



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO**

**TÍTULO**

Estudio comparativo de métodos para estimación del estado de carga en  
baterías de iones de litio

**AUTOR**

**Paccha Herrera, Edwin Bladimir**

**TRABAJO DE TITULACIÓN**

Previo a la obtención del grado académico en  
**MAGÍSTER EN ELECTRÓNICA Y AUTOMATIZACIÓN**

**TUTOR**

**Recalde Lino, Ángel Andrés**

**Santa Elena, Ecuador**

**Año 2026**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO  
TRIBUNAL DE SUSTENTACIÓN**

---

**Ing. Alicia Andrade Vera, Mgtr.  
COORDINADORA DEL  
PROGRAMA**

---

**Ing. Ángel Recalde Lino, Ph.D.  
TUTOR**

---

**Ing. Luis Chuquimarca Jiménez, Ph.D.  
DOCENTE  
ESPECIALISTA**

---

**Ing. Junior Figueroa Olmedo, Mgtr.  
DOCENTE  
ESPECIALISTA**

---

**Abg. María Rivera González, Mgtr.  
SECRETARIA GENERAL  
UPSE**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO**

**CERTIFICACIÓN**

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Edwin Bladimir Paccha Herrera, como requerimiento para la obtención del título de Magíster en Electrónica y Automatización.

**TUTOR**

---

**Ing. Ángel Recalde Lino, Ph.D.**

**Santa Elena, 18 de febrero de 2026**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO**

**DECLARACIÓN DE RESPONSABILIDAD**

Yo, Edwin Bladimir Paccha Herrera

**DECLARO QUE:**

El trabajo de Titulación, Estudio Comparativo de Métodos para estimación del Estado de Carga en Baterías de Iones de Litio, previo a la obtención del título en Magíster en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 18 de febrero de 2026

**EL AUTOR**

---

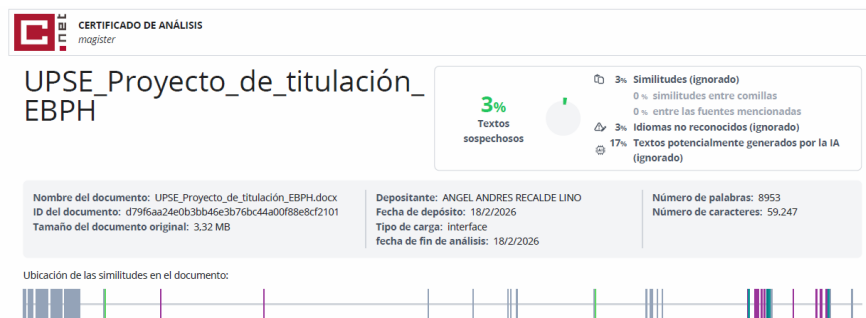
**Edwin Bladimir Paccha Herrera**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO**

**CERTIFICACIÓN DE ANTIPLAGIO**

Certifico que después de revisar el documento final del trabajo de titulación denominado Estudio Comparativo de Métodos para Estimación del Estado de Carga en Baterías de Iones de Litio, presentado por el estudiante, Edwin Bladimir Paccha Herrera fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



**TUTOR**

---

**Ing. Ángel Recalde Lino, Ph.D.**



**UNIVERSIDAD ESTATAL PENÍNSULA  
DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
INSTITUTO DE POSTGRADO**

**AUTORIZACIÓN**

Yo, Edwin Bladimir Paccha Herrera

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de este proyecto de titulación con componentes de investigación aplicada y/o de desarrollo con fines de difusión pública, además apruebo la reproducción de este proyecto de titulación dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 18 de febrero de 2026

**EL AUTOR**

---

**Edwin Bladimir Paccha Herrera**

## **AGRADECIMIENTO**

Un agradecimiento al personal académico y administrativo de la Universidad Estatal Península de Santa Elena por la impartición de sus conocimientos y la gestión durante este proceso de formación.

Gracias al Dr. Ángel Recalde, por su orientación y sugerencias que aportaron sustancialmente al desarrollo de este proyecto.

Un agradecimiento a mi amigo Dr. Franciso Jaramillo, por el continuo compartir de su conocimiento y amistad.

*Edwin Bladimir, Paccha Herrera*

## **DEDICATORIA**

A la memoria de mis padres, por su amor, continuo sacrificio y enseñanzas.

A mis hermanos, por su grato compartir.

A mi esposa Jenny, por su amor y apoyo incondicional.

A mi pequeño(a) que viene en camino. Desde siempre te hemos esperado.

*Edwin Bladimir, Paccha Herrera*

# ÍNDICE GENERAL

<b>TÍTULO.....</b>	<b>I</b>
<b>TRIBUNAL DE SUSTENTACIÓN .....</b>	<b>II</b>
<b>CERTIFICACIÓN.....</b>	<b>III</b>
<b>DECLARACIÓN DE RESPONSABILIDAD .....</b>	<b>IV</b>
<b>CERTIFICACIÓN DE ANTIPLAGIO .....</b>	<b>V</b>
<b>AUTORIZACIÓN .....</b>	<b>VI</b>
<b>AGRADECIMIENTO .....</b>	<b>VII</b>
<b>DEDICATORIA.....</b>	<b>VIII</b>
<b>ÍNDICE GENERAL .....</b>	<b>IX</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>XI</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>XII</b>
<b>RESUMEN .....</b>	<b>XIII</b>
<b>ABSTRACT.....</b>	<b>XIV</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL.....</b>	<b>5</b>
1.1. Revisión de literatura .....	5
1.2. Desarrollo teórico y conceptual .....	7
1.2.1 Métodos de estimación de SOC.....	8
1.2.2 Espectroscopía de impedancia electroquímica (EIS).....	9
1.2.3 Bosques aleatorios (RF).....	10
1.2.4 K-vecinos más cercanos (KNN) .....	11
1.2.5 Filtro de partículas (PF) .....	15

<b>CAPÍTULO 2. METODOLOGÍA.....</b>	<b>18</b>
2.1. Contexto de la investigación .....	23
2.2. Diseño y alcance de la investigación .....	23
2.3. Tipo y métodos de investigación.....	23
2.4. Población y muestra .....	24
2.5. Técnicas e instrumentos de recolección de datos.....	24
2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información .....	25
<b>CAPÍTULO 3. RESULTADOS Y DISCUSIÓN .....</b>	<b>26</b>
3.1 Métodos de estimación de estado de carga .....	26
3.1 Resultados de modelos implementados .....	27
<b>CONCLUSIONES .....</b>	<b>33</b>
<b>RECOMENDACIONES .....</b>	<b>34</b>
<b>REFERENCIAS.....</b>	<b>35</b>
<b>ANEXOS.....</b>	<b>40</b>

