



Universidad Autónoma de Querétaro
Facultad de Informática

Sistema de emulación por software embebido de
baterías Li-ion con bajo error

Tesis

Que como parte los requisitos para obtener el Grado de
Maestro en Sistemas Computacionales

Presenta

Daniel Martín Castellanos Martínez

Dirigido por:

Dr. Alberto Vázquez Cervantes

Querétaro, Qro. a 13 de agosto de 2023



Dirección General de Bibliotecas y Servicios Digitales
de Información



Sistema de emulación por software embebido de
baterías Li-ion con bajo error

por

Daniel Martin Castellanos Martinez

se distribuye bajo una [Licencia Creative Commons
Atribución-NoComercial-SinDerivadas 4.0
Internacional](#).

Clave RI: IFMAC-309321



Universidad Autónoma de Querétaro

Facultad de Informática

Maestría en sistemas computacionales

Sistema de emulación por software embebido de baterías

Li-ion con bajo error

Tesis

Que como parte los requisitos para obtener el Grado de

Maestro en Sistemas Computacionales

Presenta

Daniel Martín Castellanos Martínez

Dirigido por:

Dr. Alberto Vázquez Cervantes

Presidente: Dr. Alberto Vázquez Cervantes

Secretario: Dr. Ricardo Chaparro Sánchez

Vocal: M.C Arturo Gaona Cuadra

Suplente: Dr. Hugo Jiménez Hernández

Suplente: Dr. Julio Alejandro Romero González

Dedicatoria

A mi compañera de aventuras y apoyo inquebrantable,

En los momentos de incertidumbre, encontré en ti la fuerza que necesitaba para seguir adelante. Tu aliento constante y tu apoyo sincero han sido mi ancla en este viaje. Esta tesis no solo es el resultado de mi esfuerzo, sino también de tu confianza en mí. Gracias por todo.

A mí mamá, ejemplo de tenacidad y determinación. Tu constante apoyo moral y tus consejos valiosos me han guiado en cada paso de esta travesía académica. Sin tu amor incondicional, este logro no sería posible.

Agradecimientos

Al Dr. Alberto Vázquez Cervantes, mi director de tesis,

Agradezco su orientación en este proceso. Su tiempo y experiencia contribuyeron al enfoque que tomé en mi investigación.

Al Dr. Ricardo Chaparro Sánchez y al Maestro Arturo Gaona Cuadra,

Mi más sincero agradecimiento al Dr. Ricardo Chaparro por su inestimable ayuda en la redacción y estructuración de esta tesis. Sus sugerencias y revisión meticulosa dieron forma a mis ideas y mejoraron mi trabajo de manera significativa.

Al Maestro Arturo Gaona, cuyos conocimientos fueron un pilar fundamental en mi investigación. Sus enseñanzas abrieron puertas y permitieron la realización de esta labor.

Al Maestro Carlos Alberto Olmos Trejo, coordinador de la carrera,

Mi profundo agradecimiento por su apoyo continuo desde el inicio de mi carrera. Su guía en todo el programa ha sido esencial en este logro.

Al CONACYT (Consejo Nacional de Ciencia y Tecnología),

Deseo expresar mi sincero agradecimiento por la oportunidad que me brindaron al otorgarme la beca. Su apoyo financiero fue fundamental para poder enfocarme en mi investigación y culminar mi educación de manera exitosa.

Cada uno de ustedes ha dejado una huella indeleble en este logro, y su apoyo ha sido un faro que iluminó mi camino hacia la culminación de mi tesis.

Resumen

Las baterías de iones de litio (Li-ion) han ganado una popularidad significativa en los últimos años debido a su alta densidad de energía, ciclo de vida prolongado y baja tasa de autodescarga. Estas baterías se utilizan ampliamente en diversas aplicaciones, incluidas vehículos eléctricos, dispositivos electrónicos portátiles y sistemas de almacenamiento de energía a nivel de red (Di Rienzo et al., 2018; Tomasov et al., 2019). El rendimiento de las baterías Li-ion depende de varios factores, como la temperatura, el estado de carga y las condiciones de carga/descarga. Para garantizar el funcionamiento fiable de los sistemas alimentados por baterías, es esencial probar y validar el rendimiento de las baterías en diferentes condiciones de funcionamiento. Este proyecto de tesis presenta el desarrollo de un emulador de batería de Li-ion capaz de emular tanto el comportamiento de descarga como el de carga. Este microcontrolador mide la corriente y controla el voltaje de salida y el modo del emulador. Para emular el comportamiento de descarga, se utiliza un modelo 3RC Thévenin, mientras que se utiliza un modelo 2RC Thévenin para emular el comportamiento de carga de una batería Li-ion. Para modelar una batería real, se empleó una caracterización de potencia de pulso híbrido y los parámetros de Thévenin se estimaron utilizando el bloque de batería de circuito equivalente de Matlab. La eficacia del emulador propuesto se demuestra a través de resultados experimentales, que muestran que puede emular con precisión y eficacia el comportamiento de descarga y carga de una batería Li-ion. El emulador propuesto tiene aplicaciones potenciales para probar y validar sistemas de gestión de baterías y dispositivos que funcionan con baterías. En general, el emulador desarrollado ofrece una solución rentable y fiable para emular el comportamiento de una batería Li-ion.

Palabras clave: emulador de batería; modelado; arquitectura; Thévenin.

Abstract

Lithium-ion (Li-ion) batteries have gained significant popularity in recent years due to their high energy density, long cycle life, and low self-discharge rate. These batteries are widely used in various applications, including electric vehicles, portable electronic devices, and grid-level energy storage systems (Di Rienzo et al., 2018; Tomasov et al., 2019). The performance of Li-ion batteries is influenced by various factors, including temperature, state of charge, and charging/discharging conditions. To ensure the reliable operation of battery-powered systems, it is essential to test and validate the performance of Li-ion batteries under different operating conditions. This thesis dissertation presents the development of a Li-ion battery emulator capable of emulating both discharge and charge behavior. The emulator uses a DC-DC buck converter and an electronic load, with a proposed architecture of a personal computer running the battery model and communicating with a microcontroller via a serial interface. This microcontroller measures the current and controls the output voltage and mode of the emulator. To emulate the discharge behavior, a 3RC Thévenin model was used, while a 2RC Thévenin model was used to emulate the charging behavior of a Li-ion battery. To model a real battery, a hybrid pulse power characterization was employed, and the Thévenin parameters were estimated using the Matlab Equivalent Circuit Battery block. The effectiveness of the proposed emulator is demonstrated through experimental results, which show that it can accurately and effectively emulate the discharge and charge behavior of a Li-ion battery. The proposed emulator has potential applications in testing and validating battery management systems and battery-operated devices. Overall, the developed emulator offers a cost-effective and reliable solution for emulating the behavior of a Li-ion battery.

Keywords: battery emulator; modeling; architecture; Thévenin.

Índice

Resumen	4
Abstract.....	5
1. Introducción.....	8
1.2 Hipótesis	9
1.3 Objetivos de investigación.....	9
1.3.1 Objetivo general	9
1.3.2 Objetivos específicos.....	9
2. Revisión de literatura (estado del arte).....	11
2.1 Revisión sistemática: introducción	11
2.2 Revisión sistemática: Método.....	11
2.3 Revisión sistemática: Resultados.....	12
3. Marco teórico.....	14
3.1 Modelo de batería	14
3.2 Controlador usado.....	18
3.3 Modo de operación	19
3.4 Porcentaje de error obtenido	20
3.5 Discusión	21
4. Metodología.....	24
4.1 Campo de aplicación.....	28
4.2 Modelo de batería utilizado	30
4.3 Batería utilizada	31
4.4 Proceso de extracción de parámetros y validación	32
4.4.1 Parámetro de capacidad de la batería	32
4.4.2 Parámetros de Voc de descarga.....	34
4.4.3 Parámetros de Voc de carga	47
4.5 Arquitectura	59
4.6 Circuito utilizado	60
4.7 Programación de GUI.....	70
4.8 Programación de microcontrolador.....	75

5. Mediciones y resultados	83
5.1 Diseño experimental	83
5.2 Resultados	85
5.3 Impacto	86
5.3.1 Sistemas de gestión de batería.....	86
5.3.2 Dispositivos que funcionan con baterías	87
5.3.3 Educación e investigación	87
5.3.4 Impacto ambiental	87
5.4 Direcciones futuras	87
6. Conclusiones.....	88
7. Referencias	90

1. Introducción

En la sección de introducción de esta tesis, se expone y desarrolla la problemática actual del diseño y prueba de sistemas con baterías, resaltando la utilidad de los emuladores de batería en este contexto. A continuación, se presenta la hipótesis central de esta tesis, que propone el desarrollo de un sistema emulado de batería Li-ion para reducir los costos y riesgos asociados a estas pruebas, manteniendo una precisión adecuada.

En la sección de objetivos, se establece la meta general de implementar un emulador de batería eficiente y de bajo costo basado en un modelo Thévenin y una arquitectura distribuida y se describen varios objetivos específicos.

1.1 Planteamiento del problema

El diseño de sistemas que tengan integrada una batería es un proceso incómodo, costoso y lento. (Luna et al., 2021). Esto debido a que, para iniciar pruebas en baterías reales estas deben precargarse y precalentarse para alcanzar las condiciones iniciales definidas (König et al., 2014). De igual forma, los efectos del envejecimiento de la batería pueden limitar la repetibilidad de las pruebas. Las diferentes celdas varían ampliamente dependiendo de varios factores, como resistencia interna, estado de carga (SOC), voltaje de circuito abierto (Voc) y estado de salud (SOH). Al mismo tiempo, para probar las funciones de diagnóstico de una batería, igualmente se deben emular algunas fallas, por ejemplo, sobrecarga, sobre descarga y diferentes rangos de temperatura. Las celdas de batería reales obviamente no son adecuadas para estas pruebas, considerando los problemas de seguridad (Dai et al., 2013). Además, estas características cambian con el envejecimiento de una celda de batería por lo que, conforme el proyecto va avanzando en el proceso de desarrollo y validación, sería necesario reemplazar las baterías por otras nuevas para mantener la funcionalidad del sistema experimental (Michalczuk et al., 2014).

Por lo tanto, los emuladores de batería se han vuelto muy populares en los últimos años, ya que permiten probar la batería en diferentes condiciones sin que la preparación, seguridad, el mantenimiento o el envejecimiento del sistema sean factores por considerar. Los emuladores se pueden usar para replicar el comportamiento de baterías de diferentes químicas y potencias nominales (Hidalgo-León et al., 2020). Tal emulador de batería proporciona los siguientes beneficios (König et al., 2014):

1. Se pueden hacer pruebas del sistema completo antes de que esté disponible un prototipo de batería.
2. Se puede cambiar el estado de la batería emulada al instante y sin necesidad de preacondicionamiento o cambio de baterías voluminosas lo cual requiere mucho tiempo.
3. Se obtiene repetibilidad exacta de las pruebas.
4. Se puede cambiar de configuración dependiendo del tipo de batería, solo cambiando el modelo.
5. Se pueden probar condiciones imposibles de probar en una batería real.

1.2 Hipótesis

El desarrollo de un sistema de batería Li-ion emulado por software embebido de arquitectura distribuida permite reducir los riesgos de prueba y costos de producción de circuitos con batería, garantizando un bajo error promedio y error máximo.

1.3 Objetivos de investigación

1.3.1 Objetivo general

Implementar un emulador de batería preciso, eficiente y de bajo costo, usando un modelo Thévenin, basado en Orcioni et al. (2017), en conjunto con un dispositivo con arquitectura distribuida entre una computadora personal y un circuito con un convertidor buck y una carga electrónica variable, controlado por microcontrolador, basado en Daniil & Drury (2016) y Restrepo et al. (2012), que emule el comportamiento de voltaje y corriente en situaciones de carga y descarga de una batería real. Con el propósito de generar un banco de prueba en circuitos de aplicación de software embebido que requieren dispositivos que trabajan con baterías, con la finalidad de reducir los riesgos, logrando un bajo error promedio y error máximo, sin la necesidad de replicar la inseguridad que implica someter las baterías reales a condiciones de prueba.

1.3.2 Objetivos específicos

- Comprobar y modelar la propuesta de caracterización de Orcioni et al. (2017) como un modelo funcional para la emulación de baterías Li-ion.

- Diseñar y probar un conjunto de arquitectura distribuida entre una computadora y un microcontrolador, que controlará un circuito Buck, como elementos que permitirán la emulación del dispositivo.
- Diseñar un sistema embebido que permita la emulación práctica en un banco de pruebas de baterías Li-ion, con el fin de evaluar el error promedio y el error máximo.

2. Revisión de literatura (estado del arte)

Para la construcción de esta revisión de literatura se realizó una revisión sistemática, estructurada en las siguientes subsecciones:

- **Introducción:** Se explicará el problema principal al hacer pruebas en un sistema con una batería real y la necesidad de implementar un emulador de batería.
- **Método:** Se explicará la forma, los procedimientos y las preguntas a resolver en la realización de esta revisión sistemática.
- **Resultados:** Se expondrán las respuestas a las preguntas planteadas en el método y los resultados generales de esta revisión sistemática.
- **Discusión:** Se explicará cómo los resultados de esta revisión sistemática afectarán al proyecto planeado para la implementación de un emulador de batería.

2.1 Revisión sistemática: introducción

Ésta revisión sistemática de la implementación de un emulador de batería se enfoca en las últimas tendencias del tema, comparación de modelos, circuitos y arquitecturas utilizadas. Para así entender el contexto actual del tema y conocer la pertinencia de una implementación de un emulador de batería con estas técnicas en un entorno de desarrollo software automotriz. Conociendo las técnicas más adecuadas y los problemas para tener en cuenta a la hora de desarrollar dicho dispositivo.

2.2 Revisión sistemática: Método

En ésta de revisión de literatura se investiga el contexto del tema; implementación de un emulador de batería, a través de la técnica de revisión sistemática, realizando una búsqueda en los repositorios de investigación Redalyc, Research Gate y Google Scholar bajo el siguiente criterio de búsqueda:

- (“lithium ion” OR li ion) “battery” “emulator”

Dentro de esta búsqueda se toma en cuenta los artículos escritos en inglés y español desde los años 2012-2022 para tener un contexto amplio de los artículos relevantes en la actualidad,

antecedentes, objetivos, metodologías, resultados y conclusiones. De igual forma se toman en cuenta algunos artículos anteriores para tener un contexto del origen de ciertos conceptos.

La literatura se clasifica en publicaciones por año centradas en emulación de un entorno automotriz y se toma en cuenta el error obtenido en la implementación de un emulador.

Se identifican las variables que representen de forma cualitativa la problemática, es decir, aquellas que muestren problemas o hallazgos relevantes, áreas de oportunidad, así como modelos y circuitos utilizados.

Preguntas de investigación del análisis cuantitativo

- Número de publicaciones por año
- Número de publicaciones centradas en emulación de baterías en un entorno automotriz
- Porcentaje de error obtenido en la implementación de un emulador

Preguntas de investigación del análisis cualitativo

- Problemas relevantes
- Modelos utilizados
- Circuitos utilizados
- Hallazgos relevantes
- Áreas de oportunidad

2.3 Revisión sistemática: Resultados

Bajo esta metodología, se toman en cuenta 39 artículos, entre los que se encuentran artículos de desarrollo e implementación de dispositivos, revisiones sistemáticas e investigaciones comparativas entre diferentes modelos y circuitos.

Estos artículos se toman en cuenta para la revisión sistemática porque sus contenidos van acorde al tema y pueden sumar al conocimiento necesario para desarrollar un emulador de batería modular y de bajo costo. Dichos artículos están distribuidos en publicaciones por año como se muestra en la Tabla 2.1.

Tabla 2.1

Distribución de artículos evaluados por año

Año	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
Artículos	2	4	6	3	7	6	5	2	1	2	1

De las publicaciones tomadas en cuenta para la revisión sistemática 29 son artículos que hablan sobre la construcción de un emulador de baterías, de los cuales 10 tienen una aplicación o validan su trabajo enfocado al área automotriz. (Barreras et al., 2015; Cabello et al., 2015, 2017; Dai et al., 2013; König et al., 2013; Mathew et al., 2017; T. Mesbahi et al., 2013; Tedjani Mesbahi et al., 2018; Pebriyanti, 2013; Zhang et al., 2016).

3.Marco teórico

3.1 Modelo de batería

Así mismo se tomaron en cuenta artículos donde se implementan diferentes modelos para emular baterías, se han presentado proyectos basados en una variedad de modelos de baterías y se pueden clasificar en tres categorías: modelos electroquímicos, modelos matemáticos o experimentales y modelos de circuitos eléctricos equivalentes (Dung et al., 2016).

1. Electroquímicos: Los modelos electroquímicos son modelos precisos y complejos, estos requieren una curva de aprendizaje prolongada (Dung et al., 2016). En su mayoría utilizan modelos constituidos por ecuaciones diferenciales no lineales, por lo cual es necesario una capacidad de computación mucho mayor comparado con otros modelos.

En esta revisión sistemática se encontraron 2 proyectos que se basan en un modelo electroquímico para diseñar un emulador de baterías (Cheng & Chen, 2017; Wang et al., 2018).

2. Matemáticos o experimentales: Los modelos matemáticos estiman el tiempo de ejecución de la batería o la capacidad restante con ecuaciones matemáticas específicas que se derivan de mediciones empíricas o las reacciones cinéticas y los procesos de difusión dentro de una batería. Sin embargo, los métodos matemáticos carecen de factores importantes como el comportamiento dinámico de la batería o las respuestas transitorias de voltaje de los cambios de corriente (Dung et al., 2016).

En esta revisión sistemática se encontraron 8 proyectos que se basan en un modelo matemático para diseñar un emulador de baterías (Barreras et al., 2016; Boles et al., 2017; Cabello et al., 2015, 2017; Di Rienzo et al., 2018; Tedjani Mesbahi et al., 2018; Restrepo et al., 2012; Wu et al., 2016).

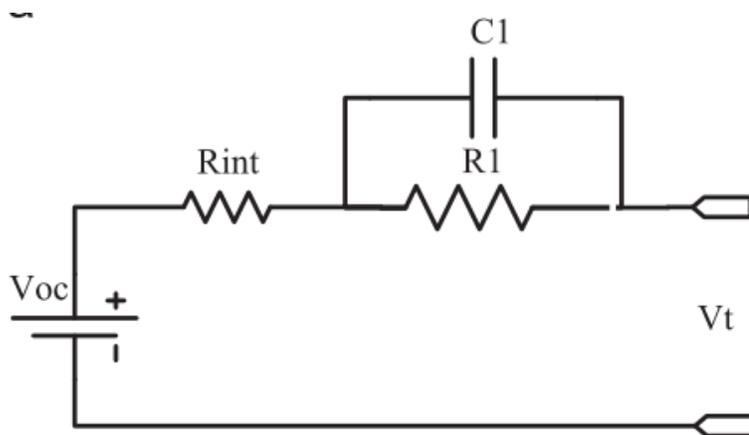
3. Circuitos eléctricos equivalentes: La mayoría de estos modelos eléctricos se clasifican en tres categorías básicas; Thévenin, impedancia y modelos basados en tiempo de ejecución (Chen & Rincón-Mora, 2006).

3.1. Thévenin: El modelo de batería de Thévenin está compuesto por una fuente de tensión ideal (V_{oc}), una resistencia interna (R_{int}), una resistencia de sobretensión (R_1) y un condensador (C_1). La configuración de este modelo se muestra en la Figura 3.1. C_1 y R_1 muestran la capacitancia de la placa paralela y la resistencia

implicada por la resistencia de contacto de la placa, respectivamente (Mousavi G. & Nikdel, 2014).

Figura 3.1

Modelo de batería Thévenin



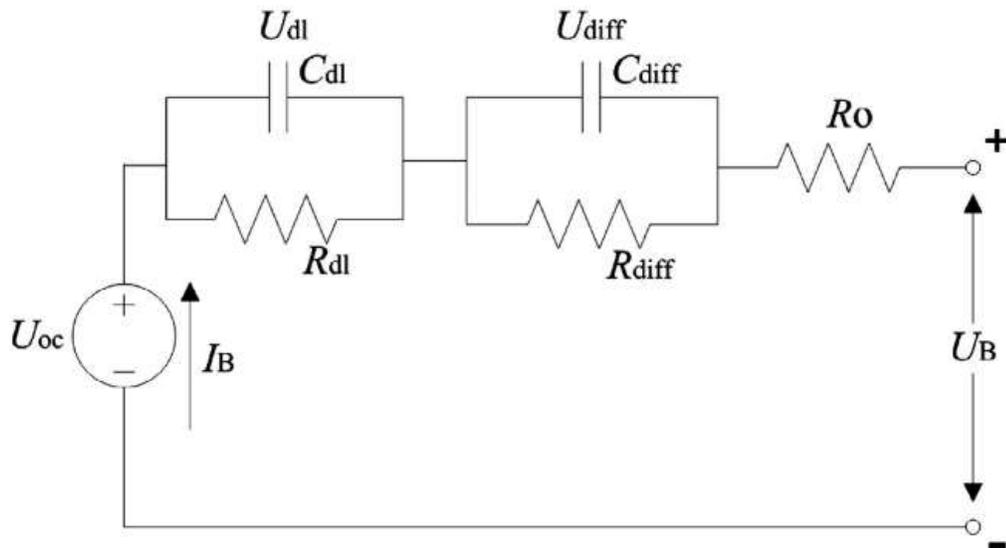
Nota. De Various battery models for various simulation studies and applications (p.479), por S.M. Mousavi & M. Nikdel, 2014, Renewable and Sustainable Energy Reviews.

En esta revisión sistemática se encontraron dos proyectos que se basan en el modelo de Thévenin para diseñar un emulador de baterías (Luna et al., 2021), (Mathew et al., 2017).

- 3.2. Impedancia: Los elementos de impedancia en este modelo se parametrizan mediante mediciones de espectroscopia de impedancia electroquímica (EIS) para lograr la respuesta de corriente alterna de una celda en ciertos intervalos de frecuencia. La configuración de este modelo se muestra en la Figura 3.2. Los resultados se representan en un gráfico llamado diagrama de Nyquist mostrado en la Figura 3.3 (Mousavi G. & Nikdel, 2014).

Figura 3.2

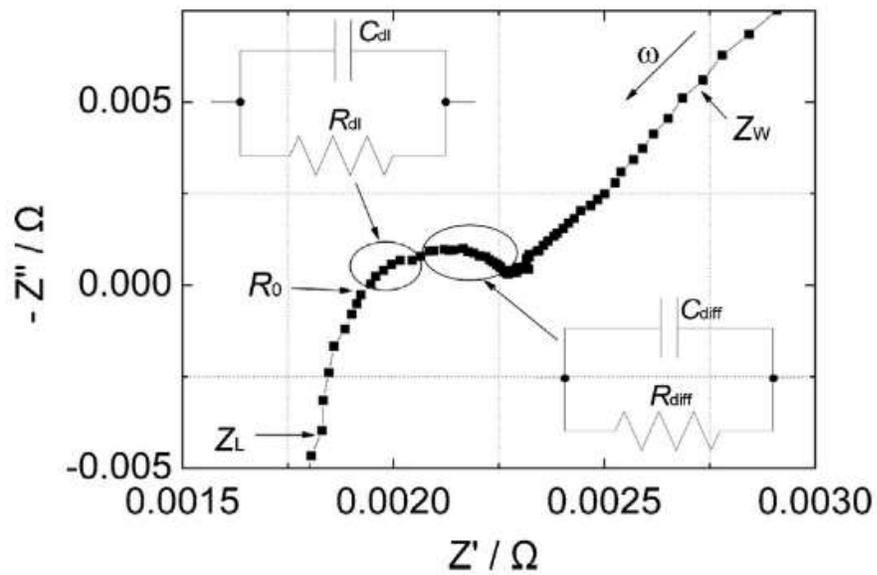
Modelo de batería de impedancia



Nota. De Cell-BMS validation with a hardware-in-the-loop simulation of lithium-ion battery cells for electric vehicles (p.177), por H. Dai et al., 2013, International Journal of Electrical Power and Energy Systems.

Figura 3.3

Diagrama de Nyquist



Nota. De Cell-BMS validation with a hardware-in-the-loop simulation of lithium-ion battery cells for electric vehicles (p.177), por H. Dai et al., 2013, International Journal of Electrical Power and Energy Systems.

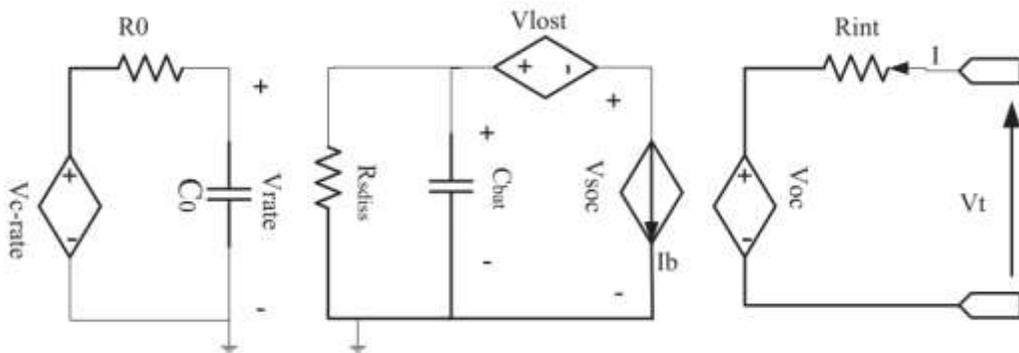
En esta revisión sistemática se encontraron 9 proyectos que se basan en el modelo de impedancia para diseñar un emulador de baterías (R. Di Rienzo et al., 2022), (Buccolini et al., 2018), (Orcioni et al., 2017), (Daniil & Drury, 2016), (Barreras et al., 2015), (Van Sterkenburg et al., 2014), (König et al., 2014), (Pebriyanti, 2013), (Dai et al., 2013).

3.3. Modelos basados en el tiempo de ejecución: Son modelos complejos para la simulación del tiempo de ejecución y la respuesta de corriente directa (DC) bajo una corriente de descarga constante.

Este modelo consta de tres circuitos separados presentados en la Figura 3.4. El circuito de la izquierda, indica la dependencia de la frecuencia de descarga. El circuito medio indica la dependencia del estado de carga y la tasa de descarga. Finalmente, el circuito derecho consta de un V_{oc} y una resistencia interna de la batería (R_{int}) (Mousavi G. & Nikdel, 2014).

Figura 3.4

Modelo de batería de tiempo de ejecución



Nota. De Various battery models for various simulation studies and applications (p.482), por S.M. Mousavi & M. Nikdel, 2014, Renewable and Sustainable Energy Reviews

En esta revisión sistemática se encontró un proyecto que se basa en el modelo de tiempo de ejecución para diseñar un emulador de baterías (König et al., 2013).

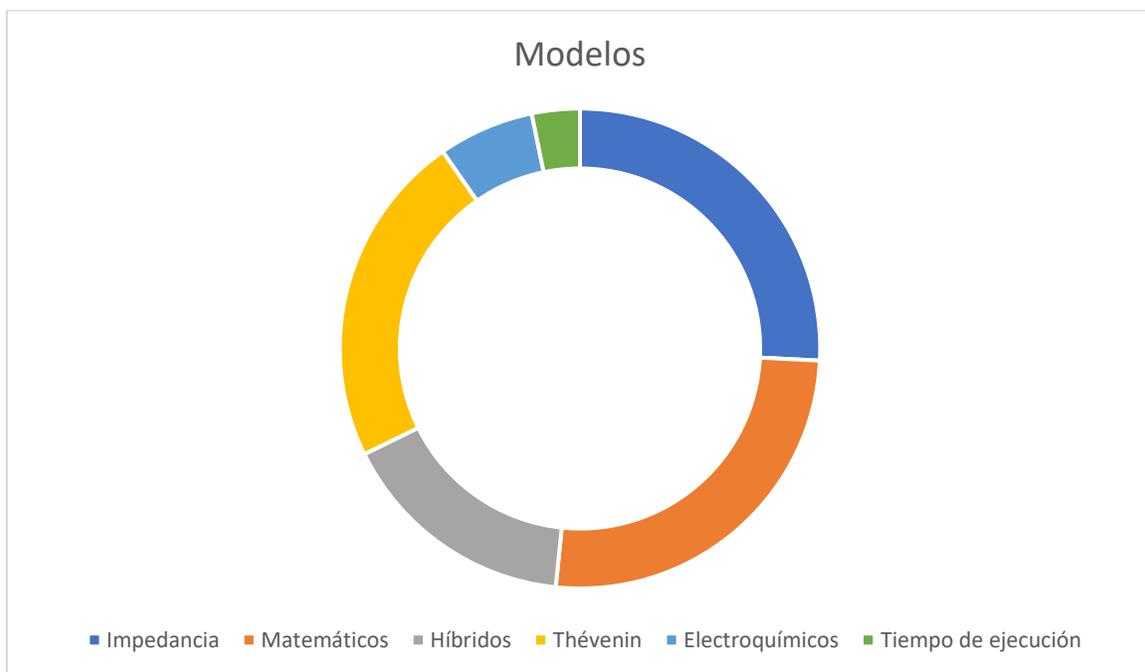
De igual manera se encuentran modelos híbridos entre diferentes tipos de modelos como:

- Thévenin con matemático (Hidalgo León et al., 2020).
- Impedancia con matemático (Dung et al., 2016).
- Thévenin con electroquímico (Michalczuk et al., 2014).
- Thévenin con tiempo de ejecución (T. Mesbahi et al., 2013).

Los resultados de distintos modelos de batería encontrados en los artículos tomados en cuenta se muestran en la Figura 3.5.

Figura 3.5

Tipo de modelos usados en revisión sistemática



3.2 Controlador usado

Por otro lado, en la revisión sistemática se toman en cuenta varios artículos que usan distintos circuitos para emular el comportamiento de una batería. Se toma en cuenta si se usan controladores enfocados para simulación tipo hardware in the loop (HIL), o un

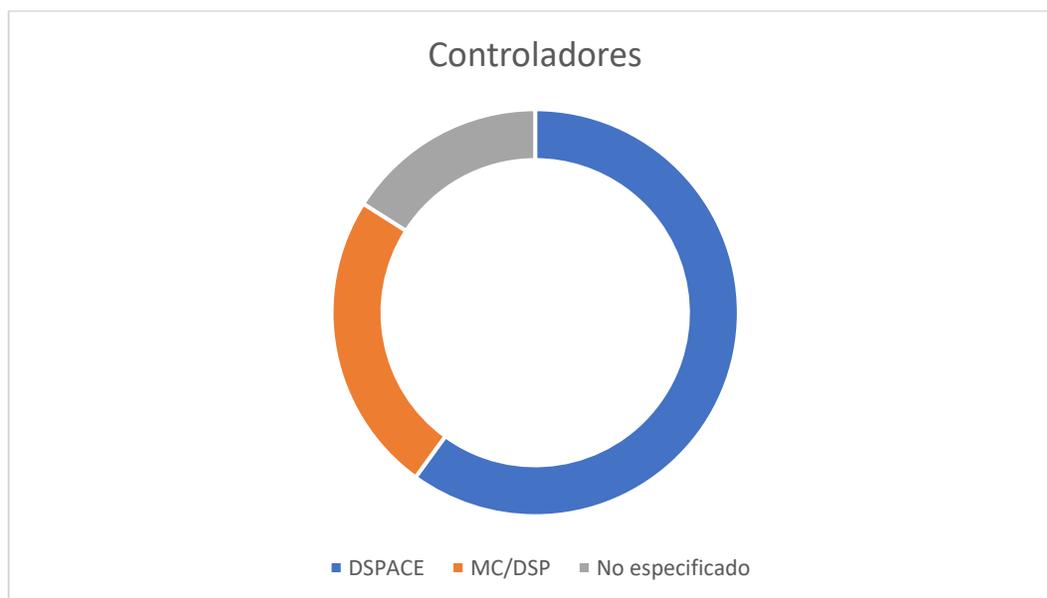
microcontrolador. Dentro de la literatura de los 25 proyectos que se tomaron en cuenta, se encontraron 15 que usan un controlador tipo HIL (Barreras et al., 2015, 2016; Cabello et al., 2015, 2017; Daniil & Drury, 2016; König et al., 2013, 2014; Lacressonnière et al., 2021; Luna et al., 2021; Mathew et al., 2017; T. Mesbahi et al., 2013; Tedjani Mesbahi et al., 2018; Michalczuk et al., 2014; Pebriyanti, 2013; Van Sterkenburg et al., 2014), contra 6 que usan microcontroladores (Buccolini et al., 2018; Dai et al., 2013; Di Rienzo et al., 2018, 2022; Hidalgo León et al., 2020; Restrepo et al., 2012).

