

ING. IVÁN ALEJANDRO MENDOZA
VARELA

"SISTEMA DE CARGA PARA CELDAS DE LITIO-
FERROFOSFATO"

2022



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE INGENIERÍA

"SISTEMA DE CARGA PARA CELDAS DE LITIO- FERROFOSFATO"

TESIS

QUE COMO PARTE DE LOS REQUISITOS PARA OBTENER EL
GRADO DE MAESTRO EN CIENCIAS EN INSTRUMENTACIÓN Y
CONTROL AUTOMÁTICO

PRESENTA:

ING. IVÁN ALEJANDRO MENDOZA VARELA

DIRIGIDO POR:

DR. MIGUEL ÁNGEL MARTÍNEZ PRADO

Santiago de Querétaro, Qro. Octubre 2021



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Instrumentación y
Control Automático

**SISTEMA DE CARGA PARA CELDAS DE LITIO-FERROFOSFATO
TESIS**

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Instrumentación y Control Automático

Presenta:

Ing. Iván Alejandro Mendoza Varela

Dirigido por:

Dr. Miguel Ángel Martínez Prado

SINODALES

Dr. Miguel Ángel Martínez Prado

Presidente

Dr. Juvenal Rodríguez Reséndiz

Secretario

C. a Dr. Alfredo Álvarez Díaz-Comas

Vocal

Dr. Ákos Odry

Suplente

Dr. Mariano Garduño Aparicio

Suplente

Centro Universitario

Querétaro, QRO

México.

Octubre 2021

© 2021 - Ing. Iván Alejandro Mendoza Varela

All rights reserved.

Abstract

A proper charge in an electric vehicle (EV) battery allows it to have a longer useful life and therefore a lower maintenance cost. For this purpose, the voltage and current supplied to the battery must be precisely regulated. In this thesis the model of a phase-shifted full-bridge (PSFB) converter is obtained, and a double control loop is designed to regulate the state of charge of a lithium ferrophosphate (LiFePO_4) battery. The autoregressive exogenous (ARX) model is used to model the system from the response of the converter to a step input. Once the plant model is obtained, it is controlled by means of a classical controller. A couple of cases are evaluated where the control parameters are modified, the cases are compared, and the best approach is selected. From the obtained model, the controller is designed for the proposed cases. The theoretical controller response is compared with the experimental response. The results show that the modeling of the converter is accurate in a 94% by comparing it with the real response. In the same way, the result obtained from the controller is accurate in a 96% by comparing it with a simulation. Both the modeling and the control obtained experimentally resemble the results obtained theoretically. The controller manages to respond as expected in all proposed cases.

Resumen

Una carga adecuada en la batería de un vehículo eléctrico (EV) le permite tener una vida útil más larga y por lo tanto un menor costo de mantenimiento. Para ello, el voltaje y la corriente suministrados a la batería deben regularse con precisión. En esta tesis se obtiene el modelo de un convertidor de puente completo con desplazamiento de fase (PSFB) y se diseña un lazo de control doble para regular el estado de carga de una batería de ferrofosfato de litio (LiFePO_4). El modelo exógeno autorregresivo (ARX) se utiliza para modelar el sistema a partir de la respuesta del convertidor a una entrada escalonada. Una vez obtenido el modelo de la planta, se controla mediante un controlador clásico. Se evalúan un par de casos donde se modifican los parámetros de control, se comparan los casos y se selecciona el mejor diseño. A partir del modelo obtenido, se diseña el controlador para los casos propuestos. La respuesta teórica del controlador se compara con la respuesta experimental. Los resultados muestran que el modelado del convertidor es preciso en un 94 % comparándolo con la respuesta real. De la misma forma, el resultado obtenido del controlador es preciso en un 96 % comparándolo con una simulación. Tanto el modelado como el control obtenido experimentalmente se asemejan a los resultados obtenidos teóricamente. El controlador logra responder como se esperaba en todos los casos propuestos.

Índice general

Abstract	
Resumen	I
Contenido	III
Lista de Figuras	VII
Lista de Tablas	IX
1. Introducción	1
1.1. Justificación	1
1.2. Planteamiento del Problema	2
1.3. Hipótesis y Objetivos	3
1.3.1. Hipótesis	3
1.3.2. Objetivo General	3
1.3.3. Objetivos Particulares	3
1.4. Estructura de Tesis	4
2. Antecedentes	5
2.1. Trabajos Relacionados	5
3. Metodología	15

3.1. Fundamentación teórica	15
3.1.1. Filtro EMI (interferencia electromagnética)	15
3.1.2. Rectificación	16
3.1.3. Convertidor CD-CD	18
3.1.4. Modelado de la Planta	21
3.1.5. Sistema de control	25
3.1.6. Cargadores eléctricos comerciales	26
3.2. Implementación	27
3.2.1. Diagrama de bloques	27
3.3. Diseño	29
3.3.1. Unidad de protección	29
3.3.2. Filtrado	29
3.3.3. Transformador	29
3.3.4. Convertidor DC-DC	33
3.3.5. Desplazamiento de Fase	34
3.3.6. Control	35
4. Resultados y Discusión	37
4.1. Resultados	37
4.1.1. Filtrado	37
4.1.2. Materiales	39
4.1.3. Eficiencia del convertidor	39
4.1.4. Implementación del modelo	43
4.1.5. Control de voltaje	46
4.2. Aporte científico de la investigación	51
4.3. Aporte tecnológico de la investigación	51
4.4. Impacto sostenible de la investigación	52
4.5. Consideraciones éticas	52

5. Conclusiones	53
References	58
Apéndice A. Banco de pruebas	59
A.1. Programación de microcontrolador	59
A.2. PCB	82

Índice de figuras

3.1. Filtro EMI en modo común.	16
3.2. Rectificador de onda completa conformado por cuatro diodos.	17
3.3. Convertidor CC-CC de puente completo.	19
3.4. Planta con entrada $u(t)$ y salida $y(t)$ con adición de ruido $v(t)$.	21
3.5. Diagrama de bloques de actividades.	27
3.6. Diagrama de bloques de funcionamiento de cargador.	29
3.7. Convertidor de puente completo.	34
4.1. Señal de 60 Hz después de filtro EMI.	38
4.2. Señal de 4 kHz después de filtro EMI.	38
4.3. Señal de 100 kHz después de filtro EMI.	39
4.4. Banco de pruebas utilizado para experimentación.	44
4.5. Diagrama del sistema en Simulink.	45
4.6. Respuesta de la planta ante entrada escalón	46
4.7. Diagrama del sistema, considerando el modelo obtenido y el controlador de voltaje.	47
4.8. Ajuste del controlador a través del sintonizador PID de MATLAB.	48
4.9. Diagrama de control en Simulink.	48
4.10. Caso 1 con $T_s = 0,02$.	49
4.11. Caso 2 con $T_s = 0,05$.	49
4.12. Respuesta a la señal de escalón aleatoria.	50

4.13. Respuesta del controlador PI ante señal de diente de sierra.	51
A.1. PCB de prototipo, cara superior.	83
A.2. PCB de prototipo, cara inferior.	84

Índice de tablas

3.1. Niveles de cargadores comerciales.	27
4.1. Componentes de banco de pruebas.	39

Introducción

Este trabajo analiza el procedimiento para obtener el modelado del convertidor PSFB, con el fin de obtener la función de transferencia del convertidor y realizar un control de tensión constante. Los resultados muestran que el modelo obtenido tiene una precisión del 94 % al compararlo con la respuesta real del convertidor. Además, el resultado obtenido del controlador es preciso en un 96 % comparándolo con una simulación.

1.1. Justificación

Una de las necesidades de los usuarios de EV es reducir el tiempo que lleva recargar la batería de sus coches. Contar con un sistema que sea capaz de recargar la batería en menor tiempo implica la demanda de mayor corriente a la línea eléctrica [1]. Al demandar mayor corriente, se requiere el uso de elementos capaces de soportar la corriente y de un control adecuado para evitar la pérdida de energía en forma de calor.

Es por esto que se requiere de la construcción de un dispositivo que sea capaz de cargar la batería del vehículo, cuidando que el voltaje y corriente sean controlados de manera correcta para que la batería tenga una vida útil óptima y tenga una alta eficiencia [2]. El sistema propuesto es capaz de entregar voltaje y corriente controlada de acuerdo con la carga de un paquete de celdas de LiFePO_4 . Se busca reducir el tiempo en el que el paquete de celdas es

cargado en su totalidad. Para esto es necesario suministrar la corriente y voltaje adecuados con base en las características de la batería utilizada. La corriente requerida para esta tarea amerita el uso de convertidores electrónicos de potencia y de un sistema embebido capaz de realizar el control adecuado para la salida del convertidor.

El uso de un doble lazo de control utilizado en un convertidor electrónico de puente completo permite controlar tanto voltaje como corriente para alimentar el paquete de celdas y así seguir su curva de carga ideal. De igual manera se necesita obtener el modelo de la planta para su control, por lo que se utiliza el modelo *autoregressive exogenous* (ARX) para describir la respuesta del sistema.

1.2. Planteamiento del Problema

Los EV son cada vez más populares y prácticos. Pueden reducir las emisiones y ser más baratos a largo plazo. Éstos ofrecen algunas ventajas que no están disponibles en los vehículos convencionales con motor de combustión interna [3]. Debido a que los motores eléctricos reaccionan rápidamente, los EV son muy sensibles y tienen un buen torque. De igual manera la eficiencia que éstos presentan es de hasta 75 %, mientras que con los automóviles convencionales llega únicamente a 25 %, pues parte de la energía se pierde en forma de calor [4] [5].

Como fuente de energía, EV utilizan baterías de iones de litio, principalmente por su alto almacenamiento de energía por unidad de masa en relación con otros sistemas de almacenamiento. Es importante realizar una carga adecuada de la batería para que ésta pueda cumplir su vida útil. Para esto se necesita un dispositivo capaz de suministrar la corriente y voltaje necesarios en la batería de litio dependiendo de su estado de carga.

La demanda de EV e híbridos hoy en día va en aumento. Según el Instituto Nacional de Estadística y Geografía (INEGI), la cantidad de vehículos que se vendieron en el 2019 (sin contar noviembre y diciembre) fueron 19595 unidades, mientras que en el 2016 fueron 8265, es decir, hubo un 237 % de aumento en la compra de vehículos el año pasado a comparación del 2016 [6]. Esto quiere decir que en México se debe de comenzar a considerar un cambio

de infraestructura para poder dar servicio a aquellos portadores de EV y PHEV. A pesar de que, en México los EV vendidos en 2019 apenas representan el 1.8 % del total de autos vendidos, la tendencia va al alza puesto que la evolución registrada el año pasado indica el comienzo de la transición a autos eléctricos.

Además de ser más eficientes, los EV no emiten gases de efecto invernadero perjudiciales para el ambiente ni óxido de nitrógeno, el cual es dañino para la salud. Esto se debe a que no ocupan gasolina. En 2016, la empresa estadounidense *Bloomberg New Energy Finance* (BNEF) realizó un estudio con respecto a la gasolina ahorrada por el uso de EV. Según este estudio, los EV evitaron el consumo de 17.800 barriles de petróleo al día, lo que representa un crecimiento del 77 % respecto a 2015.

1.3. Hipótesis y Objetivos

1.3.1. Hipótesis

El tiempo de carga de un paquete de 48 celdas de LiFePO_4 de 100 Ah por cada celda es menor a 4 horas al utilizar un doble lazo de control para regular el voltaje y la corriente de salida de un convertidor electrónico y así operar a una potencia de 4.5 kW.

1.3.2. Objetivo General

Cargar un paquete de 48 celdas de LiFePO_4 controlando el voltaje y la corriente suministrada a partir de una fuente de corriente directa.

1.3.3. Objetivos Particulares

- Acondicionar el voltaje para ser introducido a la batería, seleccionando un convertidor CD-CD adecuado.
- Controlar el voltaje y corriente de salida del convertidor para carga en corriente continua y voltaje continuo, dependiendo del estado de carga.

- Evaluar la componente tecnológica desarrollada en la batería de un EV.

1.4. Estructura de Tesis

La tesis está organizada de la siguiente manera:

- El capítulo 2
- El capítulo 3
- El capítulo 4
- El capítulo 5

Antecedentes

El tema de la contaminación ambiental ha sido estudiado de manera exhaustiva en los últimos años. Una de las soluciones es reducir la generación de dióxido de carbono por parte de los automóviles que operan con motores de combustión interna. A continuación se muestran los principales trabajos que se han realizado con respecto a la mejora de la tecnología de carga en vehículos eléctricos.

2.1. Trabajos Relacionados

En el 2011 se revisan la estructura y el rendimiento electroquímico de LiFePO_4 a la luz de los principales requisitos técnicos para las baterías de vehículos eléctricos [7]. Se describen la capacidad de velocidad, la densidad de capacidad, la vida cíclica y el rendimiento a baja temperatura de varios materiales LiFePO_4 . Se discuten los principales factores que afectan estas propiedades, que incluyen el tamaño de partícula, el dopado, el revestimiento de carbono, la carga de carbono conductor y las técnicas de síntesis.

Una de las consideraciones para una alta eficiencia, es la forma en que se convierte la energía alterna a directa. Musavi propone una nueva topología de corrección del factor de potencia entrelazado sin puente AC-DC frontal para la carga de baterías de vehículos eléctricos híbridos enchufables (PHEV) de nivel 2 [8]. La topología puede lograr una alta

eficiencia, que es fundamental para minimizar el tamaño del cargador, el tiempo de carga del PHEV y la cantidad y el costo de la electricidad extraída de la red. Además, se presenta un modelo analítico detallado para esta topología, que permite el cálculo de las pérdidas de potencia y la eficiencia del convertidor. Se incluyen resultados experimentales y de simulación para un prototipo de convertidor elevador que convierte el voltaje de entrada de CA universal (85 265 V) en una salida de 400 V de CC con una carga de hasta 3,4 kW. Los resultados experimentales demuestran un factor de potencia superior a 0,99 de 750 W a 3,4 kW, THD inferior al 5% de media carga a plena carga y una eficiencia máxima del 98,9% a una frecuencia de conmutación de 70 kHz, entrada de 265 V y carga de 1,2 kW.

Un año después, el mismo Musavi presenta un estudio de topología que evalúa las topologías para su uso en convertidores de CA-CC frontal para cargadores de baterías PHEV [9]. El estudio de topología se centra en varios convertidores con corrección del factor de potencia de refuerzo, que ofrecen alta eficiencia, alto factor de potencia, alta densidad y bajo costo. Los resultados experimentales se presentan e interpretan para cinco convertidores prototipo, convirtiendo el voltaje de entrada de CA universal a 400 V de CC. Los resultados demuestran que el convertidor elevador PFC de medio puente de fase desplazada es ideal para aplicaciones de carga residencial de nivel 1 de automoción en Norteamérica, donde el suministro típico está limitado a 120 V y 1,44 kVA o 1,92 kVA. Para aplicaciones de carga residencial de nivel 2 para automóviles en América del Norte y Europa, el convertidor elevador PFC intercalado sin puentes es un candidato de topología ideal para suministros típicos de 240 V, con niveles de potencia de 3,3 kW, 5 kW y 6,6 kW.

Cada vehículo eléctrico tiene distintas necesidades de carga en cuanto a potencia. Se presenta un estudio topológico completo de las soluciones de carga de los vehículos eléctricos disponibles actualmente [10]. Los cargadores de vehículos eléctricos, etapas de conversión (etapa única integrada o dos etapas), nivel de potencia (nivel 1, 2 o 3) y tipo de dispositivos semiconductores utilizados (silicio, carburo de silicio, o nitruro de galio) se revisan a fondo en este documento.

Es importante modelar el comportamiento de la batería a utilizar. Hongwen He presen-

ta un método de estimación en línea basado en modelos de una batería LiFePO_4 para su aplicación en vehículos eléctricos (EV) mediante el uso de un algoritmo de filtro de Kalman extendido adaptativo (AEKF) [11]. El modelo de circuito equivalente de Thevenin se selecciona para modelar la batería LiFePO_4 y sus ecuaciones matemáticas se deducen hasta cierto punto. Además, se elabora y emplea una implementación del algoritmo AEKF para la estimación de los parámetros en línea del modelo de batería LiFePO_4 . Para ilustrar las ventajas de la estimación de los parámetros en línea, se realiza un análisis de comparación en los voltajes terminales entre la estimación en línea y el cálculo fuera de línea bajo la prueba de característica de potencia de pulso híbrida (HPPC) y la prueba de horario de conducción de dinamómetro urbano (UDDS). Además, se propone un enfoque de estimación de SoC en línea eficiente basado en el resultado de la estimación en línea de voltaje de circuito abierto (OCV). Los resultados experimentales muestran que la estimación de SoC en línea basada en OCV-SoC puede limitar de manera eficiente el error por debajo de 0.041.

Dentro de los diseños de cargadores eléctricos, Oh presenta el diseño e implementación de un cargador de batería a bordo (OBC) de una etapa no aislado de alta eficiencia para vehículos eléctricos [12]. Al revisar las topologías convencionales, se determina una estructura de circuito adecuada para cargar una batería en un amplio espectro de condiciones de entrada y salida. Además, se presenta una estrategia adecuada para un OBC altamente eficiente a través del análisis de topología seleccionada. Se presenta un análisis teórico detallado, una estrategia operativa y resultados experimentales en un prototipo de 3.7 kW para evaluar el desempeño del sistema propuesto.

Yilmaz revisa el estado actual y la implementación de los cargadores de baterías, los niveles de potencia de carga y la infraestructura para vehículos eléctricos e híbridos enchufables [1]. Los sistemas de carga se clasifican en tipos integrados y externos con flujo de energía unidireccional o bidireccional. La carga unidireccional limita los requisitos de hardware y simplifica los problemas de interconexión. La carga bidireccional admite la inyección de energía de la batería a la red. Los cargadores a bordo típicos restringen la energía debido a limitaciones de peso, espacio y costo. Pueden integrarse con el accionamiento eléctrico para

evitar estos problemas. La disponibilidad de la infraestructura de carga reduce los requisitos y los costos de almacenamiento de energía a bordo. Los sistemas de carga a bordo pueden ser conductores o inductivos. Un cargador externo puede diseñarse para altas tasas de carga y está menos limitado por el tamaño y el peso. Se discuten los niveles de potencia de Nivel 1, Nivel 2 y Nivel 3. Se presentan aspectos futuros como la recarga de la plataforma. Se presentan, comparan y evalúan varios cargadores de nivel de energía y configuraciones de infraestructura en función de la cantidad de energía, el tiempo de carga y la ubicación, el costo, el equipo y otros factores.

En 2014 se presenta un cargador de batería de vehículo aislado a bordo que utiliza dispositivos de potencia de carburo de silicio (SiC) para lograr una alta densidad y alta eficiencia para su aplicación en vehículos eléctricos (EV) y EV híbridos enchufables (PHEV) [13]. El cargador de nivel 2 propuesto tiene una arquitectura de dos etapas en la que la primera etapa es un convertidor de CA dc reforzado sin puente y la segunda etapa es un convertidor CC-CC aislado de puente completo con desplazamiento de fase. Se presenta el funcionamiento de ambas topologías y se discuten las ventajas específicas obtenidas mediante el uso de dispositivos de potencia de SiC. El diseño de los componentes de la etapa de potencia, el empaquetado del módulo de potencia multichip y el empaquetado a nivel del sistema se presentan con un enfoque principal en la densidad del sistema y un enfoque secundario en la eficiencia del sistema. En este trabajo, se desarrolla un prototipo de hardware y se mide una eficiencia máxima del sistema del 95% mientras se operan ambas etapas de potencia con una frecuencia de conmutación de 200 kHz. Una potencia de salida máxima de 6,1 kW da como resultado una densidad de potencia volumétrica de 5,0 kW / L y una densidad de potencia gravimétrica de 3,8 kW / kg cuando se considera el volumen y la masa del sistema.

Hannan revisa la estimación del estado de carga (SOC) de la batería de iones de litio y su sistema de gestión para las futuras aplicaciones de vehículos eléctricos sostenibles [14]. Se presenta la importancia del sistema de gestión de baterías (BMS) que emplea baterías de iones de litio, que pueden garantizar un funcionamiento confiable y seguro y evaluar el SOC de la batería. La revisión identifica que el SOC es un parámetro crucial, ya que significa la

energía disponible restante en una batería que proporciona una idea sobre las estrategias de carga / descarga y protege la batería de la sobrecarga / descarga excesiva. También se observa que el SOC de las baterías de iones de litio existentes tiene una buena contribución para hacer funcionar los vehículos eléctricos de manera segura y eficiente con sus capacidades de carga / descarga. Sin embargo, todavía tienen algunos desafíos debido a sus complejas reacciones electroquímicas, degradación del rendimiento y falta de precisión para mejorar el rendimiento y la vida útil de la batería. Se revisa extensamente la clasificación de las metodologías de estimación para estimar el COS enfocándose con el modelo / algoritmo de estimación, los beneficios, los inconvenientes y el error de estimación.