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Parámetros del circuito equivalente ajustados a las curvas de Nyquist para SOC 0.6.....	21
<b>Tabla 2.</b> Comparativa de métricas entre los distintos algoritmos .....	31

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Métodos de estimación de estado de carga de baterías .....	8
<b>Figura 2.</b> Diagrama de Nyquist y su circuito eléctrico equivalente para un test de EIS en una batería.....	9
<b>Figura 3.</b> Diagrama del algoritmo de bosques aleatorios (RF).....	11
<b>Figura 4.</b> Esquema del algoritmo K-vecinos más cercanos (KNN).....	12
<b>Figura 5.</b> Clasificación de filtros de Kalman .....	12
<b>Figura 6.</b> Filtro de Kalman estándar para estimar SOC de una batería .....	13
<b>Figura 7.</b> Estimación de la función de densidad probabilística (PDF) mediante filtro de partículas.....	16
<b>Figura 8.</b> Descripción general para implementación de métodos de estimación de SOC .....	18
<b>Figura 9.</b> Perfil de corriente de ciclo HWFET.....	19
<b>Figura 10.</b> Perfil de voltaje de ciclo HWFET .....	19
<b>Figura 11.</b> Diagrama de Nyquist con base en experimentos de EIS.....	20
<b>Figura 12.</b> Dispositivo tipo carga programable .....	24
<b>Figura 13.</b> Estimación de SOC mediante filtro de Kalman linealizado (LKF) .....	27
<b>Figura 14.</b> Estimación de SOC mediante Filtro de Partículas (PF).....	28
<b>Figura 15.</b> Estimación de SOC mediante bosques aleatorios (RF).....	28
<b>Figura 16.</b> Estimación de SOC mediante K-vecinos más cercanos (KNN) .....	29
<b>Figura 17.</b> Voltaje estimado mediante filtro de Kalman.....	29
<b>Figura 18.</b> Voltaje estimado mediante Filtro de Partículas.....	30
<b>Figura 19.</b> Resistencia interna de la batería estimada mediante Filtro de Partículas.....	30

## RESUMEN

La estimación del estado de carga (SOC) en dispositivos de almacenamiento de energía es un factor clave para su adecuado funcionamiento. Este trabajo presenta un estudio comparativo de métodos de estimación de SOC en baterías de iones de litio. Por una parte, se implementaron modelos basados en un enfoque bayesiano: filtro de Kalman linealizado y el filtrado de partículas (FP). La ecuación de estado de dichos modelos incorpora a la resistencia de la batería como un parámetro de evolución artificial. Para este enfoque bayesiano se empleó una batería con química LCO, tipo 26650. Además, se implementaron dos modelos de aprendizaje automático: bosques aleatorios y K-vecinos cercanos. Para estos dos últimos enfoques se entrenó un dataset público de experimentos de espectroscopía de impedancia electroquímica. Los resultados indican un mejor desempeño de estimación del SOC por parte de los modelos bayesianos, siendo el FP quien presenta mejores métricas de desempeño.

**Palabras claves:** SOC, batería, estimador bayesiano.

## ABSTRACT

Estimating the state of charge (SOC) of energy storage devices is a key factor for their proper functioning. This work presents a comparative study of SOC estimation methods in lithium-ion batteries. First, models based on a Bayesian approach were implemented: linearized Kalman filter and particle filtering (PF). The equation of state of these models incorporates the battery resistance as an artificial evolution parameter. For this Bayesian approach, a 26650-battery cell type with LCO chemistry was used. Additionally, two machine learning models were implemented: random forests and K-nearest neighbors. A public dataset of electrochemical impedance spectroscopy experiments was used to train these two approaches. The results indicate better SOC estimation performance by the Bayesian models, with PF exhibiting the best performance metrics.

**Keywords:** SOC, battery, Bayesian estimator.

# INTRODUCCIÓN

Las baterías de iones de litio son componentes críticos en una gran variedad de sistemas eléctricos y electrónicos, que incluyen electrodomésticos, dispositivos médicos, computadoras portátiles, vehículos eléctricos y sistemas de almacenamiento de energía (Hema & Venkatarangan, 2024). Gracias a su alta densidad energética, larga vida útil, excelente eficiencia y operación eco amigable, las baterías de iones de litio se han convertido en una de las tecnologías de almacenamiento de energía más competitiva (D.V.S.R. et al., 2023). El estado de carga (SOC) es un parámetro clave que proporciona una medida indirecta de la capacidad útil restante de una batería.

Para garantizar un funcionamiento seguro y eficiente, las baterías de iones de litio se basan en sistemas de gestión de baterías (BMS), que monitorizan continuamente variables esenciales como el voltaje, corriente, temperatura y la capacidad para estimar tanto el SOC como el estado de salud (SOH) (Tavakol-Moghaddam et al., 2024).

La monitorización de estas variables permite al BMS proteger la batería contra condiciones no favorables tales como sobrecarga, sobredescarga, temperaturas extremas y sobrecorrientes, que pueden provocar degradación física y reducir significativamente su vida útil (Demirci et al., 2024).

Existen varios métodos directos de medición del SOC, entre estos se pueden mencionar métodos directos, como el método de conteo de Coulomb, el voltaje de circuito abierto (OCV), la espectroscopia de impedancia electroquímica (EIS) y los métodos de resistencia interna. Además, existen métodos de estimación indirecta, que implican la predicción del valor del SOC de la batería mediante modelos y algoritmos matemáticos, con una mayor precisión en comparación con los métodos basados en mediciones directas. Los enfoques de análisis indirecto se pueden clasificar en: métodos basados en modelos, basados en filtros adaptativos, basados en inteligencia artificial adaptativa, algoritmos avanzados y otros métodos. Los métodos basados en modelos se basan en algoritmos que crean una representación matemática del comportamiento y las características eléctricas de la batería. En este mismo enfoque de métodos basados en modelos se incluyen el modelo de circuito eléctrico (ECM) y el modelo electroquímico (EChM), ampliamente utilizados para la estimación del SOC y que sirven de base para diversas técnicas de modelado de baterías (Wang et al., 2017).

Tabine et al. (2025) proponen un modelo que emplea un ajuste polinomial. Este enfoque no requiere un modelo térmico, lo que permite corregir los errores inducidos por la temperatura. Por otra parte, en términos de enfoques basados en modelos, Zou et al. (2018) manifiestan que los modelos fraccionarios pueden lograr una mayor precisión en comparación con sus contrapartes de orden entero, aunque conllevan una mayor complejidad.