Igualmente se encontraron 4 artículos que no especifican el tipo de controlador usado (Boles et al., 2017; Cheng & Chen, 2017; Dung et al., 2016; Orcioni et al., 2017).

Los resultados de distintos controladores encontrados en los artículos tomados en cuenta se muestran en la Figura 3.6.

Figura 3.6

Controladores usados en revisión sistemática



3.3 Modo de operación

Así mismo, otro aspecto tomado en cuenta es la capacidad del circuito de emular carga y descarga de una batería real en la operación normal o sólo la descarga. En la revisión sistemática se encuentran 10 proyectos que solamente pueden emular la operación de

descarga de una batería (Barreras et al., 2015; Buccolini et al., 2018; Cheng & Chen, 2017; Dai et al., 2013; Di Rienzo et al., 2022; Dung et al., 2016; Hidalgo León et al., 2020; T. Mesbahi et al., 2013; Tedjani Mesbahi et al., 2018; Pebriyanti, 2013; Van Sterkenburg et al., 2014) y 21 proyectos que pueden emular carga y descarga (Barreras et al., 2015, 2016; Boles et al., 2017; Cabello et al., 2015, 2017; Daniil & Drury, 2016; Di Rienzo et al., 2018; Guo et al., 2019; König et al., 2013, 2014; Lacressonnière et al., 2021; Luna et al., 2021; Mathew et al., 2017; Michalczuk et al., 2014; Orcioni et al., 2017; Restrepo et al., 2012; Sarikurt et al., 2014; Topan et al., 2016; Wang et al., 2018; Wu et al., 2016; Zhang et al., 2017).

Los resultados de distintos modos de operación de los emuladores encontrados en los artículos tomados en cuenta se muestran en la Figura 3.7.

Figura 3.7

Tipos de operación en revisión sistemática



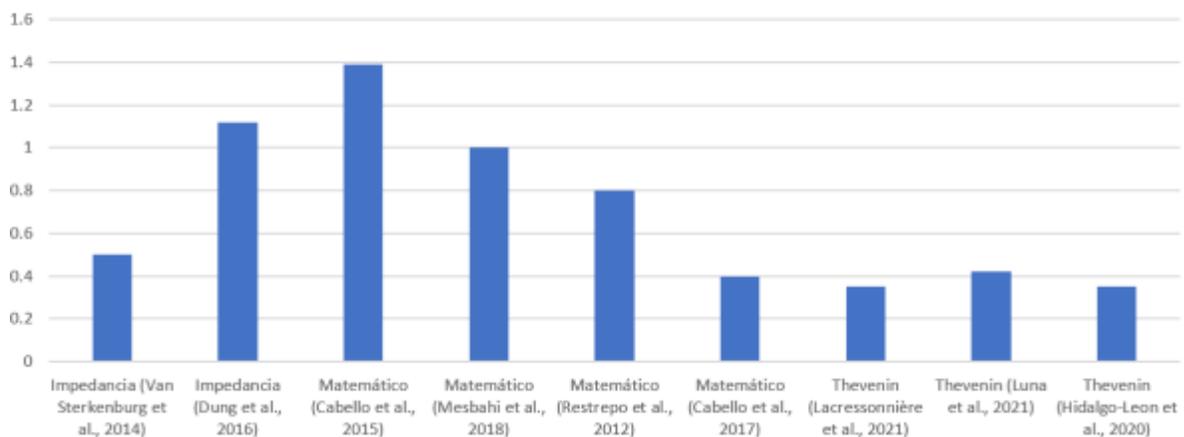
3.4 Porcentaje de error obtenido

En los artículos analizados se encuentra que, dentro de los emuladores que utilizan un modelo matemático, el error promedio varía entre 0.4% y 1.39. En los modelos de impedancia el error promedio varía entre 0.5% y 1.12%. En los modelos de Thévenin el error promedio

varía entre 0.35 % y 0.42%. En la Figura 3.8 se puede encontrar como se distribuyen estos valores tomando en cuenta los artículos donde lo mencionan.

Figura 3.8

Error promedio logrado con diferentes modelos en los emuladores implementados

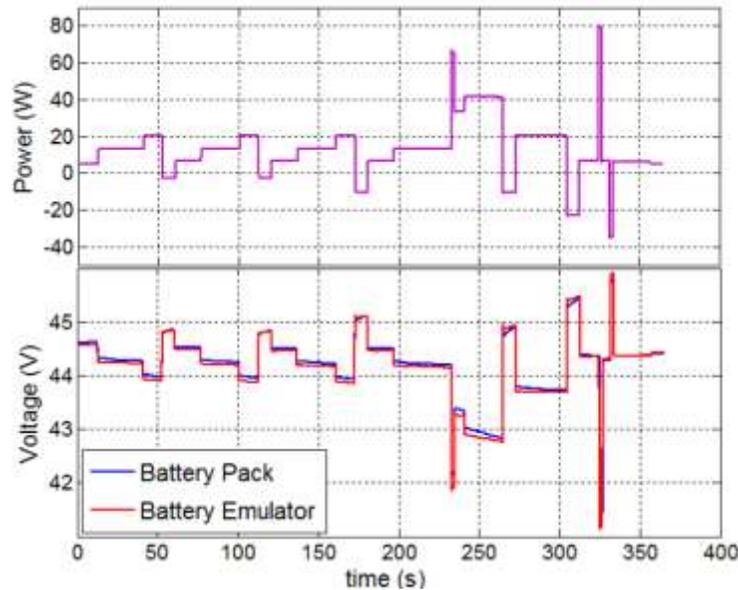


3.5 Discusión

Se encontró que la emulación de batería es un tema muy utilizado en la industria automotriz, encontrando un 34% de artículos aplicados en dicha área. Es necesario tomar en cuenta las dificultades descritas anteriormente para hacer pruebas en un sistema que incluya una batería, para así entender la utilidad de un emulador de batería y los diferentes parámetros que utilizan diferentes artículos sobre el desarrollo y la implementación de un dispositivo así. Aplicaciones con necesidad de alta precisión usarán un modelo más complejo para que se apege con más cercanía a la realidad, por lo que, a su vez, necesitaran un controlador con mayor capacidad de computación. En la Figura 3.9 se puede notar como el sistema implementado por Daniil & Drury, (2016) con un controlador tipo HIL, se compara con las mediciones de una batería real.

Figura 3.9

Voltajes comparados entre emulador y batería real



Nota. De Investigation and validation of methods to implement a two-quadrant battery emulator for power Hardware-in-the-Loop Simulation (p.2074), por N. Daniil & D. Drury, 2016, IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society.

El modelo en el que cada artículo se base es un parámetro crucial del proyecto, ya que el emulador solo puede ser tan preciso como lo es su modelo. En la mayoría de los artículos que desarrolla un emulador de batería utilizan modelos de circuitos equivalentes basados en impedancia, ya que es un modelo bastante preciso para su relativa baja complejidad y dentro de un laboratorio de pruebas es relativamente sencillo caracterizar una batería por medio de mediciones de EIS. Esto puede ser un problema para la viabilidad de un proyecto basado en un modelo de impedancia si no se tiene el equipo para hacer este tipo de mediciones.

Por esta razón gran parte de los artículos que no se basan en impedancia, se basan en modelos matemáticos o de Thévenin los cuales pueden caracterizar la batería por medio de su respuesta a pulsos de corriente y adaptando el modelo a la medición.

En la literatura se encuentra el uso de dos tipos de controladores. Los controladores tipo HIL tienen una gran capacidad de computación y de igual manera se puede tener el modelo de

batería embebido dentro de ellos, de igual forma estos controladores por lo general son muy costosos y la viabilidad de tener varios en un proceso de desarrollo dentro de un proyecto que los necesite es baja. Así mismo se usan microcontroladores, normalmente procesadores digitales de señales, lo cuales pueden ayudar a controlar el circuito ya que, a pesar de su baja capacidad de computación, comparados con un controlador tipo HIL, pueden recibir instrucciones por un protocolo de comunicación (Normalmente UART) desde una computadora con mayor capacidad de procesamiento, como una computadora personal, que puede correr el modelo de batería. Incluso hay artículos que utilizan un modelo lo suficientemente bajo en complejidad como para correr el modelo embebido dentro del microcontrolador.

Al comparar los modelos de batería en este artículo de revisión sistemática se considera que los modelos basados en Thévenin son bastante precisos (capturan el comportamiento dinámico de la batería y las respuestas transitorias), sencillos (no necesitan gran poder de computación) y no es necesario tener un equipo especializado para parametrizar la batería.

La elección de un controlador para un emulador de batería dependerá del modelo elegido y el presupuesto que tenga el proyecto. Si se quiere desarrollar un emulador económico para utilizar en un proceso de desarrollo en el que varios desarrolladores tengan un emulador para hacer pruebas, se puede utilizar una arquitectura en donde el modelo de batería se ejecute en una computadora personal comunicada con un microcontrolador, el cual controlará la salida y entrada de voltajes y corrientes.

Por último, el modo de operación dependerá de las necesidades del proyecto y se deberá de tomar en cuenta que al simular carga y descarga de una batería se necesita un circuito más complejo y tener en cuenta más variables a medir en la operación.

En conclusión, una posible implementación de un emulador de batería modular y de bajo costo para la industria automotriz se podría realizar utilizando un modelo de Thévenin y utilizando una arquitectura en donde el modelo se ejecute en una computadora personal comunicada con un microcontrolador que controle la salida de voltajes y corrientes de un convertidor reductor DC/DC y la entrada de voltaje y corriente a una resistencia electrónica

variable. De esta forma se tendría un circuito relativamente de bajo costo, con una buena precisión y capaz de emular el comportamiento de carga y descarga de una batería real.

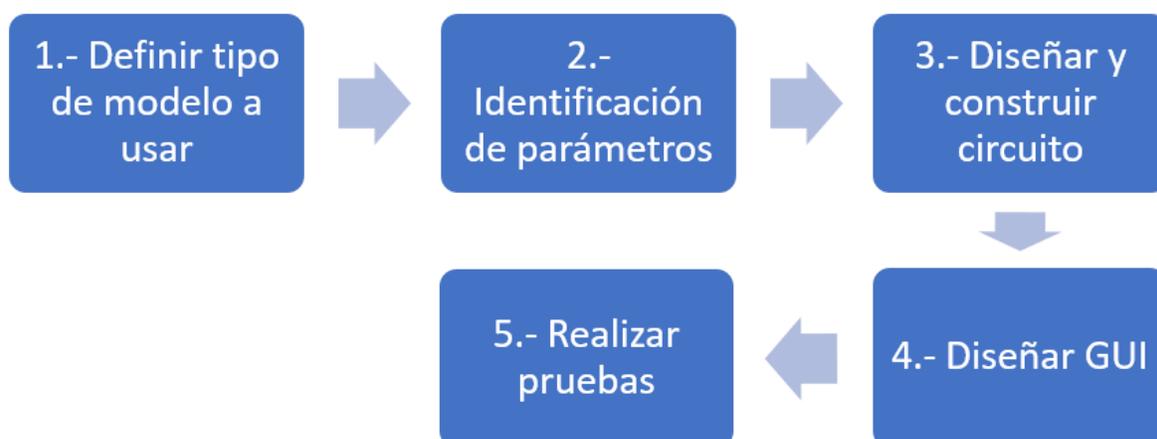
4. Metodología

Se propone la siguiente metodología para el desarrollo del emulador basada en la metodología de desarrollo en cascada (Kramer, M, 2018) y en la implementación de un dispositivo de emulador por (Luna et al., 2021).

En la figura 4.1 se muestra la metodología de desarrollo del prototipo dividida en pasos, cada paso tiene sus acciones específicas, descritas a continuación.

Figura 4.1

Metodología de desarrollo de proyecto

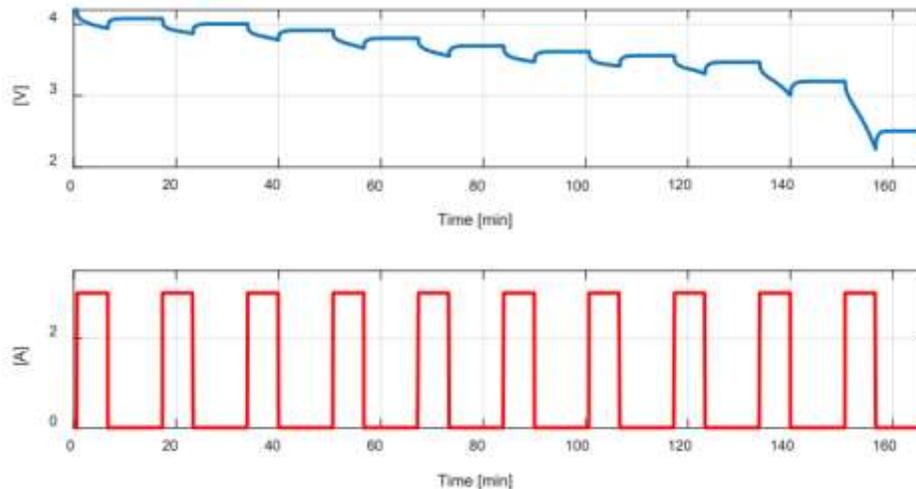


1. Definir tipo de modelo a usar: Es necesario seleccionar un modelo no tan complejo como para necesitar una computadora especializada en HIL, pero lo suficientemente robusto como para alcanzar el nivel de error buscado. En este proyecto se utilizará un modelo de Thévenin, basado en Luna et al., (2021).

2. Identificación de parámetros: Se extraerán los parámetros necesarios de la batería Li-ion A2C0215090000 para emular su comportamiento desde un modelo de Thévenin. La extracción de parámetros se realizará con una resistencia electrónica variable con corriente pulsada (figura 4.2), basado en Nutricato (2022) y Campagna (2020).

Figura 4.2

Experimentos de descarga y carga con corriente pulsada utilizados para el procedimiento de parametrización

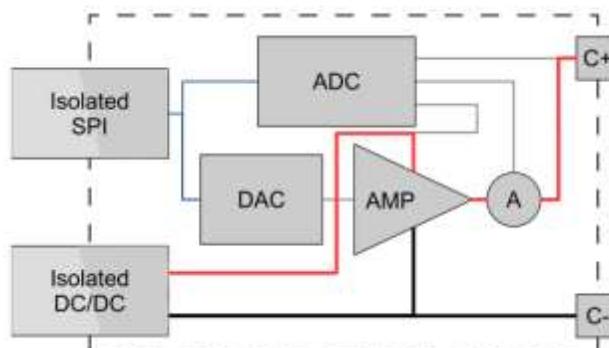


Nota. De Battery models for battery powered applications: A comparative study (p. 4085), por Campagna, N, et al., 2020, Edición especial de "Battery Management for Electric Vehicles".

3. Diseñar y construir circuito: Se construirá un circuito con convertidor reductor y una resistencia variable controlado por un microcontrolador basado en R. Di Rienzo et al. (2022) y A. Restrepo et al. (2012), estos circuitos son mostrados en la figura 4.3 y figura 4.4 respectivamente.

Figura 4.3

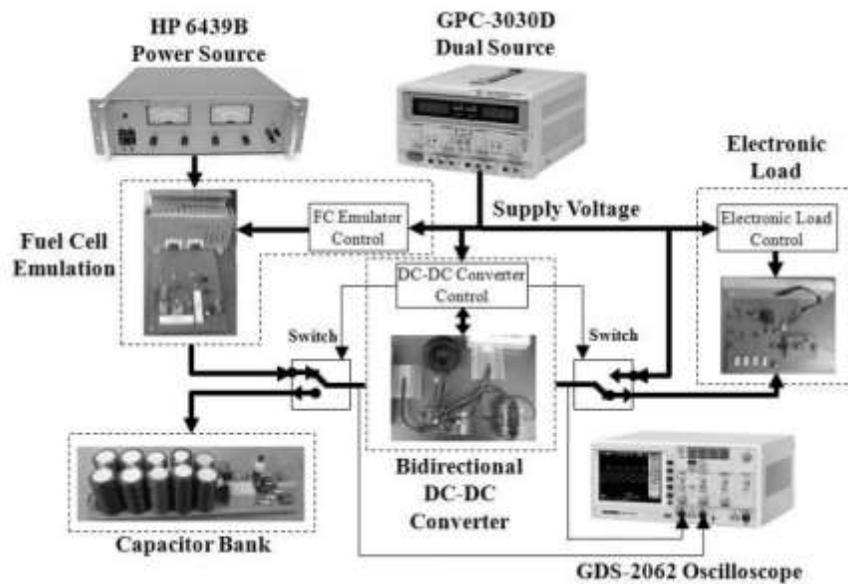
Circuito propuesto por R. Di Rienzo et al.



Nota. De Modular Battery Emulator for Development and Functional Testing of Battery Management Systems: The Cell Emulator (p. 1215), por Di Rienzo, R., et. al., 2022, Edición especial de " Condition Monitoring and Diagnostics of Energy Storage Systems Components".

Figura 4.4

Circuito propuesto por A. Restrepo et al.

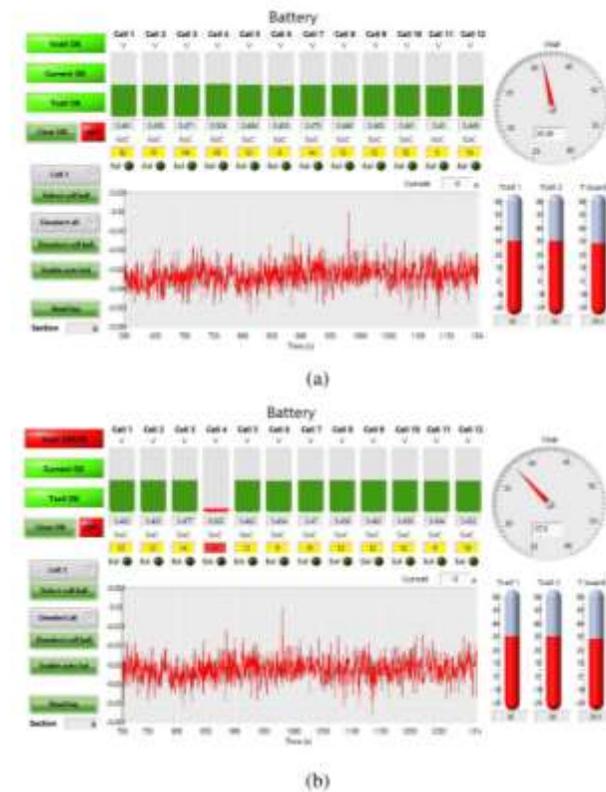


Nota. Power control of a bidirectional dc bus for fuel cells applications (pp. 159-170), por Restrepo, A., et. al., 2012 Revista EIA.

4. Diseñar GUI: en distintos artículos se propone una GUI implementada en LabView u otro programa de simulación de paga, en este proyecto se optará por diseñar una GUI desde python para no depender de un programa de paga. La estructura de dicha GUI será basada en R. Di Rienzo et al. (2022), y está ilustrada en la figura 4.5.

Figura 4.5

GUI propuesta por R. Di Rienzo et al.

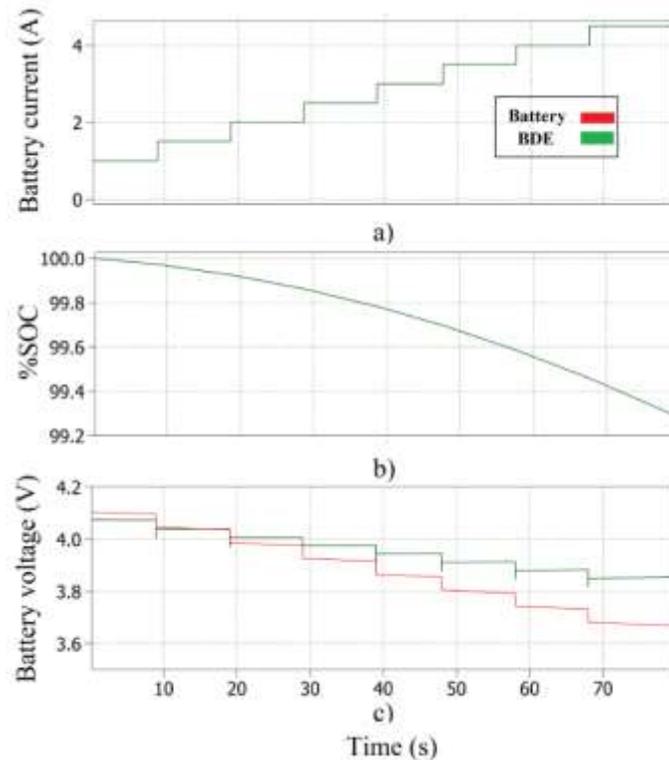


Nota. De Modular Battery Emulator for Development and Functional Testing of Battery Management Systems: The Cell Emulator (p. 1215), por Di Rienzo, R., et al., 2022, Edición especial de " Condition Monitoring and Diagnostics of Energy Storage Systems Components".

5. Realizar pruebas: Se harán pruebas para validar el prototipo y la integración del sistema. Estas pruebas al estar basadas en un sistema que depende de varios aspectos diferentes y al no poder tener una predicción 100% exacta, estarán basadas en la ley de los grandes números y teorema del límite central. A su vez basándose en como pruebas realizadas a los emuladores en diferentes artículos como R Hidalgo-Leon, et al (2020) (figura 4.6) y J. Barreras, et al (2016).

Figura 4.6

Prueba de pulsos de corriente de descarga a) SOC b) Corriente c) Voltaje (Hidalgo-Leon et al., 2020)



Nota. De Simulation of battery discharge emulator using power electronics device with cascaded P-I control (pp. 959-964), por Hidalgo-León, R., et. al., 2022, 2020 IEEE International Conference on Industrial Technology (ICIT).

4.1 Campo de aplicación

El emulador de batería propuesto tiene varias aplicaciones potenciales en varios campos. Algunas de las aplicaciones potenciales incluyen:

- **Sistemas de gestión de baterías (BMS):** el emulador propuesto se puede utilizar para probar y validar el software y el hardware de BMS en condiciones en las que no se pueden poner a prueba baterías reales por motivos de seguridad, garantizando su fiabilidad y seguridad.

- Dispositivos con batería incluida: el emulador se puede usar para probar dispositivos con batería, como teléfonos móviles, computadoras portátiles, unidades de telemática y otros dispositivos electrónicos, para mejorar su eficiencia, confiabilidad y duración de la batería.
- Vehículos eléctricos: El emulador propuesto se puede utilizar para simular el comportamiento de las baterías de los vehículos eléctricos, lo que permite probar y optimizar varios algoritmos y estrategias de control.
- Sistemas de energía renovable: el emulador se puede usar para simular el comportamiento de las baterías en sistemas de energía renovable, como los sistemas de energía solar y eólica, lo que permite probar y optimizar varios algoritmos y estrategias de control.
- Investigación de baterías: el emulador se puede utilizar en la investigación de baterías para probar el comportamiento de las baterías en diferentes condiciones, como temperatura, corriente y voltaje, lo que permite el desarrollo de baterías más eficientes y confiables.

El desarrollo de un emulador de batería puede ser una herramienta valiosa para varias industrias debido a los riesgos y costos reducidos asociados con las pruebas de baterías reales. Por ejemplo, la industria del BMS requiere muchas pruebas de baterías para evaluar el rendimiento y la seguridad de sus productos. Sin embargo, el uso de baterías reales puede ser costoso, llevar mucho tiempo y ser peligroso, especialmente cuando las pruebas involucran situaciones de sobre corriente, sobre voltaje o alta temperatura. Mediante el uso de un emulador de batería, los fabricantes de BMS pueden reducir los riesgos asociados con las pruebas de baterías reales y al mismo tiempo obtener datos precisos sobre el rendimiento de sus productos. De la misma manera, el emulador propuesto se puede utilizar en el desarrollo y prueba de varios dispositivos que requieren baterías, como dispositivos médicos, teléfonos móviles y vehículos eléctricos. En estos casos, el emulador puede proporcionar una forma segura y rentable de evaluar el comportamiento de la batería del dispositivo, lo que puede mejorar el rendimiento y la seguridad del producto final, así como acelerar los tiempos de entrega. En general, el emulador propuesto tiene el potencial de reducir significativamente

los riesgos y los costos asociados a las pruebas de baterías, lo que lo convierte en una herramienta valiosa.

4.2 Modelo de batería utilizado

Para emular con precisión y eficacia el comportamiento de descarga y carga de una batería, es esencial seleccionar un modelo de batería adecuado. El modelo de batería debe capturar el comportamiento dinámico de la batería y tener en cuenta la resistencia interna, la capacitancia, el SOC y el voltaje de la batería. El modelo de Thévenin se utiliza para representar una batería ya que puede aumentar la precisión de emulación al considerar las características de polarización de la batería (Fotouhi et al., 2016).

El modelo de batería de Thévenin se compone de una fuente de tensión ideal (V_{oc}), una resistencia interna (R_0), una resistencia de sobretensión (R_1) y un condensador (C_1). La configuración de este modelo se muestra en la Figura 1. Estos modelos se utilizan a menudo como una alternativa a los modelos de impedancia, ya que no requieren ninguna maquinaria especial para el proceso de parametrización. En su lugar, se pueden parametrizar mediante un procedimiento de caracterización de potencia de pulso híbrido aplicando diferentes corrientes pulsadas a la batería real, como se explicará en la sección de proceso de extracción de parámetros.

En este proyecto de tesis, se propone un emulador de batería Li-ion que utiliza un modelo 3RC Thévenin para simular el comportamiento de descarga y un modelo 2RC para emular el comportamiento de carga de una batería. Los modelos Thévenin 3RC y 2RC fueron elegidos para el emulador de batería propuesto debido a su simplicidad, precisión, efectividad y capacidad para capturar el comportamiento dinámico de la batería. Estos modelos se utilizan ampliamente en la literatura y se han validado experimentalmente para diferentes químicas y capacidades de baterías. La razón por la cual se eligió esta cantidad de pares para cada modo se explica más en la sección de proceso de extracción de parámetros.

Para emular el comportamiento de la batería con el modelo de batería de Thévenin, se deben calcular parámetros como el V_{oc} , R_0 , R_1 , C_1 y otros, dependiendo de cuántos pares RC tenga el circuito. Los parámetros de capacitancia C_n también se pueden calcular como parámetros de tiempo, T_n :

$$T_n = R_n * C_n$$

La ecuación para calcular el voltaje que cae en cada par RC es:

$$\frac{dV_n}{dt} = -\frac{V_n}{T_n} + \frac{I * R_n}{T_n}$$

Teniendo en cuenta que n es el número de pares RC en el circuito, la ecuación para calcular el voltaje de salida de un modelo de batería de Thévenin es:

$$V_{out} = V_{oc} - (R_{int} * I) - V_n - V_{n-1} - V_{n-2} \dots$$

Así que las fórmulas para el modelo Thévenin 3RC serán:

$$\frac{dV_1}{dt} = -\frac{V_1}{T_1} + \frac{I * R_1}{T_1}$$

$$\frac{dV_2}{dt} = -\frac{V_2}{T_2} + \frac{I * R_2}{T_2}$$

$$\frac{dV_3}{dt} = -\frac{V_3}{T_3} + \frac{I * R_3}{T_3}$$

$$V_{out} = V_{oc} - (R_{int} * I) - V_3 - V_2 - V_1$$

Las fórmulas para el modelo Thévenin 2RC serán:

$$\frac{dV_1}{dt} = -\frac{V_1}{T_1} + \frac{I * R_1}{T_1}$$

$$\frac{dV_2}{dt} = -\frac{V_2}{T_2} + \frac{I * R_2}{T_2}$$

$$V_{out} = V_{oc} - (R_{int} * I) - V_2 - V_1$$

4.3 Batería utilizada

La batería escogida para este estudio es una batería Li-ion A2C0215090000 con un voltaje nominal de 3.2V y una capacidad de 1100mAh. Los parámetros de la batería elegida se pueden ver en la Tabla. 4.1. Para extraer los parámetros del modelo de batería, se realizó una prueba de descarga de corriente pulsada en la batería real

Tabla 4.1*Parámetros de A2C0215090000*

Characteristic	Value	Unit
Nominal Voltage	3.2	V
Nominal End of Charge Voltage	3.6	V
Maximum End of Charge Voltage	3.65	V
End of Discharge Voltage	2	V
Nominal Charge Current	400	mA
Maximum Charge Current	1.35	A
Nominal Capacity (measured from 3.7V down to 2.0V) from a fresh battery	1350	mAh
Initial DC Equivalent Series Resistance of Battery Pack at 25C	≤70	mOhm
Charging Temperature	0 to +60	°C
Discharging Temperature	-40 to +85	°C
Storage Temperature	-40 to +85	°C

4.4 Proceso de extracción de parámetros y validación

Como se explicó en secciones anteriores se aplicó un modelo Thévenin 3RC y 2RC para emular la descarga y la carga respectivamente. Para esto es necesario extraer los parámetros de circuito de la batería real. Se extraerán 8 parámetros para el modelo 3RC (V_{oc} , R_{int} , R_1 , R_2 , R_3 , T_1 , T_2 , T_3) y 6 para el modelo 2RC (V_{oc} , R_{int} , R_1 , R_2 , T_1 , T_2).

Se utilizó el bloque de batería de circuito equivalente (Equivalent Circuit Battery block) de Matlab para hacer el proceso de extracción de variables, para esto es necesario ingresar una curva de descarga generada a partir de pulsos controlados. Este proceso se explicará a detalle a continuación.