Una de las técnicas que ayudan a reducir las pérdidas es la conmutación a cero voltaje (ZVS). Prasetya logra la ZVS cuando se suma la inductancia de fuga de un transformador de $14.12 \mu\text{H}$ con una inductancia externa de $24.29 \mu\text{H}$ que se instala en serie con el transformador cuando la corriente primaria del transformador es superior a 1.289 A [15].

En 2019 se presenta un análisis comparativo sobre diversas topologías de convertidores CC-CC aislados y no aislados [16]. Aquí, el enfoque principal permanece en los convertidores CC-CC sin transformador (TL), basados en el convertidor elevador básico convencional. Además, para lograr una alta ganancia de voltaje, se ha presentado una clasificación de convertidores no aislados basada en diseño extensible y no extensible. Para el análisis comparativo y teórico, los parámetros elegidos son el número de componentes utilizados por cada topología de convertidor, la alta ganancia de voltaje ofrecida, las tensiones de voltaje en cada componente involucrado y la eficiencia de las topologías de alta ganancia. Para los convertidores en discusión, también se ha destacado el funcionamiento en condiciones ideales y no ideales.

Khaligh en 2019 ofrece una revisión y un análisis exhaustivos del estado de la técnica y las tendencias futuras de los cargadores a bordo (OBC) de alta potencia para vehículos eléctricos [17]. Para proporcionar un contexto global, se presenta un resumen de los estándares de carga globales y las tendencias relacionadas con los vehículos eléctricos (EV), lo que demuestra el impulso hacia los OBC con mayor potencia nominal. Los OBC de alta potencia

son unidireccionales o bidireccionales, y tienen una arquitectura de sistema integrada o no integrada. Los OBC de alta potencia no integrados se estudian tanto en la industria como en el mundo académico, y los primeros se utilizan para ilustrar el estado actual de la técnica. Estos últimos se clasifican sobre la base del enfoque de diseño del convertidor, se estudian por su principio de funcionamiento y se comparan en función de la densidad de potencia, el peso, la eficiencia y otras métricas. Además de los OBC no integrados, también se presentan los avances recientes en las soluciones OBC integradas en máquinas de propulsión. También se analizan otras técnicas OBC integradas, como la integración del sistema con el módulo de potencia auxiliar del EV y los sistemas de carga inalámbrica. Por último, se abordan las estrategias y funcionalidades de carga futuras en las infraestructuras de carga y se resumen las tendencias globales de OBC.

La distorsión por armónicos es un problema que afecta la eficiencia del cargador. Se presenta el modelo de una estación de carga rápida de vehículos eléctricos conectada a la red que garantiza una transferencia de energía de calidad con corrientes armónicas reducidas [18]. La estación de carga consiste en un convertidor que conecta la red a un bus de CC donde los vehículos eléctricos se conectan a través de cargadores de batería. El control del proceso de carga del vehículo individual está descentralizado y se proporciona un control separado para manejar la transferencia de energía de la red de CA al bus de CC. También se propone una estrategia de gestión energética basada en un flujo de energía óptimo mediante la integración de un sistema de generación solar fotovoltaica con una estación de carga para aliviar el impacto de la carga rápida en la red. El sistema combinado junto con la potencia de salida de las baterías de la flota de vehículos eléctricos disponibles en la estación de carga reduce la energía neta proporcionada por la red, lo que reduce la carga total en la red y minimiza las pérdidas de conversión.

Ahn describe una estrategia de carga de la batería para reducir las pérdidas de carga en una batería de iones de litio para vehículos eléctricos [19]. La estrategia de carga propuesta utiliza un perfil de corriente adaptativo basado en variaciones de la resistencia interna de la batería en función del estado de carga y la tasa de carga. Para abordar el problema de

encontrar el conjunto de corrientes óptimo para la estrategia propuesta, se aplica un algoritmo evolutivo, que es un tipo de enfoque estocástico. También se presenta una estrategia para seleccionar el número óptimo de intervalos de carga para alcanzar un compromiso entre la reducción de pérdidas y la carga computacional. Los resultados experimentales obtenidos con celdas de batería de 34 Ah de litio-níquel-manganeso-óxido de cobalto han demostrado que la estrategia de carga propuesta reduce la pérdida de carga de la celda de la batería en un 40,1 % en comparación con una estrategia de carga convencional de corriente constante. Además, se ha construido un prototipo de cargador de a bordo de 3,3 kW para investigar la reducción total de pérdidas en un sistema de carga de vehículos eléctricos que incluye una batería de 12 kWh. La estrategia de carga de corriente adaptativa propuesta reduce las pérdidas totales de carga, incluida la pérdida de batería y la pérdida del cargador de vehículos eléctricos en un 7,2 %, 11,2 % y 21,2 % en sistemas de carga con potencias nominales de 3,3 kW, 6,6 kW y 13,2 kW, respectivamente.

Se necesita hacer una revisión de la infraestructura de carga de vehículos eléctricos de última generación enfocados en la tecnología emergente de carga rápida extrema(XFC), que será necesaria para respaldar las necesidades actuales y futuras de reabastecimiento de combustible de vehículos eléctricos. Se presentan las consideraciones de diseño de las estaciones XFC y revisan las topologías típicas de convertidores de electrónica de potencia adecuadas para entregar XFC [20]. Se consideran los beneficios de usar los transformadores de estado sólido (SST) en las estaciones XFC para reemplazar los transformadores de frecuencia de línea convencionales y además proporcionan una revisión completa de los diseños de SST de voltaje medio para la aplicación XFC.

En 2020 se presenta una revisión crítica de la literatura sobre las topologías de conversión de energía de los vehículos eléctricos y los sistemas de almacenamiento de energía con desafíos, oportunidades y direcciones futuras mediante la clasificación sistemática de los vehículos eléctricos y el almacenamiento de energía [21]. El documento contempla la necesidad de fuentes que tengan alta potencia y densidad de energía para disminuir el tiempo de carga.

Kumar propone un convertidor con menos elementos de conmutación a diferencia de los

convencionales. El convertidor presentado actúa como un convertidor CC-CC frontal para el inversor CC-CA, y suministra energía a las cargas CC auxiliares o para cargar la batería auxiliar en EV [22]. El convertidor tiene una celda multiplicadora de voltaje de una etapa con un circuito de pinza de voltaje para mejorar la ganancia de voltaje con conmutación suave para reducir la pérdida y, por lo tanto, la eficiencia. El convertidor propuesto se opera con un solo interruptor MOSFET que entrega tres niveles de salida diferentes con menos tensión de voltaje a través del interruptor, es la característica principal del convertidor propuesto. El análisis teórico se valida a través de los resultados experimentales, y los resultados verifican que el convertidor propuesto es el más adecuado para aplicaciones de vehículos eléctricos.

Choudhury lleva a cabo una revisión completa de las topologías de convertidores CC-CC de alta ganancia existentes (tecnología de inductores en cascada, intercalados y acoplados) [23]. Consiste en el estudio cuantitativo y cualitativo de todos los convertidores revisados. Además, el método de selección de convertidores para aplicaciones basadas en fotovoltaica (PV) también se logra teniendo en cuenta el concepto de relación de trabajo crítica y ganancia de voltaje práctica. Dado que la relación de servicio crítico de cualquier convertidor CC-CC depende de la resistencia en serie equivalente (ESR) de los componentes del circuito, los métodos disponibles para determinar los valores de ESR también se describen en este artículo. Una vez más, el estudio de viabilidad de dichas topologías de convertidor revisadas se ejecuta teniendo en cuenta los campos de aplicación fotovoltaica, como el sistema de almacenamiento de energía (ESS), el suministro residencial, la carga de vehículos eléctricos (EV). Por último, en este artículo se resume la revisión de los estándares disponibles en relación con el sistema de generación distribuida de CC.

Finalmente, se consideran los diferentes métodos de control para la carga de las baterías. Se propone un cargador de batería (BC) para vehículos eléctricos basado en el método de corriente ondulada sinusoidal (SRC) [24]. El método SRC se utiliza como un método avanzado de carga y descarga. Se sugiere una topología particular y luego se diseña con precisión un sistema de control para implementar el método de carga-descarga SRC y ajustar las potencias activas y reactivas de salida para la tecnología de vehículo a red sin el problema de

calidad de la energía. El método SRC también se puede utilizar en espectroscopía de impedancia electroquímica (EIS) en línea para estimar el estado de carga y el estado de salud en la gestión de baterías. El sistema de control propuesto también puede generar las señales SRC, adecuadas para EIS en línea, de forma controlada a diferencia de los BC recientemente propuestos. La simulación y los resultados experimentales demuestran la eficacia de la solución propuesta.

Metodología

3.1. Fundamentación teórica

3.1.1. Filtro EMI (interferencia electromagnética)

El filtro EMI o filtro de interferencia electromagnética, es un circuito eléctrico que mitiga el ruido electromagnético de alta frecuencia presente en las líneas de alimentación y señal. El ruido de alta frecuencia es generado por una variedad de dispositivos eléctricos y electrónicos tales como motores, controles electrónicos, fuentes de alimentación, inversores, circuitos de reloj, microprocesadores, electrodomésticos, dispositivos electrónicos, etc. Este ruido está típicamente en el rango de frecuencia de 9KHz a 10GHz y puede degradar o prevenir las transmisiones de señal o el rendimiento previsto de un equipo electrónico. Los componentes de baja frecuencia del ruido electromagnético también pueden afectar la calidad de la energía. Los filtros industriales EMI proporcionan una ruta de baja impedancia al ruido de alta frecuencia y lo reducen cancelando su línea y componentes neutros o conectándolo a tierra. La efectividad de un filtro EMI se mide como pérdida de inserción (en dB sobre el rango de frecuencia). Los filtros EMI se usan ampliamente en aplicaciones como electrodomésticos, sistemas y subsistemas militares y aeroespaciales, sistemas de gestión de energía, computadoras, equipos de automatización de fábricas, equipos industriales, dispo-

sitivos de diagnóstico médico, equipos de ejercicio, cargador de baterías automotrices, salas de resonancia magnética, entre otros. El filtro propuesto es un filtro en modo común, debido a que éste suprime principalmente el ruido electromagnético y las señales de desorden de la fuente de alimentación de entrada para evitar interferencias en la fuente de alimentación y también evita el desorden de alta frecuencia generado por la fuente de alimentación en sí de interferir con la red eléctrica. Está compuesto de 2 inductores representados por L1 y L2, y 2 capacitores representados por C1 y C2. El filtro se muestra en la Figura 3.1.

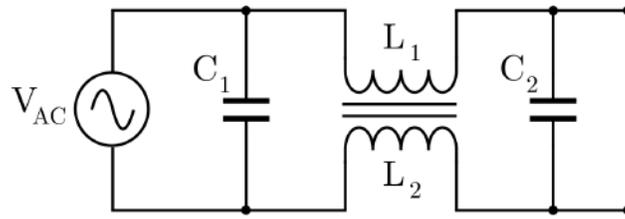


Figura 3.1: Filtro EMI en modo común.

El valor de importancia para diseñar el filtro es la frecuencia de corte a la que trabaje. Para calcular esto se utiliza la fórmula:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \quad (3.1)$$

Para esta aplicación se propone un valor de inductancia y de frecuencia de corte para después calcular el valor de la capacitancia. Con un despeje se obtiene la fórmula de la capacitancia:

$$C = \frac{1}{4\pi^2 f_c^2 L} \quad (3.2)$$

3.1.2. Rectificación

Para este proyecto, la entrada del sistema estará dada en forma de corriente alterna de 60 Hz proporcionada por el sistema de distribución. Esta corriente alterna debe ser transformada en corriente continua y para ello se utilizan los rectificadores. Los rectificadores se caracterizan por transformar la corriente alterna en continua. De esta manera permiten la

conversión directa desde un circuito alimentado con voltaje alterno, poder alimentar a la carga con corriente continua. Los rectificadores no controlados están formados exclusivamente por diodos, no necesitando circuitos de mando, por lo que los diodos conmutan de manera natural forzados por la fuente de alimentación.

Rectificación de onda completa

El rectificador de onda completa es un circuito formado por un arreglo de cuatro diodos llamado puente de diodos, el cual se muestra en la Figura 3.2. Éste rectificador es el utilizado en la conversión de la línea alterna para alimentar el sistema.

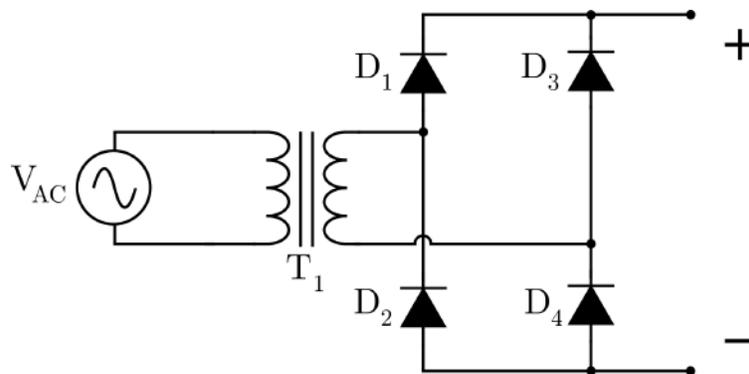


Figura 3.2: Rectificador de onda completa conformado por cuatro diodos.

Supóngase como entrada al puente de diodos una señal sinusoidal con amplitud de 127 volts y frecuencia de 60 Hz. Al pasar por el arreglo de diodos, la parte positiva de la señal sinusoidal se mantendrá sin alteraciones, pero la parte negativa será invertida, teniendo como resultado la señal rectificada.

Al momento de realizar la rectificación, hay que tener en cuenta dos valores: voltaje máximo y voltaje eficaz, también conocido como voltaje RMS (Root Mean Square). El voltaje máximo es es voltaje real que se tiene de la línea de alimentación, mientras que el voltaje eficaz es cantidad de corriente alterna que produce el mismo efecto de calentamiento que una corriente continua equivalente. La relación de estos dos valores se muestra a continuación:

$$V_{max} = V_{rms} \sqrt{2} \quad (3.3)$$

El voltaje eficaz en México es generalmente de 127 volts, por lo que el voltaje máximo llega a ser de 180 volts.

3.1.3. Convertidor CD-CD

Los convertidores CD-CD se utilizan ampliamente para regular voltaje de manera eficiente. Son circuitos de potencia que trabajan a altas frecuencias. Están compuestos de capacitores, inductores, interruptores y en ocasiones de transformadores. Generalmente son mucho más eficientes y de menor tamaño que los reguladores lineales, sin embargo el circuito de control llega a ser más complejo.

Los convertidores CD-CD pueden ser aislados o no aislados. Los convertidores aislados se derivan de convertidores no aislados mediante la incorporación de transformadores y circuitos rectificadores de salida. Los convertidores CD-CD básicos no aislados más utilizados son el convertidor Buck, Boost, Buck-Boost, Cuk, Zeta y SEPIC. Los convertidores aislados derivados del convertidor Buck incluyen el convertidor Forward, Push-Pull, medio puente y puente completo. Los convertidores Forward pueden contar con un interruptor único o interruptor doble.

Los interruptores del convertidor Forward con interruptor único y el convertidor Push-Pull tienen que soportar el doble del voltaje de entrada, mientras que los interruptores del convertidor Forward de interruptor doble, el convertidor de medio puente y el convertidor de puente completo deben soportar la tensión de entrada. Es por esto que el convertidor Forward de interruptor doble, el convertidor de medio puente y el convertidor de puente completo son utilizados cuando se trabajan altos voltajes.

En un convertidor aislado se utiliza un transformador entre los interruptores y el rectificador. Se utiliza un circuito encargado de rectificar la señal que entrega el segundo devanado del transformador. Los circuitos más utilizados son: rectificador de media onda, rectificador de onda completa, rectificador de puente completo y rectificador doblador de corriente.

La ventaja de utilizar un rectificador de puente completo, es que en el devanado secundario del transformador no se necesita de una derivación central. El voltaje de salida del

rectificador de puente completo es:

$$V_O = \frac{2DV_P}{K} \quad (3.4)$$

donde V_P es el voltaje del devanado primario, D es el ciclo de trabajo y K es la relación de vueltas entre el devanado primario y secundario.

El voltaje que soportan los diodos rectificadores es V_P/K y la frecuencia de rizo tanto del voltaje rectificado como de la corriente del inductor de filtrado es $2f_s$.

Convertidor de puente completo

El convertidor utilizado es el convertidor de puente completo, debido a que se va a trabajar con voltajes y corrientes altas y se busca reducir el voltaje con el que se alimentará la batería. Éste se muestra en la Figura 3.3.

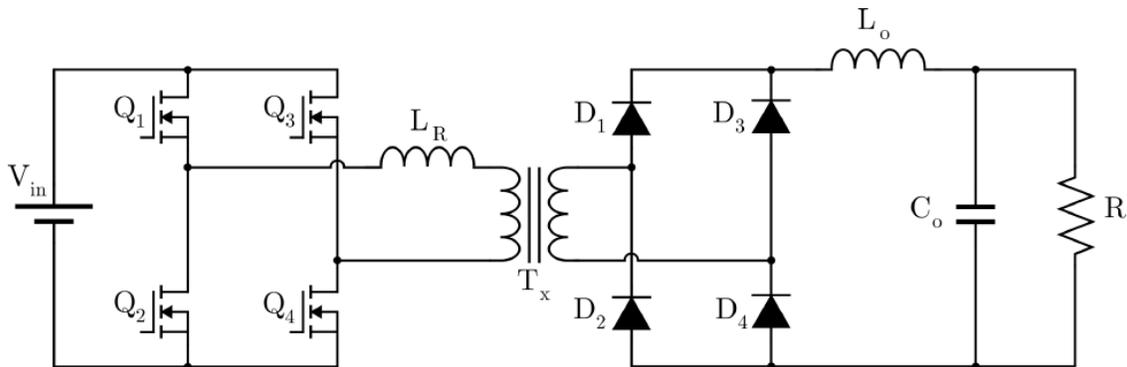


Figura 3.3: Convertidor CC-CC de puente completo.

El voltaje de salida se obtiene de la siguiente ecuación:

$$V_{out} = 2 \times V_{in} \frac{N_2}{N_1} \times D \quad (3.5)$$

donde D es el ciclo de trabajo, el cual debe permanecer entre un rango de $0 < D < 0.5$ para la mayor potencia transferida.

Desplazamiento de fase

Los convertidores CC-CC de puente completo con desplazamiento de fase (PSFB) se utilizan con frecuencia para reducir los altos voltajes de bus de CC y/o proporcionar aislamiento en aplicaciones de potencia media a alta, como fuentes de alimentación de servidor, rectificadores de telecomunicaciones, sistemas de carga de baterías y sistemas de energía renovable. Tradicionalmente, los microcontroladores se han limitado a realizar únicamente tareas de supervisión o comunicaciones en estos sistemas. Con la disponibilidad de dispositivos de microcontrolador de alto rendimiento, ahora es posible utilizar microcontroladores para cerrar lazos de control en estos sistemas, además de manejar las funciones tradicionales de microcontrolador. La transición al control de potencia digital significa que las funciones que antes se implementaban en hardware ahora se implementan en software. Además de la flexibilidad que esto agrega al sistema, esto lo simplifica considerablemente. Estos sistemas pueden implementar estrategias de control avanzadas para controlar de manera óptima la etapa de potencia en diferentes condiciones y también proporcionar inteligencia a nivel del sistema. Un convertidor PSFB consta de cuatro interruptores electrónicos de potencia (como MOSFET o IGBT) que forman un puente completo en el lado primario del transformador de aislamiento y rectificadores de diodos o interruptores MOSFET para rectificación síncrona (SR) en el lado secundario. Esta topología permite que todos los dispositivos de conmutación conmuten con conmutación de voltaje cero (ZVS), lo que resulta en menores pérdidas de conmutación y un convertidor eficiente. Para tal ello, se requiere una rectificación de señal en el lado secundario. Para sistemas con voltaje de salida bajo y / o clasificaciones de corriente de salida altas, la implementación de la rectificación síncrona en lugar de la rectificación de diodos logra el mejor rendimiento posible al evitar las pérdidas de rectificación de diodos.