Además, se podrían considerar parámetros físicos en la estimación del SOC. Por ejemplo, la dependencia de la resistencia interna de la batería para la estimación del SOC puede aprovecharse para obtener resultados más precisos (Xu et al., 2023). Además, el efecto de la temperatura también puede causar inconsistencia en la estimación del SOC de un módulo de baterías (Ren et al., 2023).

La espectroscopia de impedancia electroquímica (EIS) es una técnica eficaz para caracterizar una batería. El experimento de EIS se basa en la aplicación de una señal alterna en un rango de frecuencias. Los datos de corriente y voltaje resultantes se correlacionan para identificar impedancias, lo que proporciona información sobre los procesos electroquímicos en la batería, como la cinética de transferencia de carga, la difusión de iones y las reacciones interfaciales (Lazanas & Prodromidis, 2023). El enfoque de EIS puede emplearse para mejorar la estimación del SOC y SOH (Mustafa et al., 2024). Una desventaja de la EIS es que requiere equipos de prueba específicos y control de temperatura para obtener mejores resultados.

Por otro lado, existen modelos de estimación de SOC basados en datos, incluyendo mediciones de temperatura, voltaje y corriente. Estos enfoques pueden utilizar modelos más simples o incluso funcionar sin un modelo, pero su precisión disminuye cuando los datos de entrada son ruidosos o incompletos. Sin embargo, estos modelos son más flexibles porque no requieren un modelo de batería específico y consumen menos recursos. El proceso de estimación del SOC basado en datos comprende tres etapas importantes: recopilación de datos, entrenamiento del modelo y estimación del SOC. Algunas de las técnicas más utilizadas son las redes neuronales artificiales (RNA), la regresión de vectores de soporte (SVR), las máquinas de vectores de soporte (SVM) y la lógica difusa (D.V.S.R. et al., 2023).

Este trabajo presenta una comparación de diferentes métodos de estimación del estado de carga (SOC). También se aborda un enfoque particular en la influencia de la resistencia interna en el SOC.

La comprensión e implementación de modelos lineales y no lineales, así como enfoques basados en inteligencia artificial, ofrecen una nueva perspectiva para explorar las limitaciones de los métodos actuales y mejorar su robustez y adaptabilidad en diversas condiciones de operación. En este estudio se presentan modelos directos e indirectos. El enfoque directo se basa en mediciones de EIS, donde el SOC se predice mediante algoritmos de árboles aleatorios (RF) y K-vecinos cercanos (KNN). Los enfoques indirectos propuestos son: el filtro de Kalman linealizado (LKF) y el filtro de partículas (PF), donde se emplea un perfil de corriente del ciclo de conducción para estimar el SOC de la batería. Al comparar diferentes métodos de estimación del SOC, es posible identificar el modelo más adecuado y robusto para cada condición de operación, lo que mejora las decisiones del BMS. Este estudio facilita la optimización del rendimiento y la prolongación de la vida útil de las baterías de iones de litio, beneficiando a múltiples sectores y contribuyendo al desarrollo de tecnologías más sostenibles y eficientes.

### **Formulación del problema de investigación**

El presente proyecto está enfocado en un análisis comparativo de métodos de estimación del SOC en una batería de iones de litio. De esta manera, el problema general se plantea a continuación:

¿Cómo determinar la precisión y robustez en la estimación del estado de carga en baterías de iones de litio con énfasis en el comportamiento de la resistencia interna, lo cual afecta la eficiencia y vida útil de la batería?

### **Objetivo General:**

Desarrollar un análisis comparativo entre métodos de estimación del estado de carga para una batería de iones de litio.

### **Objetivos Específicos:**

1. Identificar y clasificar los métodos de estimación de estado de carga más relevantes para baterías de iones de litio para establecer un marco de referencia en la elección de técnicas según su aplicabilidad y contexto.
2. Implementar algoritmos de estimación de estado de carga con énfasis en el comportamiento de la resistencia interna para analizar cómo afecta esta característica a la precisión y estabilidad de la estimación en distintos modelos de batería.
3. Determinar la robustez y adaptabilidad de los métodos de estimación en diferentes escenarios operativos para evaluar la eficacia de cada técnica frente a variaciones en condiciones de carga y otros factores.
4. Establecer las métricas de análisis comparativo, tales como error cuadrático medio y tiempo de convergencia computacional, para cuantificar y comparar objetivamente el rendimiento de cada método en la estimación del estado de carga

### **Planteamiento hipotético**

La implementación de métodos de estimación del estado de carga de una batería de iones de litio, adaptables a diversas condiciones operativas con enfoque en comportamiento de la resistencia interna, puede aportar al estudio del sistema de gestión de baterías, buscando mejorar el rendimiento y la vida útil de estas.

# CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

## 1.1. Revisión de literatura

Los enfoques más comunes para la estimación del SOC son el método de conteo de Coulomb o la integración de amperios-hora, el método OCV, el enfoque basado en datos y los métodos basados en modelos (Wang et al., 2025).

Paccha-Herrera et al. (2020) utilizaron el método de conteo de Columb para calcular el SOC de una celda de litio y posterior a ello, estimar su generación de calor interna y temperatura superficial. El modelo de estimación de temperatura empleado presenta un bajo nivel de error RMS con respecto de los resultados experimentales. Al utilizar el método de conteo de Coulomb, el error de SOC debido a la medición e integración de la corriente crece con el tiempo, mientras que el error de debido a la incertidumbre en la capacidad de la batería es proporcional al estado de carga acumulado (Movassagh et al., 2021).

Para abordar el error acumulativo del método de conteo de Coulomb, se usan enfoques basados en filtros bayesianos. Por ejemplo, Cui et al. (2022) formuló un filtro de Kalman extendido (EKF) utilizando un modelo de circuito equivalente de Thevenin, El fundamento del EKF es usar una ecuación de estado para estimar el estado del sistema en el siguiente instante para resolver un problema no lineal. Se emplea la diferencia entre las observaciones reales y la salida predicha para calcular la estimación de la varianza mínima y actualizar el coeficiente de ganancia de Kalman. Luego, se emplea dicho coeficiente para corregir las ponderaciones de los valores observados y predichos en la estimación de la variable de estado. Los resultados muestran que este modelo presenta un mejor desempeño con respecto a los métodos tradicionales de estimación de SOC.