4.4.1 Parámetro de capacidad de la batería

El primer paso para el modelado de la batería es calcular la capacidad real, esto se hace siguiendo los siguientes pasos:

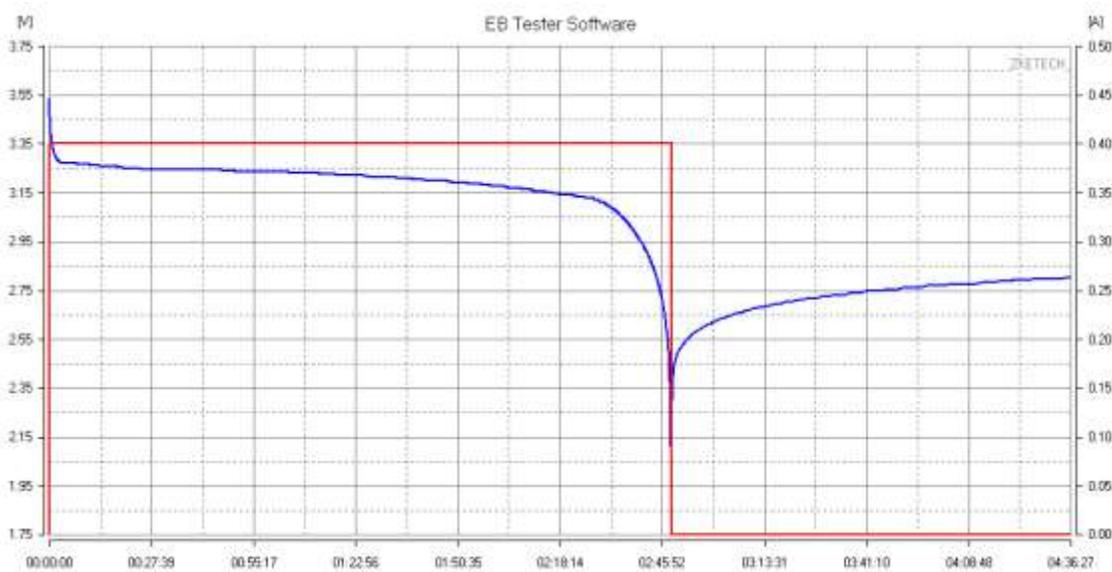
1. Cargar batería con CC (corriente constante) corriente nominal de carga (.4A) hasta el voltaje nominal de fin de carga (3.6V).
2. Cargar la batería con CV (voltaje constante) 3.6V hasta 2% de C (capacidad nominal en amperios) (.022A)

3. Descargar la batería hasta el voltaje de fin de descarga (2V) con corriente constante de corriente de descarga nominal (.4A)
4. Medir el tiempo que tardó en llegar el voltaje de fin de descarga (2 horas 48 minutos)
5. Obtener la capacidad de descarga real

El proceso de extracción de la capacidad de descarga de la batería genera la curva en la Figura 4.7.

Figura 4.7

Curva de descarga



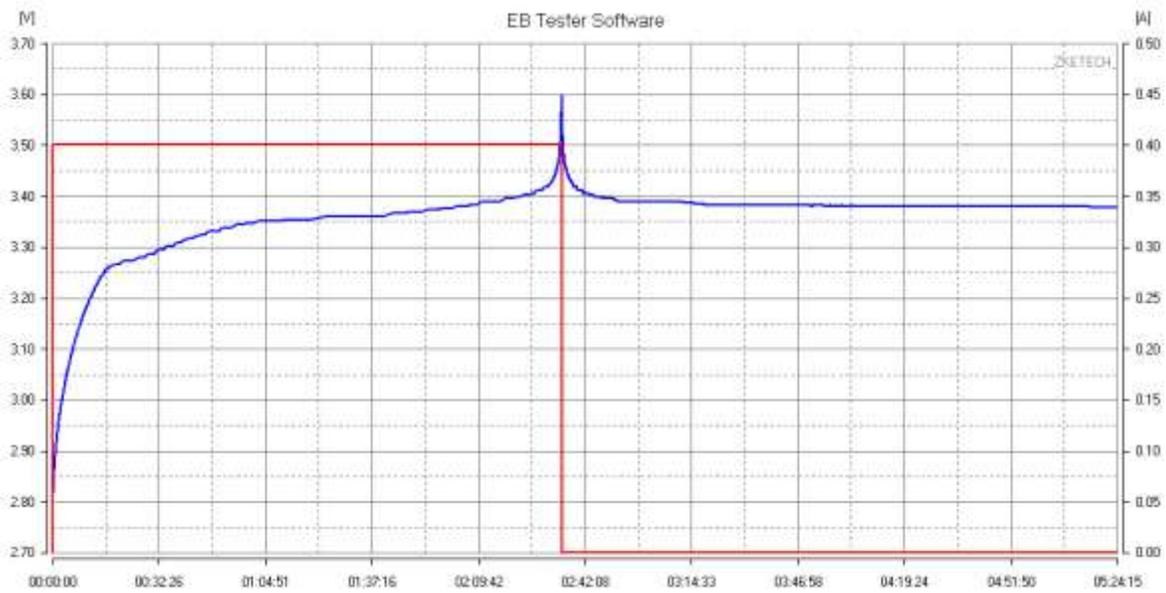
De la misma forma se calcula la capacidad de la batería en el modo de carga con los siguientes pasos:

1. Descargar batería hasta el voltaje de fin de descarga (2V)
2. Esperar a que el voltaje se estabilice (2 horas)
3. Cargar la batería en CC con corriente nominal de carga (.4A) hasta llegar al voltaje nominal de fin de carga (3.6V).
4. Medir el tiempo que tardó en llegar el voltaje de fin de carga (2 horas 41 minutos)
5. Obtener la capacidad de carga real

El proceso de extracción de la capacidad de carga de la batería genera la curva en la Figura 4.8.

Figura 4.8

Curva de carga



La capacidad real se calcula tomando en cuenta el tiempo que se tardó en descargarse con la corriente aplicada

$$C_d = I * T = 0.4A * 2.8116 = 1.1246Ah$$

$$C_c = I * T = 0.4A * 2.6907 = 1.0762Ah$$

C_d = Capacidad de descarga (Ah)

C_c = capacidad de carga (Ah)

I = corriente de descarga/carga (A)

T = tiempo transcurrido (h)

4.4.2 Parámetros de V_{oc} de descarga

El voltaje de circuito abierto (V_{oc}) se define como el voltaje a través de los terminales de la batería sin que fluya corriente. Este se tiene que medir después de un tiempo de relajación.

Para determinar el Voc de descarga de la batería en este estudio, se siguieron los siguientes pasos:

1. Cargar batería con CC corriente nominal de carga (.4A) hasta el voltaje nominal de fin de carga (3.6V).
2. Cargar la batería con CV 3.6V hasta 2% de C (.022A)
3. Descargar la batería con corriente de descarga nominal (.4A) durante 101 segundos (cantidad de tiempo para reducir 1% de carga real calculada en la sección anterior)
4. Esperar 15 min para obtener el voltaje de circuito abierto después de relajación
5. Repetir paso 3 y 4 100 veces

El proceso de extracción de parámetros de Voc de la batería genera la curva en la Figura 4.9. Se pueden apreciar los pulsos con mayor detalle en la Figura 4.10.

Figura 4.9

Curva de Voc de descarga

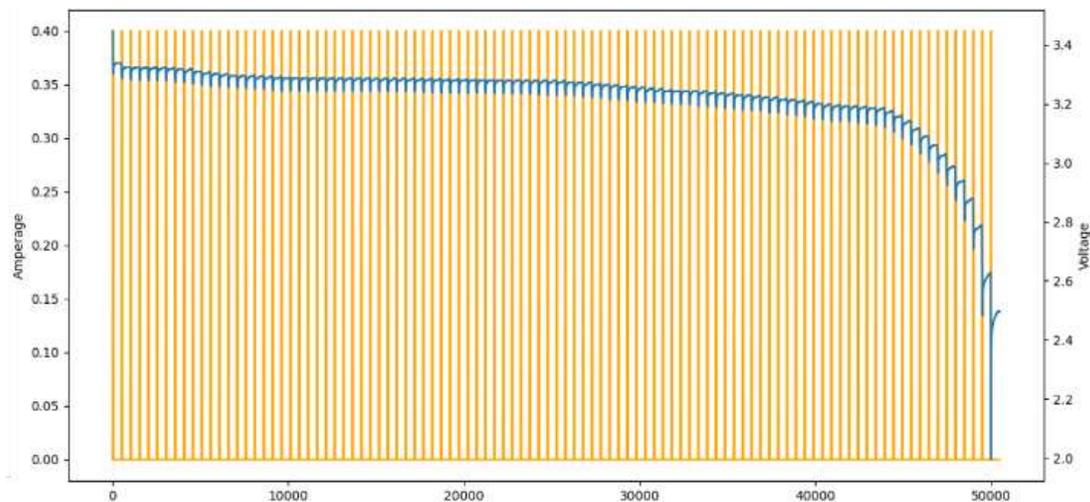
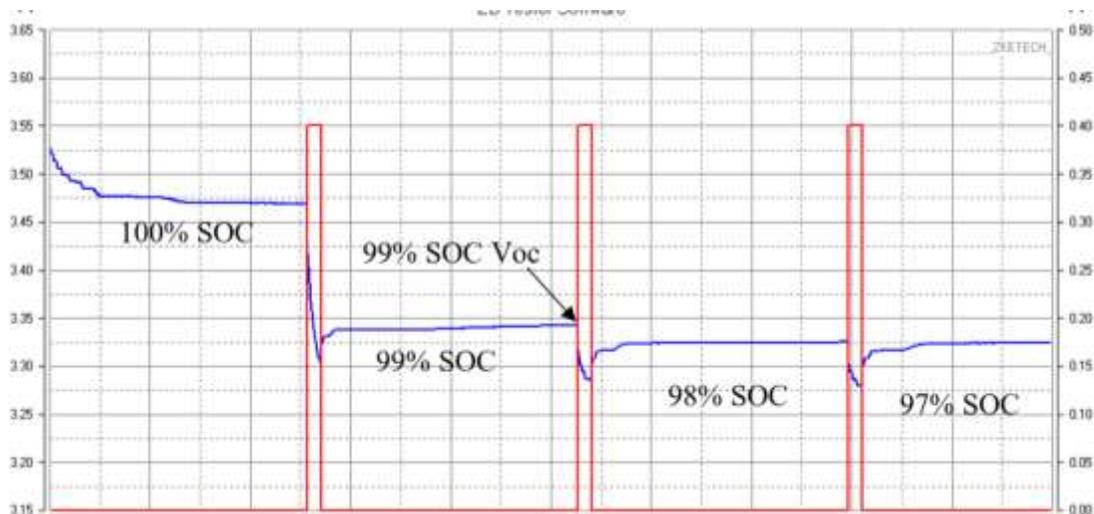


Figura 4.10

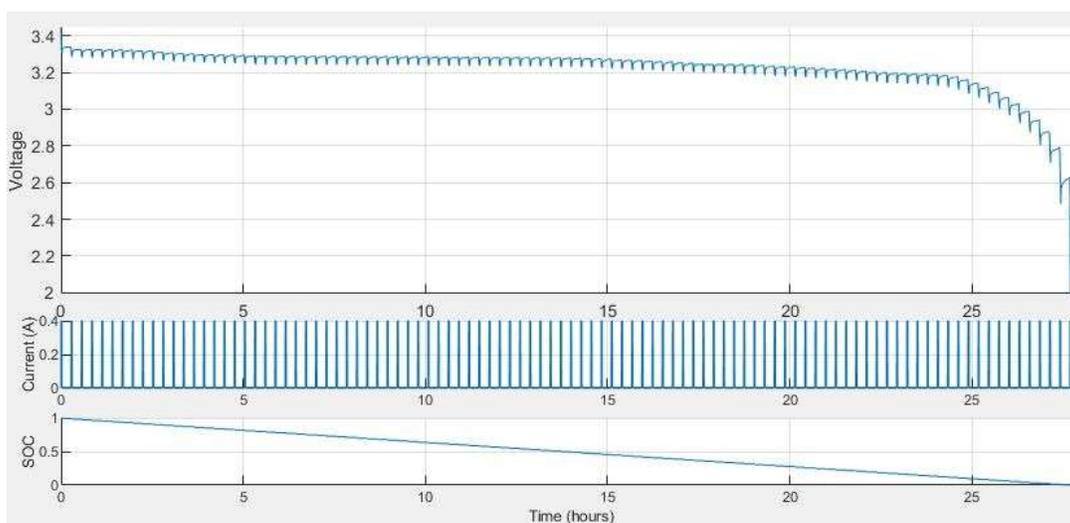
Pulsos de extracción de Voc de descarga



Posteriormente estos pulsos fueron utilizados para el proceso de extracción de parámetros e introducidos en el bloque de batería de circuito equivalente en Matlab basándose en Ahmed et al. (2015) y Nutricato et al. (2022). Se puede ver el código introducido en Matlab para cargar los datos de descarga en el Anexo 1 sección 1 y se ven los datos graficados en Matlab en la Figura 4.11.

Figura 4.11

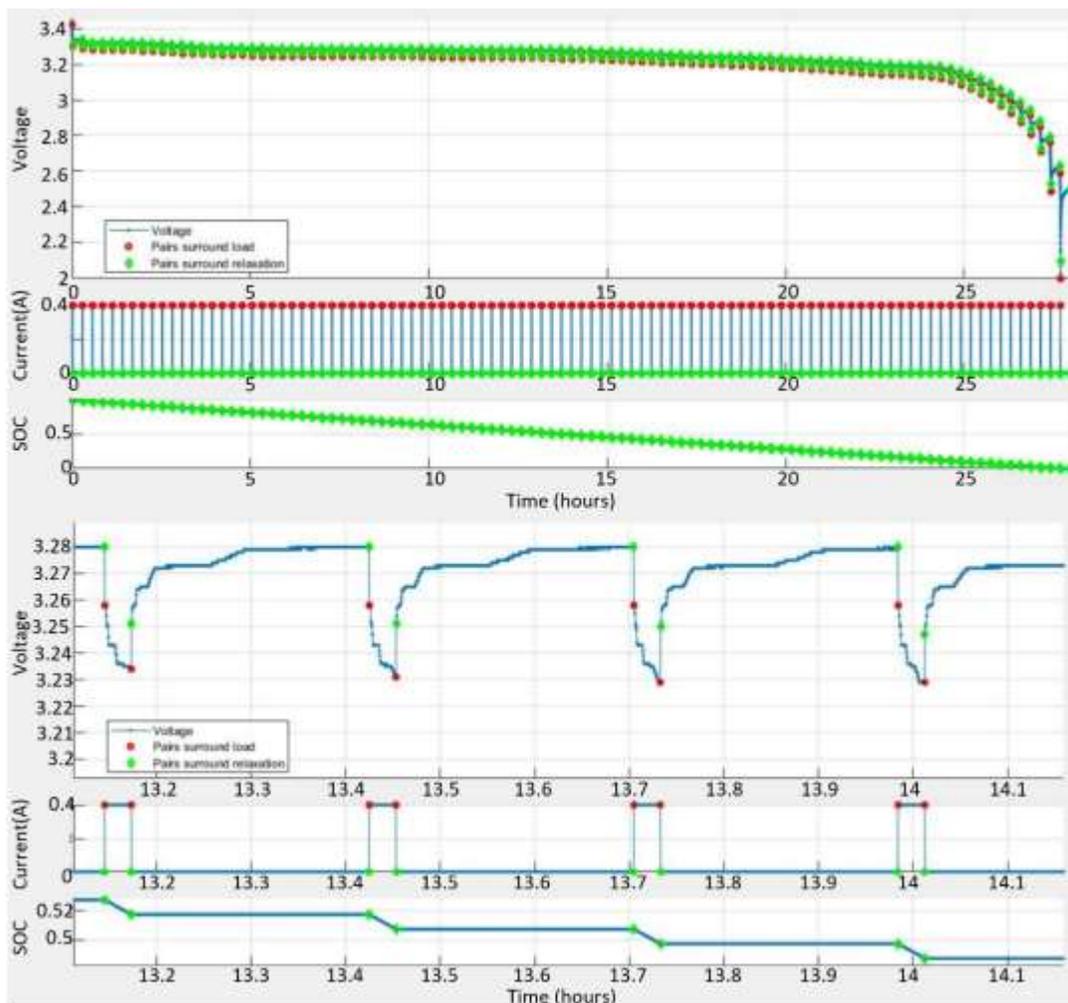
Curva de descarga en Matlab



Después hay que dividir los datos en objetos de pulsos. Para esto, hay que crear los objetos Pulse dentro de PulseSequence. Esto crea un objeto de parámetro que contiene tablas de consulta dado el número de pulsos. El cambio en SOC para cada pulso no debe ser superior al 5%. En este caso el cambio en SOC de cada pulso es de 1%. Se puede ver el código en el Anexo 1 sección 2. La gráfica de los pulsos identificados se puede ver en la Figura 4.12.

Figura 4.12

Pulsos de descarga identificados



En Matlab, se puede utilizar el mismo bloque para comparar el error al emplear diferentes cantidades de pares RC, desde 1 hasta 5. Es importante tener en cuenta que a medida que se utilizan más pares RC, se requiere mayor computación para calcular el voltaje final. Como

se puede observar en la Figura 4.13, 4.14 y 4.15, la mejor manera de emular el comportamiento de descarga es utilizando 3 pares RC. Utilizar 1 ó 2 pares ofrece más error, mientras que 4 ó 5 pares no proporcionan muchos beneficios en comparación con la cantidad de computación necesaria para ejecutarlos. Se puede ver el código en el Anexo 1 sección 3.

Figura 4.13

Determinación de pulsos de descarga 1, 10, 20 y 30

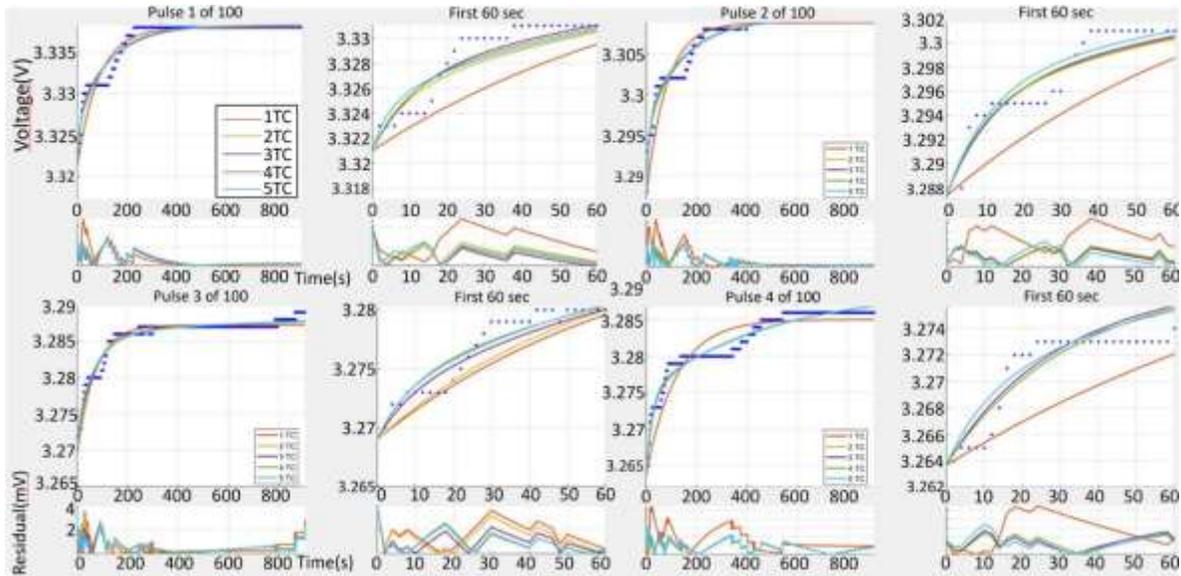


Figura 4.14

Determinación de pulsos de descarga 40, 50, 60, y 70

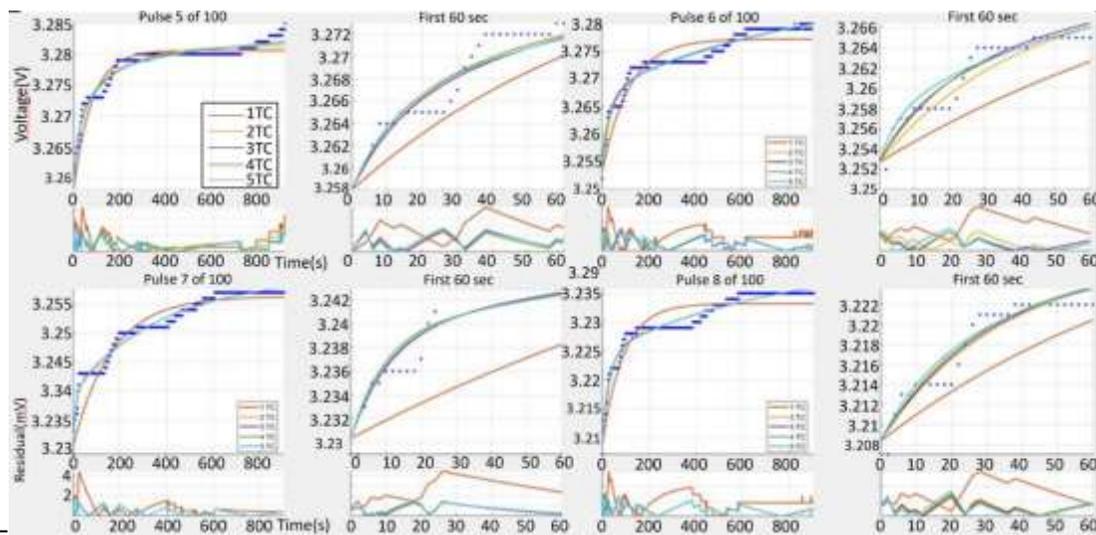
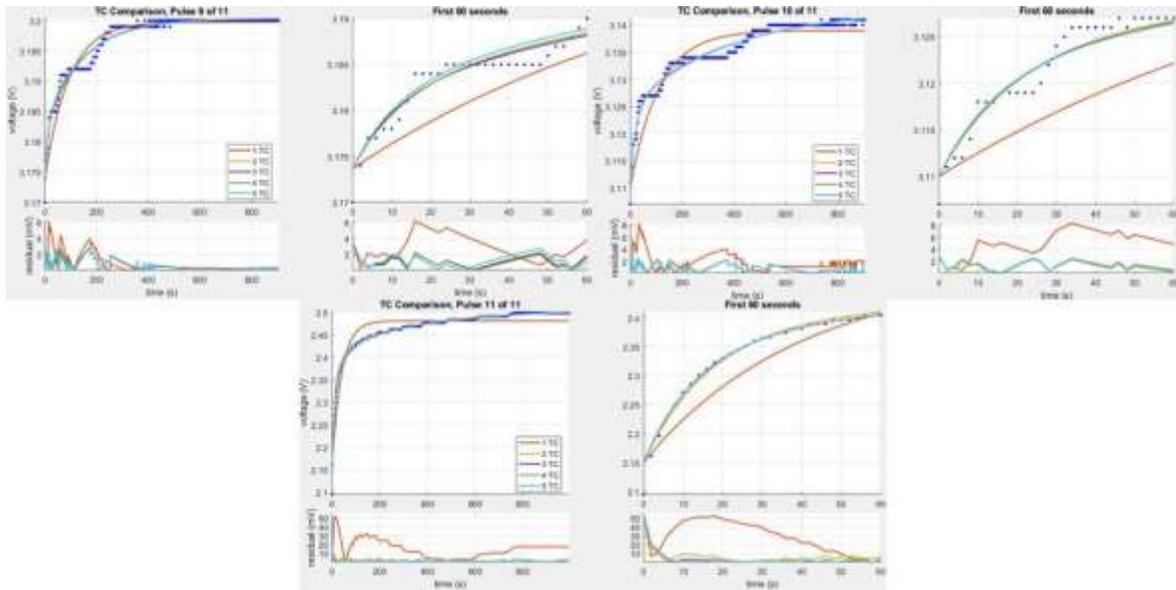


Figura 4.15

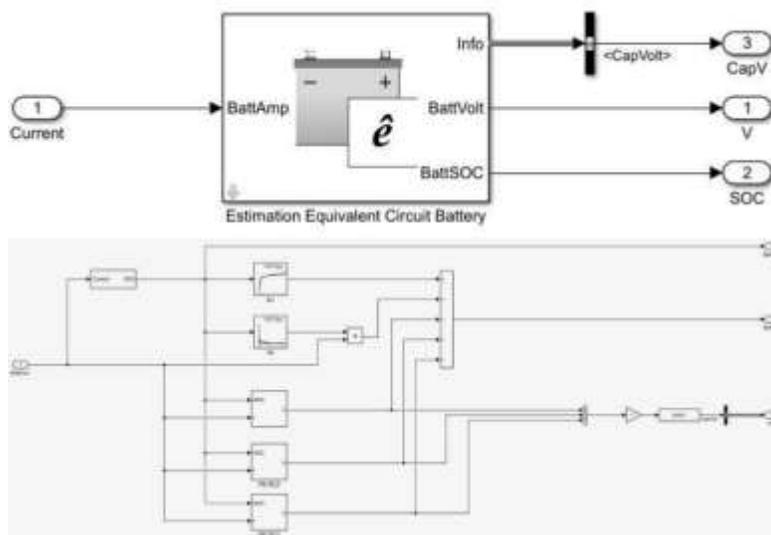
Determinación de pulsos de descarga 80, 90 y 100



Del mismo modo, se tiene que escoger un modelo de batería adecuado para el modelo escogido, en este caso Matlab tiene un modelo Thévenin 3RC ‘BatteryEstim3RC_PTBS’ el cual se puede ver en la Figura 4.16.

Figura 4.16

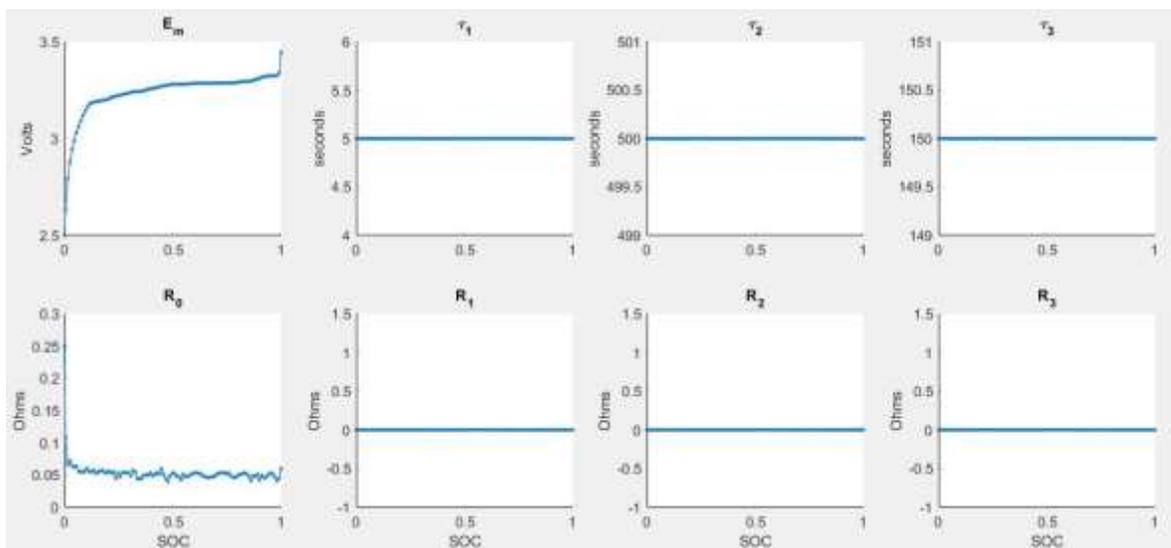
Modelo Thévenin 3RC “BatteryEstim3RC_PTBS”



Una vez se ha seleccionado el modelo adecuado y se han identificado los pulsos, se puede iniciar el proceso de estimación de parámetros. Como primer paso, se asignan valores iniciales, máximos y mínimos a los parámetros T1, T2 y T3 (Tx). Es necesario variar estos valores y realizar varias estimaciones para alcanzar el menor error posible. Con los valores asignados, se puede estimar Voc (Em) y Rint (R0). La gráfica de resultados de esta primera estimación se muestra en la Figura 4.17. Se puede ver el código en el Anexo 1 sección 4.

Figura 4.17

Primera estimación de Voc y Rint de descarga



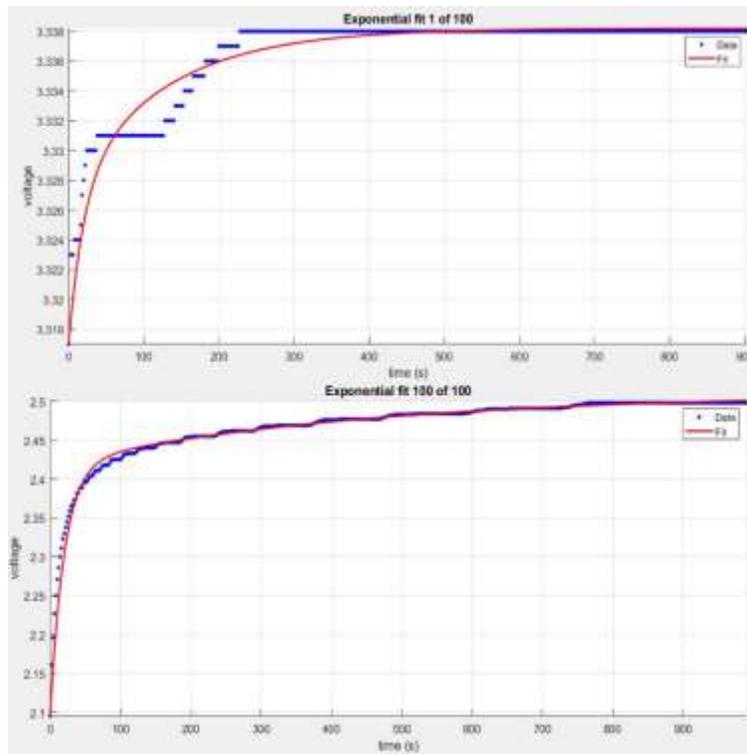
A continuación, se puede hacer una estimación de los valores Tx que anteriormente tenían valores constantes. Esta estimación realiza un ajuste exponencial, es decir, una técnica de ajuste de curvas sobre relajación de impulsos para una estimación inicial de las constantes de tiempo RC (Tau) en cada SOC. Esta operación usa la expresión exponencial estandarizada de la forma:

$$a_1 e^{-t/a_2} + b_1 e^{-t/b_2} + c_1 e^{-t/c_2} + V_i - a_1 - b_1 - c_1$$

Donde: a1, b1 y c1 son los voltajes iniciales del capacitor, a2, b2 y c2 son las constantes de tiempo, t es el tiempo y Vi es la tensión en el inicio de la relajación. Se puede ver la estimación de los pulsos 1 y 100 en la Figura 4.18. Se puede ver el código en el Anexo 1 sección 5.

Figura 4.18

Estimación Tx pulsos de descarga 1 y 100



Después de la estimación se puede graficar los valores de Tx y los resultados de la simulación comparados con los pulsos originales. En este caso la estimación genera un error promedio de 7.3306mV y un error máximo de 508.0771mV. Se pueden ver la estimación de los parámetros Tx y el resultado de la simulación en la Figura 4.19 y 4.20 respectivamente.