El convertidor mostrado en la Figura [3.3](#) puede controlarse con un desplazamiento de fase. Los interruptores MOSFET Q_1 , Q_2 , Q_3 y Q_4 forman el puente completo en el lado primario del transformador T_x . Q_1 y Q_2 tienen el 50 % del ciclo de trabajo y están 180 grados desfasados entre sí. De manera similar, Q_3 y Q_4 se encuentran al 50 % de su ciclo de trabajo y 180 grados desfasados entre sí. Las señales de conmutación PWM para el tramo Q_3 -

Q_4 del puente completo están desfasadas con respecto a las del tramo Q_1-Q_2 . La cantidad de este desfase determina la cantidad de superposición entre los interruptores diagonales, que a su vez decide la cantidad de energía transferida. D_1 , D_2 , D_3 y D_4 proporcionan una rectificación en el devanado secundario, mientras que L_O y C_O forman el filtro de salida.

3.1.4. Modelado de la Planta

La identificación de procesos dinámicos puede basarse en diferentes modelos que describen diferentes entornos estocásticos. Los modelos de error de ecuación describen una categoría muy útil de modelos debido a su amplia aplicabilidad en la predicción y el control.

El modelo ARX (autoregressive exogenous) constituye una forma sencilla de representar un proceso dinámico impulsado por una entrada y ruido. Este modelo describe la salida observada del proceso como la suma de una regresión en observaciones anteriores de entrada y salida y de un proceso que describe el error de ecuación.

Estructura del modelo ARX

En la Figura [3.4](#) se muestra un sistema con adición de ruido.

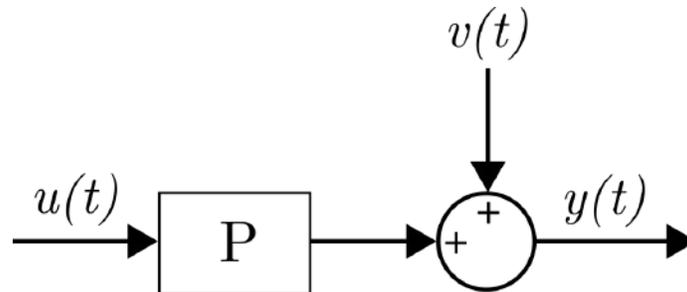


Figura 3.4: Planta con entrada $u(t)$ y salida $y(t)$ con adición de ruido $v(t)$.

En este modelo se conoce tanto la entrada, como la salida, pero el ruido en la práctica es desconocido y puede presentarse de diversas maneras. El considerar este elemento permite que el análisis sea más cercano a la realidad.

El modelo lineal del sistema se escribe de la siguiente manera:

$$y(k) = \sum_{j=1}^n a_j y(k-j) + \sum_{j=0}^n b_j u(k-j) + e(k) \quad (3.6)$$

El ruido del modelo, al que llamaremos $e(k)$ será un ruido blanco, es decir, que la media de sus valores será igual a cero.

$$E\{e(k)\} = 0 \quad (3.7)$$

En cuanto a la varianza, ésta se puede expresar de la siguiente manera:

$$E\{e^2(k)\} = \sigma^2 \quad (3.8)$$

Ya que se definieron las propiedades estadísticas de $e(k)$, podemos reescribir el diagrama de bloques de la Figura [3.4](#) como:

$$y(k) = \hat{y}(k) + e(k) \quad (3.9)$$

y partiendo del modelo lineal del sistema, se puede observar que:

$$\hat{y}(k) = \sum_{j=1}^n a_j y(k-j) + \sum_{j=0}^n b_j u(k-j) \quad (3.10)$$

Si a estas sumatorias se les aplica la transformada Z, pueden expresarse de la siguiente manera:

$$A(z) = a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n} \quad (3.11)$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_n z^{-n} \quad (3.12)$$

Considerando entonces los términos anteriores y el modelo ARX, éste se puede escribir

de la siguiente manera:

$$Y(z) = \frac{B(z)}{1 - A(z)}U(z) + V(z) \quad (3.13)$$

$V(z)$ representa el ruido blanco que es introducido a la salida de la planta, el cual puede expresarse como:

$$V(z) = \frac{1}{1 - A(z)}E(z) \quad (3.14)$$

Si a (3.14) se le aplica la transformada Z inversa y se considera a (3.11), $v(k)$ puede expresarse como:

$$v(k) = \sum_{i=1}^n a_i v(k - i) + e(k) \quad (3.15)$$

Para que el modelo ARX pueda ser implementado en los algoritmos recursivos de identificación, se le tiene que dar una forma distinta. A las sumatorias de la ecuación (3.6) se les va a reescribir como vectores. Ahora se separarán los parámetros y las mediciones. El vector de parámetros quedará:

$$P^T = [b_0, a_1, b_1, \dots, a_n, b_n] \quad (3.16)$$

y el vector de mediciones será:

$$z^T(k) = [u(k), y(k - 1), u(k - 1), \dots, y(k - n), u(k - n)] \quad (3.17)$$

Tanto P^T como $z^T(k)$ tendrán una dimensión de $1 \times (2n + 1)$. Estos dos vectores se sustituyen en (3.6) para quedar de la siguiente manera:

$$y(k) = P^T z(k) + e(k) \quad (3.18)$$

Este modelo puede modificarse para que el valor k ahora sea el número de observaciones t con respecto a las entradas y salidas. Al hacer el cambio, queda:

$$Y^T(t) = P^T Z^T(t) + E^T(t) \quad (3.19)$$

Esto significa que ahora se considerarán sistemas multivariados, por lo que $Y^T(t)$ será:

$$Y^T(t) = [y(1), y(2), \dots, y(t)] \quad (3.20)$$

El error tendrá un cambio parecido:

$$E^T(t) = [e(0), e(1), \dots, e(t)] \quad (3.21)$$

Finalmente, $Z^T(t)$ será ahora una matriz de dimensión $(2n + 1) \times t$ y se verá de la siguiente manera:

$$Z^T(t) = \begin{bmatrix} u(1) & u(2) & \dots & u(t) \\ y(0) & y(1) & \dots & y(t-1) \\ u(0) & u(1) & \dots & u(t-1) \\ \vdots & & & \vdots \\ y(1-n) & y(2-n) & \dots & y(t-n) \\ u(1-n) & u(2-n) & \dots & u(t-n) \end{bmatrix} \quad (3.22)$$

Con ésta forma del modelo ARX, se puede comenzar a implementar en un algoritmo recursivo de identificación.

Método de mínimos cuadrados para la estimación de parámetros del modelo ARX

Como se vió en la estructura del modelo ARX, se cuenta con una perturbación en forma de ruido blanco, por lo que la salida $Y^T(t)$ en (3.19) tendrá propiedades estadísticas parecidas al error en donde la media de la salida será:

$$E\{Y^T(t)\} = P^T Z^T(t) \quad (3.23)$$

y la covarianza de la salida será:

$$cov\{Y(t)\} = cov\{E(t)\} = \sigma^2 I \quad (3.24)$$

donde I está definida por la matriz identidad.

Ahora se busca minimizar la forma cuadrática dependiendo de la estimación de los parámetros contenidos en P . Esto se realiza con el método de mínimos cuadrados.

$$J = (Y(t) - Z(t)P(t))^T W (Y(t) - Z(t)P(t)) \quad (3.25)$$

En esta ecuación, W será llamada matriz de peso. Ésta matriz puede definirse de distintas maneras, sin embargo, para este caso, se considerará a $W = I$. Con ésta consideración, (3.25) quedará como:

$$J = Y^T(t)Y(t) - 2Y^T(t)Z(t)P(t) + P^T(t)Z^T(t)Z(t)P(t) \quad (3.26)$$

Ahora lo que tenemos que hacer es derivar con respecto a $P(t)$ para encontrar el estimador mínimo cuadrático y así aplicando el método de mínimos cuadrados. La derivada queda:

$$\frac{\partial J}{\partial P(t)} = -2Z^T(t)Y(t) + 2Z^T(t)Z(t)P(t) = 0 \quad (3.27)$$

Despejando para obtener el estimador mínimo cuadrático:

$$\hat{P}^T(t) = (Z^T(t)Z(t))^{-1}Z^T(t)Y(t) \quad (3.28)$$

3.1.5. Sistema de control

Un sistema de control puede ser representado gráficamente por un diagrama de bloques, tales diagramas de bloques indican la interrelación existente entre los distintos componentes del sistema. En un diagrama de bloques, todas las variables del sistema se enlazan entre sí por medio de bloques funcionales. El bloque funcional, o simplemente bloque, es un símbolo de la operación matemática que el sistema produce a la salida sobre la señal de entrada. Una flecha hacia adentro del bloque indica la entrada y la que se aleja del bloque indica la salida. Debe notarse que la magnitud de la señal de salida del bloque será la señal de

entrada multiplicada por la función de transferencia del bloque. Existen tres principales tipos de controladores: Controladores PI, PD y PID. En este caso se utiliza el controlador PI para el control de la corriente. La potencia de salida del controlador es igual a la suma de la proporción y los coeficientes de integración. Cuanto mayor sea el coeficiente de proporción, menor será la potencia de salida en el mismo error de control. Cuanto mayor sea el coeficiente de integración, más lento será el coeficiente de integración acumulado.

Control PI

El control PI proporciona un error de control cero y es insensible a la interferencia del canal de medición. La desventaja del control de PI es la reacción lenta a las perturbaciones. Para ajustar el controlador PI, primero debe establecer el tiempo de integración igual a cero y el tiempo de proporción máxima. Luego, al disminuir el coeficiente de proporcionalidad, lograr oscilaciones periódicas en el sistema. Cerca del valor óptimo del coeficiente de proporcionalidad es dos veces más alto que aquel en el que cualquier vacilación, y cerca del valor óptimo de la constante de tiempo de integración, es un 20% menos que el período de oscilación. Su función de transferencia es la siguiente:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_N s} \right) \quad (3.29)$$

Donde K_p es la ganancia proporcional y T_N se denomina tiempo de acción integral. Ambos valores son ajustables. El tiempo integral regula la velocidad de acción de control, mientras que una modificación en K_p afecta tanto a la parte integral como a la parte proporcional de la acción de control.

3.1.6. Cargadores eléctricos comerciales

Es muy importante conocer los tipos de cargadores para coches eléctricos que existen en el mercado, ya que todo aquel que se posea un vehículo eléctrico tendrá diferentes necesidades. Para comprender un poco mejor los automóviles eléctricos hay que conocer qué tipos de carga tienen los cargadores para coches eléctricos, puesto que según su velocidad varía el

tiempo en recargar las baterías, que depende directamente de la potencia disponible. Los cargadores para vehículos eléctricos se clasifican en tres niveles, en donde cada nivel tendrá un nivel de carga más rápido que el anterior. En la Tabla 3.1 se muestra el voltaje que utiliza, la corriente que consume y la potencia que suministra a la batería del coche eléctrico.

Tabla 3.1: Niveles de cargadores comerciales.

Nivel	Voltaje(V)	Corriente(A)	Potencia(kW)	Tiempo de carga(horas)
1	120	12-15	1-2	8-12
2	120-240	15-80	3.5-20	4-6
3	480	40-200	20-100	0.5

3.2. Implementación

3.2.1. Diagrama de bloques

El diagrama de bloques del sistema se muestra en la Figura 3.5, en donde se realiza una breve descripción de cada bloque.

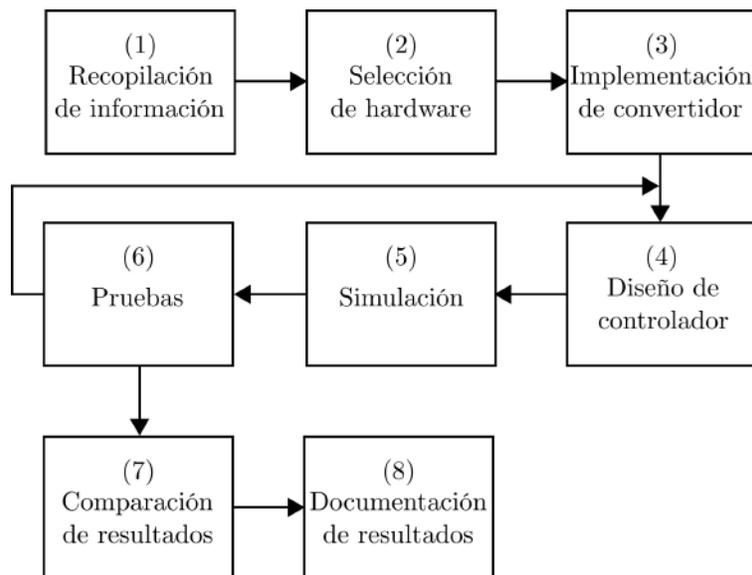


Figura 3.5: Diagrama de bloques de actividades.

1. Se buscan antecedentes relacionados con el tema de carga de celdas de litio en diversas fuentes de investigación científica y en patentes. También se analizan los posibles métodos de control que puedan ser implementados en el sistema.
2. Seleccionar los componentes electrónicos utilizados en el convertidor CD-CD que sean capaces de operar en voltajes mayores a 180 Volts y corriente de hasta 20 Amperes. Se selecciona el microcontrolador que controlará el ancho de pulso del convertidor.
3. Se conecta físicamente el convertidor CD-CD y se evalúa su eficiencia dependiendo del ciclo de trabajo al que opere.
4. Se diseña un controlador con doble lazo de control, en donde un lazo se encarga de controlar el voltaje y otro de controlar corriente.
5. Modelar el sistema para simular el comportamiento del controlador y verificar que el control se realiza de manera correcta.
6. Realizar las pruebas del sistema para evaluar los resultados y comprobar su desempeño con cargadores de autos eléctricos comerciales.
7. Comparar los resultados de la simulación con los experimentales y asegurarse de que sean lo más parecido entre ellos.
8. Documentar los resultados obtenidos y describir con detalle cada paso realizado en la investigación.

En esta sección se observa el proceso que sigue el sistema desde la corriente alterna hasta la batería, así como los componentes electrónicos que se van a utilizar. El sistema mostrado en la Figura [3.6](#) se puede visualizar en dos secciones: una se encarga de suministrar el voltaje y corriente necesaria y la otra del control de las mismas. Existen varias opciones que pueden controlar este tipo de sistemas, pero este trabajo hará uso de un microcontrolador dsPIC y un microcontrolador STM32 Nucleo debido a que éstos cuentan con opciones para el control por modulación de ancho de pulso, desfase y alta frecuencia.

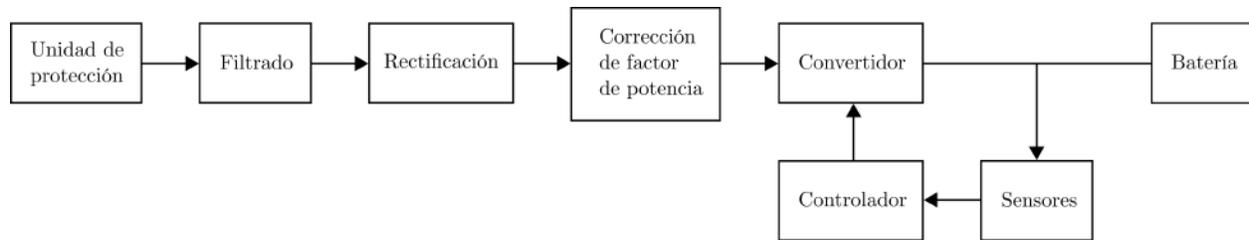


Figura 3.6: Diagrama de bloques de funcionamiento de cargador.

3.3. Diseño

3.3.1. Unidad de protección

En la unidad de protección se considera únicamente un fusible, el cual permitirá que se deje de suministrar energía al circuito si la corriente de demanda excede a la tolerada por el cargador. La corriente de interrupción seleccionada del fusible es de 20 Amperes.

3.3.2. Filtrado

El filtro de interferencia electromagnética es del tipo pasa bajas, el cual rechaza la energía de RF no deseada por encima de una frecuencia de corte deseada, pasando las frecuencias por debajo de este punto con poca o ninguna pérdida de inserción. En este caso la frecuencia de corte es de 4 kHz y el filtro pasa bajas es operado en modo común.

Proponiendo una inductancia de 28 mH, una frecuencia de corte de 4 kHz y despejando la ecuación para la capacitancia, se obtiene lo siguiente:

$$C = \frac{1}{4\pi^2 f_c^2 L} = \frac{1}{4\pi^2 (4kHz)^2 (28mH)} = 56nF \quad (3.30)$$

3.3.3. Transformador

Lo primero es definir los parámetros de la fuente de alimentación para el diseño del transformador. En este caso se tienen los siguientes valores:

$$V_{INmin} = 370 \text{ V}$$

$$V_{INmax} = 380 \text{ V}$$

$$V_O = 180 \text{ V}$$

$$I_O = 20 \text{ A}$$

$$f_S = 100 \text{ kHz}$$

$$T_S = 10 \text{ } \mu\text{s}$$

$$W_{loss} = 0.5 \text{ W}$$

$$T_{max} = 40^\circ\text{C}$$

También se define el límite absoluto del ciclo de trabajo D_{lim} , el ciclo de trabajo normal D_{max} y en base a esto se calcula V_{INmin} y $V_{IN} \cdot D$:

$$D_{lim} = 0.47$$

$$D_{max} = 0.42$$

$$V_{IN} \cdot D = V_{INmin} \cdot D_{max} = 155,4V \quad (3.31)$$

$$V_{INmax} \cdot D_{lim} = 178,6V \quad (3.32)$$

Calculando los voltajes de salida más las caídas de diodos y la caída de voltaje en el devanado secundario con carga máxima se obtiene lo siguiente:

$$V'_O = 180V + 0,8V = 180,8V \quad (3.33)$$

Ahora se calcula la relación de vueltas deseada:

$$n = \frac{N_P}{N_S} = \frac{V_{IN} \cdot D}{V'_O} = \frac{155,4V}{180,8V} = 0,859 \quad (3.34)$$

Las posibles relaciones son 43:50, 35:40, 4:5, 9:10, 21:25.

Se selecciona el material del núcleo, la forma y el tamaño tentativo, utilizando la hoja de datos del fabricante o usando la fórmula del producto de área.

Material de núcleo = Ferrita

Tipo de núcleo = ETD

Tamaño del núcleo = ETD49

Para el núcleo específico seleccionado, se toma el área efectiva del núcleo (A_e), volumen (V_e) y longitud de devanado en centímetros (l_e).

$$A_e = 2.11 \text{ cm}^2$$

$$V_e = 24 \text{ cm}^3$$

$$l_e = 11.4 \text{ cm}$$

También se toma el área de la ventana, ancho, altura y longitud media por vuelta y la resistencia térmica.

$$A_W = 2.73 \text{ cm}^2$$

$$b_W = 3.27 \text{ cm}$$

$$MLT = 8.5 \text{ cm}$$

$$R_{th} = 8 \text{ K/W}$$

Se calcula el límite de pérdida según el aumento máximo de temperatura.

$$P_{lim} = \frac{T_{max}}{R_{th}} = 5W \quad (3.35)$$

Aproximadamente la mitad de la pérdida se irá en el núcleo, mientras que la otra mitad de la pérdida en el embobinado.

$$P_{Clim} = 2.5 \text{ W}$$

$$P_{Wlim} = 2.5 \text{ W}$$

Se calcula la pérdida máxima del núcleo por cm^3 .

$$C_{loss} = \frac{P_{Clim}}{V_e} = 0,1042 \frac{W}{\text{cm}^3} \quad (3.36)$$

Usando este valor de pérdida de núcleo, se toma la curva de pérdida de núcleo para el material P seleccionado y se encuentra la densidad de flujo. Se duplica para obtener la

variación de densidad de flujo de pico pico limitada por pérdida, ΔB , con 0.1042 W/cm^3 y una frecuencia de 100kHz .

$$\Delta B = 2 \cdot 1100G = 2200G = 0,22T \quad (3.37)$$

Usando la ley de Faraday, se calcula el número de vueltas secundarias:

$$\int E_S dt = V'_O \cdot T_S \quad (3.38)$$

$$N_S = \frac{\int E_S dt}{\Delta \Phi} = \frac{V'_O \cdot T_S}{\Delta \Phi} \quad (3.39)$$

$$N_S = \frac{V'_O \cdot T_S}{\Delta B \cdot A_e} = \frac{(180,8) \cdot (10 \times 10^{-6})}{(0,22) \cdot (2,11 \times 10^{-4})} = 38,94 \approx 39 \quad (3.40)$$

Ahora se realiza la elección de las vueltas en el devanado primario. Una relación de vueltas mayor da como resultado una corriente pico más baja y más pérdida de cobre. De las posibilidades definidas anteriormente, las soluciones de prueba muestran que la mejor opción es 35:40.