Para estudiar la estimación del SOC como un problema no lineal, también se emplea el filtro de partículas (PF). Guerrab y Lakhdari (2025) presentaron un filtro de partículas

extendido (EPF) que mejora el desempeño de un PF convencional. El PF es un enfoque bayesiano que emplea un conjunto de partículas derivadas de una distribución a priori, que se actualiza a través de mediciones, para de esta manera aproximar la función de densidad de probabilidad del estado del sistema. A diferencia del filtro de Kalman, que tiene una base lineal y ruido gaussiano, el PF puede manejar sistemas no lineales con distribuciones de ruido no gaussiano. El filtro EKF se desarrolló empleando un circuito eléctrico equivalente RC e incorpora la inicialización guiada de partículas, reducción de varianza y un método de estimación de ruido de medición de manera adaptativa. Este enfoque presenta mejor desempeño y robustez en comparación a los métodos de estimación de SOC basados en EKF y PF.

Otro estudio reciente que se enfoca en el mejoramiento del PF para estimación del SOC es presentado por Chen et al. (2023). Los autores manifiestan que el PF puede sufrir degeneración de partículas en donde los pesos se desbalancean y para contrarrestar ello proponen un filtro de partículas H-infinito usando distintos ciclos de conducción estandarizados. Este enfoque mejora notablemente el rendimiento del PF comparado con el caso cuando los pesos de las partículas no son tratados.

En cuanto a enfoques basados en inteligencia artificial, Almaita et al. (2022) estimaron el SOC de un sistema de almacenamiento de baterías de iones de litio de alta capacidad, utilizado para mejorar el desempeño de una red de distribución. Los autores afirman que el modelo de red neuronal de memoria a largo plazo (LSTM) utilizado, muestra una alta precisión para estimar el SOC de un grupo de baterías en comparación con otros modelos de redes neuronales, presentando un error estándar máximo (MSE) inferior al 0.62 %. Además, un modelo LSTM puede considerar distintos rangos de temperatura, así como condiciones de envejecimiento para mejorar la estimación del SOC (Chen et al., 2024).

Los modelos de circuitos equivalentes (ECM) con frecuencia se utilizan junto con mediciones de EIS, combinando componentes eléctricos y ajustando sus parámetros para minimizar el error entre los datos obtenidos y los espectros de impedancia modelados

(Meddings et al., 2020). Por ejemplo, Buchicchio et al. (2023) desarrollaron un método de estimación del SOC de una batería utilizando un ECM basado en mediciones de EIS combinado con algoritmos de aprendizaje automático. Se emplearon siete parámetros, incluyendo elementos de orden fraccionario. Este método reduce el tiempo de entrenamiento y es adecuado para la estimación del SOC en línea llegando a una exactitud superior al 93% cuando se emplea modelos basados en KNN.

El estudio desarrollado por Zhang et al. (2025) presenta un método de estimación de SOC empleando técnicas de aprendizaje de máquinas: bosques aleatorios (RF) combinado con una red neuronal convolucional (CNN) a través de un modelo de ECM y datos EIS. La red CNN se emplean para extraer características de los diagramas de Nyquist generados a partir de los datos de EIS. Estos resultados en conjunto con los parámetros obtenidos del ECM sirven como entradas para entrenar el algoritmo RF. Mediante este enfoque se alcanzó un coeficiente de determinación de 0.9926.

## 1.2. Desarrollo teórico y conceptual

El SOC de una batería representa la relación entre la capacidad residual  $Q_t$  y la capacidad nominal  $Q_n$ , y puede escribirse según lo indica la ecuación (1):

$$\text{SOC}(t) = \frac{Q_t}{Q_n} \times 100 \quad (1)$$

De acuerdo con Miao y Gao (2022), el SOC se puede expresar en términos de la integración de la corriente  $i(t)$  según la ecuación (2):

$$\text{SOC}(t) = \text{SOC}_0 - \frac{\eta \int_{t_0}^t i(t) dt}{Q_n}, \quad (2)$$

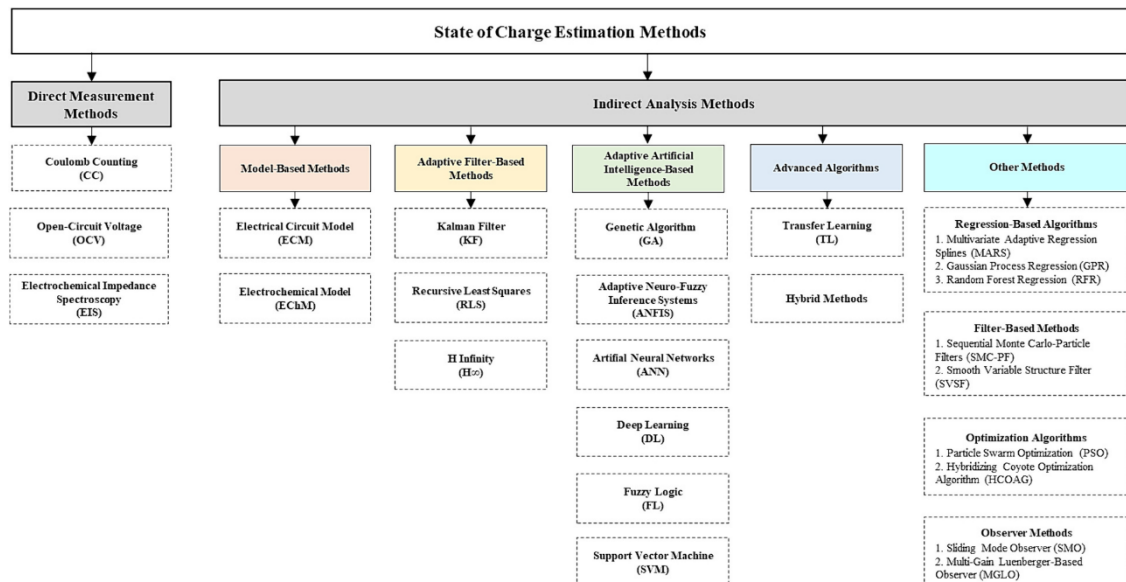
Donde  $\text{SOC}_0$  es el valor inicial del estado de carga, la corriente  $i(t) < 0$  para carga e  $i(t) > 0$  para descarga, y  $\eta$  es la eficiencia de Coulomb, que representa la relación entre la carga total extraída de la batería y la carga total suministrada a la misma durante un ciclo completo.

### 1.2.1 Métodos de estimación de SOC

Existen algunos enfoques al realizar la clasificación de métodos de estimación de estado de carga. Una forma es clasificarlos como métodos directos e indirectos, tal como se muestra en la Figura 1. En los métodos directos se encuentran el conteo de Coulumb, voltaje de circuito abierto y EIS. En los métodos indirectos se encuentran métodos basados en modelos, métodos con filtros adaptativos, algoritmos avanzados y otros métodos basados en regresiones, filtros, algoritmos de optimización y métodos de observación.