Figura 4.19

Estimación Tx de descarga

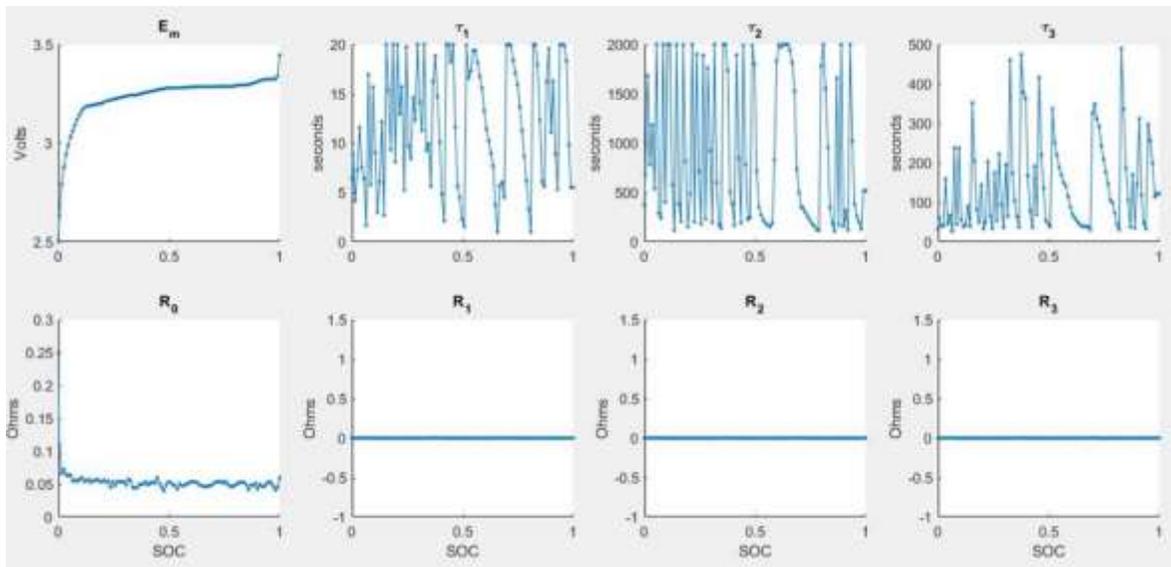
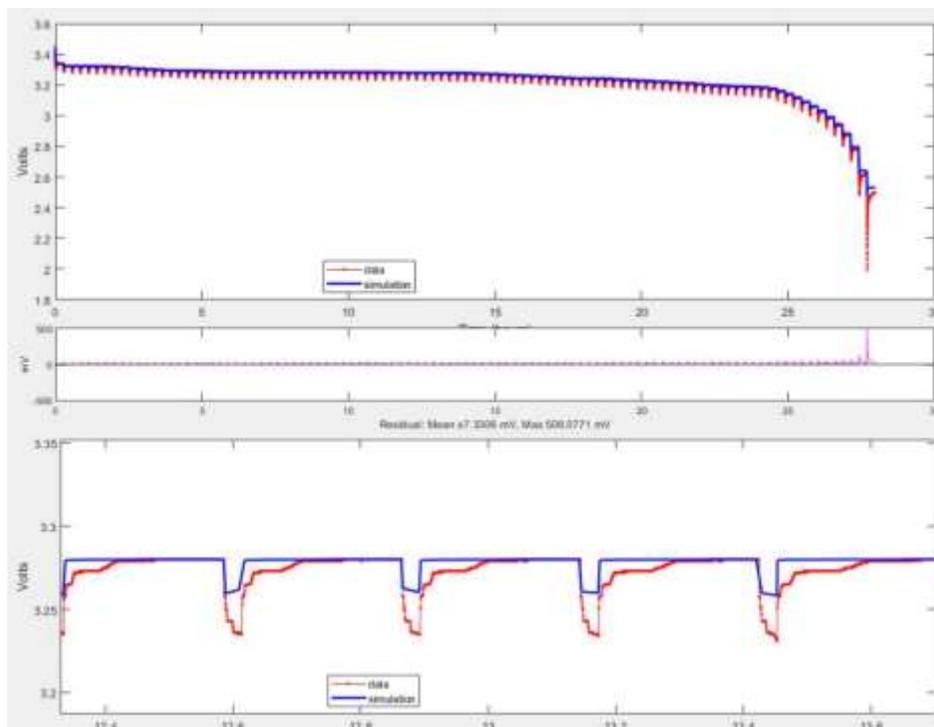


Figura 4.20

Resultados de simulación de descarga



Con estos valores de Tx se puede calcular los valores de Rx. Para esta estimación se realiza un ajuste lineal, que identifica los parámetros y establece los valores iniciales mediante un enfoque de sistema lineal, pulso a pulso. Con estos nuevos valores la simulación reduce el error promedio a 3.2758mV y el error máximo a 111.4646mV. Se puede ver la estimación de los pulsos 1 y 100 en la Figura 4.21. Los valores de Rx y los nuevos resultados de simulación se pueden ver en la Figura 4.22 y 4.23 respectivamente. Se puede ver el código en el Anexo 1 sección 6.

Figura 4.21

Estimación Rx pulsos de descarga 1 y 100

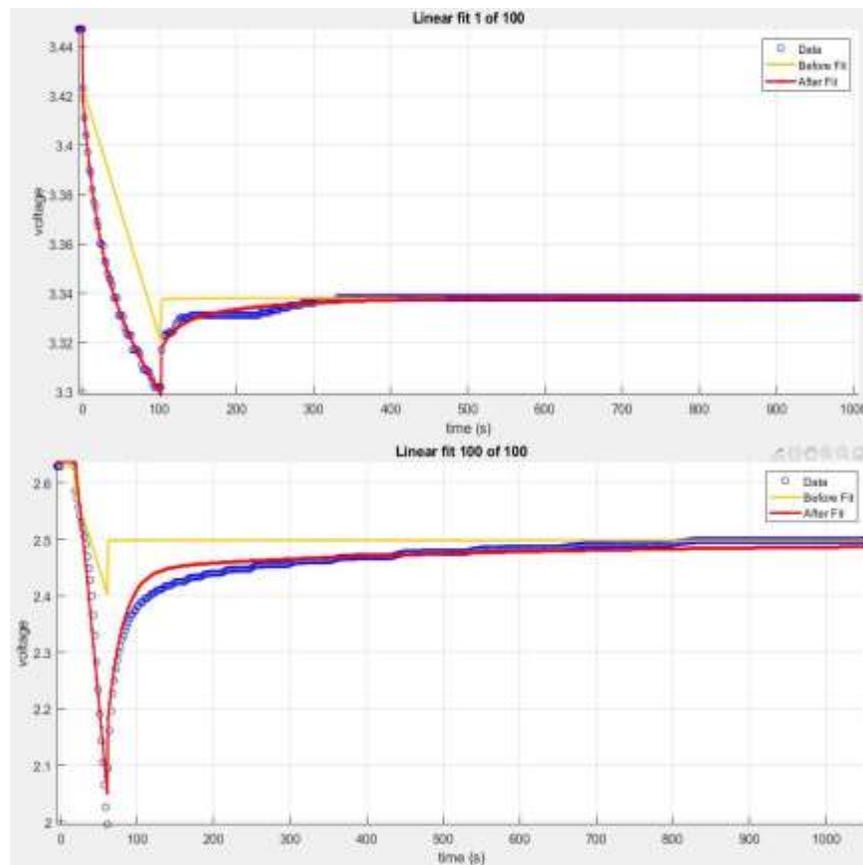


Figura 4.22

Estimación Rx de descarga

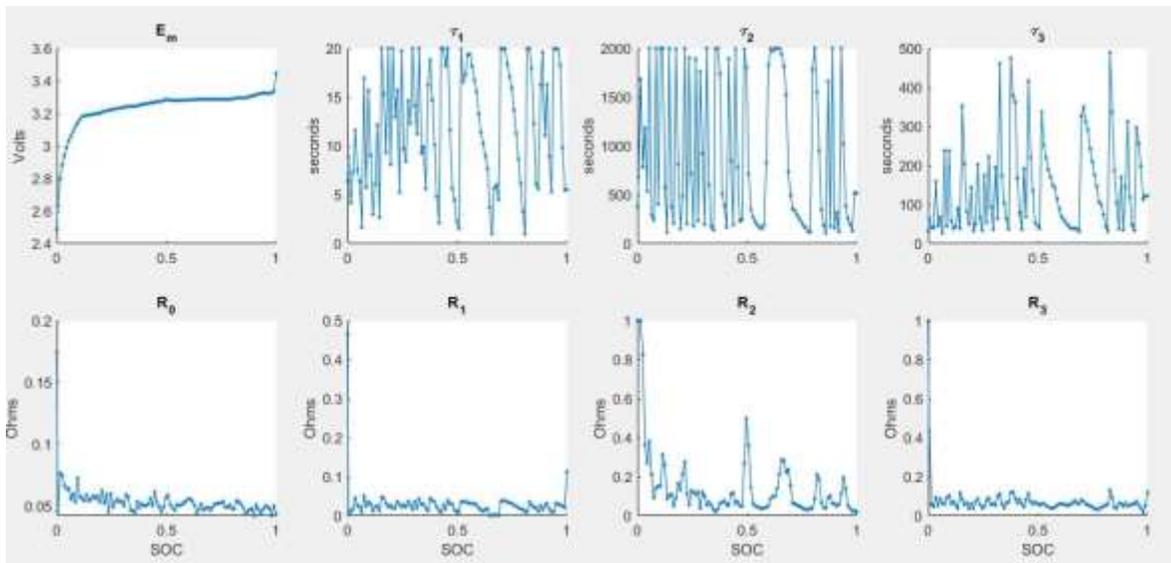
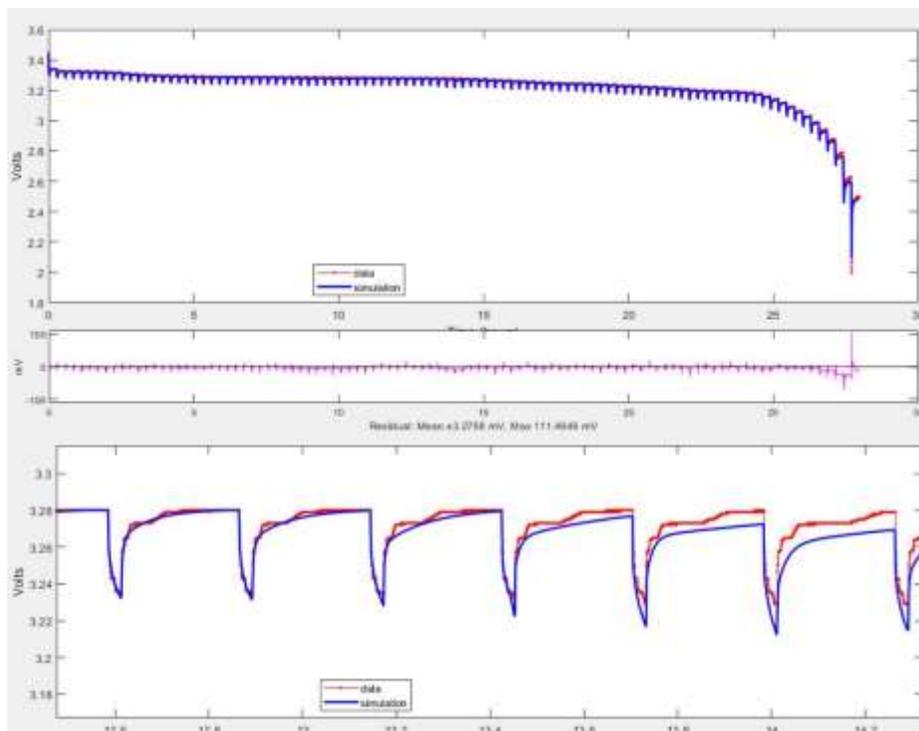


Figura 4.23

Resultados de simulación de descarga



Por último, se aplicó una optimización de SDO (Simulink Design Optimization), la cual ayudó a reducir el error de la simulación a 2.3182mV (0.0719%) de error promedio y 285.5815mV de error máximo. Se puede ver la optimización de los pulsos 1 y 100 en la Figura 4.24. Los valores de finales de todos los parámetros de Voc y los resultados finales de simulación se pueden ver en la Figura 4.25 y 4.26 respectivamente. Se puede ver el código en el Anexo 1 sección 7.

Figura 4.24

Optimización SDO pulsos de descarga 1 y 100

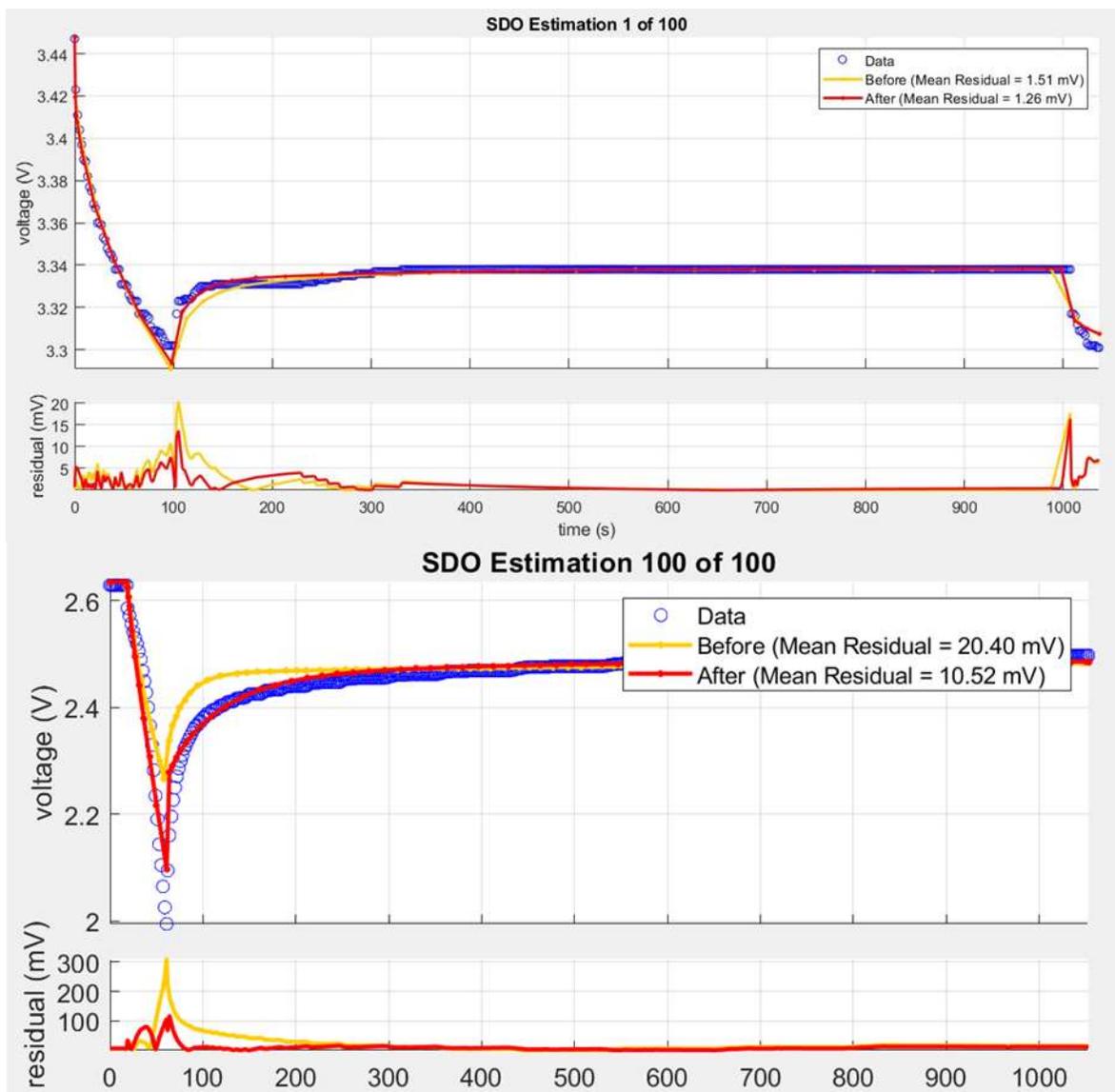


Figura 4.25

Parámetros finales Voc de descarga

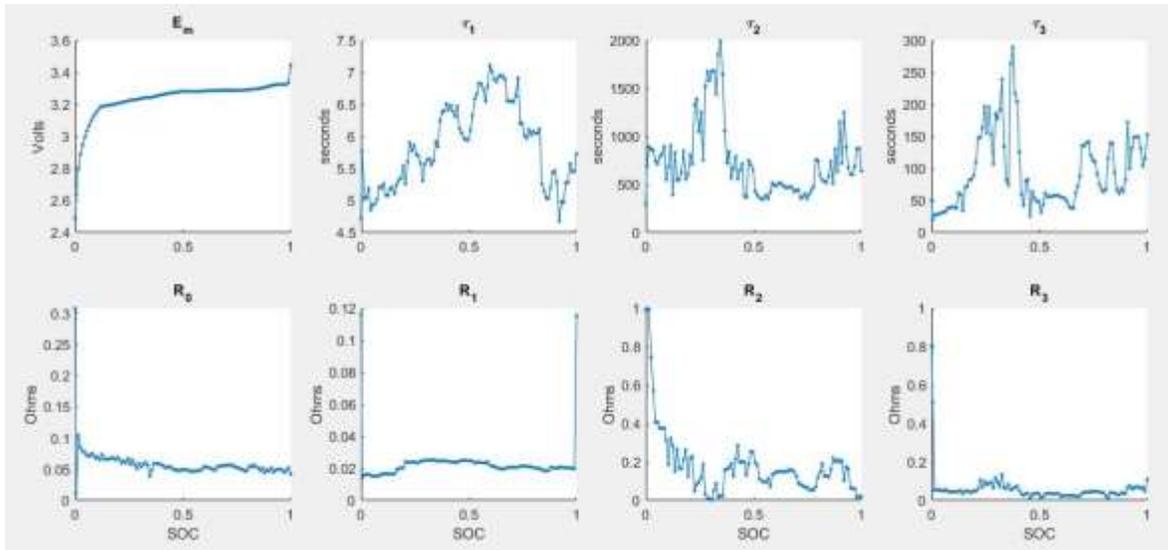
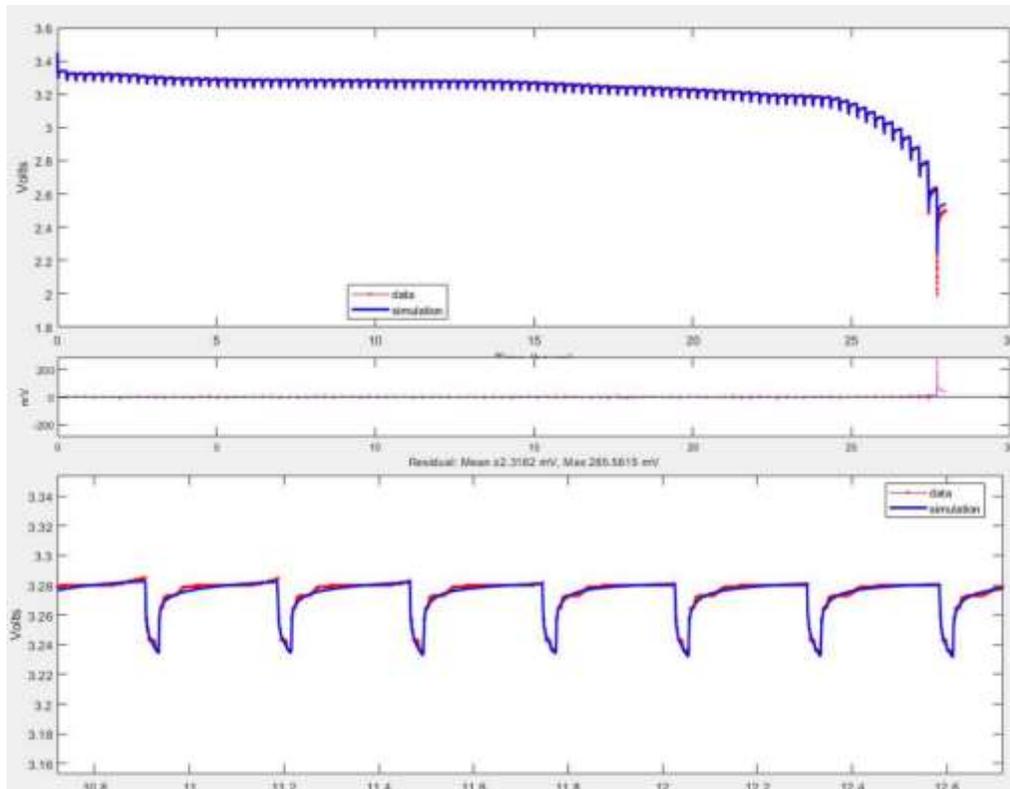


Figura 4.26

Resultados finales de simulación de descarga



Los valores finales de descarga generados por este proceso de extracción de parámetros se pueden ver en el Anexo 3. Estos serán los valores que se usarán en la tabla de búsqueda en el programa final para estimar el voltaje de salida al emular el comportamiento de descarga.

4.4.3 Parámetros de Voc de carga

Para determinar el Voc de carga de la batería en este estudio, se siguieron los siguientes pasos:

1. Descargar batería hasta el voltaje de fin de descarga (2V)
2. Esperar a que el voltaje se estabilice (2 horas)
3. Cargar la batería con corriente de descarga nominal (.4A) durante 97.5 segundos (cantidad de tiempo para cargar 1% de carga real calculada en la sección anterior)
4. Esperar 30 min para obtener el voltaje de circuito abierto
5. Repetir paso 3 y 4 100 veces

El proceso de extracción de parámetros de Voc de la batería genera la curva en la Figura 4.27. Se pueden apreciar los pulsos con mayor detalle en la Figura 4.28.

Figura 4.27

Curva de Voc carga

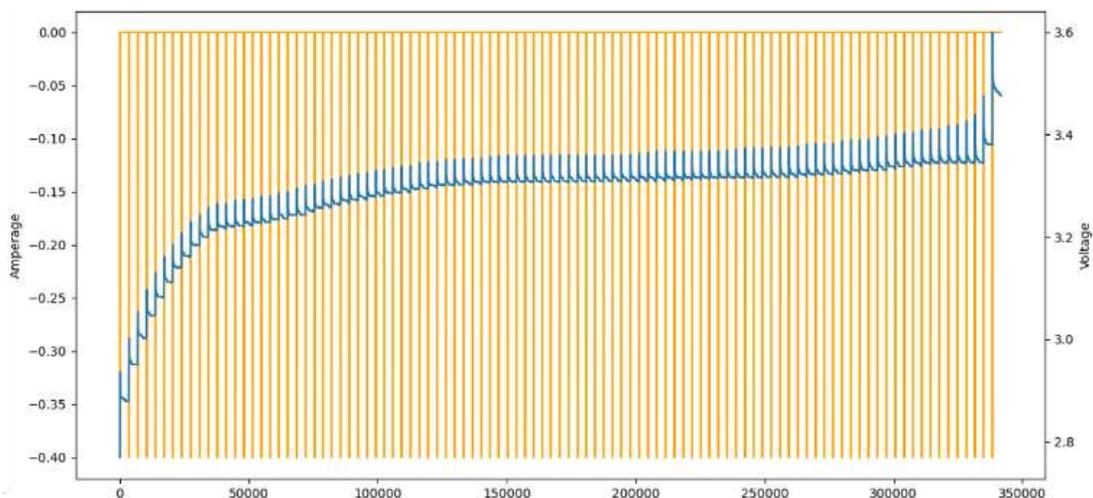
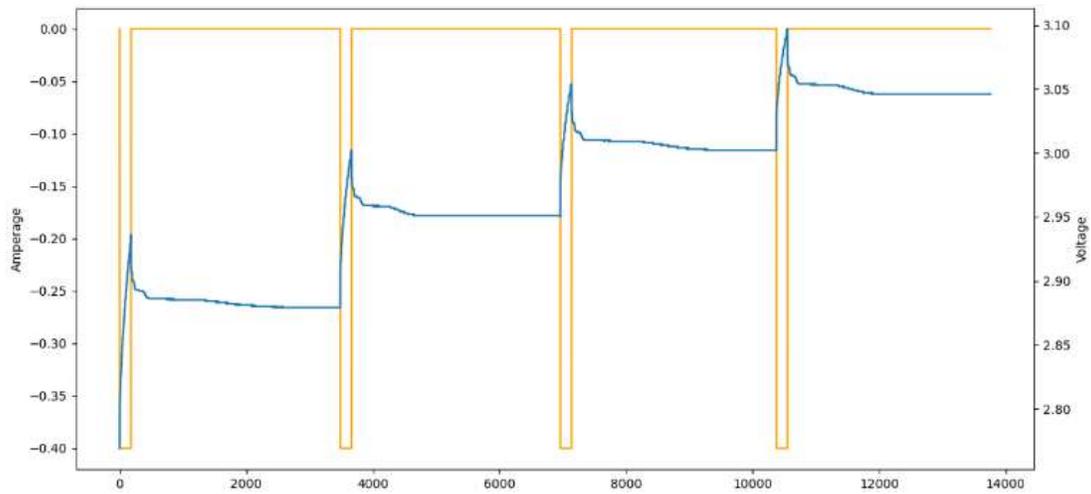


Figura 4.28

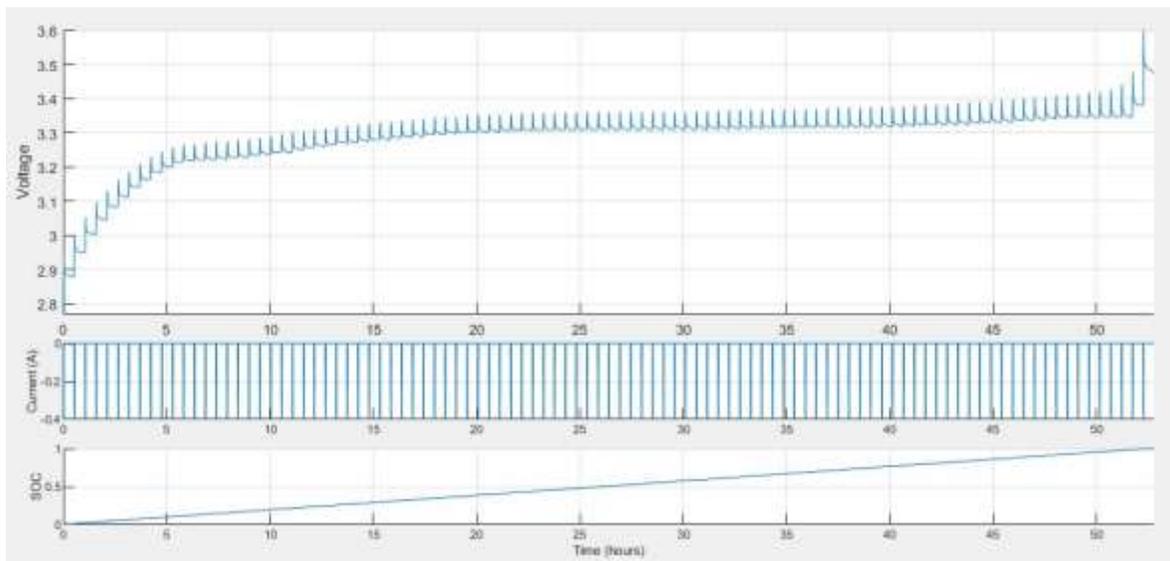
Pulsos de extracción de Voc carga



Estos datos fueron utilizados para el proceso de extracción de parámetros e introducidos en el bloque de batería de circuito equivalente en Matlab. Se puede ver los datos graficados en Matlab en la Figura 4.29. Se puede ver el código en el Anexo 2 sección 1.

Figura 4.29

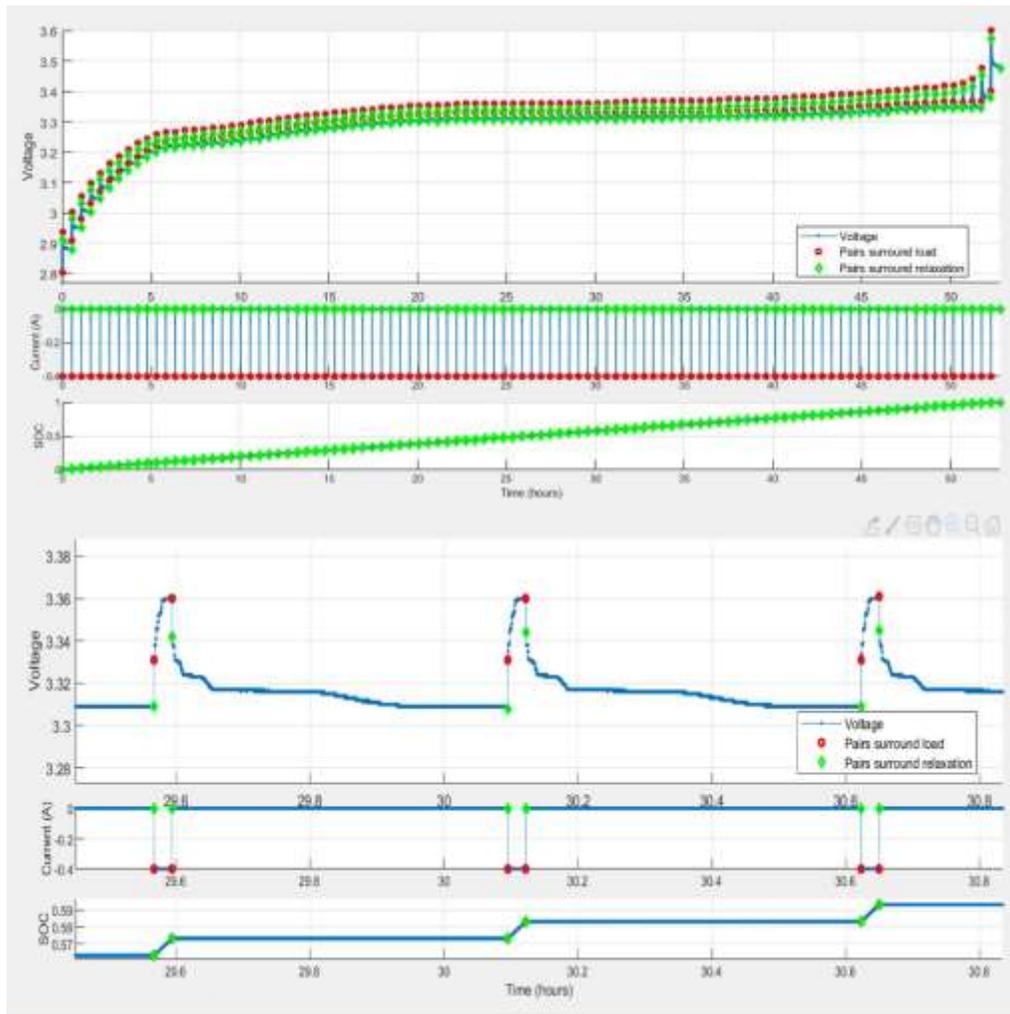
Curva de carga en Matlab



Después se dividieron los datos en objetos de pulsos. La gráfica de los pulsos identificados se puede ver en la Figura 4.30. Se puede ver el código en el Anexo 2 sección 2.