Recalculando V_{IND} y la densidad del flujo en las peores condiciones de $V_{INmax} D_{lim}$:

$$V_{IND} = n \cdot V'_O = 0,875 \cdot 180,8 = 158,2V \quad (3.41)$$

$$\Delta B_{lim} = \frac{\Delta B \cdot V_{INmax}}{V_{IND}} = \frac{(0,22) \cdot (178,6)}{(158,2)} \quad (3.42)$$

Ahora se calcula el efecto pelicular para una frecuencia de 100kHz .

$$\delta = \sqrt{\frac{\rho}{\pi f \mu}} = \sqrt{\frac{2,3 \times 10^{-8}}{\pi \cdot (100 \times 10^3) \cdot (4\pi \times 10^{-7})}} = 0,024\text{cm} \quad (3.43)$$

Se calculan las corrientes de CC y RMS de CA en cada devanado con V_{INmin} y D_{max} :

$$I_{SDC} = I_0 \cdot D_{max} = I_0 \cdot \frac{V_{IND}}{V_{INmin}} = (20) \cdot \frac{(158,2)}{(370)} = 8,55A \quad (3.44)$$

$$I_{SAC} = I_{SDC} \sqrt{\frac{1 - D_{max}}{D_{max}}} = (8,55) \sqrt{\frac{1 - 0,427}{0,427}} = 9,89A \quad (3.45)$$

$$I_{PDC} = \frac{I_{SDC}}{n} = \frac{8,55}{0,875} = 9,77A \quad (3.46)$$

$$I_{PAC} = \frac{I_{SAC}}{n} = \frac{9,89}{0,875} = 11,3A \quad (3.47)$$

Se define el devanado primario. Una capa de 35 vueltas repartidas en el ancho de devanado disponible de 3.27 cm permite un diámetro máximo de alambre de:

$$AWG_{Pmax} = \frac{b_W}{N_P} = \frac{3,27}{35} = 0,93mm \quad (3.48)$$

Se utilizará un alambre AWG 20 - 0.81 mm. Debido al efecto pelicular, se utiliza un hilo de Litz calibre AWG 20 con 50 alambres de 0.1 mm.

Se define el devanado secundario. El secundario consta de 40 vueltas, por lo que el diámetro máximo de alambre será:

$$AWG_{Smax} = \frac{b_W}{N_S} = \frac{3,27}{40} = 0,817mm \quad (3.49)$$

3.3.4. Convertidor DC-DC

El convertidor que se utiliza es el convertidor de puente completo mostrado en la Figura 3.7, debido a que puede entregar un voltaje y corriente más elevado en comparación a otros convertidores.

El voltaje de entrada es convertido a analógico por medio de los MOSFETs Q_1 , Q_2 , Q_3 y Q_4 . El propósito de esto es tener una señal de alta frecuencia, la cual puede ser reducida con

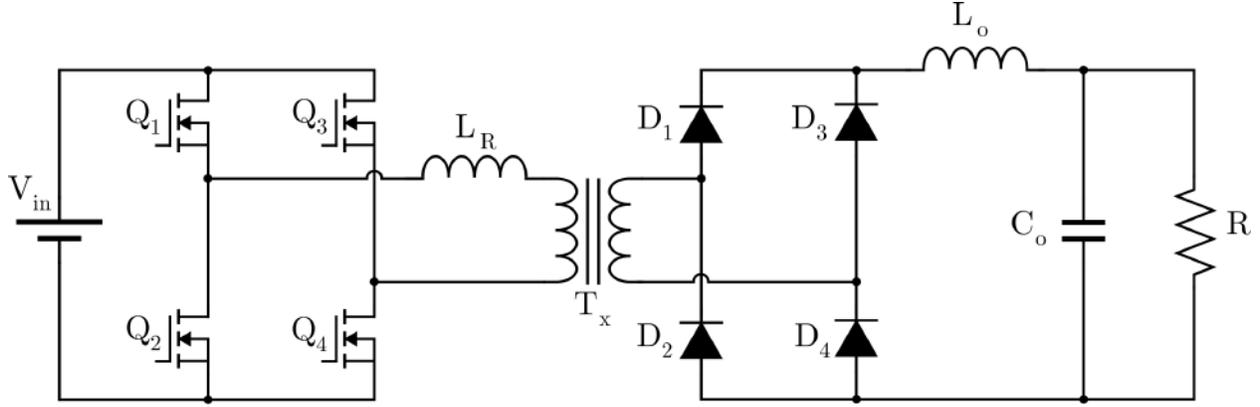


Figura 3.7: Convertidor de puente completo.

mayor facilidad por un transformador sin perder la corriente que se necesita. La frecuencia a la que los MOSFETs conmutan es a 100 kHz.

El transformador T_x es el encargado de aislar las dos secciones del convertidor. La señal alterna vuelve a convertirse a continua por medio de un puente de diodos. Esto permite que se tenga nuevamente una señal de corriente continua capaz de entregar un voltaje adecuado al paquete de celdas. D_1 , D_2 , D_3 y D_4 proporcionan una rectificación en el devanado secundario, mientras que L_o y C_o forman el filtro de salida.

L_o se calcula tomando en cuenta lo siguiente:

$$L_o = \frac{n \cdot V_{IN} - V_O}{\Delta I_{L_o}} \left(\frac{D \cdot T_S}{2} \right) = \frac{1,145(380) - 180}{5} \left(\frac{5 \times 10^{-6}}{2} \right) = 127,5 \mu H \quad (3.50)$$

El segundo elemento para el filtro de salida, el capacitor, es obtenido de la siguiente manera:

$$C_o = \frac{1}{(2\pi f_C)^2 L_o} = \frac{1}{(2\pi \cdot 10 \times 10^3)^2 (127,5 \times 10^{-6})} = 2 \mu F \quad (3.51)$$

3.3.5. Desplazamiento de Fase

Los convertidores CC-CC de puente completo con desplazamiento de fase (PSFB) se utilizan con frecuencia para reducir los altos voltajes de bus de CC y/o proporcionar aislamiento en aplicaciones de potencia media a alta, como fuentes de alimentación de servidor,

rectificadores de telecomunicaciones, sistemas de carga de baterías y sistemas de energía renovable.

El convertidor mostrado en la Figura 3.7 puede controlarse con un desplazamiento de fase. Los interruptores MOSFET Q_1 , Q_2 , Q_3 y Q_4 forman el puente completo en el lado primario del transformador T_x . Q_1 y Q_2 tienen el 50% del ciclo de trabajo y están 180 grados desfasados entre sí. De manera similar, Q_3 y Q_4 se encuentran al 50% de su ciclo de trabajo y 180 grados desfasados entre sí. Las señales de conmutación PWM para el tramo Q_3 - Q_4 del puente completo están desfasadas con respecto a las del tramo Q_1 - Q_2 . La cantidad de este desfase determina la cantidad de superposición entre los interruptores diagonales, que a su vez decide la cantidad de energía transferida.

Para el control del voltaje de salida se emplea un microcontrolador dsPIC33EV256GM102. Este se encarga de entregar una señal PWM a una frecuencia de 100kHz y con una resolución de 1400 escalones. Este mismo controlador cuenta con un módulo ADC, que es el encargado de medir el voltaje de salida y así tener un lazo cerrado, donde el voltaje de salida ajusta el desfase entre las señales PWM.

3.3.6. Control

Para esta planta se utiliza un controlador clásico, para ajustar el voltaje del convertidor. Al ser un control discreto, la función de transferencia del controlador puede expresarse de diferentes maneras en la parte integral y derivativa.

El término integral puede expresarse de las siguientes formas:

$$\frac{K_i T_S}{z - 1} \quad \text{Euler hacia adelante} \quad (3.52)$$

$$\frac{K_i T_S z}{z - 1} \quad \text{Euler hacia atrás} \quad (3.53)$$

$$\frac{K_i T_S}{2} \frac{z + 1}{z - 1} \quad \text{Trapezoidal} \quad (3.54)$$

El término derivativo también puede expresarse de distintas maneras:

$$\frac{N(z-1)}{1+NT_S} \quad \text{Euler hacia adelante} \quad (3.55)$$

$$\frac{N(z-1)}{1+NT_S z} \quad \text{Euler hacia atrás} \quad (3.56)$$

$$\frac{2N(z-1)}{1+NT_S(z+1)} \quad \text{Trapezoidal} \quad (3.57)$$

En ambos casos se considera la forma de Euler hacia adelante, sin embargo el controlador que se propone para esta aplicación es un controlador PI, por lo que el término derivativo no es contemplado.

La función de transferencia del controlador quedaría de la siguiente manera:

$$G_C(z) = \frac{K_p(z-1) + K_i T_S}{z-1} \quad (3.58)$$

Resultados y Discusión

4.1. Resultados

4.1.1. Filtrado

Tomando en cuenta la configuración de la Figura [3.1](#) y los valores obtenidos, se realiza una prueba con múltiples señales de diferentes frecuencias. A continuación se muestran 3 de esas señales con diferentes frecuencias, donde la señal de color amarillo es antes del filtro y la de color verde después del filtro.

Estas 3 señales son las que más nos interesan. La línea eléctrica tiene una frecuencia de 60 Hz y en la Figura [4.1](#) se observa del lado derecho una atenuación de 0 dB, por lo que esta frecuencia no se ve afectada.

La frecuencia de corte, que son 4 kHz, se observa en la Figura [4.2](#). En esta Figura ya se observa una atenuación de 3.2 dB y un desfase de aproximadamente 45° , que es lo esperado en la frecuencia de corte.

Finalmente, en la Figura [4.3](#) se observa una señal de 100 kHz. Esta frecuencia es de interés, debido a que es la frecuencia a la que opera el convertidor de puente completo y se desea la menor cantidad de ruido en dicha frecuencia para un correcto funcionamiento del circuito.

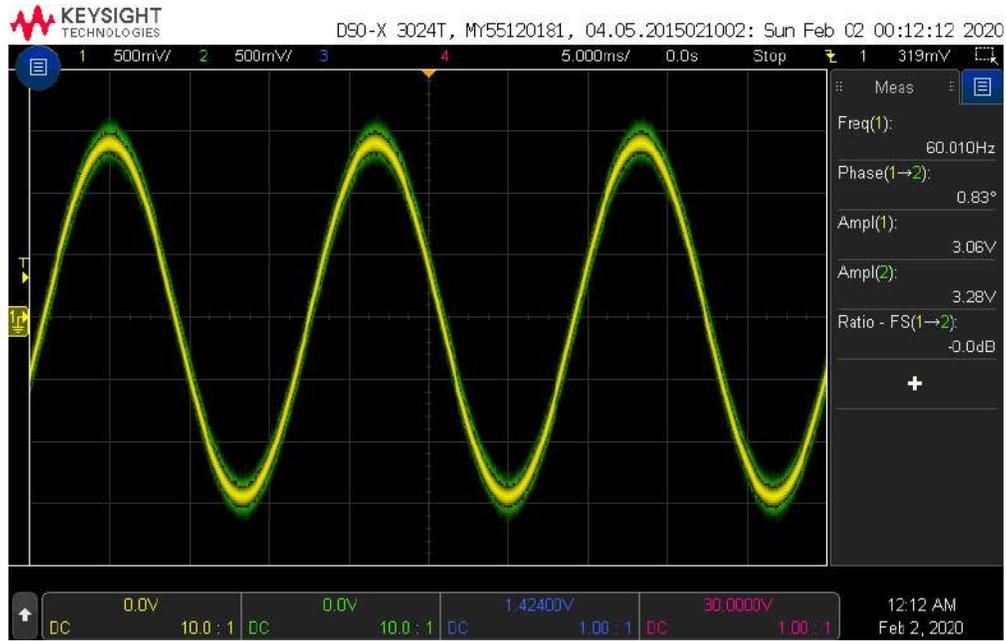


Figura 4.1: Señal de 60 Hz después de filtro EMI.

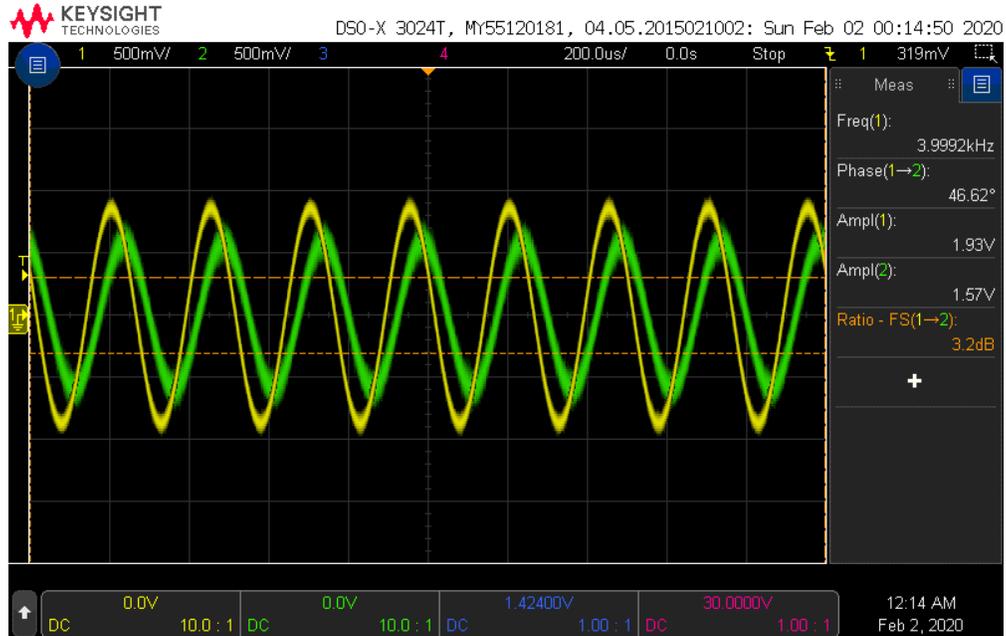


Figura 4.2: Señal de 4 kHz después de filtro EMI.

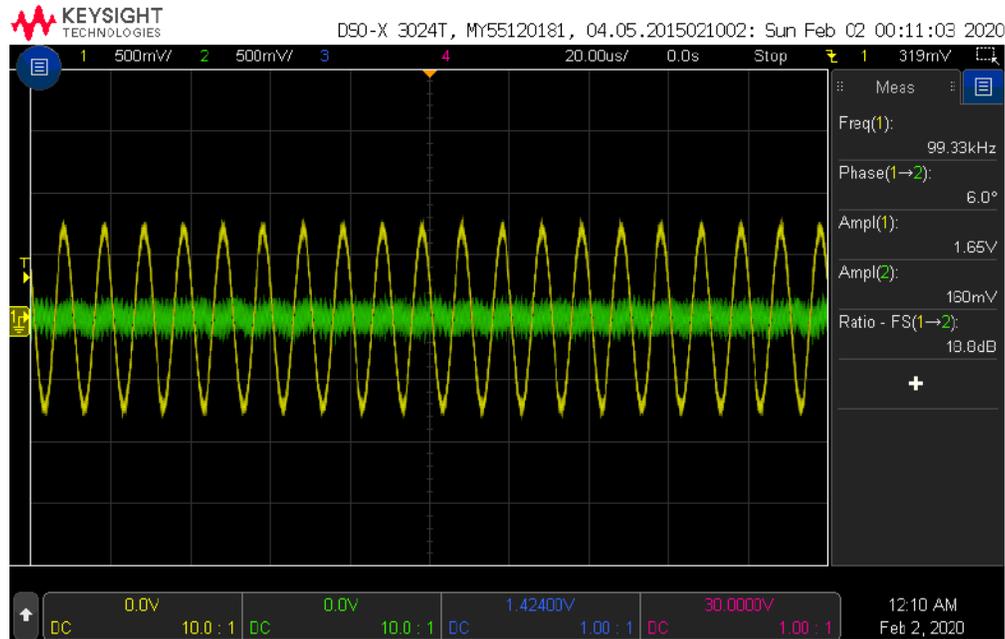


Figura 4.3: Señal de 100 kHz después de filtro EMI.

4.1.2. Materiales

Se construye un banco de pruebas para realizar la experimentación pertinente. El banco se muestra en la Figura 4.4. Está conformado por los siguientes componentes:

Tabla 4.1: Componentes de banco de pruebas.

Componente	Modelo
Microcontrolador	B-G474E-DPOW1
Inductor de salida	100 μ H
Capacitor de salida	3000 μ F
Núcleo de ferrita	ETD49
MOSFET	STP36N55M5
Diodo	RHRP3060

4.1.3. Eficiencia del convertidor

Para determinar la eficiencia del convertidor se realizan los cálculos correspondientes a las pérdidas de los componentes.

Transformador

La corriente en el devanado secundario se obtiene mediante la siguiente ecuación:

$$I_S = \sqrt{I_{S1}^2 + I_{S2}^2 + I_{S3}^2} = \sqrt{1,2^2 + 0,55^2 + 0,2^2} = 1,31[\text{A}] \quad (4.1)$$

donde:

I_{S1} = Corriente secundaria cuando la energía fluye en el secundario.

I_{S2} = Corriente secundaria cuando el par de diodos está conduciendo

I_{S3} = Corriente secundaria causada durante el período de marcha libre

La corriente primaria se obtiene mediante la siguiente ecuación:

$$I_P = \sqrt{I_{P1}^2 + I_{P2}^2} = \sqrt{0,7^2 + 0,4^2} = 0,81[\text{A}] \quad (4.2)$$

donde:

I_{P1} = Corriente primaria cuando la energía fluye en el secundario.