**Figura 1**

*Métodos de estimación de estado de carga de baterías*



*Nota.* Tomado de Demirci et al. (2024).

Entre los modelos de estimación de SOC, uno de los más simples es el conteo de Coulomb, aunque no es recomendable en condiciones dinámicas. El filtro de Kalman y sus variantes han sido utilizados a gran escala en diversas aplicaciones incluyendo vehículos eléctricos. Los métodos basados en EIS caracterizan el comportamiento electroquímico, pero requieren equipos especiales y no se puede realizar en línea. Los métodos basados en ECM son usados con frecuencia en sistemas BMS, pudiendo operar en línea. Las técnicas con base en inteligencia artificial pueden adaptarse a diversas

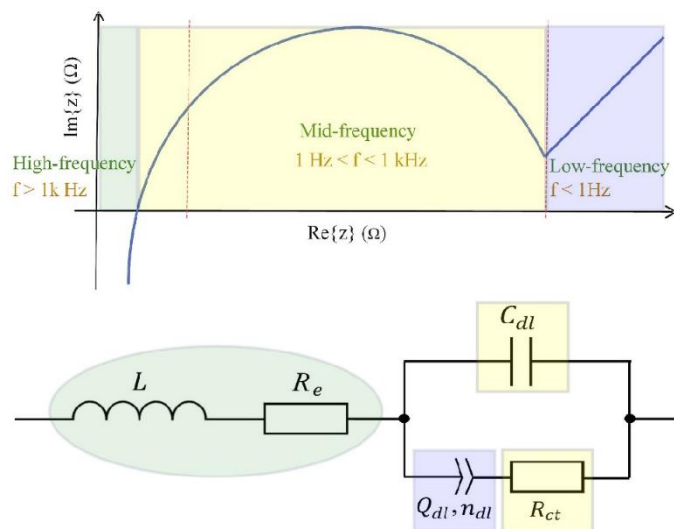
condiciones y mejorar la estimación del SOC, pero requieren datos de entrada precisos y entrenamiento cuidadoso. Los esfuerzos actuales se enfocan en capturar el comportamiento no lineal de las baterías y en refinar los algoritmos existentes, incluyendo la implementación de métodos híbridos (Demirci et al., 2024).

### 1.2.2 Espectroscopía de impedancia electroquímica (EIS)

La EIS caracteriza una batería mediante la aplicación de una tensión senoidal de pequeña amplitud (entre 5 mV y 50 mV) en un rango de frecuencia de entre 0.001 Hz y 100 kHz (Pisani et al., 2025). Para ello, a través de la medición de la amplitud de la corriente y el desfase en cada frecuencia, se determinan los componentes resistivos y reactivos de la impedancia de la batería para un determinado estado de carga. Los resultados de los experimentos de EIS se presentan comúnmente mediante un diagrama de Nyquist, como se muestra en la Figura 2 (parte superior). El espectro de EIS puede comportarse diferente cuando cambian las condiciones de temperatura y envejecimiento de la batería.

**Figura 2**

*Diagrama de Nyquist y su circuito eléctrico equivalente para un test de EIS en una batería*



*Nota.* Los colores de las zonas de alta, media y baja frecuencia se corresponden con sus respectivos componentes del circuito equivalente. Adaptado de Zhang et al. (2025).

Se puede formular un ECM para los respectivos espectros de EIS. La resistencia óhmica pura  $R_e$  está dada por el valor en la intersección de la curva y el eje horizontal; la inductancia  $L$  modela los fenómenos electromagnéticos en la batería. La zona de frecuencia media representa la región de transferencia de carga, incluyendo la resistencia de transferencia de carga  $R_{ct}$  y la capacitancia de doble capa  $C_{dl}$ . La  $R_{ct}$  refleja la resistencia de los iones de litio que se mueven entre el electrodo y el electrolito (Thomas y Weatherspoon, 2025). Finalmente, la zona de baja frecuencia representa el proceso de difusión de los iones de litio, modelado por el elemento de fase constante  $Q_{dl}$ , donde el exponente  $n$  varía entre 0 (comportamiento resistivo) y 1 (comportamiento capacitivo). El ECM de la Figura 2 se caracteriza por la impedancia dada por:

$$Z(\omega) = R_e + \frac{1}{j\omega C_{dl}} + \left[ \frac{1}{Q_{dl}(j\omega)^{-n}} + R_{ct} \right]^{-1} + j\omega L \quad (3)$$

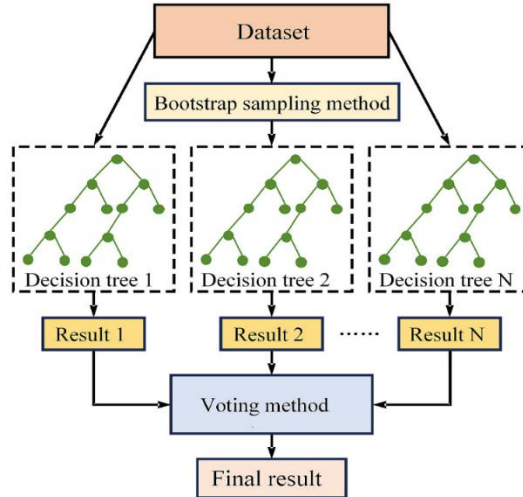
Los parámetros de la ecuación (3) se corresponden a un determinado valor de SOC y pueden emplearse para entrenar algoritmos basados en inteligencia artificial.

### 1.2.3 Bosques aleatorios (RF)

El algoritmo de bosques aleatorios (RF) utiliza un conjunto de árboles de decisión y se emplea para abordar problemas de regresión y clasificación. Los árboles de decisión muestran resultados con base en reglas que se aplican jerárquicamente. En RF, a partir de los datos de entrada se seleccionan muestras aleatorias, este proceso se conoce como *bootstrapping*, de tal forma que se entrenan distintos árboles de manera independiente. El mecanismo de voto para asignar un resultado al entrenamiento puede ser mediante votación mayoritaria (en problemas de clasificación) o promedio de valores calculados por cada árbol (problemas de regresión). Este proceso se ilustra en la Figura 3.