Figura 4.30

Pulsos de carga identificados



Se realizó una comparación para determinar la cantidad óptima de pares RC, y como se puede observar en la Figura 4.31, 4.32 y 4.33, la mejor manera de emular el comportamiento de descarga es utilizando 2 pares RC. Se puede ver el código en el Anexo 2 sección 3.

Figura 4.31

Determinación de pulsos de carga 1, 10, 20 y 30

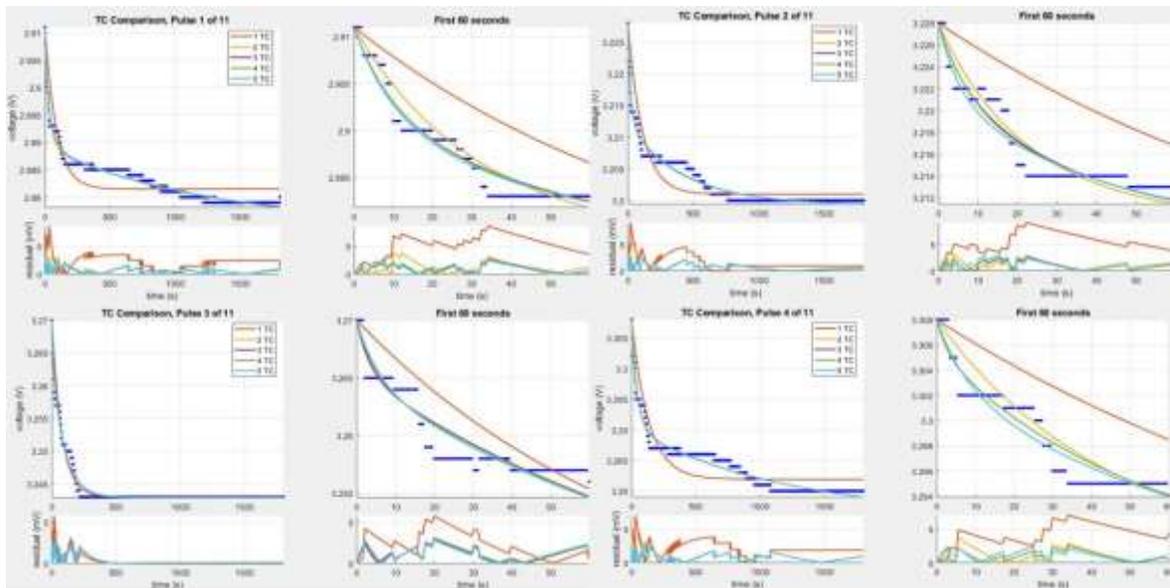


Figura 4.32

Determinación de pulsos de carga 40, 50, 60, y 70

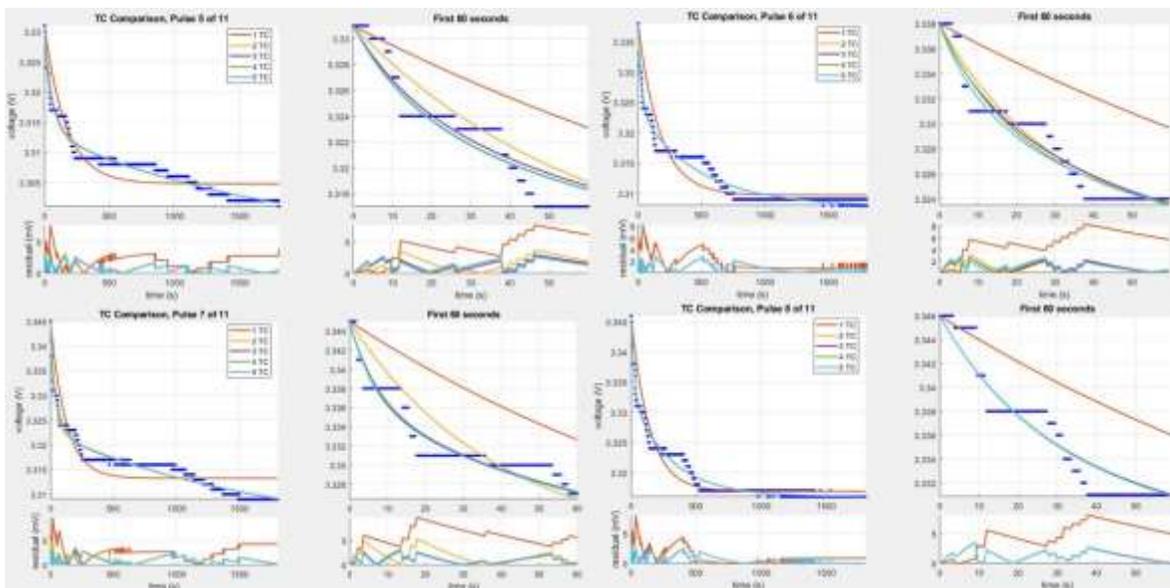
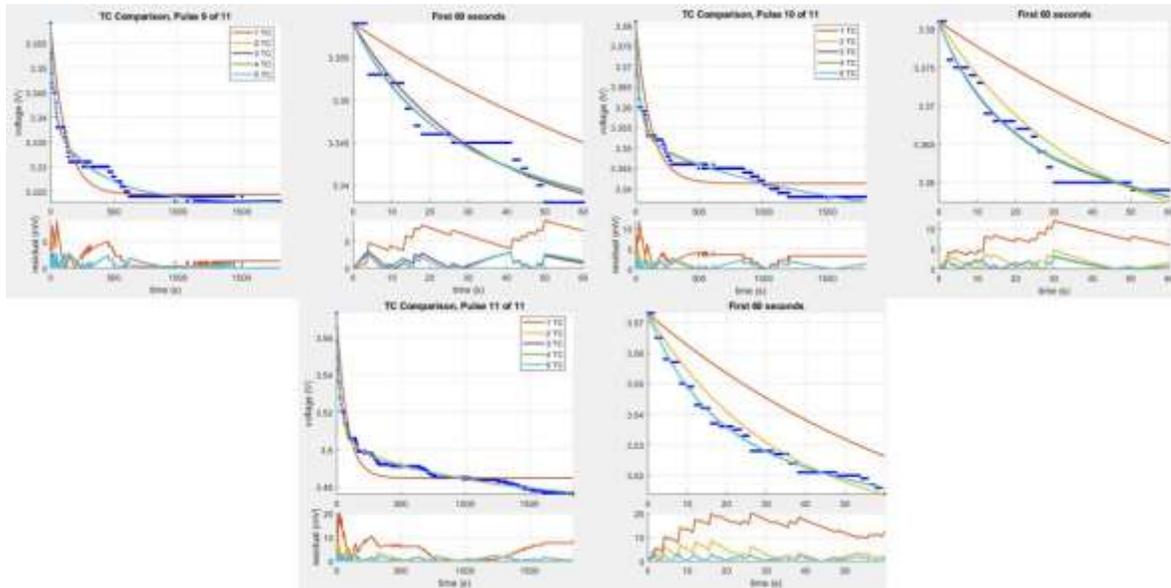


Figura 4.33

Determinación de pulsos de carga 80, 90 y 100



A continuación, se tiene que escoger un modelo de batería adecuado para el modelo escogido, en este caso se modificó, como se muestra en la Figura 4.34, el modelo Thévenin 3RC que proporciona Matlab; ‘BatteryEstim3RC_PTBS’ para que solo tenga 2 pares RC y se llamó ‘BatteryEstim2RC_PTBS’ el cual se puede ver en la Figura 4.35.

Figura 4.34

Modificación al modelo “BatteryEstim3RC_PTBS”

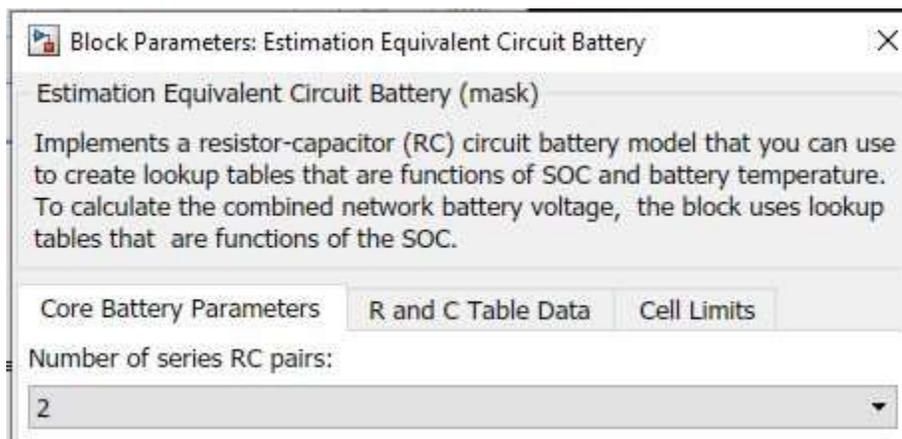
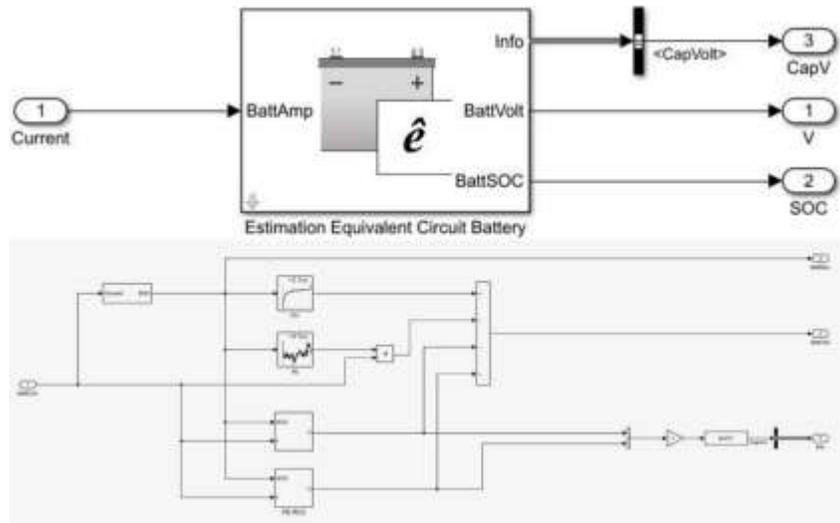


Figura 4.35

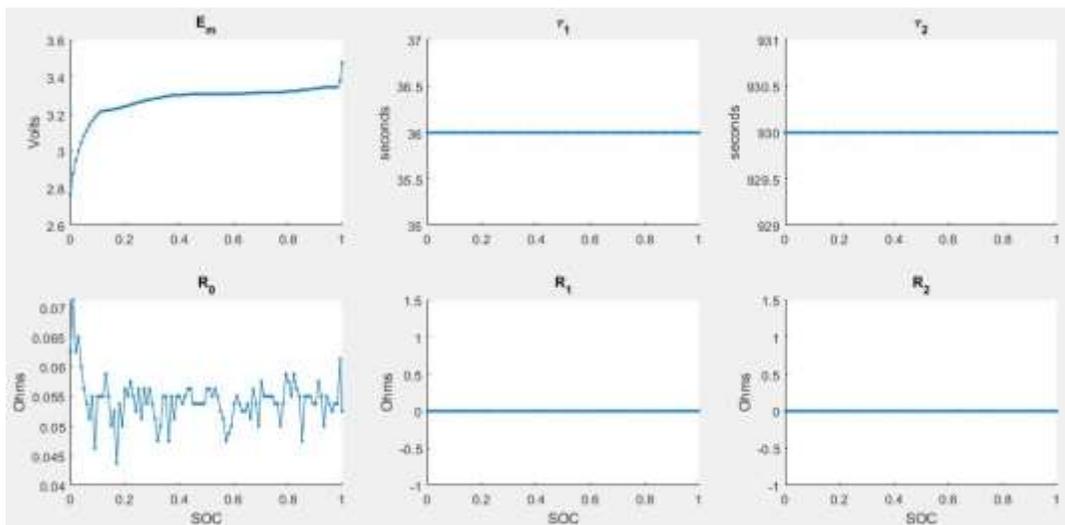
Modelo Thévenin 2RC “BatteryEstim2RC_PTBS”



Se asignaron valores iniciales, máximos y mínimos a los parámetros T1, T2 (Tx). Con los valores asignados, se puede estimó Voc (Em) y Rint (R0). La gráfica de resultados de esta primera estimación se muestra en la Figura 4.36. Se puede ver el código en el Anexo 2 sección 4.

Figura 4.36

Primera estimación de Voc y Rint de carga



A continuación, se estimaron los valores de Tx que anteriormente tenían valores constantes. Se puede ver la estimación de los pulsos 1 y 100 en la Figura 4.37. En este caso la estimación genera un error promedio de 6.3781mV y un error máximo de 139.1345mV. Se pueden ver la estimación de los parámetros Tx y el resultado de la simulación en la Figura 4.38 y 4.39 respectivamente. Se puede ver el código en el Anexo 2 sección 5.

Figura 4.37

Estimación Tx pulsos de carga 1 y 100

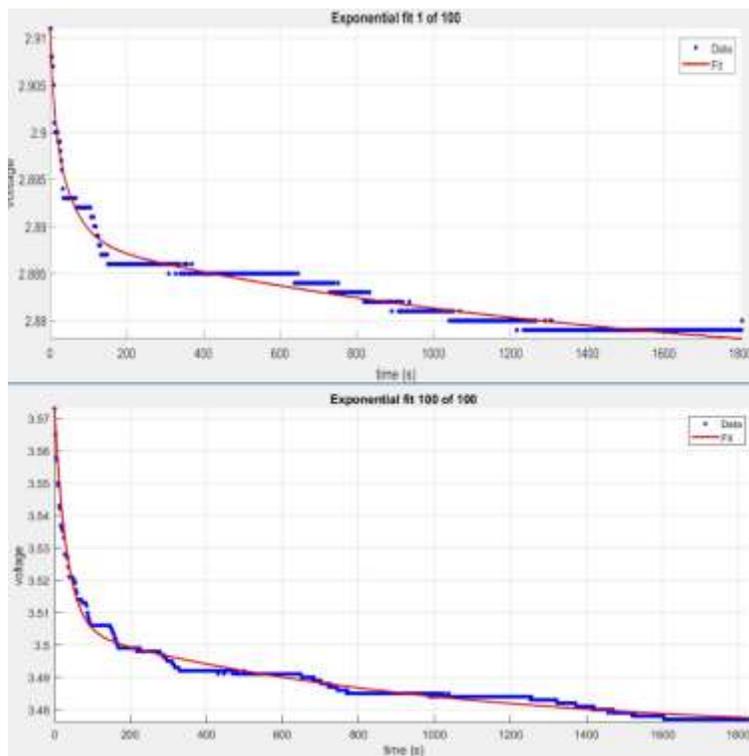


Figura 4.38

Estimación Tx de carga

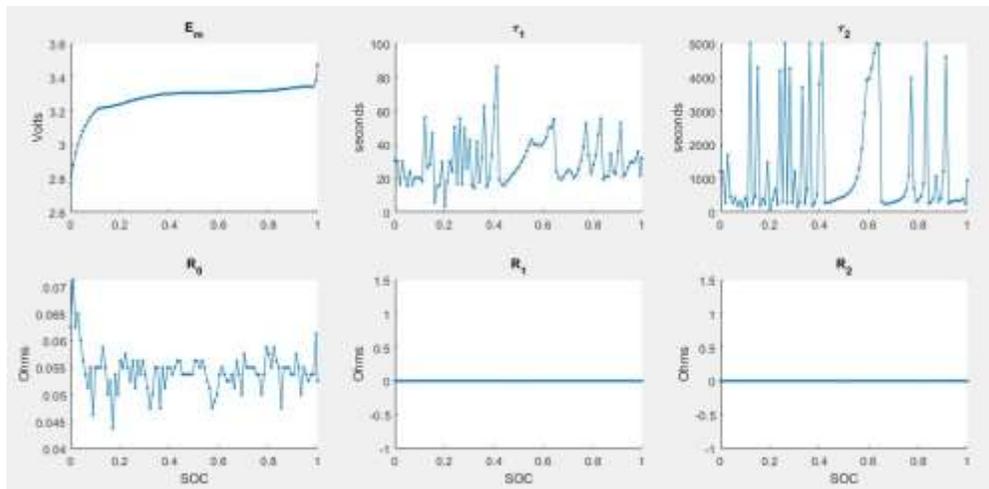
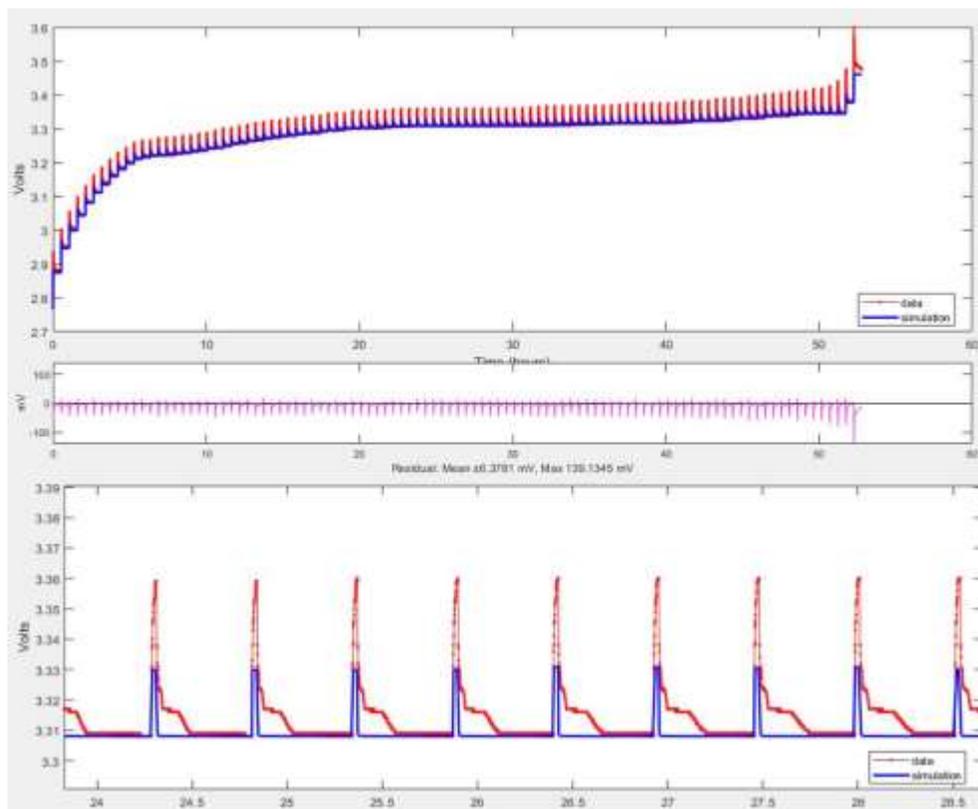


Figura 4.39

Resultados de simulación de carga



Con estos valores de Tx se puede calcular los valores de Rx. Con estos nuevos valores la simulación reduce el error promedio a 3.2758mV y el error máximo a 111.4646mV. Se puede ver la estimación de los pulsos 1 y 100 en la Figura 4.40. Los valores de Rx y los nuevos resultados de simulación se pueden ver en la Figura 4.41 y 4.42 respectivamente. Se puede ver el código en el Anexo 2 sección 6.

Figura 4.40

Estimación Rx pulsos de carga 1 y 100

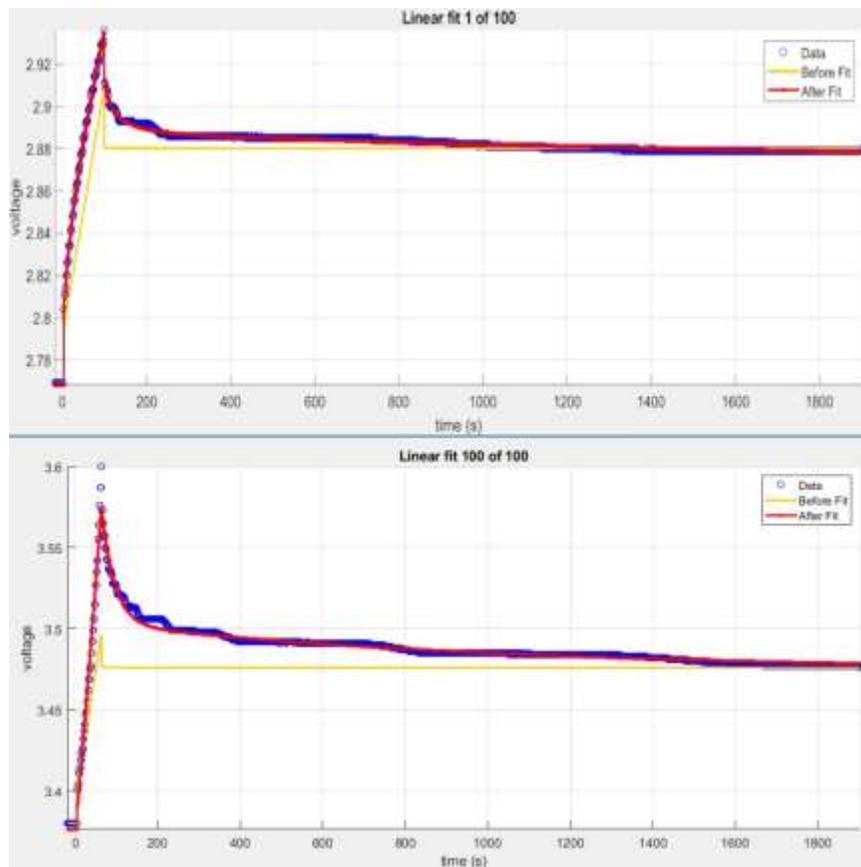


Figura 4.41

Estimación Rx de carga

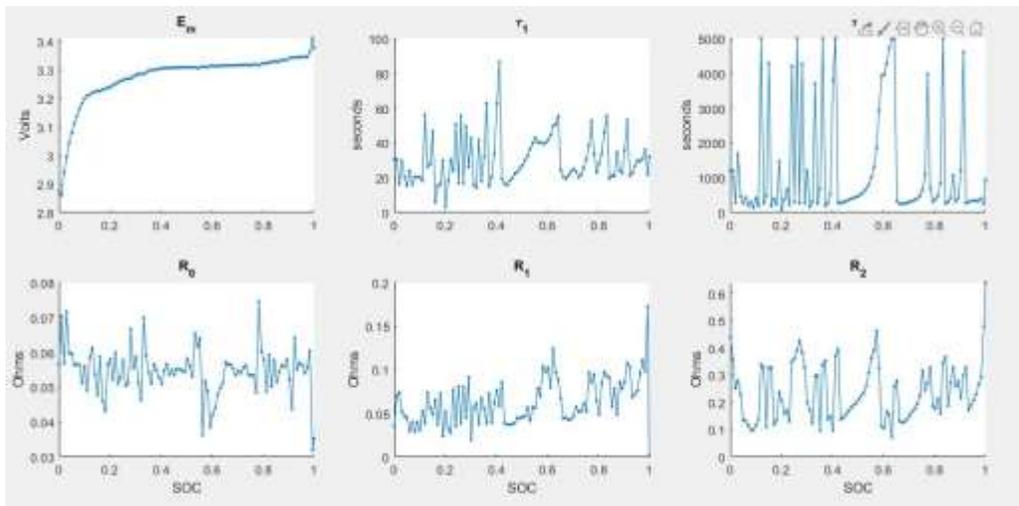
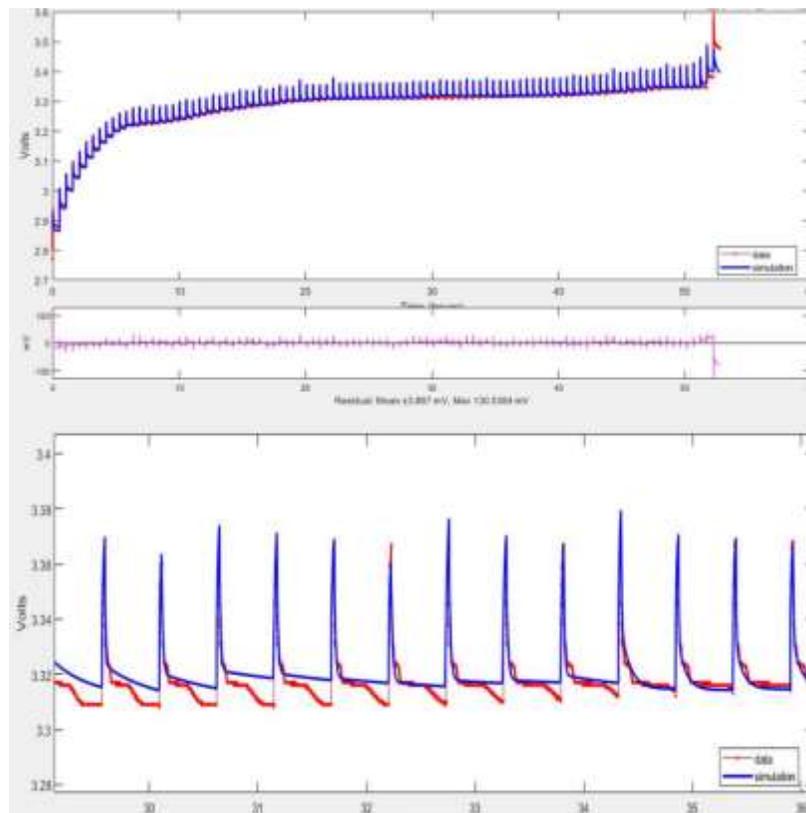


Figura 4.42

Resultados de simulación de carga



Por último, se aplicó una optimización de SDO, la cual ayudó a reducir el error de la simulación a 1.5041mV (0.0457%) de error promedio y 60.4095mV de error máximo. Se puede ver la optimización de los pulsos 1 y 100 en la Figura 4.43. Los valores finales de todos los parámetros de Voc y los resultados finales de simulación se pueden ver en la Figura 4.44 y 4.45 respectivamente. Se puede ver el código en el Anexo 2 sección 7.

Figura 4.43

Optimización SDO pulsos de carga 1 y 100

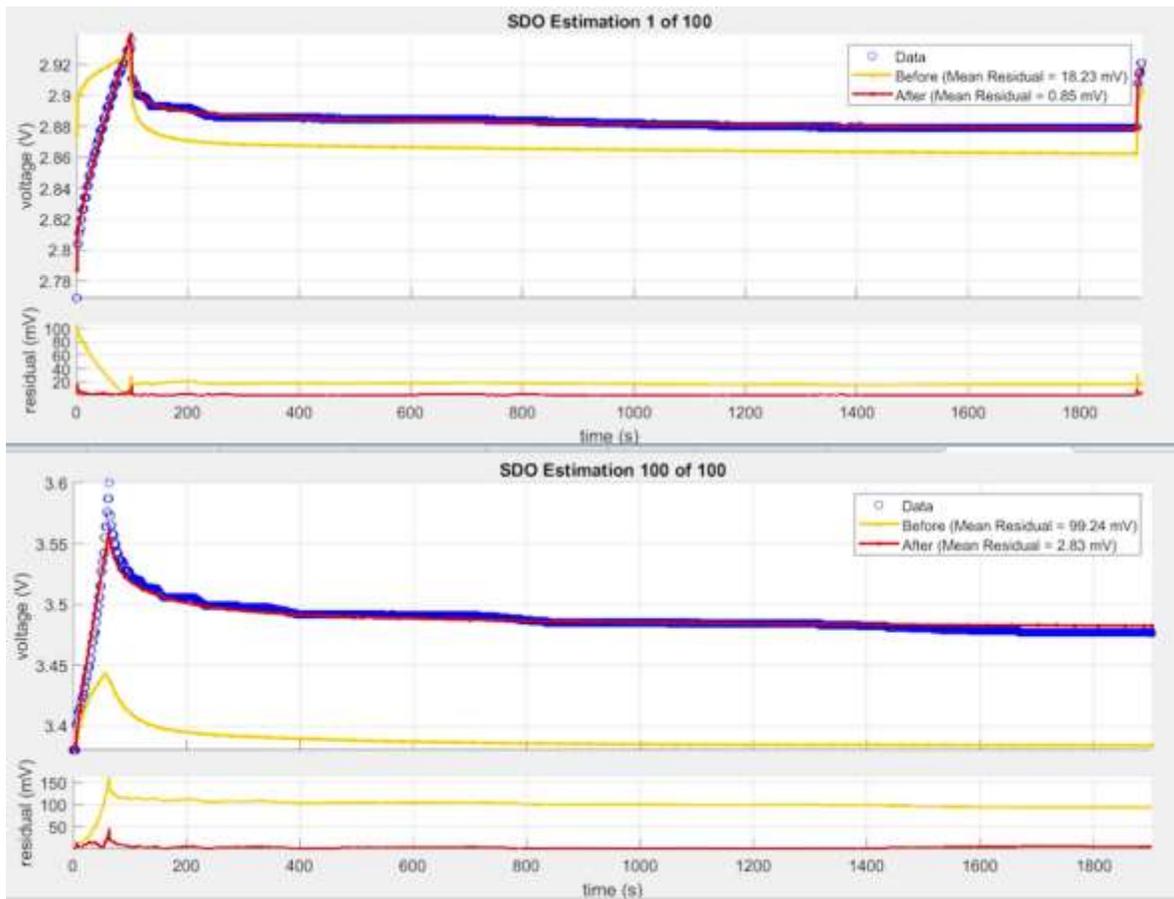


Figura 4.44

Parámetros finales Voc de carga

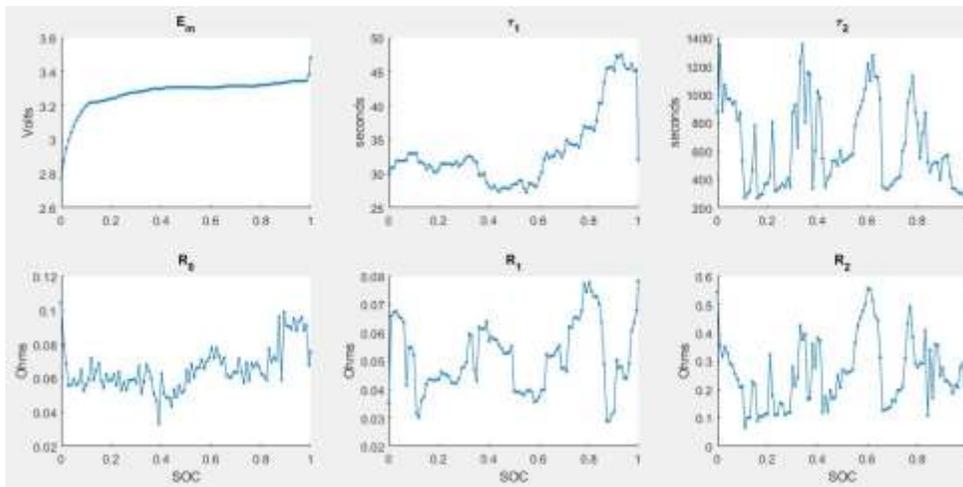
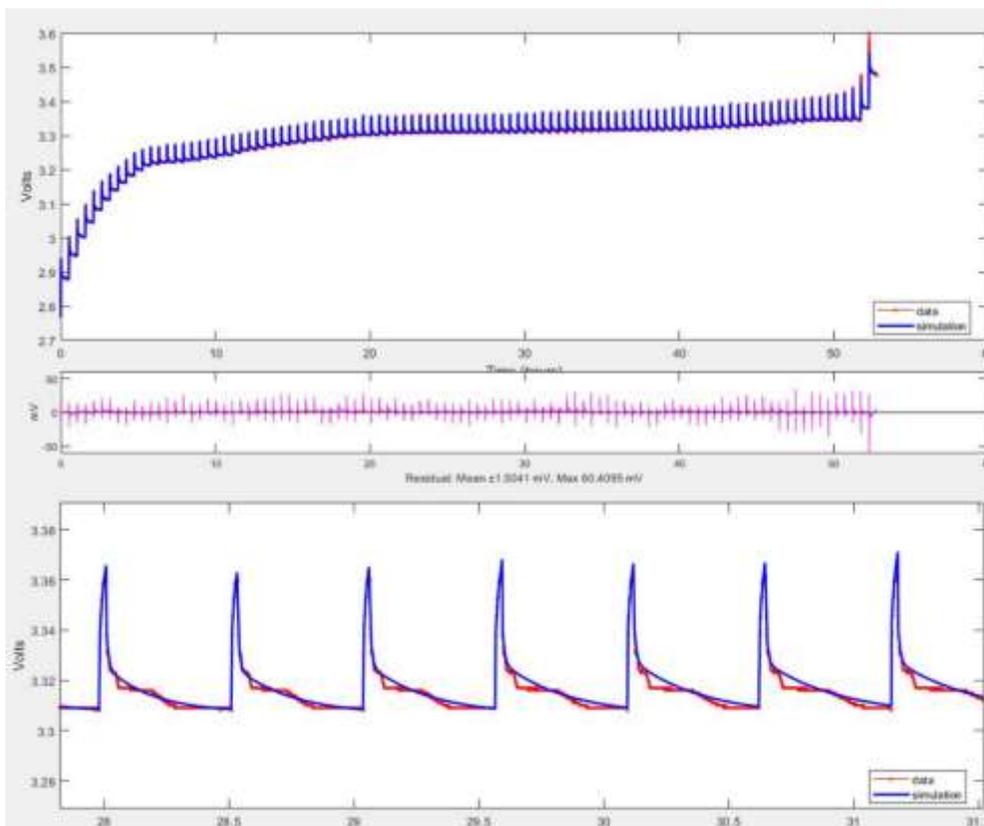


Figura 4.45

Resultados finales de simulación de carga



Los valores finales de carga generados por este proceso de extracción de parámetros se pueden ver en el Anexo 4. Estos serán los valores que se usarán en la tabla de búsqueda en el programa final para estimar el voltaje de salida al emular el comportamiento de carga.