I_{P2} = Corriente primaria cuando el par de diodos está conduciendo

Las pérdidas estimadas del transformador se obtienen considerando las pérdidas por conducción y las pérdidas del núcleo.

$$P_T \approx I_P^2 \cdot DCR_P + I_S^2 \cdot DCR_S + P_{core} \approx 0,81^2 \cdot 0,12 + 1,31^2 \cdot 0,25 + 5,2 \approx 5,7[\text{W}] \quad (4.3)$$

donde:

P_T = Pérdidas de transformador

I_P = Corriente primaria

DCR_P = Resistencia CD primaria

I_S = Corriente secundaria

DCR_S = Resistencia CD secundaria

P_{core} = Pérdidas de núcleo

MOSFETs

Las pérdidas de conmutación ocurren cuando el MOSFET cambia de un estado encendido a un estado apagado y viceversa. La frecuencia juega un papel importante ya que el MOSFET cambiará de estado dependiendo de qué tan rápido cambie la señal de control. Para obtener las pérdidas de los MOSFET, se tuvo en cuenta el Informe de Aplicación PSFB de Texas Instruments [25].

$$P_Q \approx I_P^2 \cdot R_{ds(on)} + 2 \cdot Q A_g \cdot V_{gs} \cdot \frac{f_s}{2} \quad (4.4)$$

$$P_Q \approx 0,81^2 \cdot 0,08 + 2 \cdot (62 \cdot 10^{-9}) \cdot 12 \cdot \frac{(100 \cdot 10^3)}{2} \approx 0,06[\text{W}] \quad (4.5)$$

donde:

P_Q = Pérdidas de MOSFET

$R_{ds(on)}$ = Resistencia drenador-fuente

$Q A_g$ = Carga de puerta

V_{gs} = Voltaje puerta-fuente

Diodos

En este caso, la ecuación para determinar las pérdidas de diodos es la siguiente:

$$P_D \approx 2 \cdot V_D \cdot I_D \approx 2 \cdot 0,6 \cdot 1,31 \approx 1,57[\text{W}] \quad (4.6)$$

Inductor

Para obtener las pérdidas del inductor de salida, es necesario calcular la corriente pico-pico del inductor. Está diseñado para una corriente pico-pico del inductor del 10%.

$$\Delta I_{Lout} = \frac{P_{out} \cdot 0,1}{V_{out}} = \frac{125 \cdot 0,1}{25} = 0,5[\text{A}] \quad (4.7)$$

donde:

ΔI_{Lout} = Corriente pico-pico del inductor

P_{out} = Potencia de salida

V_{out} = Voltaje de salida

Calculando la corriente del inductor de salida:

$$I_{Lout} = \sqrt{\left(\frac{P_{out}}{V_{out}}\right)^2 + \left(\frac{\Delta I_{Lout}}{\sqrt{3}}\right)^2} = 5,01[\text{A}] \quad (4.8)$$

Finalmente, las pérdidas del inductor de salida se obtienen como una estimación del doble de las pérdidas por conducción.

$$P_{Lout} = 2 \cdot I_{Lout}^2 \cdot DCR_{Lout} \approx 2 \cdot 5,01^2 \cdot 0,033 \approx 1,65[\text{W}] \quad (4.9)$$

donde:

DCR_{Lout} = Resistencia CD del inductor de salida

Capacitor

Para obtener las pérdidas del condensador de salida, se debe calcular su corriente.

$$I_{Cout} = \frac{\Delta I_{Lout}}{\sqrt{3}} = 0,28[\text{A}] \quad (4.10)$$

donde:

I_{Cout} = Corriente del capacitor de salida

Las pérdidas del capacitor de salida se obtienen a partir de la siguiente ecuación:

$$P_{Cout} \approx (I_{Cout})^2 \cdot ESR_{Cout} \approx 0,28^2 \cdot (73 \cdot 10^{-3}) \approx 0,01[\text{W}] \quad (4.11)$$

donde:

ESR_{Cout} = Capacitancia de salida efectiva

Pérdidas totales

Para obtener las pérdidas totales, se realiza una suma de todas las pérdidas de los componentes anteriores. El resultado es el siguiente:

$$P_{loss} \approx P_T + 4 \cdot P_Q + P_D + P_{Lout} + P_{Cout} \approx 9,17[\text{W}] \quad (4.12)$$

Considerando la siguiente ecuación para determinar la eficiencia:

$$\eta = \frac{\frac{P_{out}}{P_{loss}}}{1 + \frac{P_{out}}{P_{loss}}} = \frac{\frac{125}{9,17}}{1 + \frac{125}{9,17}} = 0,931 \quad (4.13)$$

4.1.4. Implementación del modelo

Se prueba el método y se toman 9000 valores con un tiempo de muestreo de 40 μs . Este tiempo de muestreo se selecciona para obtener una señal más detallada. Se utiliza el siguiente archivo de MATLAB.

```
1 clc
2 clear all
3 T = 0.00001; %tiempo de muestreo
4 n = 2; %orden del sistema
```

```

5 [nmuestra,u,YT] = textread('MMCC.txt','%f %f %f');
6 N = length(nmuestra); %numero de muestras
7 Y = YT'; %Salida
8 y = Y;
9 ZT(1,:) = u; %Entrada
10 for i = 1 : n
11     for k = 1 : N - i
12         ZT(2*i,k+i) = y(k); %MMCC
13         ZT((2*i)+1,k+i) = u(k);
14     end
15 end
16 P =inv(ZT*ZT')*ZT*Y' %vector de parametros

```

El vector de parámetros obtenido de MATLAB es el siguiente:

$$P^T = [0, -0,38132, 0,004, -0,31958, 0,00247, -0,29436, -0,00181] \quad (4.14)$$

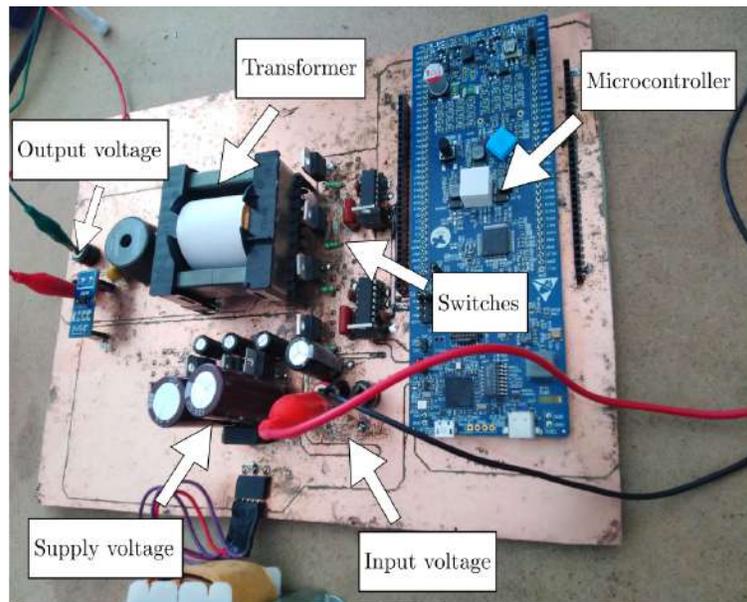


Figura 4.4: Banco de pruebas utilizado para experimentación.

La función de transferencia de un sistema discreto de tercer orden se describe como:

$$G(z) = \frac{b_0z^3 + b_1z^2 + b_2z + b_3}{z^3 + a_1z^2 + a_2z + a_3} \quad (4.15)$$

La ecuación (4.15) se sustituye luego con el vector de parámetros de la ecuación (4.14). Luego se obtiene el modelo, dando como resultado la siguiente función de transferencia:

$$G(z) = \frac{0,004z^2 + 0,00247z - 0,00181}{z^3 - 0,38132z^2 - 0,31958z - 0,29436} \quad (4.16)$$

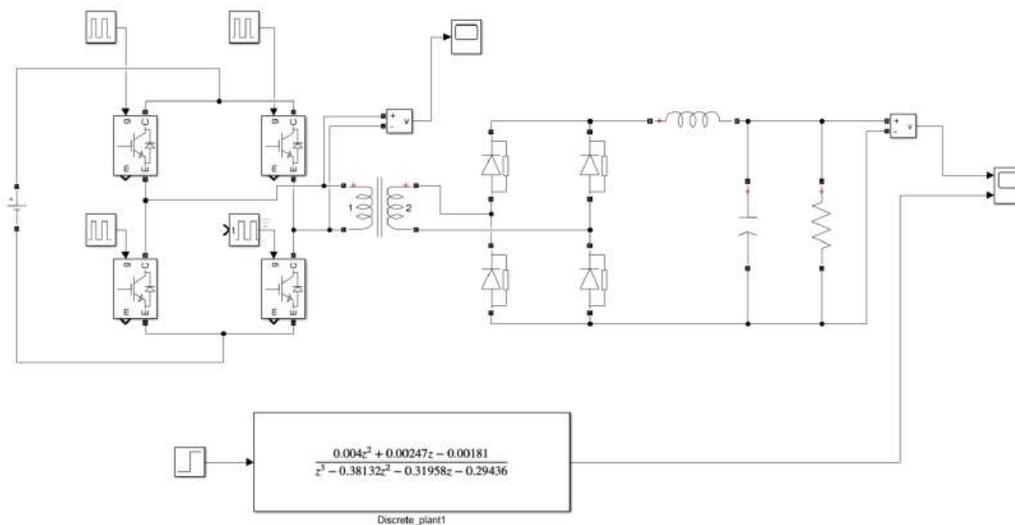


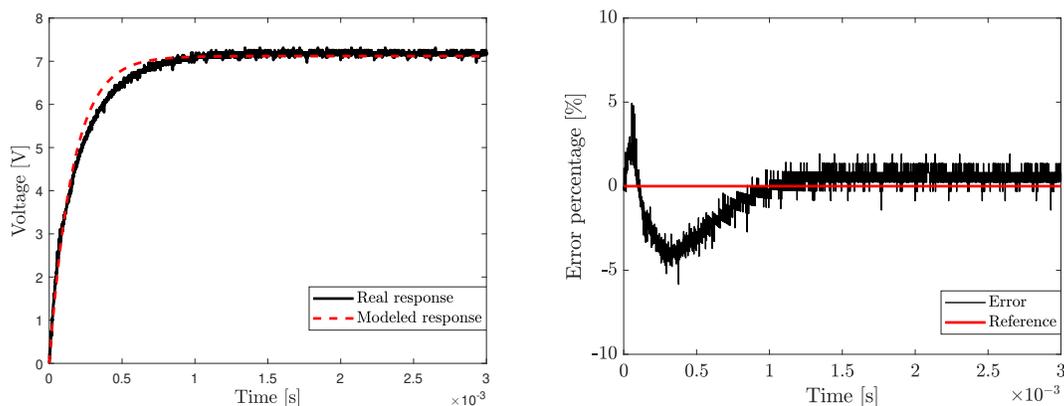
Figura 4.5: Diagrama del sistema en Simulink.

El porcentaje de error se obtiene con la siguiente ecuación:

$$\text{error} = \left(\frac{\text{modelado} - \text{real}}{\text{real}} \right) * 100 \quad (4.17)$$

Este modelo se compara con la respuesta real de la planta. La comparación se realiza considerando una entrada escalón en Simulink mostrada en la Figura 4.5. En la Figura 4.6a se observa el parecido del modelo obtenido con el real. A pesar de que las señales no son exactas, el método de mínimos cuadrados da una aproximación cercana a la realidad. En la Figura 4.6b se muestra el porcentaje de error. El porcentaje mostrado demuestra un error

de no más del 6 %, lo que significa una precisión del modelo del 94 %.



(a) Comparación entre respuesta real y mode- (b) Error obtenido al comparar la respuesta
lada. real y la respuesta modelada.

Figura 4.6: Respuesta de la planta ante entrada escalón

4.1.5. Control de voltaje

Para esta planta, se utiliza un controlador clásico para ajustar el voltaje del convertidor. Al ser un control discreto, la función de transferencia del controlador se puede expresar de diferentes formas en la parte integral y derivada.

El término integral se puede expresar de las siguientes formas:

$$\frac{K_i T_S}{z - 1} \quad \text{Euler hacia adelante} \quad (4.18)$$

$$\frac{K_i T_S z}{z - 1} \quad \text{Euler hacia atrás} \quad (4.19)$$

$$\frac{K_i T_S}{2} \frac{z + 1}{z - 1} \quad \text{Trapezoidal} \quad (4.20)$$

Se considera la forma de Euler hacia adelante y el controlador propuesto para esta aplicación es un controlador PI.

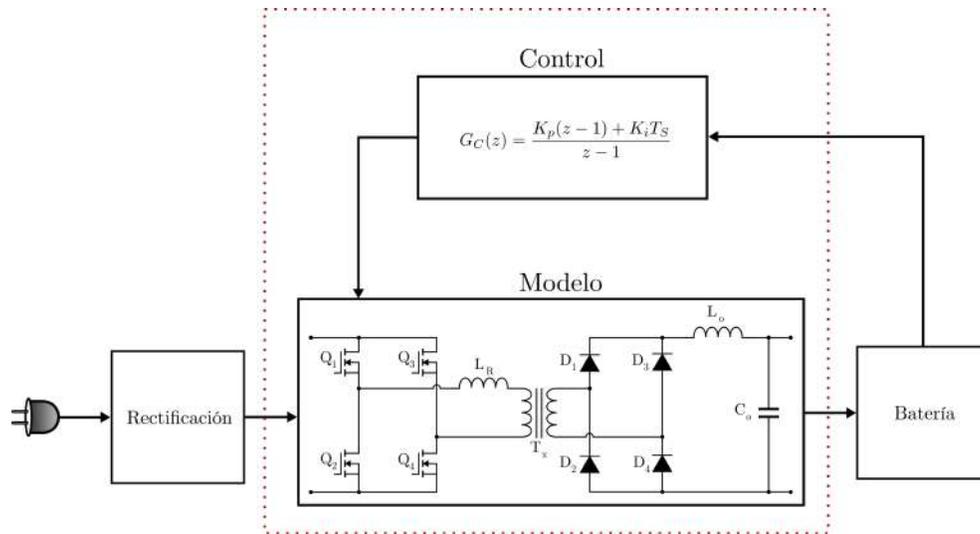


Figura 4.7: Diagrama del sistema, considerando el modelo obtenido y el controlador de voltaje.

La función de transferencia del controlador es la siguiente:

$$G_C(z) = \frac{K_p(z - 1) + K_i T_S}{z - 1} \quad (4.21)$$

El diagrama donde se contempla el modelo y el controlador se muestra en la Figura 4.7. En el diagrama, la carga es la batería y el controlador se encarga de regular el voltaje suministrado a la batería. Sin embargo, para ajustar el controlador, se usa una resistencia para observar la variación de voltaje en el sistema y el tiempo de respuesta del controlador.

La respuesta del sistema está determinada por los valores que se asignan a K_p y K_i . Se consideran dos posibles enfoques de control. La primera prueba se realiza considerando un tiempo de subida de 20 ms, que se muestra en la Figura 4.10. Para determinar los valores de K_p y K_i , se utilizó el sintonizador PID de MATLAB, donde se especificaron los parámetros deseados. El sintonizador se muestra en la Figura 4.8. Los valores obtenidos son:

$$K_p = 0,011945 \quad K_i = 238,83 \quad (4.22)$$

Los resultados reales se comparan con los resultados simulados. Se utiliza Simulink para realizar las simulaciones. El diagrama utilizado se muestra en la Figura 4.9.

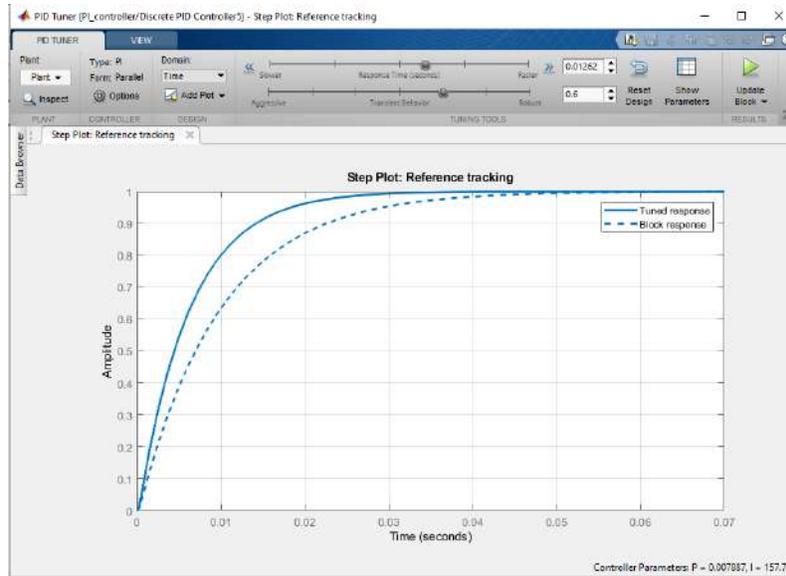


Figura 4.8: Ajuste del controlador a través del sintonizador PID de MATLAB.

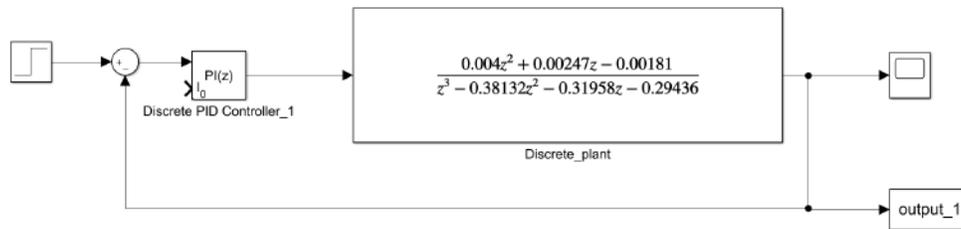


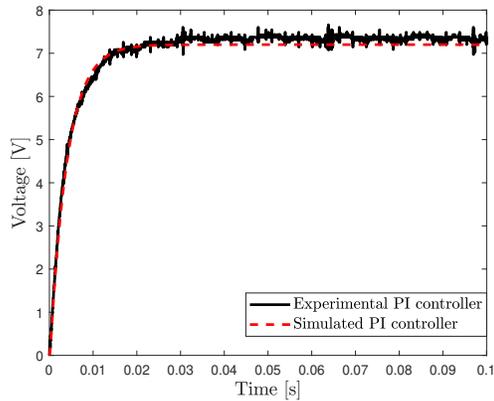
Figura 4.9: Diagrama de control en Simulink.

La respuesta esperada y la obtenida son bastante similares, lo que es un indicador de que el modelado se realizó correctamente.

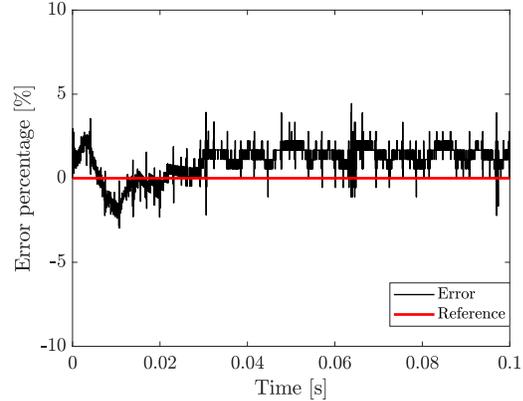
Sin embargo, al proponer una respuesta del sistema más rápida, la parte integral no permite que la planta mantenga un voltaje constante. Por esta razón, se considera una respuesta más lenta, ya que en esta aplicación una respuesta de 20 ms no es crítica, porque el voltaje de la batería no cambia más rápido que la respuesta del sistema.

Se realiza una segunda prueba, pero esta vez considerando un tiempo de subida de 50 ms. En este caso los valores de los términos proporcional e integral son:

$$K_p = 0,0049 \quad K_i = 99,73 \quad (4.23)$$

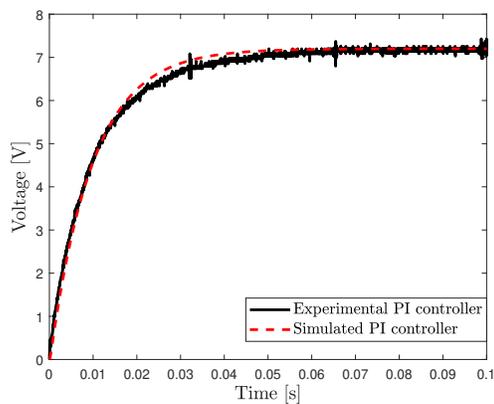


(a) Respuesta del controlador, caso 1.

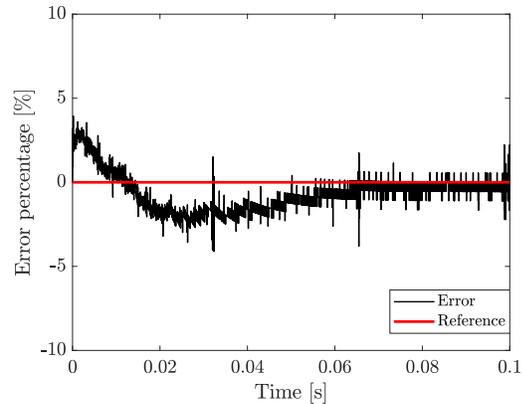


(b) Error obtenido al comparar el controlador simulado y el controlador experimental, caso 1.

Figura 4.10: Caso 1 con $Ts = 0,02$.



(a) Respuesta del controlador, caso 2.



(b) Error obtenido al comparar el controlador simulado y el controlador experimental, caso 2.

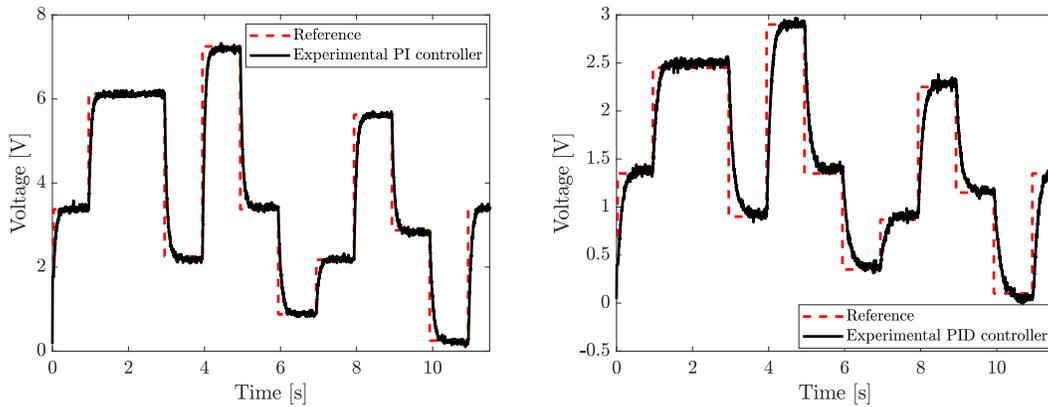
Figura 4.11: Caso 2 con $Ts = 0,05$.

El controlador se comporta como se muestra en la Figura [4.11](#). En este caso, el voltaje se ajusta con un rango de error más bajo y la respuesta teórica se acerca bastante a la real. Aunque la respuesta del controlador es más lenta, no afecta a la aplicación en cuestión ya que el voltaje de la batería también cambia lentamente.