**Figura 3**

*Diagrama del algoritmo de bosques aleatorios (RF)*



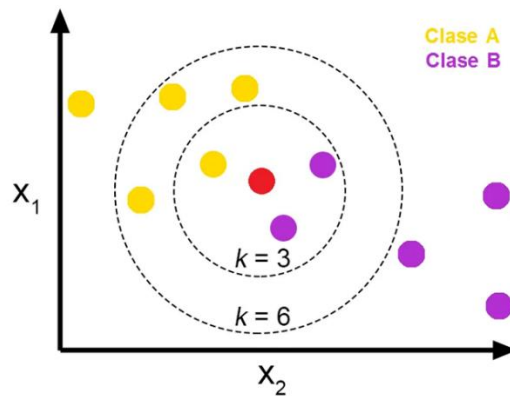
*Nota.* Tomado de Zhang et al. (2025).

#### 1.2.4 K-vecinos más cercanos (KNN)

Si se considera la Figura 4, en donde el círculo rojo es una clase no conocida, mientras que los círculos amarillo y violeta son clases conocidas. Para asociar la clase desconocida (rojo) a una de las otras dos conocidas, se toma un número  $K$  de vecinos cercanos. Por ejemplo, si se toman  $K = 3$  vecinos cercanos, la clase predominante está dada por los círculos violetas, entonces el círculo rojo es asignado a la clase violeta, mientras que si se toma  $K = 6$  vecinos, la clase predominante de la clase desconocida es la amarilla. Luego, para problemas de clasificación se consideran los votos mayoritarios de los vecinos, mientras que, para regresión, se toma el promedio de los valores vecinos.

**Figura 4**

*Esquema del algoritmo K-vecinos más cercanos (KNN)*



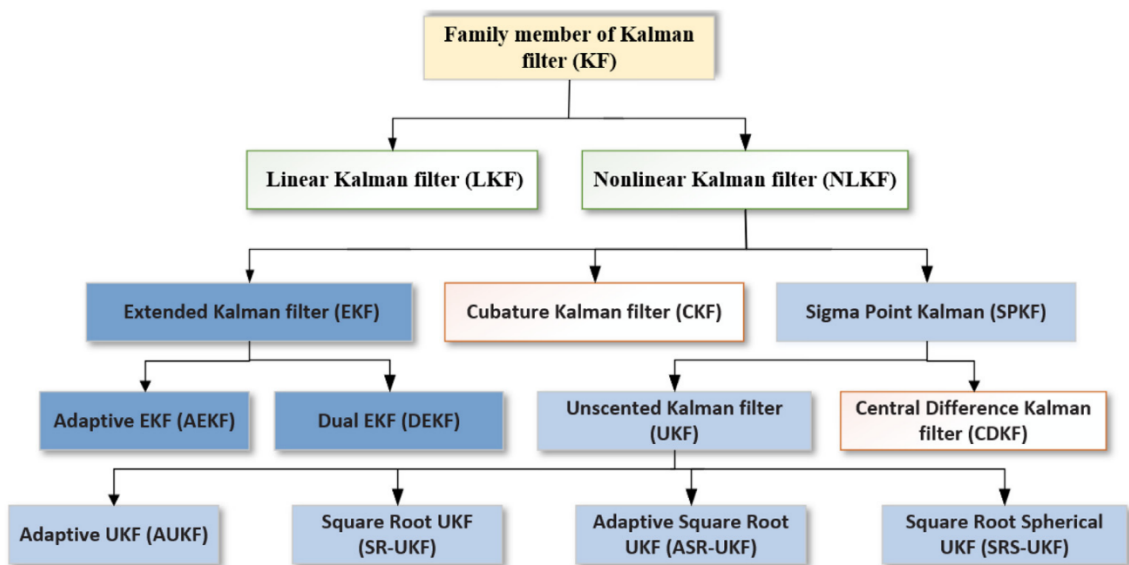
*Nota.* Tomado de Ítalo (2018).

### 1.2.5 Filtro de Kalman

El filtro de Kalman es un método recursivo para estimar estados en un sistema. Se basa en el teorema de Bayes que implica la predicción de un estado futuro y la posterior corrección de este a través de mediciones que pueden incorporar ruidos.

**Figura 5**

*Clasificación de filtros de Kalman*



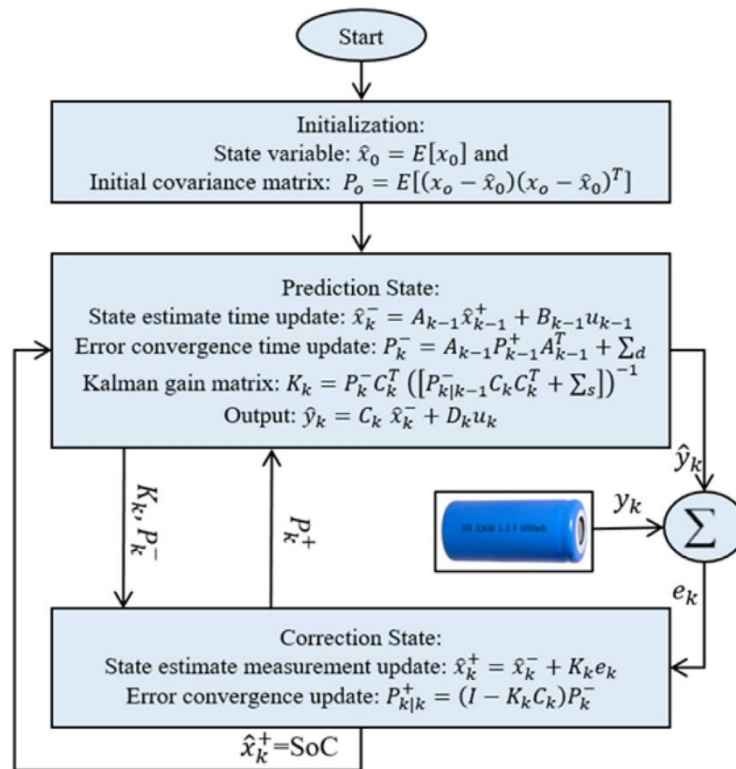
*Nota.* Tomado de Hossain et al. (2022).

Como se observa en la Figura 5, los filtros de Kalman pueden tener algunas variantes, que incluyen dos grandes familias como el KF lineal y KF no lineal. De esta última variante se desprenden las familias de filtro de Kalman extendido (EKF), filtro de Kalman de cubatura (CKF) y filtro de Kalman de puntos sigma (SPKF), existiendo a su vez otros subgrupos de filtros para estas familias.

En la Figura 6 se muestra un esquema del algoritmo de filtro de Kalman estándar, en donde se describen las etapas de inicialización, predicción y corrección para estimar el SOC en una batería.

**Figura 6**

*Filtro de Kalman estándar para estimar SOC de una batería*



*Nota.* Tomado de Hossain et al. (2022).