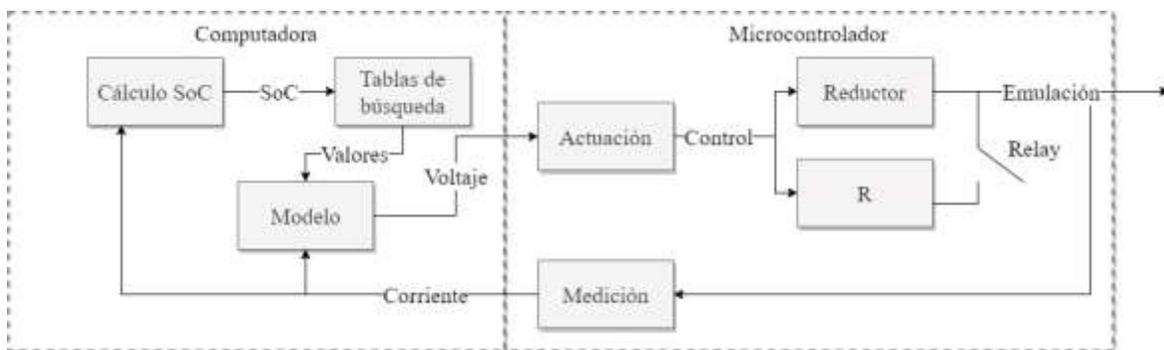
4.5 Arquitectura

En este proyecto de tesis se propone una arquitectura distribuida que consta de una computadora personal con dos tablas de búsqueda (descarga y carga), encargada de calcular el voltaje de salida mediante las ecuaciones diferenciales del modelo Thévenin adecuado. La corriente medida por el circuito se recibirá por medio de comunicación serial, lo que permitirá a la computadora calcular el SOC de la batería. Una vez obtenido el SOC, se utilizarán los valores adecuados de la tabla de búsqueda para calcular el voltaje del modelo.

El circuito medirá la corriente, el voltaje y la dirección de la corriente para determinar si el sistema está en descarga o carga. Estos datos se enviarán por UART a la computadora. El circuito controlará el convertidor reductor y la resistencia variable utilizando un DAC (convertidor digital analógico) mediante un control PID y cambiará entre los modos de carga y descarga utilizando un relay. Se puede ver una representación gráfica de la arquitectura en la Figura 4.46.

Figura 4.46

Arquitectura de emulador



La arquitectura distribuida propuesta en este proyecto de tesis tiene varios beneficios y ventajas en comparación con otros enfoques. En primer lugar, esta arquitectura permite una mayor flexibilidad en el diseño y la implementación del sistema de carga y descarga de la

batería. Al utilizar una computadora personal como el cerebro del sistema, se puede personalizar y adaptar el software para cumplir con los requisitos específicos del proyecto e incluso incluir nuevos módulos como SOH o temperatura e incluir diferentes químicas de batería.

En segundo lugar, la arquitectura distribuida ofrece una mayor precisión y confiabilidad en el cálculo del voltaje de salida, al procesar las ecuaciones diferenciales del modelo adecuado en una computadora, con mayor precisión de punto flotante, a diferencia de la precisión de un microcontrolador.

Finalmente, esta arquitectura presenta ventajas en cuanto a la reducción de costos y la escalabilidad. Al correr el modelo en una computadora personal, no es necesario utilizar una máquina tipo HIL, como se requiere en otros enfoques. Al emplear un microcontrolador sencillo y económico, los costos se reducen, permitiendo la producción más económica del dispositivo. Y así permitir que se utilice por varios desarrolladores dentro de un proyecto que tenga una batería.

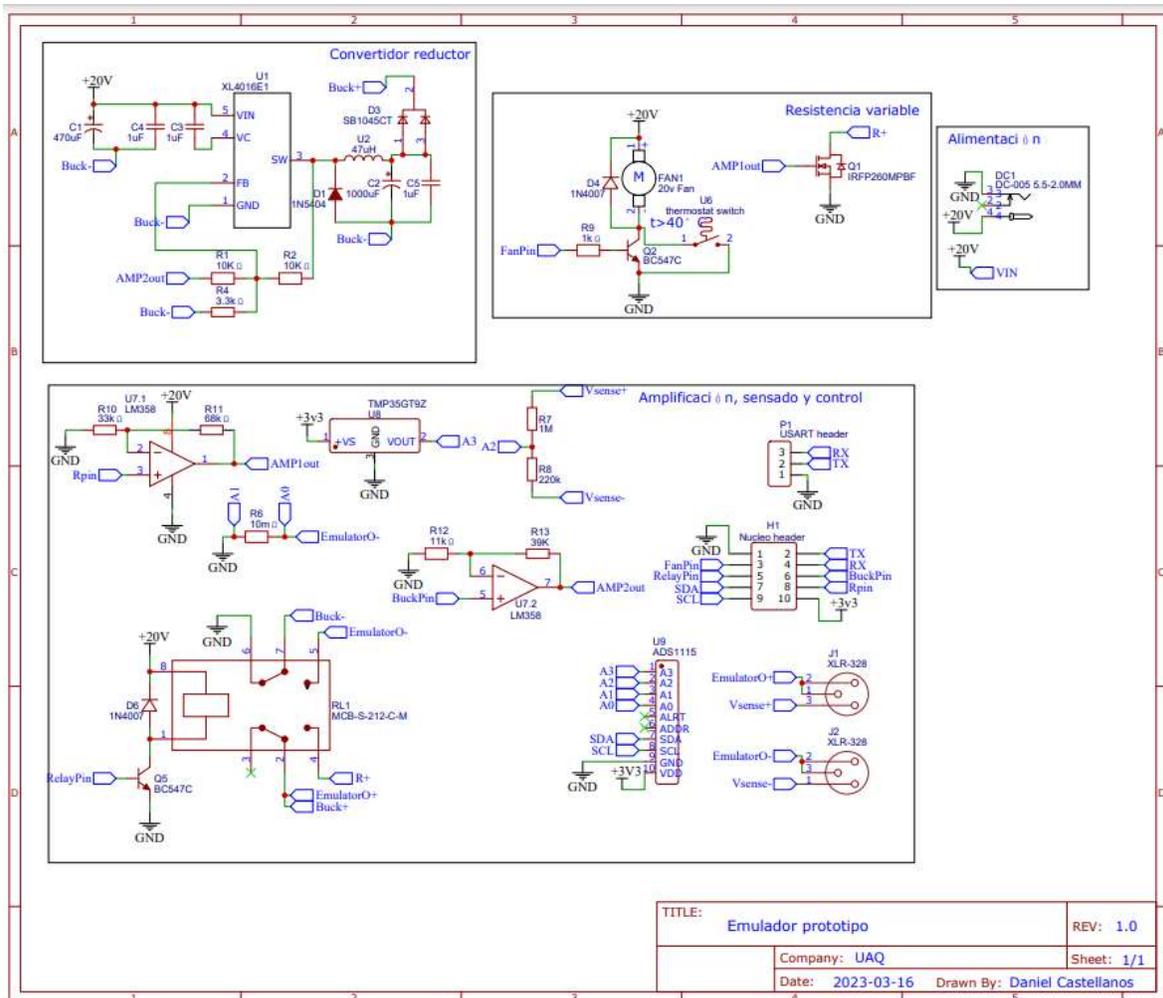
En resumen, la arquitectura distribuida propuesta en este proyecto de tesis ofrece una solución flexible, precisa y económica para el diseño y la implementación de sistemas de carga y descarga de baterías. Al utilizar una computadora personal, microcontrolador y un circuito de medición y control, se puede obtener un sistema que cumpla con los requisitos específicos de diversos proyectos.

4.6 Circuito utilizado

Como se explicó previamente, el proyecto utiliza un convertidor reductor y una resistencia variable, ambos controlados mediante un relay. El esquemático completo se muestra en la Figura 4.47 y cada sección se detalla en las Figuras 4.48, 4.49, 4.50 y 4.51.

Figura 4.47

Esquemático de circuito de prototipo

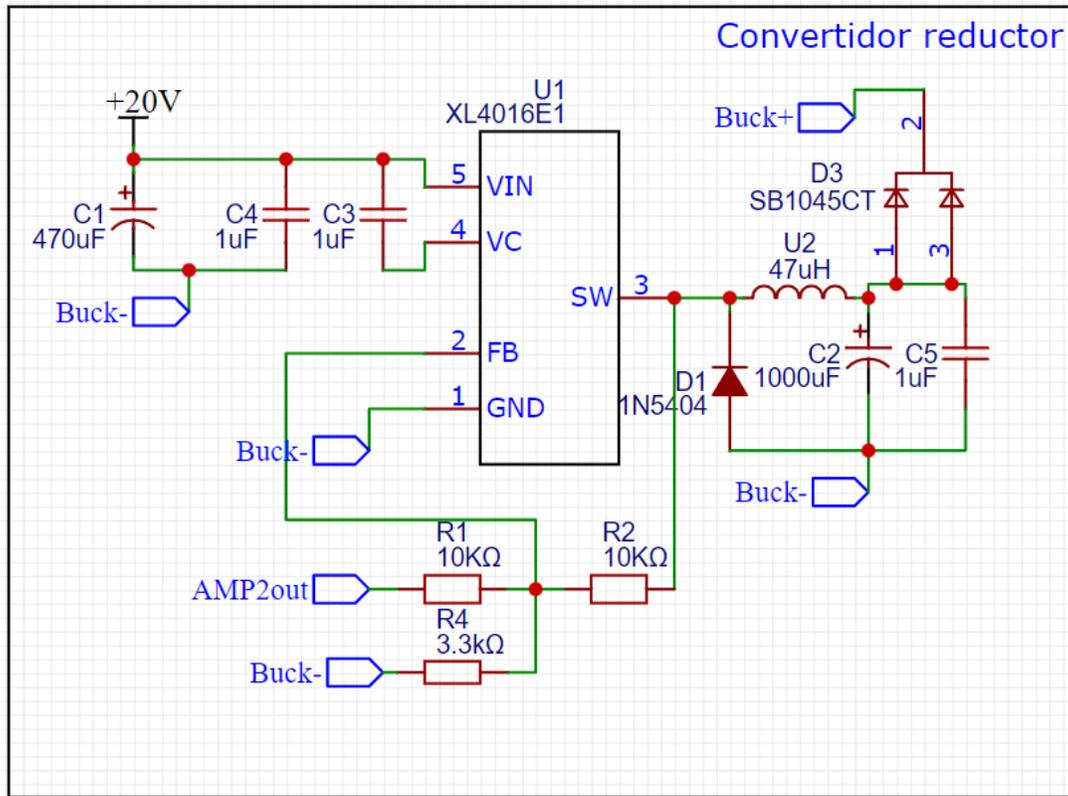


En la Figura 4.48 se muestra el convertidor reductor XL4016E1 utilizado en el proyecto, capaz de proporcionar hasta 8A de corriente de salida y una salida de 1.25V a 36V. Dado que la batería seleccionada a emular solo genera un voltaje nominal de entre 2V y 3.6V, este convertidor es adecuado para el proyecto y puede ser utilizado para emular sistemas de baterías que tengan un voltaje y corriente de salida mayor. El voltaje de control del convertidor se aplica al pin FB y se puede generar utilizando el DAC del microcontrolador, amplificado por un amplificador operacional en configuración de amplificador no inversor. Alternativamente, se podría utilizar un filtro RC con PWM para aplicar el voltaje de control

en lugar del DAC. Además, se usa un doble diodo SB1045CT para proteger el convertidor en contra de corriente inversa.

Figura 4.48

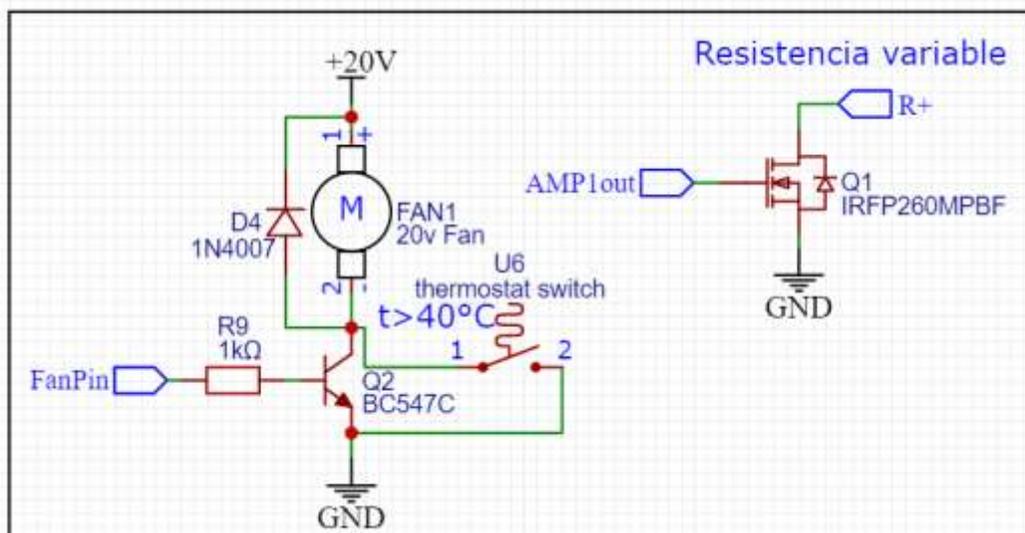
Esquemático de convertidor reductor



De la misma forma, se emplea un MOSFET IRFP260 como resistencia variable, que permite regular la corriente que fluye entre el drenador y la fuente al variar el voltaje en su compuerta entre 0V y 10V (voltaje de saturación), funcionando efectivamente como una resistencia variable. De igual modo se ha incluido un ventilador controlado por el microcontrolador y equipado con un termistor de seguridad que se activa a 40°C.

Figura 4.49

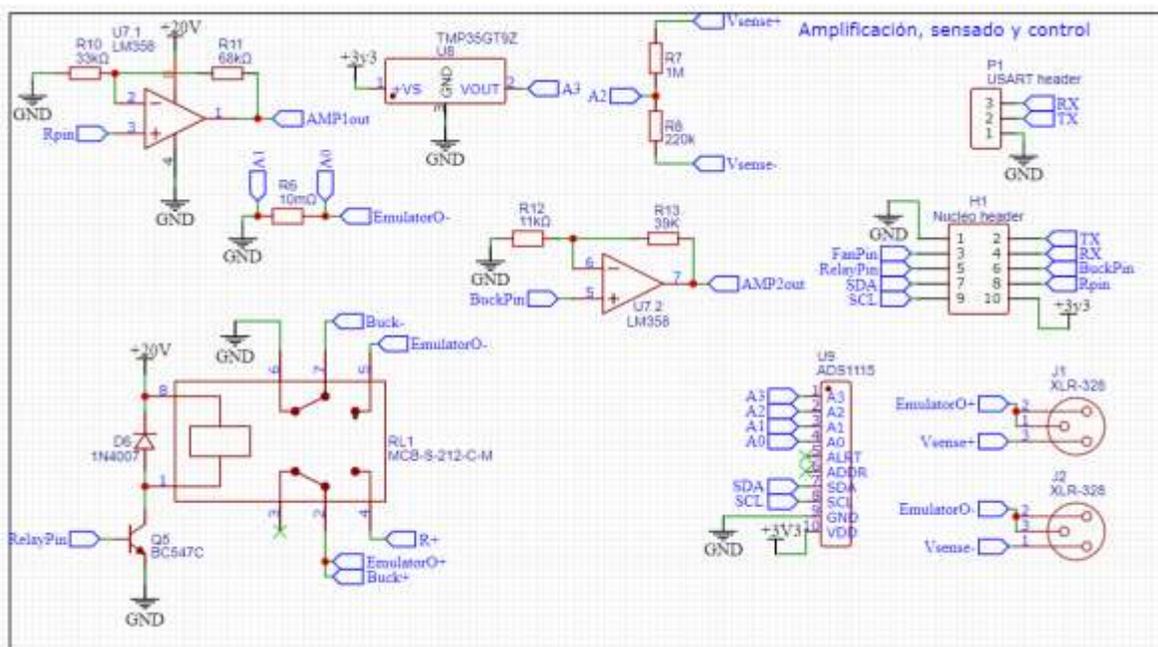
Esquemático de resistencia variable



Para facilitar la conexión externa del microcontrolador en este circuito de prototipado, se dejaron headers disponibles. De la misma manera, se incluyeron dos amplificadores operacionales para amplificar las señales del DAC hacia el convertidor reductor y la resistencia variable. Para cambiar entre los modos de descarga y carga, se utilizó un relay doble, en el cual una mitad activa la resistencia variable y la otra mitad cambia las tierras para emular el comportamiento de carga y obtener una medición precisa de la corriente. La medición de corriente, voltaje y temperatura del disipador de calor se realiza a través de un ADS1115, un ADC de 16 bits.

Figura 4.50

Esquemático de amplificación, medición y control



Se puede ver en la Figura 4.51 el circuito simplificado del emulador. En la Figura 4.52 se puede ver el circuito generado al momento de descarga y en la Figura 4.53 el circuito simplificado generado al momento de carga.

Figura 4.51

Circuito simplificado de emulador

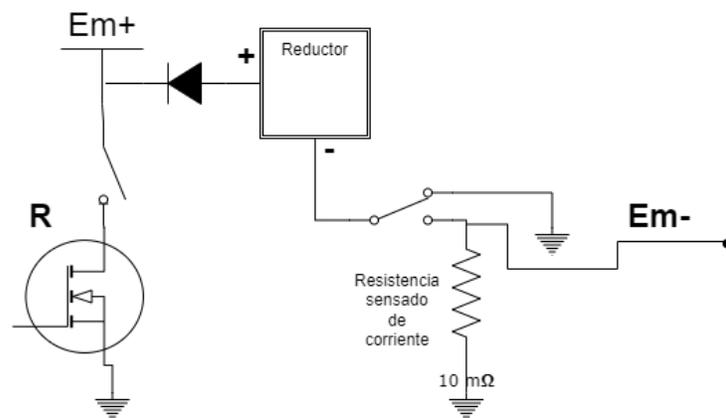


Figura 4.52

Circuito simplificado de descarga

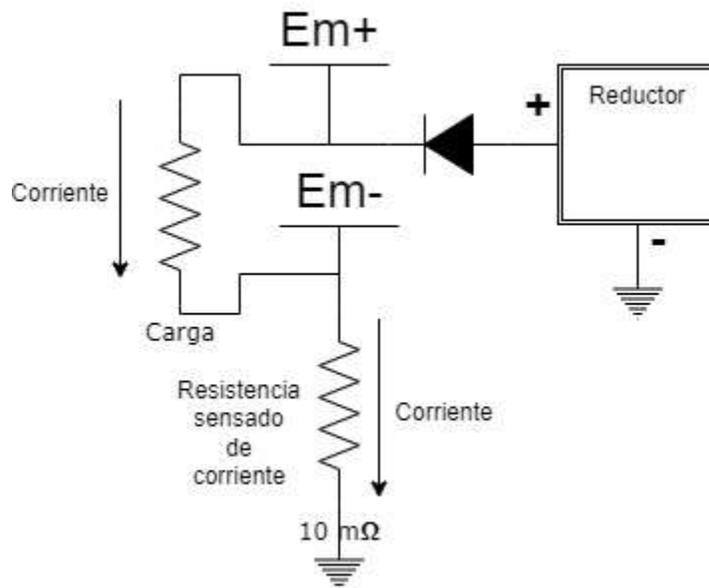
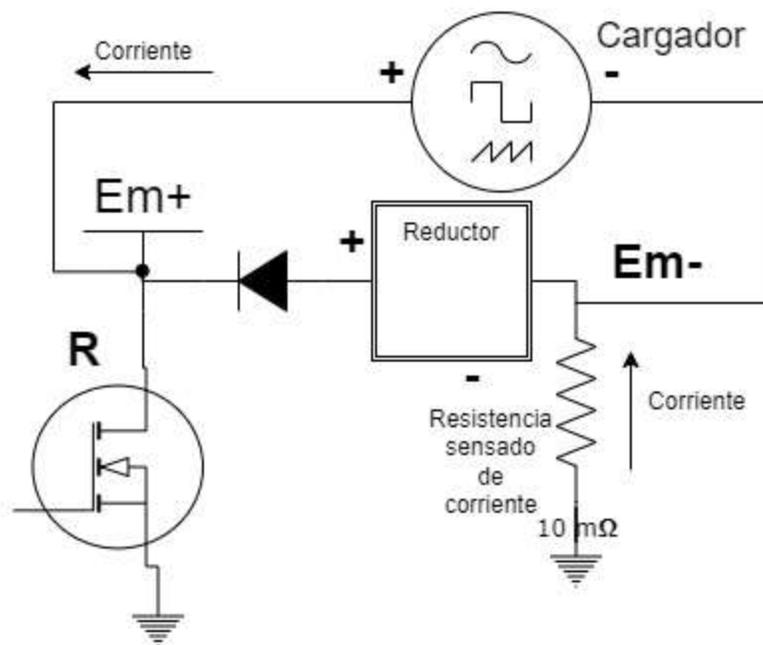


Figura 4.53

Circuito simplificado de carga



Se hizo un circuito impreso (PCB) del esquemático anterior para realizar las pruebas. En la Figura 4.54 se puede ver la cara superior y en la Figura 4.55 la cara inferior del circuito. En la Figura 4.56 se puede ver el circuito armado en físico. El PCB mide 66.04mm por 45.212mm

Figura 4.54

Cara superior de PCB de prototipo

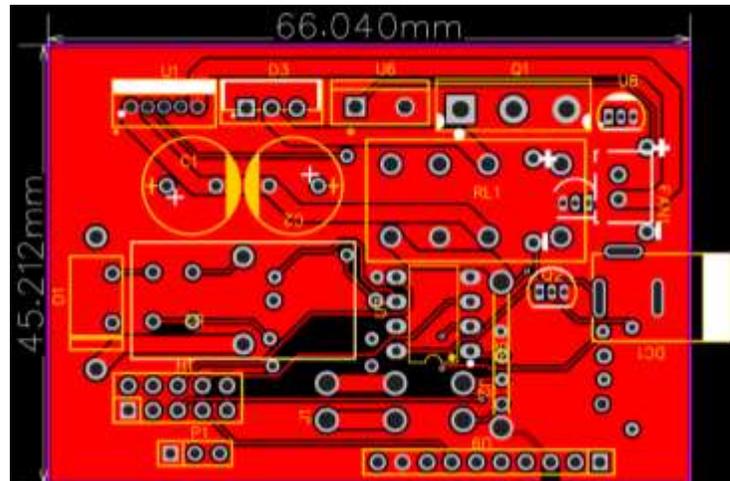


Figura 4.55

Cara inferior de PCB de prototipo

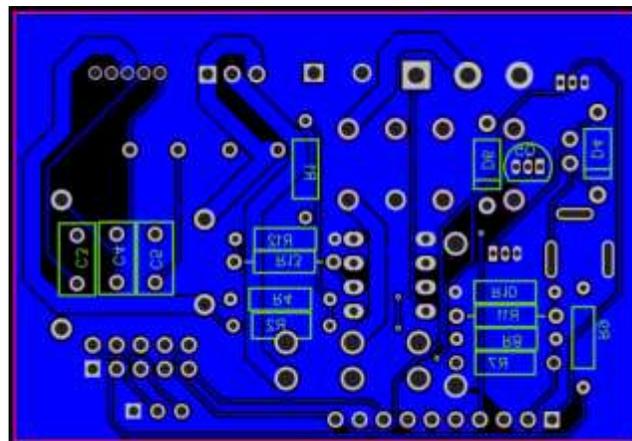
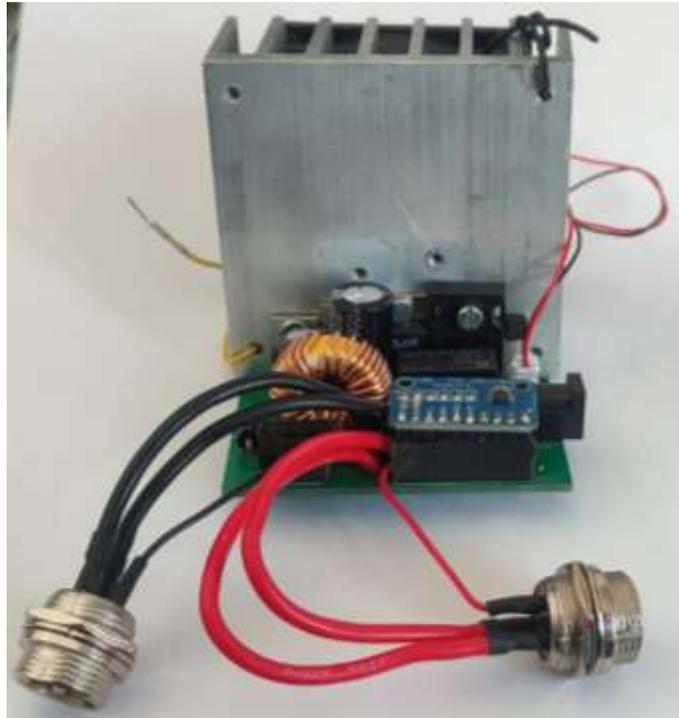


Figura 4.56

Circuito físico de prototipo



De igual manera se planeó un esquemático para un producto final con el microcontrolador montado en la placa. Se puede ver en la Figura 4.57 el esquemático, en la Figura 4.58 la cara superior, en la Figura 4.59 la cara inferior y en la Figura 4.60 un modelo 3D del circuito final. En el Anexo 5 se puede ver la lista de materiales (BOM) del circuito final.