Por tanto, se puede concluir que el controlador debe considerar un tiempo de subida y estabilización superior a 50 ms para evitar tener un gran margen de error.

Verificados los valores proporcionales e integrales del segundo caso, es necesario probar la respuesta dinámica del sistema para verificar la capacidad de regulación del controlador. La primer prueba es cambiar el punto de ajuste a una señal escalón aleatoria y la segunda es cambiar el punto de ajuste a una señal de diente de sierra.

Al cambiar el punto de ajuste a una señal de escalón aleatoria, se obtienen los siguientes resultados:



(a) Respuesta del controlador, caso PI. (b) Respuesta del controlador, caso PID.

Figura 4.12: Respuesta a la señal de escalón aleatoria.

El controlador se comporta de la misma manera que cuando se le introduce una señal escalón. La diferencia en esta prueba es que el controlador debe ajustarse constantemente a diferentes puntos de ajuste y el controlador logra ajustarse correctamente a las diferentes variaciones.

Para una comparación más precisa, el controlador PI se compara con un controlador PID. En este caso, el controlador PID está diseñado para tener el mismo tiempo de subida que el controlador PI. También se utiliza una señal escalón aleatoria para analizar el comportamiento del controlador. La respuesta se muestra en la Figura [4.12b](#).

Al mirar el gráfico del controlador PID, se observa una pequeña distorsión en el voltaje de salida. Esto se debe principalmente a que se ha incorporado la parte derivada al controlador, lo que introduce ruido en la señal, por lo que se considera que el controlador PI es más adecuado para esta aplicación.

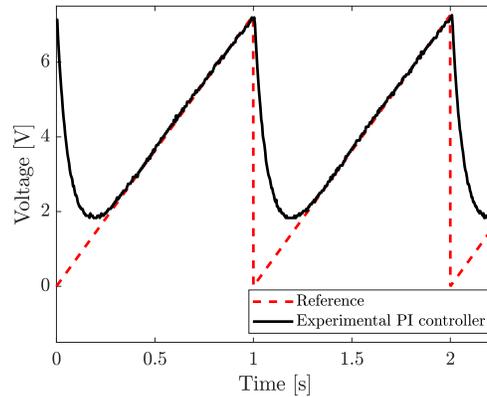


Figura 4.13: Respuesta del controlador PI ante señal de diente de sierra.

Finalmente, la referencia se cambia a una señal de diente de sierra. En este caso particular, es más difícil para el controlador PI ajustarse a la referencia cuando cambia drásticamente. Es posible que se necesite un ajuste diferente o un controlador más complejo para poder adaptarse a estos cambios abruptos.

4.2. Aporte científico de la investigación

El control que se realizó en la etapa de carga de la batería permitió suministrar el voltaje y corriente necesaria para una batería de litio-ferrofosfato. De igual manera funciona para carga de baterías de distintos materiales, con el único requisito de especificar el voltaje y corriente que se requiera.

4.3. Aporte tecnológico de la investigación

Se diseñó un prototipo funcional que permite realizar pruebas para la carga de baterías. Este prototipo permitió corroborar el modelado propuesto y probarlo con celdas para observar su eficiencia y ajuste de parámetros de voltaje y corriente.

4.4. Impacto sostenible de la investigación

El prototipo permitió que las baterías fueran cargadas de forma adecuada, aumentando así la vida útil de la batería y por lo tanto reducir los desechos por compra de un mayor número de baterías.

4.5. Consideraciones éticas

El trabajo de investigación no involucró riesgos visibles para la salud, en virtud de que se trata únicamente de un circuito electrónico. No existe riesgo de vulnerar derechos ni hay información de terceras partes que pueda ponerse en riesgo. Tampoco se han considerado riesgos biológicos o químicos dado que no se planea el desecho de ninguna de sus partes.

Conclusiones

Este trabajo presenta el diseño de un cargador para celdas de LiFePO_4 . Se propuso un modelado y control de un convertidor PSFB, utilizando el modelo ARX y un controlador clásico. El modelado y el control cumplen con los requisitos de carga de una batería LiFePO_4 . La hipótesis se aprueba, al comprobar que el convertidor logra suministrar el voltaje y corriente adecuados a las celdas de litio-ferrofosfato. El modelado llega a describir la planta con una precisión del 94%, por lo que el control diseñado llega a parecerse al obtenido de forma práctica en los dos casos propuestos. El controlador se comporta con precisión, alcanzando una precisión del 96% y controlando la tensión de salida como se esperaba. El controlador de la misma forma puede ser ajustado por la parte integral para modificar el tiempo de respuesta del sistema, aunque hay que tener en cuenta que reducir el tiempo de subida impide que la tensión se regule correctamente. Se facilitan los cálculos a la hora de controlar un sistema, concretamente un convertidor DC-DC, simplificando la obtención de la planta mediante ecuaciones complejas. El uso del modelo también facilita la implementación del controlador en cualquier sistema embebido, ya que desde el inicio las ecuaciones utilizadas son en tiempo discreto.

Como trabajo futuro se pretende hacer una comparativa con un controlador difuso para observar las mejoras que se pueden tener en este tipo de sistemas. De igual manera se tiene contemplado incorporar un control de temperatura para evitar que las celdas se sobrecalienten.

ten.

Bibliografía

- [1] M. Yilmaz and P. T. Krein, “Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles,” *IEEE Transactions on Power Electronics*, vol. 28, no. 5, pp. 2151–2169, 2013.
- [2] Y. Du, S. Lukic, B. Jacobson, and A. Huang, “Review of high power isolated bi-directional dc-dc converters for phev/ev dc charging infrastructure,” in *2011 IEEE Energy Conversion Congress and Exposition*, pp. 553–560, 2011.
- [3] e. a. Ehsani M., *Modern Electric, Hybrid Electric, and Fuel Cell Vehicles*. CRC Press, 2018.
- [4] United States Department of Energy, “Determining electric motor load and efficiency.”
- [5] M. L. Baglione, *DEVELOPMENT OF SYSTEM ANALYSIS METHODOLOGIES AND TOOLS FOR MODELING AND OPTIMIZING VEHICLE SYSTEM EFFICIENCY*. PhD thesis, University of Michigan, 2007.
- [6] Instituto Nacional de Estadística y Geografía, “Registro administrativo de la industria automotriz de vehículos ligeros,” 2020.
- [7] W.-J. Zhang, “Structure and performance of lifepo₄ cathode materials: A review,” *Journal of Power Sources*, vol. 196, no. 6, pp. 2962 – 2970, 2011.

- [8] F. Musavi, W. Eberle, and W. G. Dunford, "A high-performance single-phase bridgeless interleaved pfc converter for plug-in hybrid electric vehicle battery chargers," *IEEE Transactions on Industry Applications*, vol. 47, no. 4, pp. 1833–1843, 2011.
- [9] F. Musavi, M. Edington, W. Eberle, and W. G. Dunford, "Evaluation and efficiency comparison of front end ac-dc plug-in hybrid charger topologies," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 413–421, 2012.
- [10] A. Khaligh and S. Dusmez, "Comprehensive topological analysis of conductive and inductive charging solutions for plug-in electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 8, pp. 3475–3489, 2012.
- [11] H. He, R. Xiong, and H. Guo, "Online estimation of model parameters and state-of-charge of lifepo4 batteries in electric vehicles," *Applied Energy*, vol. 89, no. 1, pp. 413–420, 2012.
- [12] C. Oh, D. Kim, D. Woo, W. Sung, Y. Kim, and B. Lee, "A high-efficient nonisolated single-stage on-board battery charger for electric vehicles," *IEEE Transactions on Power Electronics*, vol. 28, no. 12, pp. 5746–5757, 2013.
- [13] B. Whitaker, A. Barkley, Z. Cole, B. Passmore, D. Martin, T. R. McNutt, A. B. Lostetter, J. S. Lee, and K. Shiozaki, "A high-density, high-efficiency, isolated on-board vehicle battery charger utilizing silicon carbide power devices," *IEEE Transactions on Power Electronics*, vol. 29, no. 5, pp. 2606–2617, 2014.
- [14] M. Hannan, M. Lipu, A. Hussain, and A. Mohamed, "A review of lithium-ion battery state of charge estimation and management system in electric vehicle applications: Challenges and recommendations," *Renewable and Sustainable Energy Reviews*, vol. 78, pp. 834 – 854, 2017.
- [15] T. Prasetya, D. Wijaya, and E. Firmansyah, "Design of full-bridge dc-dc converter 311/100 v 1kw with pspwm method to get zvs condition," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 8, p. 59, 03 2017.

- [16] A. Amir, A. Amir, H. S. Che, A. Elkhateb, and N. A. Rahim, “Comparative analysis of high voltage gain dc-dc converter topologies for photovoltaic systems,” *Renewable Energy*, vol. 136, pp. 1147 – 1163, 2019.
- [17] A. Khaligh and M. D’Antonio, “Global trends in high-power on-board chargers for electric vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3306–3324, 2019.
- [18] W. Khan, F. Ahmad, and M. S. Alam, “Fast ev charging station integration with grid ensuring optimal and quality power exchange,” *Engineering Science and Technology, an International Journal*, vol. 22, no. 1, pp. 143 – 152, 2019.
- [19] J. Ahn and B. K. Lee, “High-efficiency adaptive-current charging strategy for electric vehicles considering variation of internal resistance of lithium-ion battery,” *IEEE Transactions on Power Electronics*, vol. 34, no. 4, pp. 3041–3052, 2019.
- [20] H. Tu, H. Feng, S. Srdic, and S. Lukic, “Extreme fast charging of electric vehicles: A technology overview,” *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 861–878, 2019.
- [21] D. Kumar, R. K. Nema, and S. Gupta, “A comparative review on power conversion topologies and energy storage system for electric vehicles,” *International Journal of Energy Research*, vol. 44, no. 10, pp. 7863–7885, 2020.
- [22] P. R. Kumar and C. G. Shankar, “High-performance single-input three-output dc–dc high gain converter for fuel cell-based electric vehicles,” *Electrical Engineering*, vol. 102, no. 3, pp. 1715–1737, 2020.
- [23] e. a. Choudhury, “A comprehensive review and feasibility study of dc–dc converters for different pv applications: Ess, future residential purpose, ev charging,” *Energy Systems*, vol. 11, no. 3, p. 641–671, 2020.

- [24] M. Bayati, M. Abedi, G. B. Gharehpetian, and M. Farahmandrad, “Sinusoidal-ripple current control in battery charger of electric vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7201–7210, 2020.
- [25] M. O’Loughlin, “Phase-Shifted, Full-Bridge Application Report,” Tech. Rep. SLUA560C, Texas Instruments, 06 2011.

Banco de pruebas

En esta sección se muestra información referente al prototipo utilizado como banco de pruebas. Con la información mostrada es posible replicar los resultados obtenidos.

A.1. Programación de microcontrolador

A continuación se muestra el código principal utilizado en el microcontrolador B-G474E-DPOW1. En este código se hizo uso del sistema operativo en tiempo real FreeRTOS para realizar tareas y ejecutarlas por prioridad. La tarea de mayor prioridad es el control de voltaje y corriente, mientras que la transmisión de datos por comunicación UART y el cambio del punto de referencia para pruebas tienen menor prioridad.

```
1 /* USER CODE BEGIN Header */
2 /**
3  * *****
4  * @file      : main.c
5  * @brief     : Main program body
6  * *****
7  * @attention
8  *
9  * <h2><center>&copy; Copyright (c) 2021 STMicroelectronics.
10 * All rights reserved.</center></h2>
11 *
12 * This software component is licensed by ST under BSD 3-Clause license,
13 * the "License"; You may not use this file except in compliance with the
14 * License. You may obtain a copy of the License at:
```

```

15      *                               opensource.org/licenses/BSD-3-Clause
16      *
17      ****
18      */
19  /* USER CODE END Header */
20  /* Includes ...
   -----*/
21  #include "main.h"
22  #include "cmsis_os.h"
23
24  /* Private includes ...
   -----*/
25  /* USER CODE BEGIN Includes */
26
27  #include "string.h"
28  #include "stdio.h"
29
30  /* USER CODE END Includes */
31
32  /* Private typedef ...
   -----*/
33  /* USER CODE BEGIN PTD */
34
35  float iref = 0;
36
37  /* USER CODE END PTD */
38
39  /* Private define ...
   -----*/
40  /* USER CODE BEGIN PD */
41
42
43  /* USER CODE END PD */
44
45  /* Private macro ...
   -----*/
46  /* USER CODE BEGIN PM */
47
48  /* USER CODE END PM */
49
50  /* Private variables ...
   -----*/
51  ADC_HandleTypeDef hadc1;
52
53  HRTIM_HandleTypeDef hrtim1;
54
55  SPI_HandleTypeDef hspi1;
56
57  TIM_HandleTypeDef htim8;
58
59  UART_HandleTypeDef huart3;

```

```

60
61 PCD_HandleTypeDef hpcd_USB_FS;
62
63 /* Definitions for Transmit */
64 osThreadId_t TransmitHandle;
65 const osThreadAttr_t Transmit_attributes = {
66     .name = "Transmit",
67     .priority = (osPriority_t) osPriorityNormal,
68     .stack_size = 128 * 4
69 };
70 /* Definitions for Blinky */
71 osThreadId_t BlinkyHandle;
72 const osThreadAttr_t Blinky_attributes = {
73     .name = "Blinky",
74     .priority = (osPriority_t) osPriorityLow,
75     .stack_size = 128 * 4
76 };
77 /* Definitions for Control */
78 osThreadId_t ControlHandle;
79 const osThreadAttr_t Control_attributes = {
80     .name = "Control",
81     .priority = (osPriority_t) osPriorityHigh,
82     .stack_size = 128 * 4
83 };
84 /* Definitions for InputChange */
85 osThreadId_t InputChangeHandle;
86 const osThreadAttr_t InputChange_attributes = {
87     .name = "InputChange",
88     .priority = (osPriority_t) osPriorityLow,
89     .stack_size = 128 * 4
90 };
91 /* USER CODE BEGIN PV */
92
93 /* USER CODE END PV */
94
95 /* Private function prototypes ...
96 -----*/
97 void SystemClock_Config(void);
98 static void MX_GPIO_Init(void);
99 static void MX_ADC1_Init(void);
100 static void MX_USART3_UART_Init(void);
101 static void MX_USB_PCD_Init(void);
102 static void MX_TIM8_Init(void);
103 static void MX_HRTIM1_Init(void);
104 static void MX_SPI1_Init(void);
105 void TransmitTask(void *argument);
106 void BlinkyTask(void *argument);
107 void ControlTask(void *argument);
108 void InputChangeTask(void *argument);
109 /* USER CODE BEGIN PFP */

```

```

110
111 /* USER CODE END PFP */
112
113 /* Private user code ...
-----*/
114 /* USER CODE BEGIN 0 */
115
116
117
118 /* USER CODE END 0 */
119
120 /**
121  * @brief The application entry point.
122  * @retval int
123  */
124 int main(void)
125 {
126     /* USER CODE BEGIN 1 */
127
128
129     /* USER CODE END 1 */
130
131     /* MCU Configuration-----*/
132
133     /* Reset of all peripherals, Initializes the Flash interface and the ...
        Systick. */
134     HAL_Init();
135
136     /* USER CODE BEGIN Init */
137
138     /* USER CODE END Init */
139
140     /* Configure the system clock */
141     SystemClock_Config();
142
143     /* USER CODE BEGIN SysInit */
144
145     /* USER CODE END SysInit */
146
147     /* Initialize all configured peripherals */
148     MX_GPIO_Init();
149     MX_ADC1_Init();
150     MX_USART3_UART_Init();
151     MX_USB_PCD_Init();
152     MX_TIM8_Init();
153     MX_HRTIM1_Init();
154     MX_SPI1_Init();
155     /* USER CODE BEGIN 2 */
156
157
158

```

```

159 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
160
161
162
163
164 /*
165 HAL_TIM_PWM_Start(&htim8, TIM_CHANNEL_1);
166 HAL_TIMEx_PWMN_Start(&htim8, TIM_CHANNEL_1);
167
168 HAL_TIM_PWM_Start(&htim8, TIM_CHANNEL_2);
169 HAL_TIMEx_PWMN_Start(&htim8, TIM_CHANNEL_2);
170
171
172 SET_BIT(TIM8->CR1, TIM_CR1_UDIS);
173
174 TIM8->CCR1 = 700 - phase;
175 TIM8->CCR2 = 700 + phase;
176
177 CLEAR_BIT(TIM8->CR1, TIM_CR1_UDIS);
178
179 */
180
181
182 HRTIM_DeadTimeCfgTypeDef pDeadTimeCfg = {0};
183 pDeadTimeCfg.Prescaler = HRTIM_TIMDEADTIME_PRESCALERRATIO_MUL8;
184 pDeadTimeCfg.RisingValue = 5000;
185 pDeadTimeCfg.FallingValue = 5000;
186 HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_A, ...
187     &pDeadTimeCfg);
188 HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
189     &pDeadTimeCfg);
190
191 //int timerC = 54400;
192 //int timerA = 54400;
193
194 /* ----- Timer C initialization ...
195 ----- */
196 HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_C].TIMxCR = ...
197     HRTIM_TIMCR_CONT + HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU;
198
199 /* ----- Timer A initialization ...
200 ----- */
201 HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].TIMxCR = ...
202     HRTIM_TIMCR_CONT + HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU;
203
204 /* Enable TA1, TA2, TD1 and TD2 outputs */
205 HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TA1OEN + HRTIM_OENR_TA2OEN + ...
206     HRTIM_OENR_TC1OEN + HRTIM_OENR_TC2OEN;
207
208 /* Start Timer A and Timer D */
209 HRTIM1->sMasterRegs.MCR = HRTIM_MCR_TACEN + HRTIM_MCR_TCCEN;

```

```

203
204
205
206 /* USER CODE END 2 */
207
208 /* Init scheduler */
209 osKernelInitialize();
210
211 /* USER CODE BEGIN RTOS_MUTEX */
212 /* add mutexes, ... */
213 /* USER CODE END RTOS_MUTEX */
214
215 /* USER CODE BEGIN RTOS_SEMAPHORES */
216 /* add semaphores, ... */
217 /* USER CODE END RTOS_SEMAPHORES */
218
219 /* USER CODE BEGIN RTOS_TIMERS */
220 /* start timers, add new ones, ... */
221 /* USER CODE END RTOS_TIMERS */
222
223 /* USER CODE BEGIN RTOS_QUEUES */
224 /* add queues, ... */
225 /* USER CODE END RTOS_QUEUES */
226
227 /* Create the thread(s) */
228 /* creation of Transmit */
229 TransmitHandle = osThreadNew(TransmitTask, NULL, &Transmit_attributes);
230
231 /* creation of Blinky */
232 BlinkyHandle = osThreadNew(BlinkyTask, NULL, &Blinky_attributes);
233
234 /* creation of Control */
235 ControlHandle = osThreadNew(ControlTask, NULL, &Control_attributes);
236
237 /* creation of InputChange */
238 InputChangeHandle = osThreadNew(InputChangeTask, NULL, ...
    &InputChange_attributes);
239
240 /* USER CODE BEGIN RTOS_THREADS */
241 /* add threads, ... */
242 /* USER CODE END RTOS_THREADS */
243
244 /* USER CODE BEGIN RTOS_EVENTS */
245 /* add events, ... */
246 /* USER CODE END RTOS_EVENTS */
247
248 /* Start scheduler */
249 osKernelStart();
250
251 /* We should never get here as control is now taken by the scheduler */
252 /* Infinite loop */

```

```

253  /* USER CODE BEGIN WHILE */
254  while (1)
255  {
256
257      /* USER CODE END WHILE */
258
259      /* USER CODE BEGIN 3 */
260  }
261  /* USER CODE END 3 */
262  }
263
264  /**
265   * @brief System Clock Configuration
266   * @retval None
267   */
268  void SystemClock_Config(void)
269  {
270      RCC_OscInitTypeDef RCC_OscInitStruct = {0};
271      RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
272      RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};
273
274      /** Configure the main internal regulator output voltage
275       */
276      HAL_PWREx_ControlVoltageScaling(PWR_REGULATOR_VOLTAGE_SCALE1_BOOST);
277      /** Initializes the RCC Oscillators according to the specified parameters
278       * in the RCC_OscInitTypeDef structure.
279       */
280      RCC_OscInitStruct.OscillatorType = ...
281          RCC_OSCILLATORTYPE_HSI|RCC_OSCILLATORTYPE_HSI48;
282      RCC_OscInitStruct.HSISState = RCC_HSI_ON;
283      RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
284      RCC_OscInitStruct.HSI48State = RCC_HSI48_ON;
285      RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
286      RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
287      RCC_OscInitStruct.PLL.PLLM = RCC_PLLM_DIV4;
288      RCC_OscInitStruct.PLL.PLLN = 85;
289      RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV8;
290      RCC_OscInitStruct.PLL.PLLQ = RCC_PLLQ_DIV8;
291      RCC_OscInitStruct.PLL.PLLR = RCC_PLLR_DIV2;
292      if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
293      {
294          Error_Handler();
295      }
296      /** Initializes the CPU, AHB and APB buses clocks
297       */
298      RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
299          |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
300      RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
301      RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
302      RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
303      RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