### Filtro de Kalman linealizado

El filtro de Kalman linealizado (LKF) es un método de estimación utilizado para sistemas no lineales cuya dinámica puede aproximarse de manera adecuada mediante una expansión de primer orden alrededor de una trayectoria de referencia conocida.

Dado el sistema no lineal:

$$x_{k+1} = f(x_k, u_k) + w_k, \quad (4)$$

$$z_k = h(x_k) + v_k, \quad (5)$$

donde  $x$  es el vector de espacio de estados,  $u$  representa la entrada del sistema,  $z$  es el vector de medición,  $k$  es el instante de tiempo,  $w_k$  y  $v_k$  son ruidos de media cero con covarianzas  $Q$  y  $R$ , respectivamente. El filtro LKF asume la existencia de una trayectoria  $\bar{x}_k$  que cumple con:

$$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k), \quad (6)$$

lo cual representa la evolución esperada en ausencia de perturbaciones.

Las variables de error se definen como:

$$\delta x_k = x_k - \bar{x}_k, \quad \delta z_k = z_k - h(x_k) \quad (7)$$

El sistema es linealizado alrededor de  $\bar{x}_k$  por medio del Jacobiano:

$$F_k = \left. \frac{\partial f(x, u_k)}{\partial x} \right|_{x=\bar{x}_k}, \quad (8)$$

$$H_k = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\bar{x}_k}, \quad (9)$$

Luego se puede formular el modelo de estado con error lineal

$$\delta x_{k+1} = F_k \delta x_k + w_k, \quad (10)$$

$$\delta z_k = H_k \delta x_k + v_k, \quad (11)$$

que define un sistema lineal variable en el tiempo (LTV), adecuado para el filtrado de Kalman.

La característica principal del KF es que permite la estimación recursiva de los estados internos (no medibles) de un sistema, que, en el caso de la batería, es el SOC. Esto se logra mediante el uso de un estado previo, predicciones basadas en modelos y mediciones con ruido. Los pasos de predicción y actualización se describen a continuación (Zarchan & Musoff, 2015):

*Predicción:*

$$\delta \hat{x}_k |_{k-1} = F_{k-1} \delta \hat{x}_{k-1} |_{k-1}, \quad (12)$$

$$P_k |_{k-1} = F_{k-1} P_{k-1} |_{k-1} F_{k-1}^T + Q_{k-1}. \quad (13)$$

*Actualización:*

$$K_k = P_k |_{k-1} H_k^T (H_k P_k |_{k-1} H_k^T + R_k)^{-1}, \quad (14)$$

$$\delta \hat{x}_k |_{k-1} = \delta \hat{x}_k |_{k-1} + K_k (\delta z_k - H_k \delta \hat{x}_k |_{k-1}), \quad (15)$$

$$P_k |_{k-1} = (I - K_k H_k) P_k |_{k-1}, \quad (16)$$

donde  $K_k$  es la matriz de ganancia de Kalman,  $P_k$  es la covarianza del error e  $I$  es la matriz identidad. Entonces, el estado actual estimado  $\hat{x}_k |_{k-1}$  se puede formular de la siguiente manera:

$$\hat{x}_k |_{k-1} = \bar{x}_k + \delta \hat{x}_k |_{k-1}. \quad (17)$$

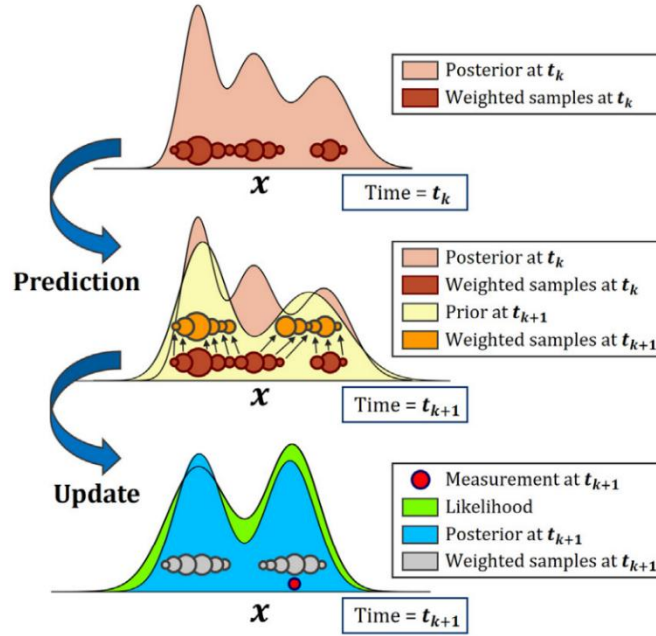
En comparación con el Filtro de Kalman Extendido (EKF), el LKF linealiza únicamente alrededor de una trayectoria nominal predefinida, lo que lo hace adecuado cuando dicha trayectoria se conoce a priori y las desviaciones son pequeñas.

### 1.2.5 Filtro de partículas (PF)

El PF es una herramienta bayesiana recursiva que por lo general se enfoca en problemas no lineales y que busca estimar la distribución de la función de densidad probabilística (PDF) posterior para un estado con base en un modelo de medición y a partir del estado previo. Para ello emplea la propagación de partículas aleatorias que poseen sus propios pesos. Un esquema general de estimación de la PDF posterior se muestra en la Figura 7.

**Figura 7**

*Estimación de la función de densidad probabilística (PDF) mediante filtro de partículas*



*Nota.* El tamaño de los círculos corresponde al peso de las partículas. Tomado de Paccha-Herrera et al. (2024).

El PF, estima recursivamente la función PDF del vector de estado desconocido  $x_k$ , a partir de mediciones  $z_{1:k}$ , siendo  $k$  el tiempo discreto. Para representar la PDF posterior  $p(x_k | z_{1:k})$  se usan las denominadas partículas, que son un conjunto de muestras aleatorias  $N_p$  con sus respectivos pesos  $w$ , descritos a continuación (Arulampalam et al., 2002; Díaz et al., 2020):

$$\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_p}, \sum_{i=1}^{N_p} w_k^{(i)} = 1. \quad (18)$$

Para formular la PDF posterior, el PF obtiene el conjunto de partículas de una PDF alternativa  $q(\cdot)$ , denominada PDF de densidad de importancia. Luego, la PDF posterior se escribe como:

$$p(x_k | z_{1:k}) = \sum_{i=1}^{N_p} w_k^{(i)} \delta(x_k - x_k^{(i)}), \quad (19)$$

donde  $\delta(\cdot)$  es la función Delta de Dirac, mientras que los pesos  $w$  se actualizan mediante la expresión:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(z_k | x_k^{(i)}) p(x_k^{(i)} | x_{k-1}^{(i)})}{q(x_k^{(i)} | x_{k-1}^{(i)}, z_k)}. \quad (20)$$