Figura 4.57

Esquemático circuito de producto final

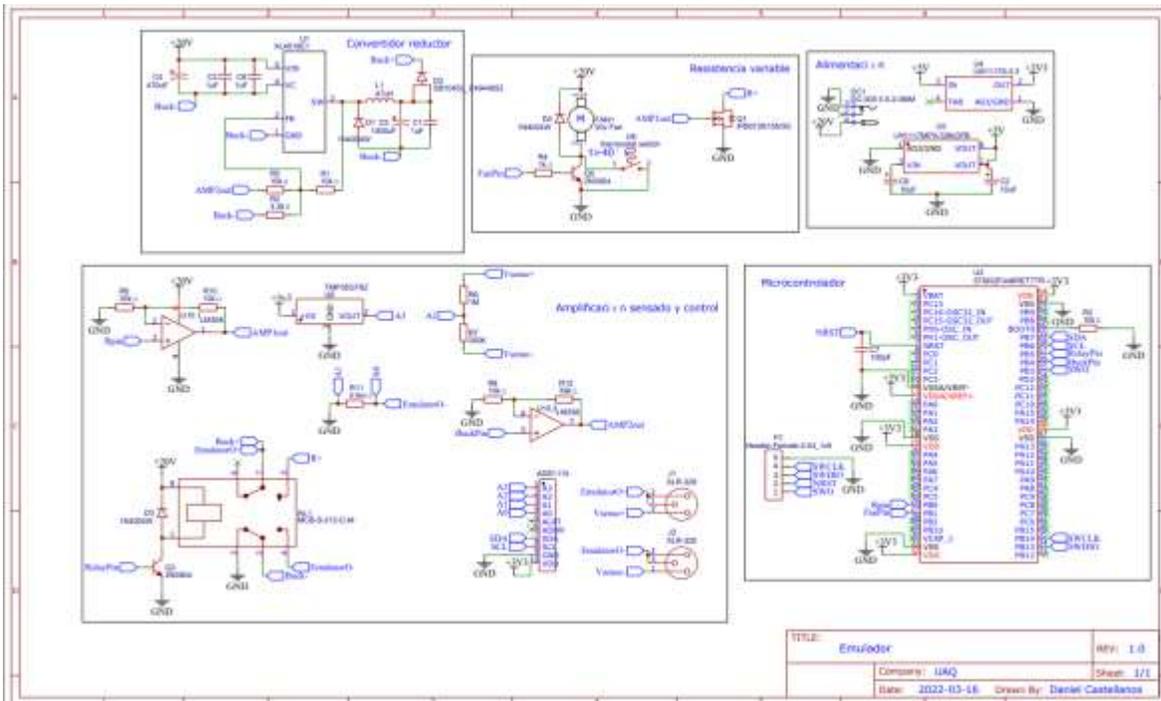


Figura 4.58

Cara superior PCB de producto final

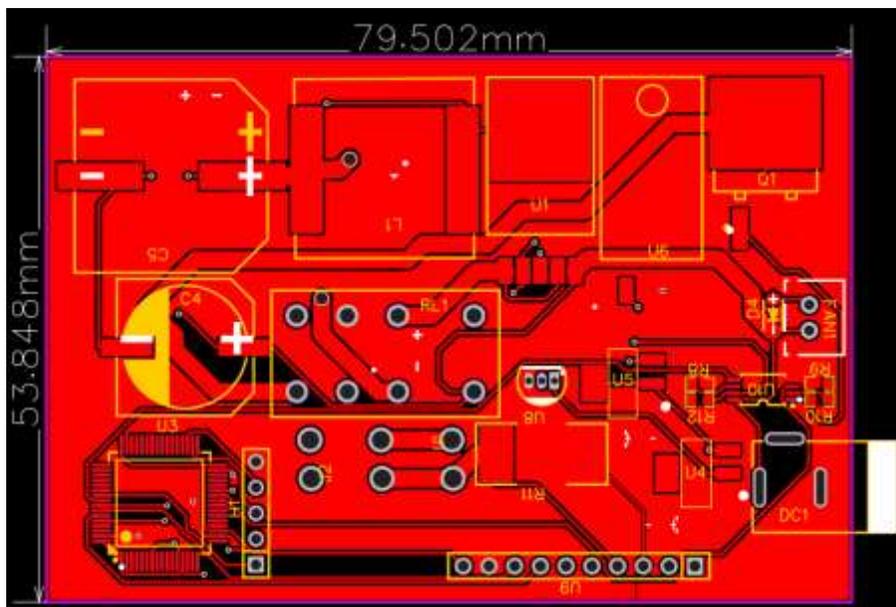


Figura 4.59

Cara inferior PCB de producto final

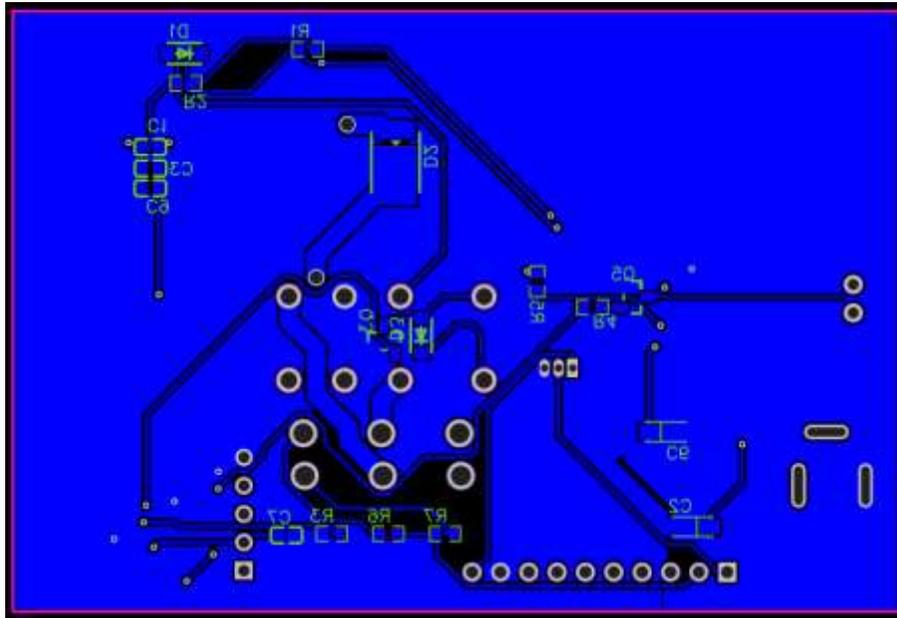
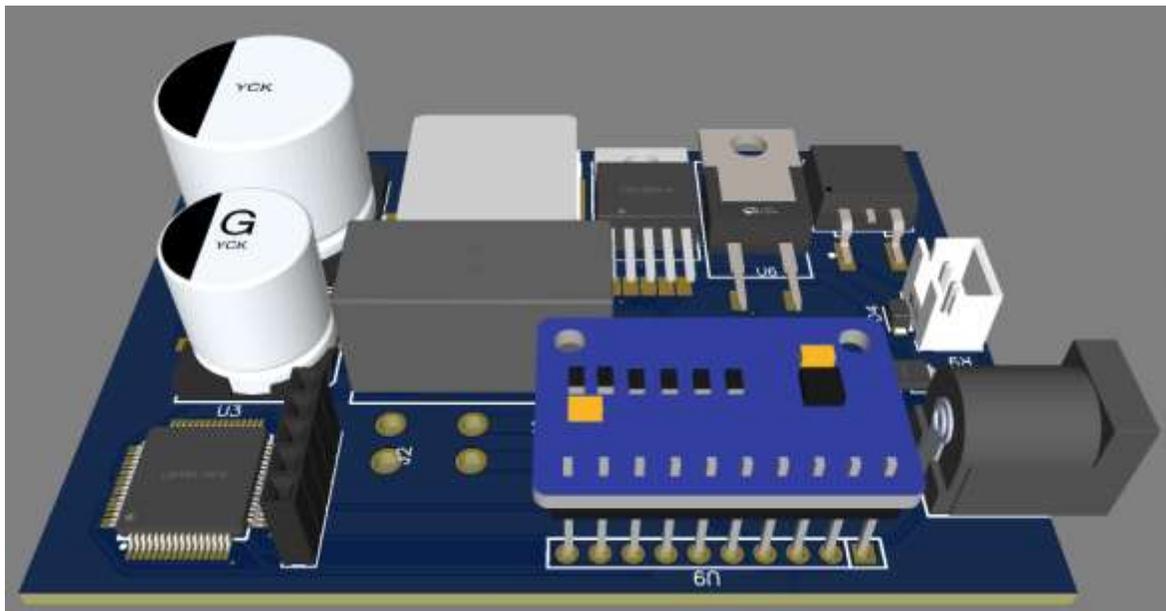


Figura 4.60

Modelo 3D PCB de producto final



4.7 Programación de GUI

La interfaz gráfica de usuario (GUI) es un componente esencial del sistema del emulador ya que proporciona una interfaz fácil de usar para controlar y monitorear el comportamiento del emulador. En esta sección, se discutirá el desarrollo de una GUI utilizando la biblioteca Tkinter de Python, que permitirá a los usuarios interactuar fácilmente con el emulador. La GUI se basará en la arquitectura Modelo-Vista-Controlador (MVC), separando la gestión de datos, la presentación y el control de la interfaz de usuario.

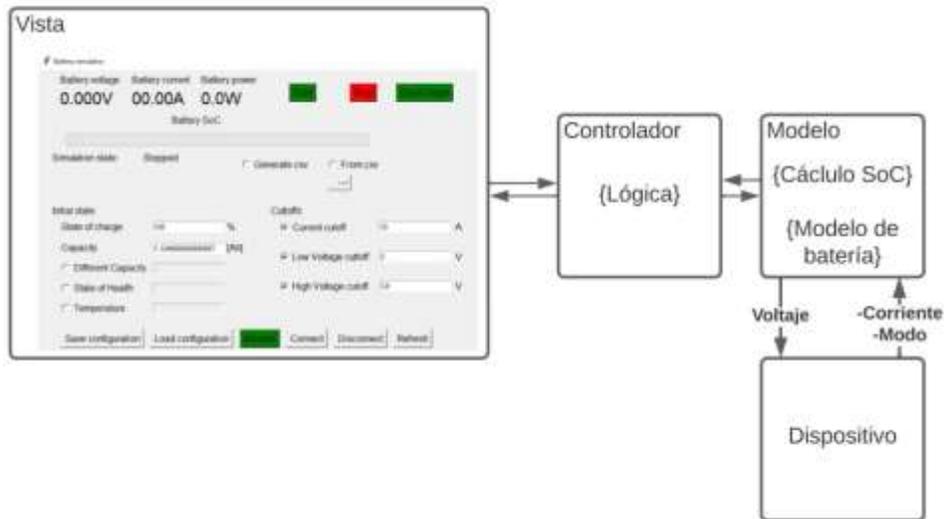
El componente de software de "Vista" de la GUI se encargará de administrar las diferentes ventanas y la interacción con el usuario a través de botones, cuadros de diálogo, etc. Como se dijo antes esto se desarrollará con la librería Tkinter. De esta forma, la GUI mostrará datos en tiempo real del emulador, incluidos el voltaje de salida, la corriente y el modo. Se usará la biblioteca Matplotlib para crear un gráfico en tiempo real del voltaje y el comportamiento actual de la batería.

La comunicación entre el componente "Vista" y el "Controlador" se activará cuando se reciban entradas del usuario por medio de la GUI. El controlador gestionará la lógica del dispositivo y enviará la información correspondiente al componente "Modelo" para obtener las cifras calculadas que se mostrarán en la "Vista".

De igual modo, el "Modelo" procesará los comandos recibidos del "Controlador" para determinar el voltaje de salida necesario, basándose en el SOC calculado, el intervalo entre cada medición, la corriente, el modo de operación detectados por el dispositivo, recibidos a través de una interfaz serial utilizando la biblioteca pySerial. Tras calcular el voltaje de salida, el "Modelo" lo enviará al dispositivo mediante la misma comunicación serial. Se puede ver en la Figura 4.61 un diagrama de la arquitectura de la GUI.

Figura 4.61

Arquitectura de GUI

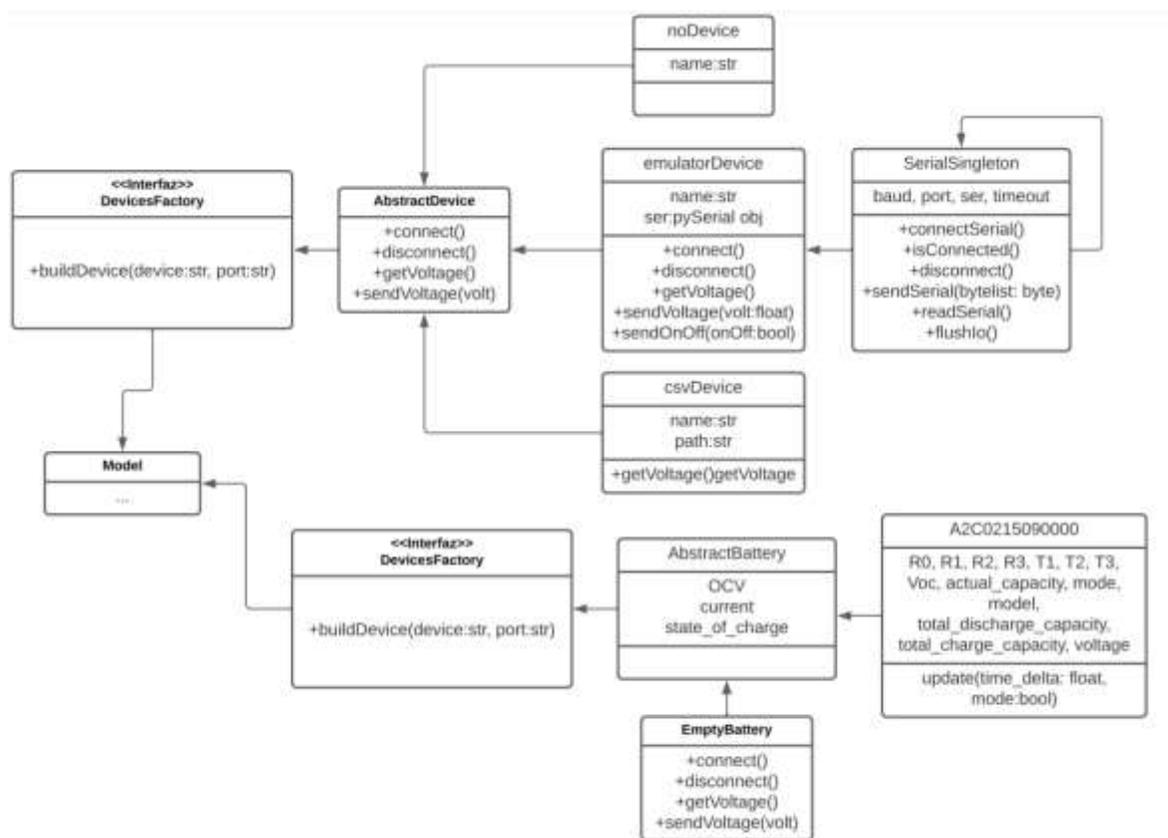


La programación de los componentes del modelo se llevó a cabo considerando diversos patrones de diseño, permitiendo que el código sea escalable y modular. Se empleó el patrón de diseño Abstract Factory con el fin de crear diferentes dispositivos. Se configuraron tres dispositivos: "No device", utilizado cuando no se ha seleccionado ningún dispositivo; "csv", que se emplea cuando los datos se obtienen de un archivo de valores separados por comas (csv); y "Emulador", destinado para el dispositivo emulador.

El patrón de diseño Abstract Factory también se empleó para añadir el modelo de la batería, lo que proporciona la capacidad de incluir diferentes modelos en futuras implementaciones. Además, se empleó el patrón de diseño Singleton para la interfaz serial, permitiendo la existencia de un único objeto encargado de gestionar la comunicación serial. Se puede ver en la Figura 4.62 un diagrama de clases de los patrones de diseño utilizados.

Figura 4.62

Diagrama de clases de patrones de diseño



El código del software de la GUI se encuentra detallado en los Anexos 6, 7, 8, 9 y 10, pero aquí se presentará un breve resumen de su funcionalidad principal. El desempeño del software se rige principalmente por dos banderas: `PlotFlag` y `SimStartedFlag`. Estas banderas se activan mediante los botones de inicio y parada de la simulación, tal como se ilustra en la Figura 4.63.

Al conectar un dispositivo, se invoca la función `readData()`. Esta función extrae el voltaje del dispositivo usando la función `getLabelData()`. Si la simulación aún no ha comenzado, `readData()` simplemente se llama a ella misma cada 500 milisegundos para actualizar la información en las etiquetas de voltaje, corriente y potencia.

La función `getLabelData()` recoge el voltaje a través de la conexión serial. Si la simulación ya está en marcha, esta función calcula el voltaje de salida con base en el modelo de la batería,

y verifica la existencia de posibles errores en la simulación, como sobrevoltaje, SOC menor a cero, entre otros. Esta función igualmente devuelve los datos leídos a través de la interfaz serial.

Por otro lado, la función `animate()` se ejecuta cada 500 milisegundos y obtiene los datos utilizando `readData()`. Es esta función, `animate()`, la que activa el cálculo del voltaje de salida cada medio segundo. El comportamiento de este programa de la GUI puede observarse con más detalle en la Figura 4.63.

Del mismo modo se agregó la opción de simular el comportamiento de una batería a partir de un archivo csv. Esta opción es muy útil para probar de manera rápida la efectividad de un modelo. Se puede ver imágenes de la GUI en las Figuras 4.64, 4.65 y 4.66.

Figura 4.63

Comportamiento programa de GUI

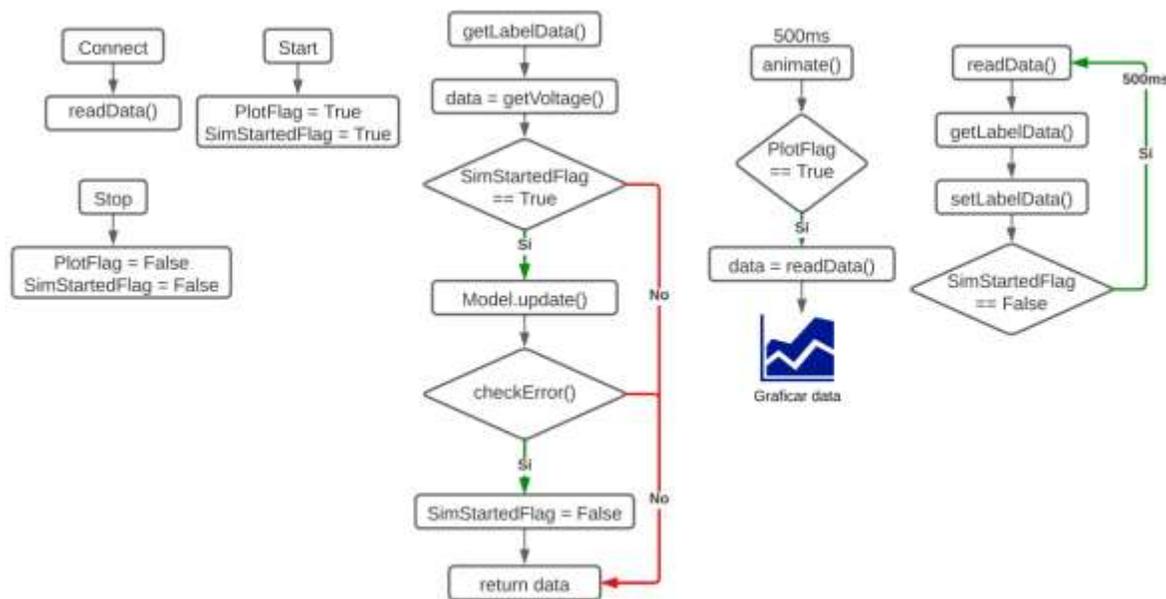


Figura 4.64

Pantalla principal de GUI

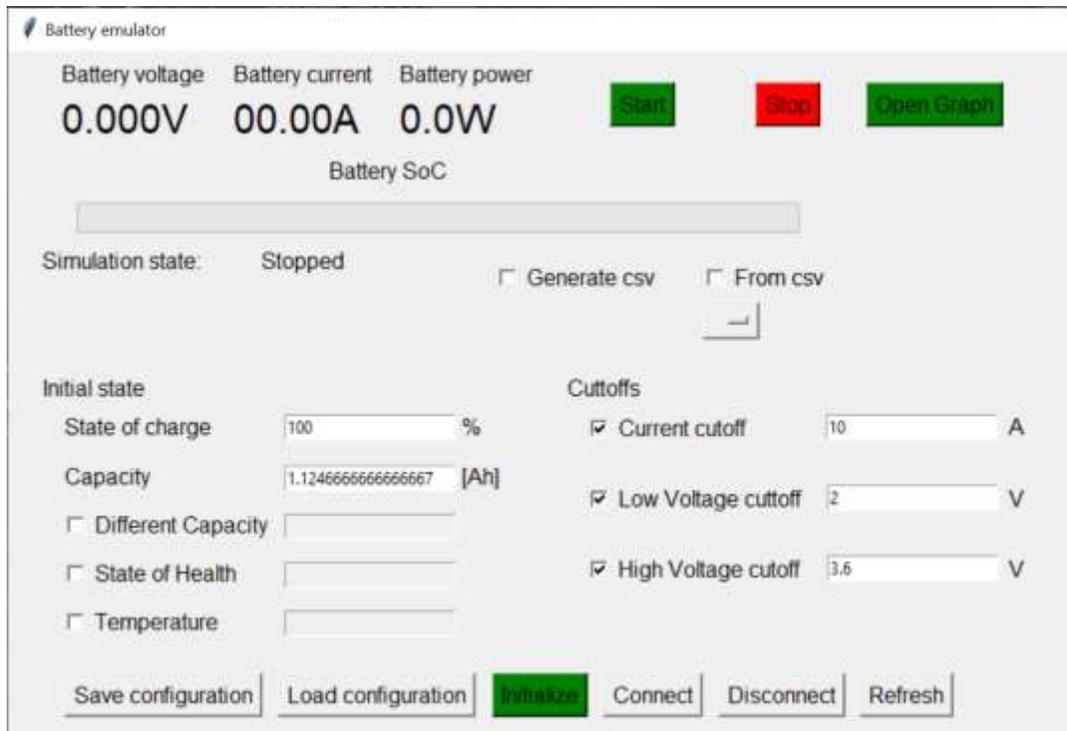


Figura 4.65

Pantalla de grafica de GUI

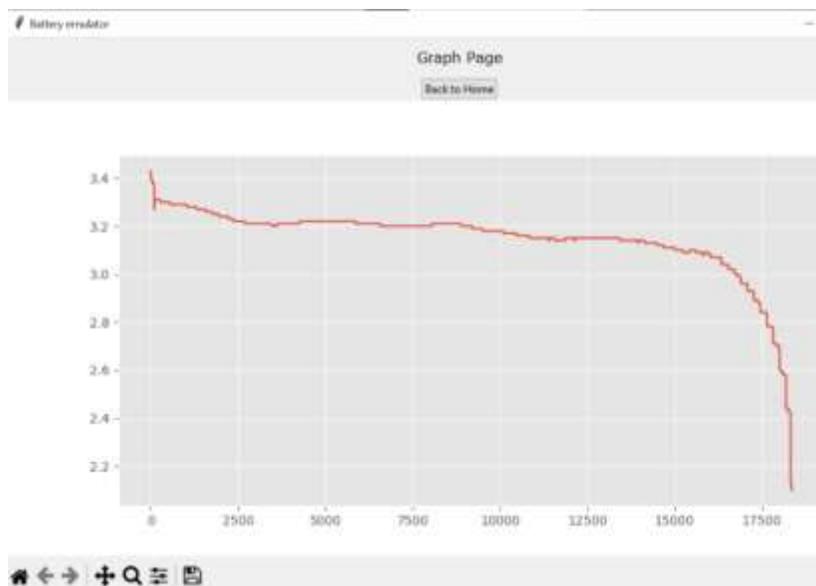
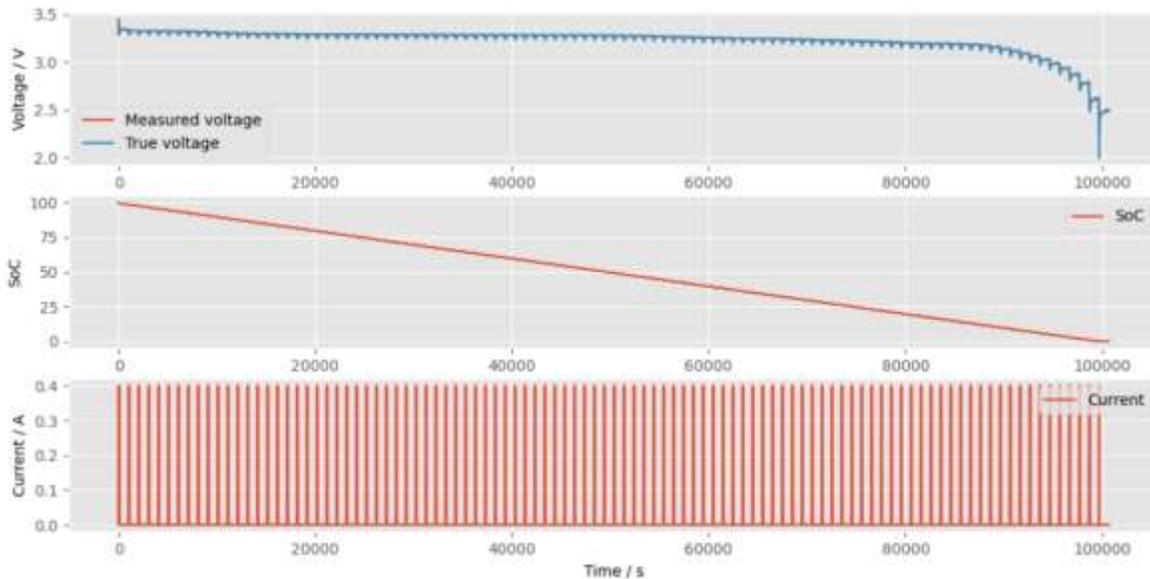


Figura 4.66

Simulación csv GUI



4.8 Programación de microcontrolador

Esta sección detallará el proceso de programación del microcontrolador STM32F446RE, que juega un papel crucial en el funcionamiento del emulador de batería propuesto. El microcontrolador es responsable de medir la corriente, controlar el voltaje de salida y el modo del emulador. Se comunicará con una computadora personal, que ejecutará el modelo de batería, utilizando el protocolo UART con interrupciones. Esta comunicación es crítica para la representación precisa del emulador del comportamiento de carga y descarga de la batería.

El ST STM32F446RE es un microcontrolador de línea avanzada de alto rendimiento de la familia STM32, construido sobre el procesador ARM Cortex-M4. Está diseñado para las capacidades en tiempo real y la operación de baja potencia y bajo voltaje, lo que lo convierte en una opción ideal para una amplia gama de aplicaciones. A los efectos de esta tesis, se eligió el STM32F446RE por su impresionante rendimiento, capacidades de E/S y funciones de conectividad avanzadas, que lo hacen adecuado para controlar el complejo funcionamiento del emulador de batería. Adicionalmente en futuros trabajos se puede probar si el alto rendimiento del procesador es lo suficiente para correr el modelo dentro del microcontrolador.

El STM32F446RE ofrece una variedad de interfaces de comunicación, que incluyen UART, SPI e I2C, lo que le permite interactuar con otros dispositivos y módulos de manera efectiva. En el sistema, la interfaz UART se utiliza para comunicarse con la computadora personal que ejecuta el modelo de batería y el I2C se utiliza para comunicarse con el ADC ADS1115.

El STM32F446RE también se destaca por sus ADC de 12 bits; sin embargo, para este proyecto, se usa un ADC externo de 16 bits, el ADS1115, para lograr una mayor precisión en las mediciones de voltaje y corriente.

El ADS1115 es un ADC de precisión, bajo consumo y 16 bits. Este ADC proporciona una interfaz para microcontroladores y microprocesadores con una interfaz compatible con I2C y ofrece alta precisión con su amplificador de ganancia programable, que es de hasta 16 veces. A pesar de que el STM32F446RE tiene su propio ADC, se eligió el ADS1115 para este proyecto debido a su resolución y precisión superiores. Es crucial en esta aplicación donde las lecturas precisas de voltaje y corriente son esenciales para una emulación confiable del comportamiento de la batería.

El ADC está conectado al microcontrolador a través de una interfaz I2C. Mide el voltaje y la corriente y comunica estas lecturas al STM32F446RE, que luego comunica estos datos a la GUI para que se calcule el voltaje de salida de acuerdo con el modelo. También monitorea la temperatura del disipador para mantener los elementos en condiciones óptimas.

En el lado STM32F446RE, el periférico I2C se configura utilizando las funciones de la biblioteca STM32Cube. La velocidad del bus I2C se establece en modo estándar (100 kHz). Se puede ver en la Figura 4.67 la configuración de I2C en STM32Cube.

Figura 4.67

Configuración I2C en STM32Cube



A continuación, el ADS1115 se configura para las medidas requeridas. Opera en modo Single-Shot, donde se realiza una conversión cada vez que se realiza una solicitud. El amplificador de ganancia programable (PGA) está configurado en 1, lo que permite un rango de voltaje de entrada de $\pm 4,096$ V. La velocidad de datos se establece en 860 muestras por segundo, lo que proporciona un equilibrio adecuado entre la velocidad y el ruido. El ADC está configurado para realizar mediciones diferenciales, midiendo la diferencia de voltaje entre sus pines de entrada, lo cual es necesario para la detección de corriente con una resistencia de derivación.

Medición de corriente y voltaje: el voltaje a través del emulador de batería se mide directamente usando un canal del ADC a través de un divisor de voltaje. Para medir la corriente, se coloca una resistencia de derivación de $10\text{m}\Omega$ en serie con la carga. La caída de voltaje a través de esta resistencia es proporcional a la corriente que fluye a través de ella y se mide usando dos canales del ADS1115 en modo diferencial. Luego, la corriente se calcula a partir de esta diferencia de voltaje utilizando la ley de Ohm.

Las lecturas de ADC son iniciadas por el STM32F446RE enviando un comando de escritura al ADS1115 con la configuración para el canal deseado. Después se espera a que la

conversión se complete y se recupera el resultado de la medición. Luego, los datos leídos se procesan para calcular el voltaje real o el valor actual. Para calcular el voltaje real a partir de la medición del ADC es necesario hacer una interpolación lineal, ya que el valor del ADC de la lectura del voltaje, cambia dependiendo de la corriente del circuito. La interpolación lineal se puede ver en el Anexo 11 en la función interpolate() y voltage().

El DAC se emplea en conjunto con un control proporcional para regular tanto la compuerta del MOSFET, utilizándolo como resistencia variable, como el convertidor reductor, el cual se utiliza como fuente de voltaje. Al incrementar el voltaje de la compuerta del MOSFET de 0V a su voltaje de saturación (10V), se aumenta la corriente que puede pasar a través de este, funcionando, así como una resistencia variable bajo la Ley de Ohm. El control del convertidor reductor, por otro lado, se puede realizar con el DAC de manera inversa: a medida que se incrementa el voltaje del DAC, el voltaje de salida del convertidor disminuye. La configuración del DAC se ilustra en la Figura 4.68.

Figura 4.68

Configuración DAC en STM32Cube



Para agilizar el proceso de prototipado se utilizó una fuente de poder RD6006 con control por comunicación serial. Esta fuente utiliza un protocolo MODBUS (Sillano, 2020) el cual se implementó con el periférico UART1 del microcontrolador. El código para controlar a la fuente se puede ver en el Anexo 17 y 18.

En la configuración de emulador de batería, la comunicación entre el microcontrolador STM32F446RE y la computadora personal que ejecuta el modelo de batería es fundamental. Para ello se ha elegido el protocolo UART (Universal Asynchronous Receiver/Transmitter) por su sencillez y fiabilidad. Utilizando interrupciones, se asegura la transmisión de datos sin desperdiciar procesamiento, esencial para mantener la velocidad necesaria para la medición y la actuación.

El periférico UART del STM32F446RE se configura mediante las funciones de la biblioteca STM32Cube. La velocidad en baudios se establece en 115200, lo que garantiza una alta velocidad de datos, y la longitud de palabra se configura en 8 bits. El modo UART está configurado para habilitar tanto el transmisor como el receptor. Es necesario activar la interrupción global de UART. La configuración de UART en STM32Cube se puede ver en la Figura 4.69.

Figura 4.69

Configuración UART en STM32Cube



Configuración de interrupciones: para que la comunicación UART no bloquee el procesamiento se utilizan interrupciones. En este caso, la interrupción UART está configurada para activarse cuando se reciben datos. Esto asegura que el microcontrolador procese rápidamente los datos entrantes desde la computadora personal. Y que no se desperdicie procesamiento verificando si hay mensajes disponibles por medio de polling.

El microcontrolador STM32F446RE transmite datos a la computadora personal a través del protocolo UART. Estos datos incluyen el modo de operación, el voltaje y la corriente medidos desde el ADC ADS1115, que se envían al modelo de batería que se ejecuta en la computadora personal. La transmisión se realiza mediante la función HAL_UART_Transmit_IT de la biblioteca STM32Cube.

El microcontrolador recibe comandos de la computadora personal a través de UART. Estos comandos tienen el formato de un protocolo definido para el emulador, el cual se puede ver a continuación. La recepción de datos se realiza mediante la función HAL_UART_Receive_IT de la biblioteca STM32Cube, que configura el periférico UART para recibir datos en modo de interrupción.

Protocolo UART

0x00: Set voltaje

GUI -> microcontrolador

Func	Byte 1	Byte 2
0x00	Voltaje entero	Voltaje decimal

0x01: Encender apagar

GUI -> microcontrolador

Func	Byte 1	Byte 2
0x01	0 = apagar 1 = encender	NC

0x02: Solicitar estado

GUI -> microcontrolador

Func	Byte 1	Byte 2
0x02	NC	NC

0x02: Regreso de solicitud de estado

Microcontrolador -> GUI

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0 = descarga 1 = carga	Voltaje entero	Voltaje decimal	mA primeros dos números	mA últimos dos números

El primer paso en la lógica de control es inicializar todos los dispositivos, cuando una interrupción es recibida se procesan los comandos recibidos desde la computadora personal. Se activará diferente código dependiendo de la información recibida como se puede ver en la Figura 4.70.