```

```

303
304 if (HAL_RCC.ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_4) != HAL_OK)
305 {
306     Error_Handler();
307 }
308 /** Initializes the peripherals clocks
309 */
310 PeriphClkInit.PeriphClockSelection = ...
311     RCC_PERIPHCLK_USART3|RCC_PERIPHCLK_USB
312     |RCC_PERIPHCLK_ADC12;
313 PeriphClkInit.Usart3ClockSelection = RCC_USART3CLKSOURCE_PCLK1;
314 PeriphClkInit.UsbClockSelection = RCC_USBCLKSOURCE_HSI48;
315 PeriphClkInit.Adcl2ClockSelection = RCC_ADC12CLKSOURCE_SYSCLK;
316 if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
317 {
318     Error_Handler();
319 }
320
321 /**
322  * @brief ADC1 Initialization Function
323  * @param None
324  * @retval None
325  */
326 static void MX_ADC1_Init(void)
327 {
328
329     /* USER CODE BEGIN ADC1_Init 0 */
330
331     /* USER CODE END ADC1_Init 0 */
332
333     ADC_MultiModeTypeDef multimode = {0};
334     ADC_ChannelConfTypeDef sConfig = {0};
335
336     /* USER CODE BEGIN ADC1_Init 1 */
337
338     /* USER CODE END ADC1_Init 1 */
339     /** Common config
340     */
341     hadcl.Instance = ADC1;
342     hadcl.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
343     hadcl.Init.Resolution = ADC_RESOLUTION_12B;
344     hadcl.Init.DataAlign = ADC_DATAALIGN_RIGHT;
345     hadcl.Init.GainCompensation = 0;
346     hadcl.Init.ScanConvMode = ADC_SCAN_DISABLE;
347     hadcl.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
348     hadcl.Init.LowPowerAutoWait = DISABLE;
349     hadcl.Init.ContinuousConvMode = DISABLE;
350     hadcl.Init.NbrOfConversion = 1;
351     hadcl.Init.DiscontinuousConvMode = DISABLE;
352     hadcl.Init.ExternalTrigConv = ADC_SOFTWARE_START;

```

```

353 hadcl.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;
354 hadcl.Init.DMAContinuousRequests = DISABLE;
355 hadcl.Init.Overrun = ADC_OVR_DATA_PRESERVED;
356 hadcl.Init.OversamplingMode = DISABLE;
357 if (HAL_ADC_Init(&hadcl) != HAL_OK)
358 {
359     Error_Handler();
360 }
361 /** Configure the ADC multi-mode
362 */
363 multimode.Mode = ADC_MODE_INDEPENDENT;
364 if (HAL_ADCEx_MultiModeConfigChannel(&hadcl, &multimode) != HAL_OK)
365 {
366     Error_Handler();
367 }
368 /** Configure Regular Channel
369 */
370 sConfig.Channel = ADC_CHANNEL_2;
371 sConfig.Rank = ADC_REGULAR_RANK_1;
372 sConfig.SamplingTime = ADC_SAMPLETIME_2CYCLES_5;
373 sConfig.SingleDiff = ADC_SINGLE_ENDED;
374 sConfig.OffsetNumber = ADC_OFFSET_NONE;
375 sConfig.Offset = 0;
376 if (HAL_ADC_ConfigChannel(&hadcl, &sConfig) != HAL_OK)
377 {
378     Error_Handler();
379 }
380 /* USER CODE BEGIN ADC1_Init 2 */
381
382 /* USER CODE END ADC1_Init 2 */
383
384 }
385
386 /**
387  * @brief HRTIM1 Initialization Function
388  * @param None
389  * @retval None
390  */
391 static void MX_HRTIM1_Init(void)
392 {
393
394     /* USER CODE BEGIN HRTIM1_Init 0 */
395
396     /* USER CODE END HRTIM1_Init 0 */
397
398     HRTIM_TimeBaseCfgTypeDef pTimeBaseCfg = {0};
399     HRTIM_TimerCtlTypeDef pTimerCtl = {0};
400     HRTIM_TimerCfgTypeDef pTimerCfg = {0};
401     HRTIM_DeadTimeCfgTypeDef pDeadTimeCfg = {0};
402     HRTIM_OutputCfgTypeDef pOutputCfg = {0};
403     HRTIM_CompareCfgTypeDef pCompareCfg = {0};

```

```

404
405 /* USER CODE BEGIN HRTIM1_Init 1 */
406
407 /* USER CODE END HRTIM1_Init 1 */
408 hhrtim1.Instance = HRTIM1;
409 hhrtim1.Init.HRTIMInterruptResquests = HRTIM_IT_NONE;
410 hhrtim1.Init.SyncOptions = HRTIM_SYNCOPTION_NONE;
411 if (HAL_HRTIM_Init(&hhrtim1) != HAL_OK)
412 {
413     Error_Handler();
414 }
415 if (HAL_HRTIM_DLLCalibrationStart(&hhrtim1, HRTIM_CALIBRATIONRATE_3) ...
416     != HAL_OK)
417 {
418     Error_Handler();
419 }
420 if (HAL_HRTIM_PollForDLLCalibration(&hhrtim1, 10) != HAL_OK)
421 {
422     Error_Handler();
423 }
424 pTimeBaseCfg.Period = 25000;
425 pTimeBaseCfg.RepetitionCounter = 0x00;
426 pTimeBaseCfg.PrescalerRatio = HRTIM_PRESCALERRATIO_MUL32;
427 pTimeBaseCfg.Mode = HRTIM_MODE_CONTINUOUS;
428 if (HAL_HRTIM_TimeBaseConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_A, ...
429     &pTimeBaseCfg) != HAL_OK)
430 {
431     Error_Handler();
432 }
433 pTimerCtl.UpDownMode = HRTIM_TIMERUPDOWNMODE_UP;
434 pTimerCtl.GreaterCMP1 = HRTIM_TIMERGTCMP1_EQUAL;
435 pTimerCtl.DualChannelDacEnable = HRTIM_TIMER_DCDE_DISABLED;
436 if (HAL_HRTIM_WaveformTimerControl(&hhrtim1, ...
437     HRTIM_TIMERINDEX_TIMER_A, &pTimerCtl) != HAL_OK)
438 {
439     Error_Handler();
440 }
441 pTimerCfg.InterruptRequests = HRTIM_TIM_IT_NONE;
442 pTimerCfg.DMARequests = HRTIM_TIM_DMA_NONE;
443 pTimerCfg.DMASrcAddress = 0x0000;
444 pTimerCfg.DMADstAddress = 0x0000;
445 pTimerCfg.DMASize = 0x1;
446 pTimerCfg.HalfModeEnable = HRTIM_HALFMODE_DISABLED;
447 pTimerCfg.InterleavedMode = HRTIM_INTERLEAVED_MODE_DUAL;
448 pTimerCfg.StartOnSync = HRTIM_SYNCSTART_DISABLED;
449 pTimerCfg.ResetOnSync = HRTIM_SYNCRESET_DISABLED;
450 pTimerCfg.DACSynchro = HRTIM_DACSYNC_NONE;
451 pTimerCfg.PreloadEnable = HRTIM_PRELOAD_DISABLED;
452 pTimerCfg.UpdateGating = HRTIM_UPDATEGATING_INDEPENDENT;
453 pTimerCfg.BurstMode = HRTIM_TIMERBURSTMODE_MAINTAINCLOCK;
454 pTimerCfg.RepetitionUpdate = HRTIM_UPDATEONREPETITION_DISABLED;

```

```

452 pTimerCfg.PushPull = HRTIM_TIMPUSHPULLMODE_DISABLED;
453 pTimerCfg.FaultEnable = HRTIM_TIMFAULTENABLE_NONE;
454 pTimerCfg.FaultLock = HRTIM_TIMFAULTLOCK_READWRITE;
455 pTimerCfg.DeadTimeInsertion = HRTIM_TIMDEADTIMEINSERTION_ENABLED;
456 pTimerCfg.DelayedProtectionMode = ...
    HRTIM_TIMER_A_B_C_DELAYEDPROTECTION_DISABLED;
457 pTimerCfg.UpdateTrigger = HRTIM_TIMUPDATETRIGGER_NONE;
458 pTimerCfg.ResetTrigger = HRTIM_TIMRESETTRIGGER_NONE;
459 pTimerCfg.ResetUpdate = HRTIM_TIMUPDATEONRESET_DISABLED;
460 pTimerCfg.ReSyncUpdate = HRTIM_TIMERESYNC_UPDATE_UNCONDITIONAL;
461 if (HAL_HRTIM_WaveformTimerConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_A, ...
    &pTimerCfg) != HAL_OK)
462 {
463     Error_Handler();
464 }
465 if (HAL_HRTIM_WaveformTimerConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
    &pTimerCfg) != HAL_OK)
466 {
467     Error_Handler();
468 }
469 pDeadTimeCfg.Prescaler = HRTIM_TIMDEADTIME_PRESCALERRATIO_DIV1;
470 pDeadTimeCfg.RisingValue = 500;
471 pDeadTimeCfg.RisingSign = HRTIM_TIMDEADTIME_RISINGSIGN_POSITIVE;
472 pDeadTimeCfg.RisingLock = HRTIM_TIMDEADTIME_RISINGLOCK_WRITE;
473 pDeadTimeCfg.RisingSignLock = HRTIM_TIMDEADTIME_RISINGSIGNLOCK_WRITE;
474 pDeadTimeCfg.FallingValue = 500;
475 pDeadTimeCfg.FallingSign = HRTIM_TIMDEADTIME_FALLINGSIGN_POSITIVE;
476 pDeadTimeCfg.FallingLock = HRTIM_TIMDEADTIME_FALLINGLOCK_WRITE;
477 pDeadTimeCfg.FallingSignLock = HRTIM_TIMDEADTIME_FALLINGSIGNLOCK_WRITE;
478 if (HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_A, ...
    &pDeadTimeCfg) != HAL_OK)
479 {
480     Error_Handler();
481 }
482 if (HAL_HRTIM_DeadTimeConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
    &pDeadTimeCfg) != HAL_OK)
483 {
484     Error_Handler();
485 }
486 pOutputCfg.Polarity = HRTIM_OUTPUTPOLARITY_HIGH;
487 pOutputCfg.SetSource = HRTIM_OUTPUTSET_TIMCMP1;
488 pOutputCfg.ResetSource = HRTIM_OUTPUTRESET_TIMCMP1;
489 pOutputCfg.IdleMode = HRTIM_OUTPUTIDLEMODE_NONE;
490 pOutputCfg.IdleLevel = HRTIM_OUTPUTIDLELEVEL_INACTIVE;
491 pOutputCfg.FaultLevel = HRTIM_OUTPUTFAULTLEVEL_NONE;
492 pOutputCfg.ChopperModeEnable = HRTIM_OUTPUTCHOPPERMODE_DISABLED;
493 pOutputCfg.BurstModeEntryDelayed = HRTIM_OUTPUTBURSTMODEENTRY_REGULAR;
494 if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_A, HRTIM_OUTPUT_TA1, &pOutputCfg) != HAL_OK)
495 {
496     Error_Handler();

```

```

497 }
498 pOutputCfg.SetSource = HRTIM_OUTPUTSET_TIMCMP2;
499 pOutputCfg.ResetSource = HRTIM_OUTPUTRESET_TIMCMP2;
500 if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_C, HRTIM_OUTPUT_TC1, &pOutputCfg) != HAL_OK)
501 {
502     Error_Handler();
503 }
504 pOutputCfg.SetSource = HRTIM_OUTPUTSET_NONE;
505 pOutputCfg.ResetSource = HRTIM_OUTPUTRESET_NONE;
506 if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_A, HRTIM_OUTPUT_TA2, &pOutputCfg) != HAL_OK)
507 {
508     Error_Handler();
509 }
510 if (HAL_HRTIM_WaveformOutputConfig(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_C, HRTIM_OUTPUT_TC2, &pOutputCfg) != HAL_OK)
511 {
512     Error_Handler();
513 }
514 if (HAL_HRTIM_TimeBaseConfig(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
    &pTimeBaseCfg) != HAL_OK)
515 {
516     Error_Handler();
517 }
518 pTimerCtl.TrigHalf = HRTIM_TIMERTRIGHALF_DISABLED;
519 if (HAL_HRTIM_WaveformTimerControl(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_C, &pTimerCtl) != HAL_OK)
520 {
521     Error_Handler();
522 }
523 pCompareCfg.CompareValue = 12000;
524 pCompareCfg.AutoDelayedMode = HRTIM_AUTODELAYEDMODE_REGULAR;
525 pCompareCfg.AutoDelayedTimeout = 0x0000;
526
527 if (HAL_HRTIM_WaveformCompareConfig(&hhrtim1, ...
    HRTIM_TIMERINDEX_TIMER_C, HRTIM_COMPAREUNIT_2, &pCompareCfg) != ...
    HAL_OK)
528 {
529     Error_Handler();
530 }
531 /* USER CODE BEGIN HRTIM1_Init 2 */
532
533 /* USER CODE END HRTIM1_Init 2 */
534 HAL_HRTIM_MspPostInit(&hhrtim1);
535
536 }
537
538 /**
539  * @brief SPI1 Initialization Function
540  * @param None

```

```

541  * @retval None
542  */
543 static void MX_SPI1_Init(void)
544 {
545
546  /* USER CODE BEGIN SPI1_Init 0 */
547
548  /* USER CODE END SPI1_Init 0 */
549
550  /* USER CODE BEGIN SPI1_Init 1 */
551
552  /* USER CODE END SPI1_Init 1 */
553  /* SPI1 parameter configuration*/
554  hspil.Instance = SPI1;
555  hspil.Init.Mode = SPI_MODE_MASTER;
556  hspil.Init.Direction = SPI_DIRECTION_2LINES;
557  hspil.Init.DataSize = SPI_DATASIZE_8BIT;
558  hspil.Init.CLKPolarity = SPI_POLARITY_LOW;
559  hspil.Init.CLKPhase = SPI_PHASE_1EDGE;
560  hspil.Init.NSS = SPI_NSS_SOFT;
561  hspil.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_256;
562  hspil.Init.FirstBit = SPI_FIRSTBIT_MSB;
563  hspil.Init.TIMode = SPI_TIMODE_DISABLE;
564  hspil.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
565  hspil.Init.CRCPolynomial = 7;
566  hspil.Init.CRCLength = SPI_CRC_LENGTH_DATASIZE;
567  hspil.Init.NSSPMode = SPI_NSS_PULSE_DISABLE;
568  if (HAL_SPI_Init(&hspil) != HAL_OK)
569  {
570    Error_Handler();
571  }
572  /* USER CODE BEGIN SPI1_Init 2 */
573
574  /* USER CODE END SPI1_Init 2 */
575
576 }
577
578 /**
579  * @brief TIM8 Initialization Function
580  * @param None
581  * @retval None
582  */
583 static void MX_TIM8_Init(void)
584 {
585
586  /* USER CODE BEGIN TIM8_Init 0 */
587
588  /* USER CODE END TIM8_Init 0 */
589
590  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
591  TIM_MasterConfigTypeDef sMasterConfig = {0};

```

```

592 TIM_OC_InitTypeDef sConfigOC = {0};
593 TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig = {0};
594
595 /* USER CODE BEGIN TIM8_Init 1 */
596
597 /* USER CODE END TIM8_Init 1 */
598 htim8.Instance = TIM8;
599 htim8.Init.Prescaler = 0;
600 htim8.Init.CounterMode = TIM_COUNTERMODE_CENTERALIGNED1;
601 htim8.Init.Period = 800;
602 htim8.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
603 htim8.Init.RepetitionCounter = 0;
604 htim8.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
605 if (HAL_TIM_Base_Init(&htim8) != HAL_OK)
606 {
607     Error_Handler();
608 }
609 sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
610 if (HAL_TIM_ConfigClockSource(&htim8, &sClockSourceConfig) != HAL_OK)
611 {
612     Error_Handler();
613 }
614 if (HAL_TIM_PWM_Init(&htim8) != HAL_OK)
615 {
616     Error_Handler();
617 }
618 sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
619 sMasterConfig.MasterOutputTrigger2 = TIM_TRGO2_RESET;
620 sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
621 if (HAL_TIMEx_MasterConfigSynchronization(&htim8, &sMasterConfig) != ...
622     HAL_OK)
623 {
624     Error_Handler();
625 }
626 sConfigOC.OCMode = TIM_OC_MODE_ASSYMETRIC_PWM2;
627 sConfigOC.Pulse = 400;
628 sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
629 sConfigOC.OCNPolarity = TIM_OCNPOLARITY_HIGH;
630 sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
631 sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
632 sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
633 if (HAL_TIM_PWM_ConfigChannel(&htim8, &sConfigOC, TIM_CHANNEL_2) != ...
634     HAL_OK)
635 {
636     Error_Handler();
637 }
638 sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;
639 sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
640 sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
641 sBreakDeadTimeConfig.DeadTime = 0;
642 sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;

```

```

641 sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
642 sBreakDeadTimeConfig.BreakFilter = 0;
643 sBreakDeadTimeConfig.BreakAFMode = TIM_BREAK_AFMODE_INPUT;
644 sBreakDeadTimeConfig.Break2State = TIM_BREAK2_DISABLE;
645 sBreakDeadTimeConfig.Break2Polarity = TIM_BREAK2POLARITY_HIGH;
646 sBreakDeadTimeConfig.Break2Filter = 0;
647 sBreakDeadTimeConfig.Break2AFMode = TIM_BREAK_AFMODE_INPUT;
648 sBreakDeadTimeConfig.AutomaticOutput = TIM_AUTOMATICOUTPUT_DISABLE;
649 if (HAL_TIMEx_ConfigBreakDeadTime(&htim8, &sBreakDeadTimeConfig) != ...
    HAL_OK)
650 {
651     Error_Handler();
652 }
653 /* USER CODE BEGIN TIM8_Init 2 */
654
655 /* USER CODE END TIM8_Init 2 */
656 HAL_TIM_MspPostInit(&htim8);
657
658 }
659
660 /**
661  * @brief USART3 Initialization Function
662  * @param None
663  * @retval None
664  */
665 static void MX_USART3_UART_Init(void)
666 {
667
668     /* USER CODE BEGIN USART3_Init 0 */
669
670     /* USER CODE END USART3_Init 0 */
671
672     /* USER CODE BEGIN USART3_Init 1 */
673
674     /* USER CODE END USART3_Init 1 */
675     huart3.Instance = USART3;
676     huart3.Init.BaudRate = 115200;
677     huart3.Init.WordLength = UART_WORDLENGTH_8B;
678     huart3.Init.StopBits = UART_STOPBITS_1;
679     huart3.Init.Parity = UART_PARITY_NONE;
680     huart3.Init.Mode = UART_MODE_TX_RX;
681     huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
682     huart3.Init.OverSampling = UART_OVERSAMPLING_16;
683     huart3.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
684     huart3.Init.ClockPrescaler = UART_PRESCALER_DIV4;
685     huart3.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
686     if (HAL_UART_Init(&huart3) != HAL_OK)
687     {
688         Error_Handler();
689     }
690     if (HAL_UARTEx_SetTxFifoThreshold(&huart3, UART_TXFIFO_THRESHOLD_1_8) ...

