid-State Circuits*, vol. 46, num. 11, pp. 2478–2487, 2011.

[35] E. Irmak, I. Colak, O. Kaplan, y N. Guler, "Design and application of a novel zero-crossing detector circuit," *International Conference on Power Engineering, Energy and Electrical Drives*, num. May, pp. 7–10, 2011.

[36] I. P. Antchev, Mihail, "Pll for Single Phase Grid Connected Inverters," *International Journal of Electrical Engineering & Technology (Ijeet)*, vol. 4, num. 5, pp. 56–77, 2013.

[37] R. Teodorescu, M. Liserre, y P. Rodríguez, *Grid Synchronization in Single-Phase Power Converters*, 2010.

[38] S. H. Salter, "Power Conversion Systems for Ducks." *IEE Conference Publication*, num. 171, pp. 100–108, 1979.

[39] Y. Yang y F. Blaabjerg, "Synchronization in single-phase grid-connected photovoltaic systems under grid faults," *Proceedings - 2012 3rd IEEE International Symposium on Power Electronics for Distributed Generation Systems, PEDG 2012*, pp. 476–482, 2012.

[40] R. Teodorescu, M. Liserre, y P. Rodríguez, *Grid Synchronization in Three-Phase Power Converters*, 2010.

[41] F. González-Espín, E. Figueres, y G. Garcerá, "An Adaptive synchronous-reference-frame phase-locked loop for power quality improvement in a polluted utility grid," *IEEE Transactions on Industrial Electronics*, vol. 59, num. 6, pp. 2718–2731, 2012.

[42] S. Gao y M. Barnes, "Phase-locked loop for AC systems: Analyses and comparisons," *IET Conference Publications*, vol. 2012, num. 592 CP, pp. 1–6, 2012.

[43] R. A. Modesto, "Single-Phase PLL Structure Applied to Utility Connected Systems Using the Instantaneous Active Power Theory," pp. 63–69, 2008.

[44] X.-q. Guo, W.-y. Wu, y H.-r. Gu, "Phase locked loop and synchronization methods for grid-interfaced converters : a review," num. 4, pp. 182–187, 2011.

[45] P. Rodríguez, J. Pou, J. Bergas, J. I. Candela, R. P. Burgos, y D. Boroyevich, "Decoupled double synchronous reference frame PLL for power converters control (IEEE Transactions of Power Electronics)," *IEEE Transactions on Power Electronics*, vol. 22, num. 3, p. 1078, 2007.

[46] M. Reyes, P. Rodriguez, S. Vazquez, A. Luna, J. M. Carrasco, y R. Teodorescu, "Decoupled Double Synchronous Reference Frame current controller for unbalanced grid voltage conditions," *2012 IEEE Energy Conversion Congress and Exposition, ECCE 2012*, pp. 4676–4682, 2012.

[47] H. A. Pereira, A. F. Cupertino, C. A. S. Ribeiro, y S. R. Silva, "Influence of PLL in wind parks harmonic emissions," *2013 IEEE PES Conference on Innovative Smart Grid Technologies, ISGT LA 2013*, num. 3, 2013.

[48] L. Hadjidemetriou, E. Kyriakides, y F. Blaabjerg, "A new hybrid PLL for interconnecting renewable energy systems to the grid," *IEEE Transactions on Industry Applications*, vol. 49, num. 6, pp. 2709–2719, 2013.

[49] C. C. Martínez y M. Universitario, "Implementación de software para la sincronización de fase en frecuencia fija para sistemas electrónicos monofásicos y trifásicos mediante la plataforma LAUNCHXL-F28027 (C2000) y el entorno de desarrollo Code Composer Studio . DIRECTOR :," vol. 28027, 2016.

[50] P. Rodríguez, A. Luna, M. Ciobotaru, R. Teodorescu, y F. Blaabjerg, "Advanced grid synchronization system for power converters under unbalanced and distorted operating conditions," *IECON Proceedings (Industrial Electronics Conference)*, num. 2, pp. 5173–5178, 2006.

[51] I. Carugati, S. Maestri, P. G. Donato, D. Carrica, S. Member, y M. Benedetti, "Variable Sampling Period Filter PLL for Distorted Three-Phase Systems," vol. 27, num. 1, pp. 321–330, 2012.

[52] P. Rodríguez, A. Luna, I. Candela, R. Mujal, R. Teodorescu, y F. Blaabjerg, "Multiresonant frequency-locked loop for grid synchronization of power converters under distorted grid conditions," *IEEE Transactions on Industrial Electronics*, vol. 58, num. 1, pp. 127–138, 2011.

[53] P. F. D. Grado, "Modelado, programación y simulación de dispositivos de medida y protección de sistemas eléctricos," 2016.

[54] U. Simcoder y F. Target, "PSIM Tutorial."

[55] E.-e. Engineering, "Using Sliding Mode Controller," pp. 472–476, 2014.

[56] A. M. Alonso, F. P. Marafao, D. I. Brandao, E. Tedeschi, y J. F. Guerreiro, "A guideline for employing PSIM on power converter applications: Prototyping and educational tool," *14th Brazilian Power Electronics Conference, COBEP 2017*, vol. 2018-Janua, pp. 1–6, 2017.

[57] "Auto Code Generation for F2833X Target," num. October, 2016.

[58] A. Port y B. S. Architecture, "TMS320F2833x , TMS320F2823x Digital Signal Controllers (DSCs)," 2019.

[59] "Using SCI for Waveform Monitoring," num. November 2016, pp. 1–14.

[60] S. K. Chung, "A phase tracking system for three phase utility interface inverters," *IEEE Transactions on Power Electronics*, vol. 15, num. 3, pp. 431–438, 2000.

[61] L. González, "Mejora de la eficiencia y de las prestaciones dinámicas en procesadores electrónicos de potencia para pequeños aerogeneradores sincrónicos operando en régimen de velocidad variable," Ph.D. dissertation, Departamento de Ingeniería Electrónica. Universidad Politécnica de Valencia, May 2011.

[62] M. Electronics, "Productos." [En línea]. Disponible: https://www.mouser.ec/electronic-cad-symbols-models/