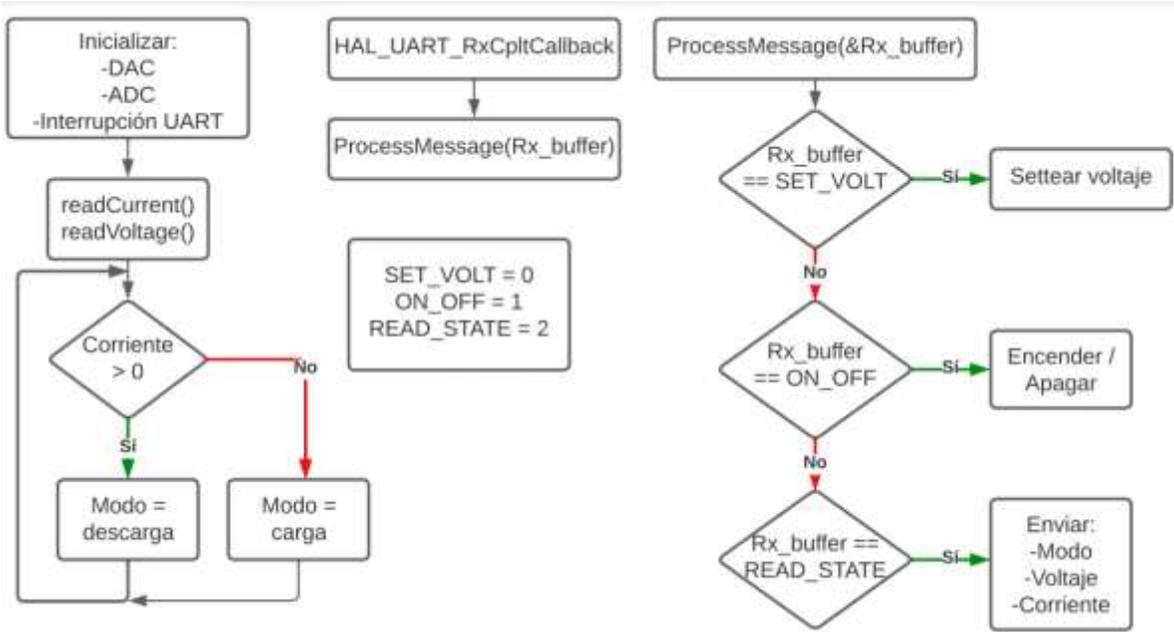
Cuando la corriente medida es positiva el emulador estará en modo de descarga, enviará este modo a la GUI y se calculará el voltaje de salida a partir del modelo Thévenin 3RC. Con este voltaje recibido, el microcontrolador ajusta el convertidor reductor CC-CC para emular el comportamiento de descarga.

Cuando la corriente medida es negativa el emulador estará en modo de carga, enviará este modo a la GUI y se calculará el voltaje de salida a partir del modelo Thévenin 2RC. Con este voltaje recibido, el microcontrolador ajusta el convertidor reductor CC-CC para emular el voltaje de carga y el MOSFET para ajustar la corriente de carga.

Al implementar la lógica de control, el microcontrolador STM32F446RE controla efectivamente el voltaje de salida y el modo del emulador de batería, lo que garantiza una representación precisa del comportamiento de carga y descarga de la batería. El código completo de la implementación del software del microcontrolador se puede ver en los Anexos 11, 12, 13, 14, 15, 16, 17, 18 y 19.

Figura 4.70

Lógica de programación



5. Mediciones y resultados

Esta sección proporciona una comprensión integral del diseño experimental, los resultados y la discusión posterior, que son fundamentales para validar la eficacia del emulador de batería. Las conclusiones extraídas de estos resultados aclararán aún más el impacto potencial del emulador en el campo de los sistemas de gestión de baterías y dispositivos que funcionan con baterías.

5.1 Diseño experimental

El diseño experimental del emulador de batería se formuló para validar su eficacia y precisión a la hora de emular el comportamiento de carga y descarga de una batería real. El enfoque principal fue evaluar la respuesta del emulador a las diferentes condiciones de operación y qué tan bien se adhiere al comportamiento de una batería real. En esta sección, se expondrá la configuración, las condiciones de prueba y la metodología de recopilación de datos utilizada en el diseño experimental.

El prototipo de emulador de batería se configuró con el circuito previamente descrito, usando la fuente de poder RD6006 y una resistencia variable. El microcontrolador se conectó a una computadora personal que ejecuta el modelo de batería. El ADC se configuró para medir el voltaje y la corriente del emulador, y el microcontrolador se programó para controlar el voltaje y el modo de salida en función de estas mediciones y los comandos recibidos de la computadora. Se utilizó un probador de batería EBC-A20 para probar el circuito tanto en modo de descarga y carga.

Características clave del probador de batería EBC-A20:

- **Amplio rango de prueba:** el EBC-A20 es capaz de probar baterías con un rango de voltaje de 0,1 V a 20 V y un rango de corriente de 0,001A a 20A. Esto lo hace adecuado para probar una amplia variedad de tamaños y tipos de baterías.
- **Medición de alta precisión:** el dispositivo ofrece mediciones de voltaje y corriente de alta precisión, lo que garantiza resultados de prueba precisos y confiables. Emplea un ADC (convertidor analógico a digital) de 16 bits para una recopilación de datos precisa.

- Pruebas de carga y descarga: el EBC-A20 admite pruebas de carga y descarga, lo que permite una evaluación completa del rendimiento del emulador. El probador puede controlar la corriente de carga/descarga y monitorear la respuesta del emulador.
- Registro y análisis de datos: el probador está equipado con funciones de registro de datos en archivos csv, lo que permite registrar y analizar el rendimiento del emulador a lo largo del tiempo. Puede rastrear parámetros como voltaje, corriente, capacidad y energía. Los datos se pueden exportar para su posterior análisis.
- Conectividad con PC: El EBC-A20 se puede conectar a una computadora personal a través de una interfaz USB. Esto permite monitorear y controlar en tiempo real el proceso de prueba utilizando el software proporcionado por EBC. El software proporciona una interfaz fácil de usar para configurar pruebas, monitorear datos en tiempo real y analizar datos registrados.
- Funciones de seguridad: el probador incorpora varias funciones de seguridad, incluida la protección contra sobretensión, sobre corriente y sobre temperatura. Estas características ayudan a garantizar condiciones de prueba seguras y evitar daños al circuito o incluso a la computadora.

La prueba implicó operar el emulador en una variedad de condiciones para imitar los diversos escenarios que podría enfrentar una batería real. El emulador se sometió a diversos grados de carga, lo que representa diferentes niveles de demanda de energía, y de igual modo se probó en diferentes estados de carga para emular los ciclos de carga y descarga de una batería.

Para validar el rendimiento del emulador, se compararon sus salidas con las pruebas hechas en la sección de proceso de extracción de parámetros. El comportamiento de la batería real sirvió como punto de referencia para evaluar la precisión y eficacia del emulador. Se investigó cualquier anomalía o valor atípico para identificar posibles problemas con el emulador o la configuración experimental.

El diseño experimental se estructuró para ofrecer una evaluación sólida del emulador de batería. La siguiente sección, "Resultados", presentará los datos recopilados durante estos experimentos y proporcionará un análisis inicial del rendimiento del emulador.

5.2 Resultados

Se validó la eficacia y precisión del emulador de batería a partir de los resultados obtenidos de las pruebas experimentales. En esta sección, se presentarán los resultados derivados de los distintos escenarios de prueba y los compararemos con el comportamiento esperado de una batería real en las mismas condiciones.

Para validar el rendimiento del emulador, sus resultados se compararon con los de una batería real los cuales se exponen en la sección de proceso de extracción de parámetros. Estos resultados se obtuvieron con el probador de batería EBC-A20 en las mismas condiciones. Las salidas de voltaje, corriente y potencia del emulador reflejaron fielmente las de la batería real, con un error de descarga promedio de 11.3997mV (0.354%), un error máximo de descarga de 431mV, un error promedio de carga de 4.8301mV (0.146%) y un error máximo de carga de 63mV. Los resultados se pueden ver en la Figura 5.1 y 5.2.

Figura 5.1

Prueba de OCV de emulador, comportamiento de descarga

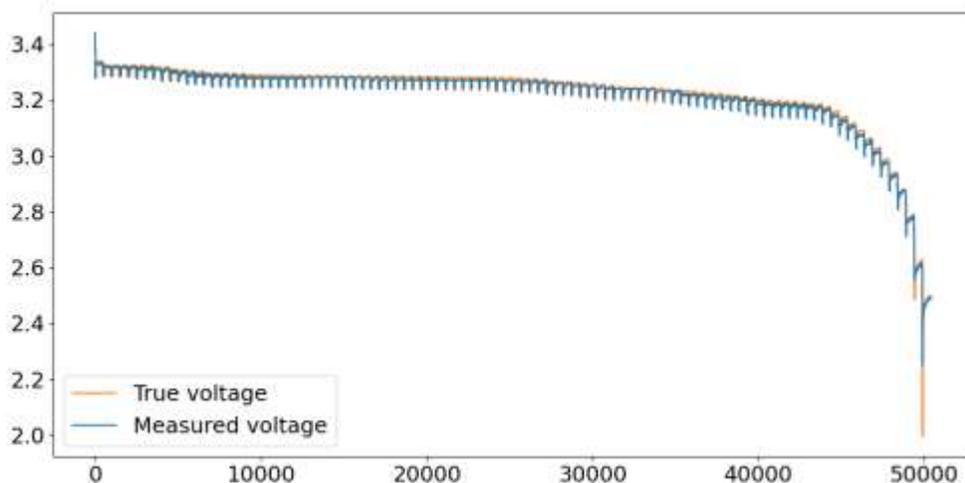
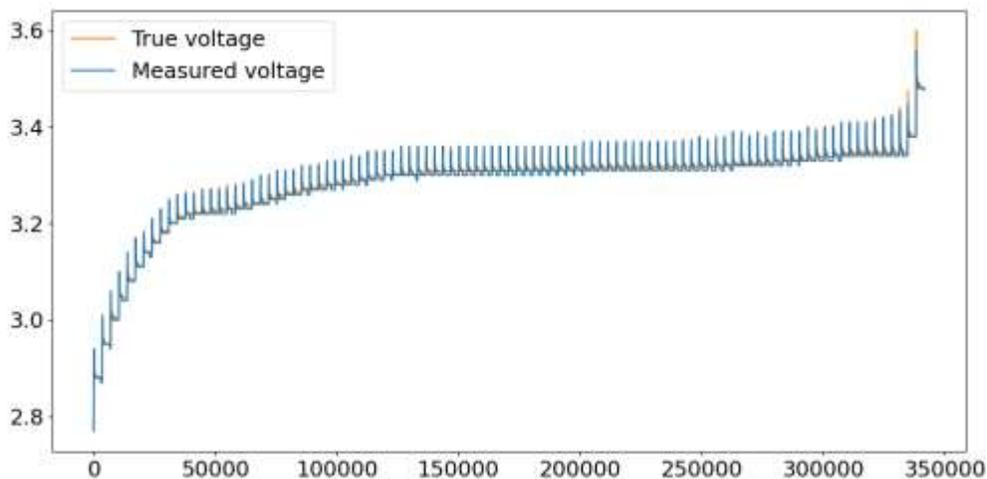


Figura 5.2

Prueba de OCV de emulador, comportamiento de carga



Estos resultados proporcionan evidencia convincente de la precisión y eficacia del emulador de batería para emular el comportamiento de carga y descarga de una batería real. La siguiente sección, "Impacto", explorará las implicaciones más amplias de estos hallazgos y las posibles aplicaciones del emulador.

5.3 Impacto

El desarrollo de un emulador de batería preciso y confiable tiene implicaciones significativas no solo para el campo de la prueba y validación de baterías, sino también para una gama más amplia de aplicaciones. Esta sección de impacto analizará los posibles efectos y contribuciones de este trabajo en el contexto de los sistemas de gestión de baterías, los dispositivos que funcionan con baterías y más.

5.3.1 Sistemas de gestión de batería

El emulador de batería se puede utilizar para probar y validar los BMS. Los BMS son cruciales para mantener el funcionamiento seguro y eficiente de los sistemas alimentados por baterías. Al proporcionar un entorno controlado que imita de cerca el comportamiento de una batería real, el emulador permite realizar pruebas exhaustivas de un BMS sin los riesgos asociados con las pruebas en una batería real.

5.3.2 Dispositivos que funcionan con baterías

El emulador también tiene aplicaciones potenciales en el desarrollo y prueba de dispositivos que funcionan con baterías. Los fabricantes pueden usar el emulador para probar sus dispositivos en una variedad de condiciones de batería, lo que ayuda a mejorar la confiabilidad y el rendimiento de sus productos.

5.3.3 Educación e investigación

El emulador sirve como una excelente herramienta educativa para estudiantes e investigadores que estudian tecnologías de baterías. Proporciona una plataforma práctica para comprender el complejo comportamiento de las baterías Li-ion. Además, al tener un código basado en patrones de diseño es fácil aumentar nuevos modelos de batería para utilizarlo como estructura base y fomentar una mayor investigación para mejorar los modelos de batería y las técnicas de emulación.

5.3.4 Impacto ambiental

Al reducir la necesidad de baterías reales en los escenarios de prueba, el emulador puede contribuir a la conservación del medio ambiente. La producción y el desecho de baterías pueden tener impactos ambientales significativos, y cualquier reducción en la cantidad de baterías requeridas puede ayudar a mitigar estos efectos.

5.4 Direcciones futuras

Este trabajo abre varias vías para futuras investigaciones. El emulador se puede refinar aún más para representar comportamientos de batería más complejos o diferentes tipos de baterías agregando alguna interfaz gráfica para agregar nuevos modelos de incluso otras químicas de batería. Por otro lado, se puede implementar un módulo para que el mismo programa extraiga los parámetros directo de una batería real para crear el modelo de batería.

Otro trabajo futuro importante sería habilitar la sección de modelado de temperatura y SOH dentro del código, ya que se agregó la parte en la GUI, pero el desarrollo y modelado de estos aspectos salen del alcance de este proyecto de tesis.

En resumen, el emulador de batería tiene el potencial de realizar contribuciones sustanciales en varios campos. Ofrece una solución rentable, segura y confiable para emular el

comportamiento de una batería, con aplicaciones prometedoras en pruebas, investigación, educación y conservación ambiental.

6. Conclusiones

El trabajo presentado en este proyecto de tesis ha explorado el desarrollo, implementación y validación de un emulador de batería. Este emulador fue diseñado para replicar con precisión y fiabilidad el comportamiento de carga y descarga de una batería real. Los hallazgos de esta investigación tienen implicaciones de gran alcance para el campo de la tecnología de baterías y más allá. Esta sección de conclusiones tiene como objetivo proporcionar un resumen completo de la investigación y su importancia, identificar áreas para el trabajo futuro y reflexionar sobre las implicaciones más amplias de esta investigación.

El objetivo principal de este proyecto de tesis era crear un emulador de batería preciso, eficiente y de bajo costo basado en un modelo de Thévenin y utilizando una arquitectura distribuida. El emulador desarrollado, como se demuestra en la investigación, ha cumplido con éxito estos criterios. Al emplear un modelo 3RC Thévenin para emular el comportamiento de descarga y un modelo 2RC Thévenin para el comportamiento de carga, el emulador exhibió una alta precisión al replicar el comportamiento de una batería de iones de litio real. Los parámetros de Thévenin se estimaron utilizando el bloque de batería de circuito equivalente de Matlab, empleando una caracterización de potencia de pulso híbrido para modelar una batería real. La gran precisión de los modelos al replicar el comportamiento de una batería confirma la validez del modelo de Thévenin propuesto por Orcioni et al. (2017).

La implementación de una arquitectura distribuida entre una computadora personal y un circuito con un convertidor reductor, y una carga electrónica variable, controlada por un microcontrolador, fue exitosa. Esta configuración basada en Daniil & Drury (2016) y Restrepo et al. (2012) fue efectiva al emular los comportamientos de voltaje y corriente durante las condiciones de carga y descarga de una batería real.

Se creó un sistema embebido para emulación práctica en un banco de pruebas de baterías de Li-ion. Esto se logró mediante el diseño de un sistema embebido que permite integrar el

emulador en un banco de pruebas de baterías de Li-ion. Los resultados del banco de pruebas demostraron que el emulador logró representar el comportamiento de una batería real, con un error promedio y un error máximo bajo.

Uno de los principales objetivos de este proyecto era reducir los riesgos de prueba y los costos de producción de los circuitos de batería. El emulador, al reemplazar las baterías reales en los escenarios de prueba, mitiga significativamente los riesgos de seguridad asociados con la prueba de la batería. Además, el uso de un emulador también reduce los costos asociados con la adquisición, el almacenamiento y la eliminación de baterías reales, especialmente en entornos de producción a gran escala.

Se llevó a cabo un riguroso proceso de prueba para validar el rendimiento del emulador. Se sometió a varias condiciones de funcionamiento y se comparó su rendimiento con una batería real probada con el probador de batería EBC-A20. El emulador se desempeñó de manera impresionante, obteniendo señales muy apegadas a las de la batería real. De la misma forma, el emulador demostró un rendimiento constante en múltiples pruebas repetidas, lo que es un testimonio de su confiabilidad y replicabilidad.

El desarrollo exitoso del emulador de batería tiene implicaciones significativas para varios campos. En el contexto de los BMS y los dispositivos que funcionan con baterías, el emulador proporciona una plataforma rentable, segura y confiable para realizar pruebas y validaciones. Al emular el comportamiento de una batería, reduce la necesidad de baterías reales en escenarios de prueba, lo cual reduce el desperdicio en las etapas de desarrollo, contribuyendo así a la conservación del medio ambiente. De la misma manera, el emulador puede servir como una valiosa herramienta educativa para estudiantes e investigadores que estudian tecnologías de baterías. Ofrece una plataforma práctica para comprender el comportamiento complejo de las baterías Li-ion, lo que puede fomentar una mayor investigación e innovación en este campo.

Si bien el emulador ha demostrado ser exitoso en su forma actual, existen numerosas oportunidades para un mayor refinamiento y expansión. El emulador podría incorporar comportamientos de batería más complejos, como efectos de envejecimiento o variaciones

de temperatura. De igual forma podría adaptarse para imitar diferentes tipos de baterías, ampliando su aplicabilidad.

A pesar de la eficacia general del emulador, existen áreas de mejora. El emulador está diseñado para emular un tipo específico de batería Li-ion y su rendimiento puede variar al emular otros tipos de baterías. Del mismo modo, el emulador no puede replicar algunos atributos físicos inherentes de una batería real, como el peso, e SOH o la generación de calor durante el funcionamiento. Las iteraciones futuras del emulador podrían abordar estas limitaciones, convirtiéndolo en una herramienta aún más versátil y precisa.

En conclusión, la investigación presentada en esta tesis validó con éxito la hipótesis y cumplió con los objetivos planteados. El trabajo presentado en esta tesis marca un avance importante en el campo de las pruebas y validación de baterías. El emulador de batería es un testimonio del potencial del diseño electrónico y el modelado de baterías para crear herramientas precisas y confiables para pruebas y validación. La precisión, confiabilidad y potencial de desarrollo del emulador lo convierten en una valiosa contribución a los esfuerzos continuos para optimizar el rendimiento, la eficiencia y la seguridad de los sistemas que funcionan con baterías. Con numerosas oportunidades para la investigación y el desarrollo futuros, este trabajo promete continuar contribuyendo al campo de la tecnología de baterías y más allá en los próximos años.

7. Referencias

- Ahmed, R., Gazzarri, J., Onori, S., Habibi, S., Jackey, R., Tjong, J., Lesage, J., Sae, S., Journal, I., July, N., Gazzarri, J., & Lesage, J. (2015). *Model-Based Parameter Identification of Healthy and Aged Li-ion Batteries for Electric Vehicle Applications*. 4(2). <https://doi.org/10.4271/2015-01-0252>
- Barreras, J. V., Fleischer, C., Christensen, A. E., Swierczynski, M., Schaltz, E., Andreasen, S. J., & Sauer, D. U. (2016). An Advanced HIL Simulation Battery Model for Battery Management System Testing. *IEEE Transactions on Industry Applications*, 52(6), 5086–5099. <https://doi.org/10.1109/TIA.2016.2585539>
- Barreras, J. V., Swierczynski, M., Schaltz, E., Andreasen, S. J., Fleischer, C., Sauer, D. U.,

- & Christensen, A. E. (2015). Functional analysis of battery management systems using multi-cell HIL simulator. *2015 10th International Conference on Ecological Vehicles and Renewable Energies, EVER 2015*. <https://doi.org/10.1109/EVER.2015.7112984>
- Boles, J. D., Ma, Y., Cao, W., Tolbert, L. M., & Wang, F. (2017). Battery energy storage emulation in a converter-based power system emulator. *Conference Proceedings - IEEE Applied Power Electronics Conference and Exposition - APEC, 2355–2362*. <https://doi.org/10.1109/APEC.2017.7931029>
- Buccolini, L., Orcioni, S., Longhi, S., & Conti, M. (2018). Cell Battery Emulator for Hardware-in-the-Loop BMS Test. *Proceedings - 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe, IEEEIC/I and CPS Europe 2018, 1–5*. <https://doi.org/10.1109/IEEEIC.2018.8493731>
- Cabello, J., Bru, E., Roboam, X., Lacressonnière, F., & Junco, S. (2015). Battery dynamic model improvement with parameters estimation and experimental validation. *8th International Conference on Integrated Modeling and Analysis in Applied Control and Automation*.
- Cabello, J., Roboam, X., Junco, S., Bru, E., & Lacressonniere, F. (2017). Scaling Electrochemical Battery Models for Time-Accelerated and Size-Scaled Experiments on Test-Benches. *IEEE Transactions on Power Systems, 32(6), 4233–4240*. <https://doi.org/10.1109/TPWRS.2017.2683398>
- Campagna, N., Castiglia, V., Miceli, R., Mastromauro, R. A., Spataro, C., Trapanese, M., & Viola, F. (2020). Battery models for battery powered applications: A comparative study. *Energies, 13(16), 4085*.
- Chen, M., & Rincón-Mora, G. A. (2006). Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance. *IEEE TRANSACTIONS ON ENERGY CONVERSION, 5(3), 178–180*.
- Cheng, M., & Chen, B. (2017). The Model Integration and Hardware-in-the-Loop (HiL) Simulation Design for the Analysis of a Power-Split Hybrid Electric Vehicle with

- Electrochemical Battery Model. *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 10(2), 275–282. <https://doi.org/10.4271/2017-01-0001>
- Dai, H., Zhang, X., Wei, X., Sun, Z., Wang, J., & Hu, F. (2013). Cell-BMS validation with a hardware-in-the-loop simulation of lithium-ion battery cells for electric vehicles. *International Journal of Electrical Power and Energy Systems*, 52(1), 174–184. <https://doi.org/10.1016/j.ijepes.2013.03.037>
- Daniil, N., & Drury, D. (2016). Investigation and validation of methods to implement a two-quadrant battery emulator for power Hardware-in-the-Loop Simulation. *IECON Proceedings (Industrial Electronics Conference)*, 2070–2075. <https://doi.org/10.1109/IECON.2016.7794013>
- Di Rienzo, R., Roncella, R., Morello, R., Baronti, F., & Saletti, R. (2018). Low-cost modular battery emulator for battery management system testing. *Proceedings - 2018 IEEE International Conference on Industrial Electronics for Sustainable Energy Systems, IESES 2018, 2018-Janua*, 44–49. <https://doi.org/10.1109/IESES.2018.8349848>
- Di Rienzo, R., Verani, A., Baronti, F., Roncella, R., & Saletti, R. (2022). *Modular Battery Emulator for Development and Functional Testing of Battery Management Systems : The Cell Emulator*.
- Dung, L. R., Yuan, H. F., Yen, J. H., She, C. H., & Lee, M. H. (2016). A lithium-ion battery simulator based on a diffusion and switching overpotential hybrid model for dynamic discharging behavior and runtime predictions. *Energies*, 9(1), 1–21. <https://doi.org/10.3390/en9010051>
- Fotouhi, A., Auger, D. J., Propp, K., Longo, S., & Wild, M. (2016). A review on electric vehicle battery modelling: From Lithium-ion toward Lithium-Sulphur. *Renewable and Sustainable Energy Reviews*, 56, 1008–1021. <https://doi.org/10.1016/j.rser.2015.12.009>
- Guo, L., Junqiu, L., & Zijan, F. (2019). Lithium-Ion Battery SOC Estimation and Hardware-in-the-Loop simulation Based on EKF. *10th International Conference on Applied Energy (ICAE2018)*, 22-25 August 2018, Hong Kong, China.

<https://doi.org/https://doi.org/10.1016/j.egypro.2019.02.009>

- Hidalgo León, R., Urquizo, J., Litardo, J., Munoz-Jadán, Y., Singh, P., & Wu, J. (2020). Simulation of battery discharge emulator using power electronics device with cascaded P-I control. *Proceedings of the IEEE International Conference on Industrial Technology*, 2020-Febru(i), 959–964. <https://doi.org/10.1109/ICIT45562.2020.9067170>
- König, O., Gregorčič, G., & Jakubek, S. (2013). Model predictive control of a DC-DC converter for battery emulation. *Control Engineering Practice*, 21(4), 428–440. <https://doi.org/10.1016/j.conengprac.2012.12.009>
- König, O., Hametner, C., Prochart, G., & Jakubek, S. (2014). Battery emulation for power-HIL using local model networks and robust impedance control. *IEEE Transactions on Industrial Electronics*, 61(2), 943–955. <https://doi.org/10.1109/TIE.2013.2253070>
- Kramer, M. (2018). Best practices in systems development lifecycle: An analyses based on the waterfall model. *Review of Business & Finance Studies*, 9(1), 77-84.
- Lacressonnière, F., Varais, A., Roboam, X., Bru, E., & Mullins, T. (2021). Scale electro-thermal model of a lithium-ion battery for time-accelerated experiments in a hardware in the loop process. *Journal of Energy Storage*, 39(February). <https://doi.org/10.1016/j.est.2021.102576>
- Luna, M., Piazza, D., C., M., La Tona, G., Accetta, A., & Pucci, M. (2021). Exploiting dynamic modeling, parameter identification, and power electronics to implement a non-dissipative Li-ion battery hardware emulator. *Mathematics and Computers in Simulation*, 183(xxxx), 48–65. <https://doi.org/10.1016/j.matcom.2020.04.028>
- Mathew, M., Kong, Q. H., McGrory, J., & Fowler, M. (2017). Simulation of lithium ion battery replacement in a battery pack for application in electric vehicles. *Journal of Power Sources*, 349, 94–104. <https://doi.org/10.1016/j.jpowsour.2017.03.010>
- Mesbahi, T., Rizoug, N., Bartholomeus, P., & Le Moigne, P. (2013). Li-ion battery emulator for electric vehicle applications. *2013 9th IEEE Vehicle Power and Propulsion Conference, IEEE VPPC 2013*, 191–198. <https://doi.org/10.1109/VPPC.2013.6671688>

- Mesbahi, Tedjani, Rizoug, N., Bartholom us, P., Sadoun, R., Khenfri, F., & Le Moigne, P. (2018). Dynamic model of li-ion batteries incorporating electrothermal and ageing aspects for electric vehicle applications. *IEEE Transactions on Industrial Electronics*, 65(2), 1298–1305. <https://doi.org/10.1109/TIE.2017.2714118>
- Michalczuk, M., Ufnalski, B., Grzesiak, L. M., & Rumniak, P. (2014). Power converter-based electrochemical battery emulator. *Przeglad Elektrotechniczny*, 90(7), 18–22. <https://doi.org/10.12915/pe.2014.07.03>
- Mousavi G., S. M., & Nikdel, M. (2014). Various battery models for various simulation studies and applications. *Renewable and Sustainable Energy Reviews*, 32, 477–485. <https://doi.org/10.1016/j.rser.2014.01.048>
- Nutricato, R., Conti, M., & Orcioni, S. (2022). *Modelli elettrici e stima dei parametri di Batterie al litio*. Universita' politecnica delle marche.
- Orcioni, S., Buccolini, L., Ricci, A., & Conti, M. (2017). Lithium-ion battery electrothermal model, parameter estimation, and simulation environment. *Energies*, 10(3). <https://doi.org/10.3390/en10030375>
- Pebriyanti, G. (2013). A lithium-ion battery modeling for a HIL-battery simulator. *Proceeding - 2013 International Conference on Computer, Control, Informatics and Its Applications: "Recent Challenges in Computer, Control and Informatics"*, IC3INA 2013, 185–190. <https://doi.org/10.1109/IC3INA.2013.6819171>
- Restrepo, A. F., Ramos-Paja, C. A., & Franco, E. (2012). Power control of a bidirectional DC bus for fuel cells applications. *Revista EIA*, 18, 159–170. <http://revista.eia.edu.co/articulos18/Revista EIA N18 art 12.pdf>
- Sarikurt, T., Ceylan, M., & Balikci, A. (2014). A hybrid battery model and state of health estimation method for lithium-ion batteries. *ENERGYCON 2014 - IEEE International Energy Conference*, 1349–1356. <https://doi.org/10.1109/ENERGYCON.2014.6850598>
- Sillano, M. (2020, March 3). *RD6006 protocol USB-serial: reverse engineering*. GitHub. <https://github.com/msillano/RD6006-Super-power-supply/tree/master>

- Tomasov, M., Kajanova, M., Bracinik, P., & Motyka, D. (2019). Overview of battery models for sustainable power and transport applications. *Transportation Research Procedia*, 40, 548–555. <https://doi.org/10.1016/j.trpro.2019.07.079>
- Topan, P. A., Nisivo, R. M., Fathoni, G., Cahyadi, A. I., & Wahyunggoro, O. (2016). State of Charge (SOC) and State of Health (SOH) Estimation on Lithium Polymer Battery via Kalman Filter. *2nd International Conference on Science and Technology-Computer (ICST), Yogyakarta, Indonesia State*, 2, 6–9.
- Van Sterkenburg, S., Fleuren, T., Veenhuizen, B., & Groenewegen, J. (2014). Design and test of a battery pack simulator. *2013 World Electric Vehicle Symposium and Exhibition, EVS 2014*, 1–12. <https://doi.org/10.1109/EVS.2013.6914804>
- Wang, J., Pan, C., Xu, H., & Xu, X. (2018). Thermo-Electric Model of the Power Battery and Module Based on AMESim. *2017 5th International Conference on Mechanical, Automotive and Materials Engineering, CMAME 2017*, 223–228. <https://doi.org/10.1109/CMAME.2017.8540167>
- Wu, T., Wang, J., Moo, C.-S., & Kawamura, A. (2016). *State-of-Charge and State-of-Health Estimating Method for Lithium-Ion Batteries*. 1–6.
- Zhang, C., Li, K., & Deng, J. (2016). Real-time estimation of battery internal temperature based on a simplified thermoelectric model. *Journal of Power Sources*, 302, 146–154. <https://doi.org/10.1016/j.jpowsour.2015.10.052>
- Zhang, C., Li, K., Deng, J., & Song, S. (2017). Improved Realtime State-of-Charge Estimation of LiFePO₄ Battery Based on a Novel Thermoelectric Model. *IEEE Transactions on Industrial Electronics*, 64(1), 654–663. <https://doi.org/10.1109/TIE.2016.2610398>