```

```

        != HAL_OK)
691  {
692    Error_Handler();
693  }
694  if (HAL_UARTEx_SetRxFifoThreshold(&huart3, UART_RXFIFO_THRESHOLD_1_8) ...
        != HAL_OK)
695  {
696    Error_Handler();
697  }
698  if (HAL_UARTEx_DisableFifoMode(&huart3) != HAL_OK)
699  {
700    Error_Handler();
701  }
702  /* USER CODE BEGIN USART3_Init 2 */
703
704  /* USER CODE END USART3_Init 2 */
705
706  }
707
708  /**
709   * @brief USB Initialization Function
710   * @param None
711   * @retval None
712   */
713  static void MX_USB_PCD_Init(void)
714  {
715
716    /* USER CODE BEGIN USB_Init 0 */
717
718    /* USER CODE END USB_Init 0 */
719
720    /* USER CODE BEGIN USB_Init 1 */
721
722    /* USER CODE END USB_Init 1 */
723    hpcd_USB_FS.Instance = USB;
724    hpcd_USB_FS.Init.dev_endpoints = 8;
725    hpcd_USB_FS.Init.speed = PCD_SPEED_FULL;
726    hpcd_USB_FS.Init.phy_itface = PCD_PHY_EMBEDDED;
727    hpcd_USB_FS.Init.Sof_enable = DISABLE;
728    hpcd_USB_FS.Init.low_power_enable = DISABLE;
729    hpcd_USB_FS.Init.lpm_enable = DISABLE;
730    hpcd_USB_FS.Init.battery_charging_enable = DISABLE;
731    if (HAL_PCD_Init(&hpcd_USB_FS) != HAL_OK)
732    {
733      Error_Handler();
734    }
735    /* USER CODE BEGIN USB_Init 2 */
736
737    /* USER CODE END USB_Init 2 */
738
739  }

```

```

740
741 /**
742  * @brief GPIO Initialization Function
743  * @param None
744  * @retval None
745  */
746 static void MX_GPIO_Init(void)
747 {
748     GPIO_InitTypeDef GPIO_InitStruct = {0};
749
750     /* GPIO Ports Clock Enable */
751     __HAL_RCC_GPIOA_CLK_ENABLE();
752     __HAL_RCC_GPIOB_CLK_ENABLE();
753     __HAL_RCC_GPIOC_CLK_ENABLE();
754     __HAL_RCC_GPIOD_CLK_ENABLE();
755
756     /*Configure GPIO pin Output Level */
757     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
758
759     /*Configure GPIO pin Output Level */
760     HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1 | GPIO_PIN_5 | GPIO_PIN_7, ...
761         GPIO_PIN_RESET);
762
763     /*Configure GPIO pin Output Level */
764     HAL_GPIO_WritePin(GPIOC, GPIO_PIN_12, GPIO_PIN_RESET);
765
766     /*Configure GPIO pin Output Level */
767     HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_RESET);
768
769     /*Configure GPIO pin : PA4 */
770     GPIO_InitStruct.Pin = GPIO_PIN_4;
771     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
772     GPIO_InitStruct.Pull = GPIO_NOPULL;
773     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
774     HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
775
776     /*Configure GPIO pins : PB1 PB5 PB7 */
777     GPIO_InitStruct.Pin = GPIO_PIN_1 | GPIO_PIN_5 | GPIO_PIN_7;
778     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
779     GPIO_InitStruct.Pull = GPIO_NOPULL;
780     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
781     HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
782
783     /*Configure GPIO pin : PC12 */
784     GPIO_InitStruct.Pin = GPIO_PIN_12;
785     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
786     GPIO_InitStruct.Pull = GPIO_NOPULL;
787     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
788     HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
789
790     /*Configure GPIO pin : PD2 */

```

```

790     GPIO_InitStruct.Pin = GPIO_PIN_2;
791     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
792     GPIO_InitStruct.Pull = GPIO_NOPULL;
793     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
794     HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);
795
796 }
797
798 /* USER CODE BEGIN 4 */
799
800
801 /* USER CODE END 4 */
802
803 /* USER CODE BEGIN Header_TransmitTask */
804 /**
805  * @brief Function implementing the Transmit thread.
806  * @param argument: Not used
807  * @retval None
808  */
809 /* USER CODE END Header_TransmitTask */
810 void TransmitTask(void *argument)
811 {
812     /* USER CODE BEGIN 5 */
813     uint16_t raw;
814     char msg[20];
815
816     uint8_t spi_buf[1];
817     uint16_t res;
818
819     /* Infinite loop */
820     for(;;)
821     {
822
823         //
824         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
825
826         HAL_ADC_Start(&hadc1);
827         raw = HAL_ADC_GetValue(&hadc1);
828
829         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
830
831         sprintf(msg, "%d\r\n", raw);
832         HAL_UART_Transmit(&huart3, (uint8_t*)msg, strlen(msg), ...
            HAL_MAX_DELAY);
833
834         //
835
836         /*
837         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
838         HAL_SPI_Receive(&hspi1, (uint8_t *)spi_buf, 2, HAL_MAX_DELAY);
839         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);

```

```

840
841     res = spi_buf[0];
842     //Store the first byte into the result variable
843     res = res << 8;
844     //Shift the
845     first byte left 8 bits.
846     res = res | spi_buf[1];
847     //Combine the two bytes into a 16 bit word (stored in a 32 bit
848     variable) using OR function
849     res = res >> 1;
850     res = res & 0b0000111111111111;
851
852     sprintf(msg, "%d \r\n", res);
853     //sprintf(msg, "%d %d %d \r\n", (unsigned int) spi_buf[0], ...
854     (unsigned int) spi_buf[1], (unsigned int) spi_buf[2]);
855     HAL_UART_Transmit(&huart3, (uint8_t*)msg, strlen(msg), ...
856     HAL_MAX_DELAY);
857
858 */
859
860     osDelay(1);
861 }
862
863 osThreadTerminate(NULL);
864
865 /* USER CODE END 5 */
866 }
867
868 /* USER CODE BEGIN Header_BlinkyTask */
869 /**
870 * @brief Function implementing the Blinky thread.
871 * @param argument: Not used
872 * @retval None
873 */
874 void BlinkyTask(void *argument)
875 {
876     /* USER CODE BEGIN BlinkyTask */
877     /* Infinite loop */
878     for(;;)
879     {
880         HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
881         osDelay(500);
882     }
883     /* USER CODE END BlinkyTask */
884 }
885
886 /* USER CODE BEGIN Header_ControlTask */
887 /**
888 * @brief Function implementing the Control thread.

```

```

885 * @param argument: Not used
886 * @retval None
887 */
888 /* USER CODE END Header_ControlTask */
889 void ControlTask(void *argument)
890 {
891     /* USER CODE BEGIN ControlTask */
892
893     float Ts = 0.0001;
894
895     float vout, vref;
896     float VNM = 20;
897     float adc_volts, Out;
898
899     float iout;
900     float INM = 5;
901     float adc_amps, OutI;
902
903
904     double Kp = 0.0119419388;
905     double Ki = 238.8387774362;
906     double Kd = 0;
907
908     double PTerm = 0.0, ITerm = 0.0, DTerm;
909     double error = 0.0, error_1 = 0.0;
910
911
912     //PI Amps
913     double KpI = 0.0049868682;
914     double KiI = 99.7373648649;
915     double KdI = 0.000024875;
916
917     double PTermI = 0.0, ITermI = 0.0, DTermI;
918     double errorI = 0.0, error_1I = 0.0;
919
920     int adcbuf, ramp = 0;
921
922     int phase = 0;
923
924     vref = 5;
925     Kp=Kp;
926     Ki = Ki*Ts;
927     Kd = Kd/Ts;
928
929     KpI=KpI;
930     KiI = KiI*Ts;
931     KdI = KdI/Ts;
932
933     /* Infinite loop */
934     for(;;)
935     {

```

```

936     adcbuf = 0;
937     for(int i = 0; i < 8; i++)
938     {
939         HAL_ADC_Start(&hadc1);
940         adcbuf = HAL_ADC_GetValue(&hadc1);
941         adc_amps = adc_amps + adcbuf;
942     }
943
944     adc_amps = adc_amps * 0.125;
945
946     /*
947     phase = adc_volts * 0.09765625; // adc_volts * (400/4096)
948
949     SET_BIT(TIM8->CR1, TIM_CR1_UDIS);
950
951     TIM8->CCR1 = 400 - phase;
952     TIM8->CCR2 = 400 + phase;
953
954     CLEAR_BIT(TIM8->CR1, TIM_CR1_UDIS);
955
956     if(phase == 400)
957     {phase = 0;}
958
959     */
960
961     /*
962     phase = adc_amps * 6.103515625; // adc_volts * (400/4096)
963
964     if(phase ≥ 25000)
965     {phase = 25000;}
966
967     if(phase < 100)
968     {phase = 100;}
969
970     __HAL_HRTIM_SETCOMPARE(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
971         HRTIM_COMPAREUNIT_2, phase);
972
973     */
974
975     //-----CURRENT CONTROL-----
976
977     adc_amps = adc_amps * 0.000805664; // adc_amps * (3.3/4096)
978     adc_amps = adc_amps * 5.1; // adjusting to real amps
979
980     error_1I = errorI;
981     errorI = iref - adc_amps;
982
983     PTermI= KpI*errorI;
984
985     if(( ITermI<INM) && (ITermI>(-INM) )

```

```

986         ITermI = ITermI + (KiI*errorI);
987     else
988     {
989         if(ITermI ≥ INM)
990             ITermI = 0.95*INM;
991         if(ITermI ≤ (-INM))
992             ITermI = -0.95*INM;
993     }
994
995     DTermI = (errorI - error_1I) * KdI;
996
997     OutI= PTermI + ITermI + DTermI;
998
999
1000     phase = OutI * 5000; //Out*(25000/Iin)
1001
1002
1003     if (phase≥25000)
1004     {phase = 25000;}
1005
1006     if (phase<100)
1007     {phase = 100;}
1008
1009     __HAL_HRTIM_SETCOMPARE(&hhrtim1, HRTIM_TIMERINDEX_TIMER_C, ...
1010         HRTIM_COMPAREUNIT_2, phase);
1011
1012     osDelay(1);
1013 }
1014 /* USER CODE END ControlTask */
1015 }
1016
1017 /* USER CODE BEGIN Header_InputChangeTask */
1018 /**
1019  * @brief Function implementing the InputChange thread.
1020  * @param argument: Not used
1021  * @retval None
1022  */
1023 /* USER CODE END Header_InputChangeTask */
1024 void InputChangeTask(void *argument)
1025 {
1026     /* USER CODE BEGIN InputChangeTask */
1027     /* Infinite loop */
1028     float random[10] = {0.5, 1, 1, 0.3, 1.2, 0.5, 0.1, 0.3, 0.9, 0.4};
1029     int irand = 0;
1030     for(;;)
1031     {
1032         //-----SAWTOOTH INPUT-----
1033
1034         /*
1035         iref = iref + 0.001;

```

```

1036     if (iref > 1)
1037         iref = 0;
1038     osDelay(1);
1039     */
1040
1041     //-----RANDOM INPUT-----
1042     //
1043     irand++;
1044     if(irand > 10)
1045         irand = 0;
1046
1047     iref = random[irand];
1048
1049     osDelay(1000);
1050     //
1051     //-----FIXED INPUT-----
1052
1053     /*
1054     iref = 0.1;
1055     osDelay(10);
1056     */
1057 }
1058 /* USER CODE END InputChangeTask */
1059 }
1060
1061 /**
1062  * @brief Period elapsed callback in non blocking mode
1063  * @note This function is called when TIM6 interrupt took place, inside
1064  * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to ...
1065  * increment
1066  * a global variable "uwTick" used as application time base.
1067  * @param htim : TIM handle
1068  * @retval None
1069  */
1069 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
1070 {
1071     /* USER CODE BEGIN Callback 0 */
1072
1073     /* USER CODE END Callback 0 */
1074     if (htim->Instance == TIM6) {
1075         HAL_IncTick();
1076     }
1077     /* USER CODE BEGIN Callback 1 */
1078
1079     /* USER CODE END Callback 1 */
1080 }
1081
1082 /**
1083  * @brief This function is executed in case of error occurrence.
1084  * @retval None
1085  */

```

```

1086 void ErrorHandler(void)
1087 {
1088     /* USER CODE BEGIN ErrorHandler.Debug */
1089     /* User can add his own implementation to report the HAL error return ...
        state */
1090
1091     /* USER CODE END ErrorHandler.Debug */
1092 }
1093
1094 #ifdef USE_FULL_ASSERT
1095 /**
1096  * @brief Reports the name of the source file and the source line number
1097  *         where the assert_param error has occurred.
1098  * @param file: pointer to the source file name
1099  * @param line: assert_param error line source number
1100  * @retval None
1101  */
1102 void assert_failed(uint8_t *file, uint32_t line)
1103 {
1104     /* USER CODE BEGIN 6 */
1105     /* User can add his own implementation to report the file name and ...
        line number,
1106     tex: printf("Wrong parameters value: file %s on line %d\r\n", ...
        file, line) */
1107     /* USER CODE END 6 */
1108 }
1109 #endif /* USE_FULL_ASSERT */
1110
1111 /***** (C) COPYRIGHT STMicroelectronics *****/
        FILE*****/

```

A.2. PCB

Se muestra la PCB utilizada para el diseño del banco de pruebas. La placa está diseñada en doble cara, donde la cara inferior es el plano tierra que separa las señales analógicas de las digitales.

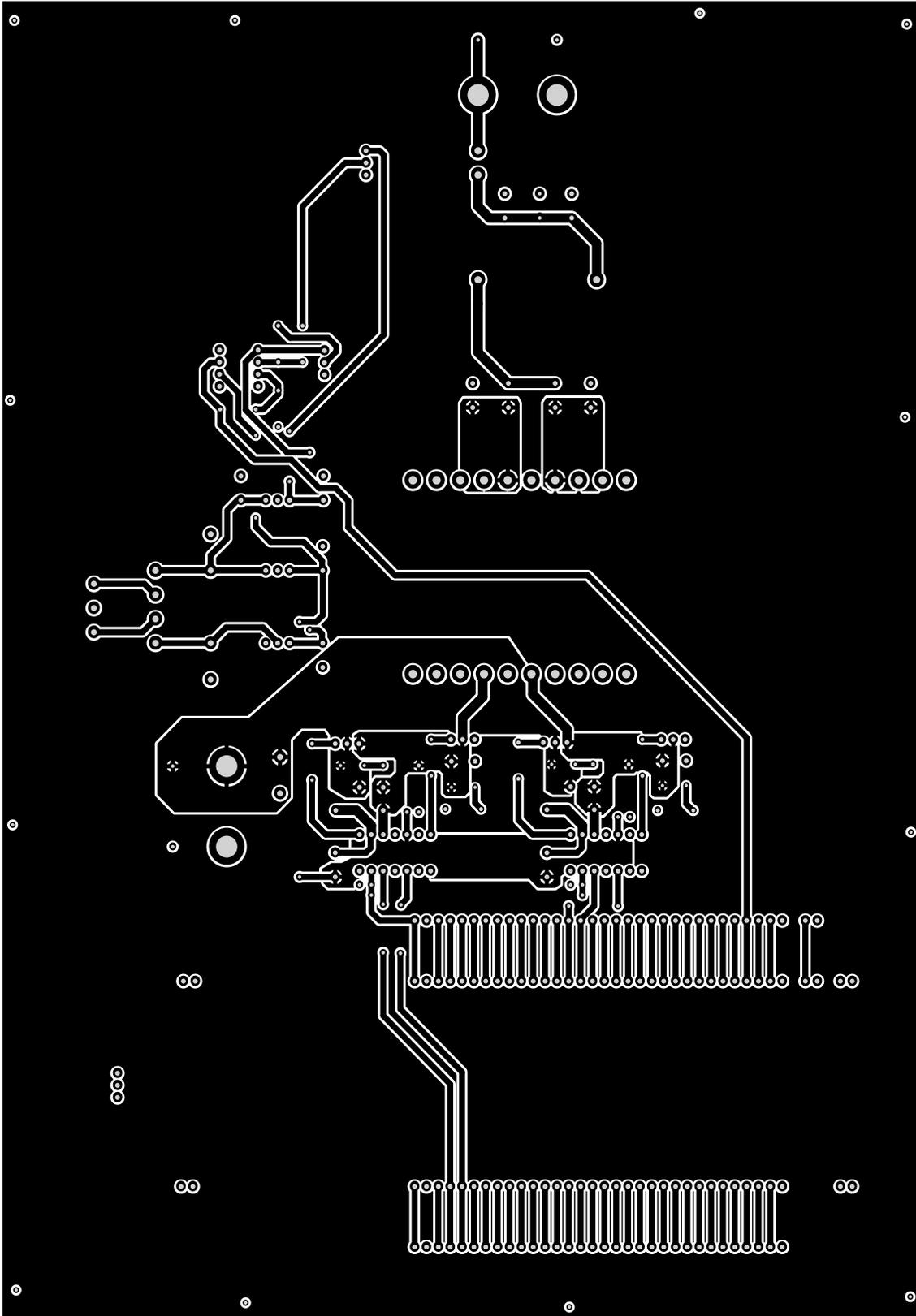


Figura A.1: PCB de prototipo, cara superior.
83

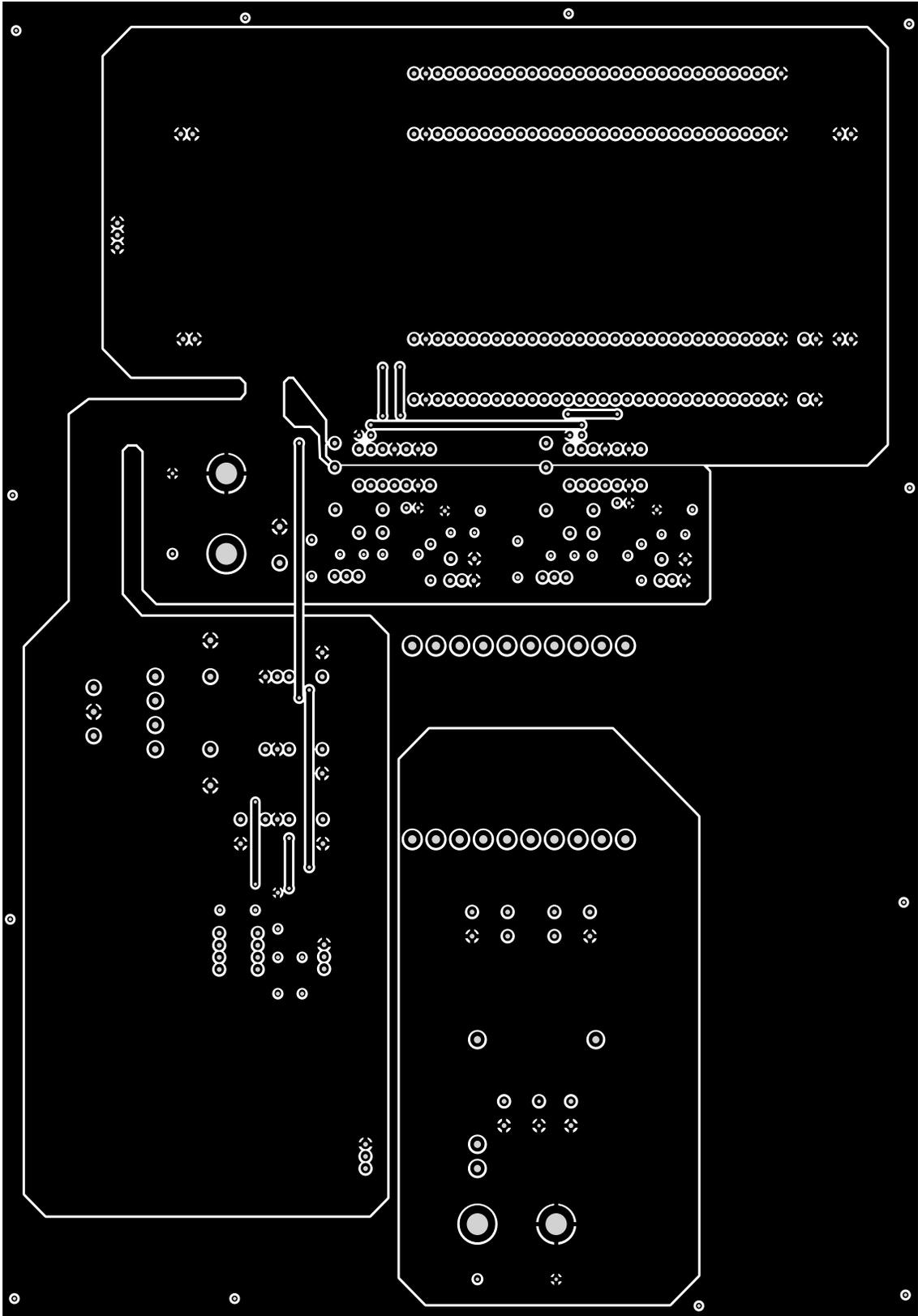


Figura A.2: PCB de prototipo, cara inferior.
84