Para la implementación del PF, la PDF de densidad de importancia se formula de la siguiente manera:

$$q(x_k | x_{k-1}^{(i)}, z_k) = p(x_k | x_{k-1}^{(i)}), \quad (21)$$

lo cual implica que la PDF de densidad de importancia es igual a la PDF a priori. Luego, los pesos pueden ser actualizados mediante la función de verosimilitud dada por:

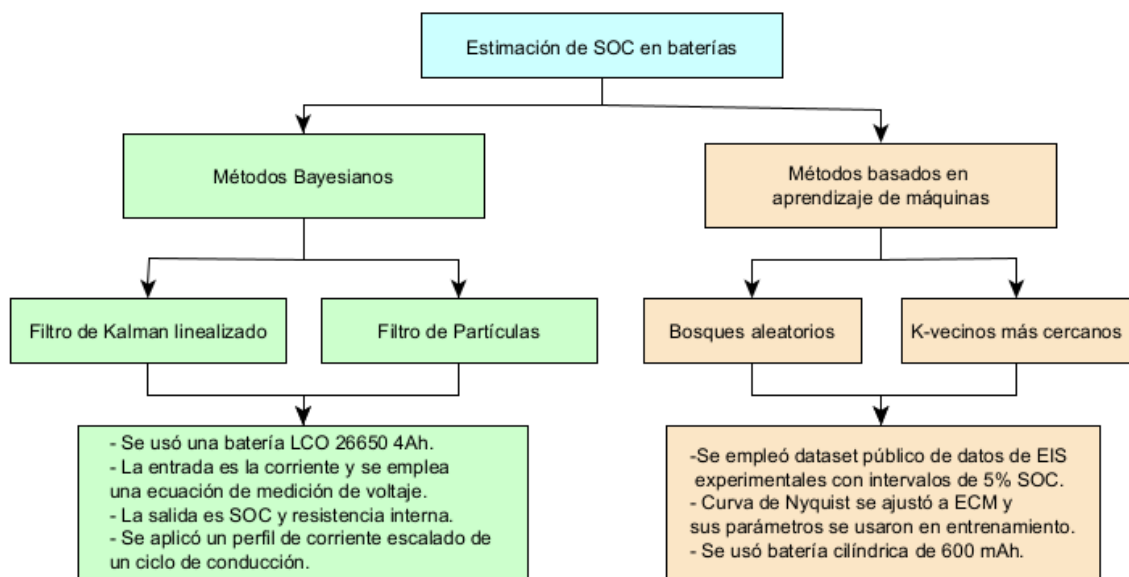
$$w_k^{(i)} = w_{k-1}^{(i)} p(z_k | x_k^{(i)}). \quad (22)$$

## CAPÍTULO 2. METODOLOGÍA

El proceso general para el desarrollo del presente estudio comparativo de métodos de estimación de estado de carga en baterías de iones de litio se muestra en la Figura 8. Se implementaron dos métodos bayesianos (filtro de Kalman linealizado y filtro de partículas) y dos métodos basados en aprendizaje automático (bosques aleatorios y KNN).

**Figura 8**

*Descripción general para implementación de métodos de estimación de SOC*

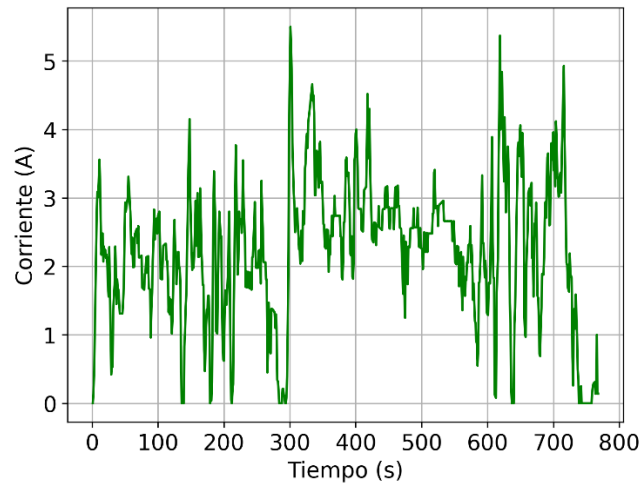


Los métodos bayesianos empleados en este estudio son el KF linealizado y el PF. En ambos casos se formula una ecuación de medición de voltaje. Para que la estimación del SOC sea generalizada, se utilizó como entrada un perfil de corriente variable HWFET que corresponde a un ciclo estandarizado de autopista de un vehículo eléctrico (Ver Figura 9). Este ciclo fue aplicado continuamente hasta alcanzar el voltaje de corte de 2.75 V. El voltaje correspondiente de la celda se muestra en la Figura 10. En este enfoque se utilizó una batería 26650 LCO con capacidad 4 Ah. La celda estuvo 100% cargada al inicio del experimento y se descargó totalmente, a temperatura ambiente.

Por otro lado, se entrenaron dos algoritmos de aprendizaje automático con base en parámetros de un circuito eléctrico, el cual se representa a partir de las curvas de EIS obtenidas de un dataset público. En este caso se utilizó una batería cilíndrica de 600 mAh.

**Figura 9**

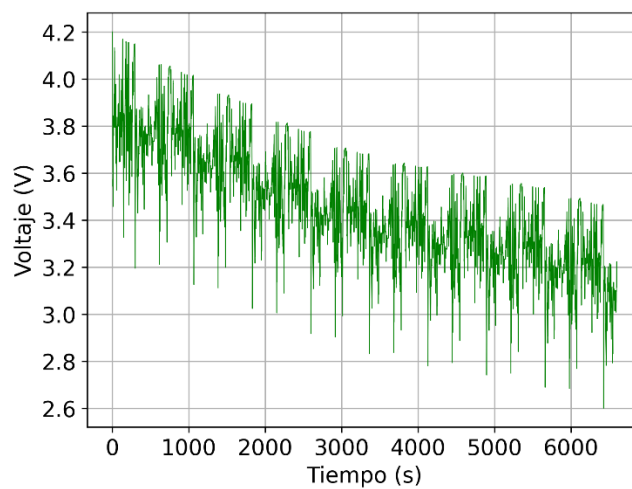
*Perfil de corriente de ciclo HWFET*



*Nota.* La corriente fue aplicada a una batería tipo 26650 de 4 Ah y el ciclo está escalado a una corriente máxima de 5.5 A. Adaptado de Paccha-Herrera et al. (2020).

**Figura 10**

*Perfil de voltaje de ciclo HWFET*



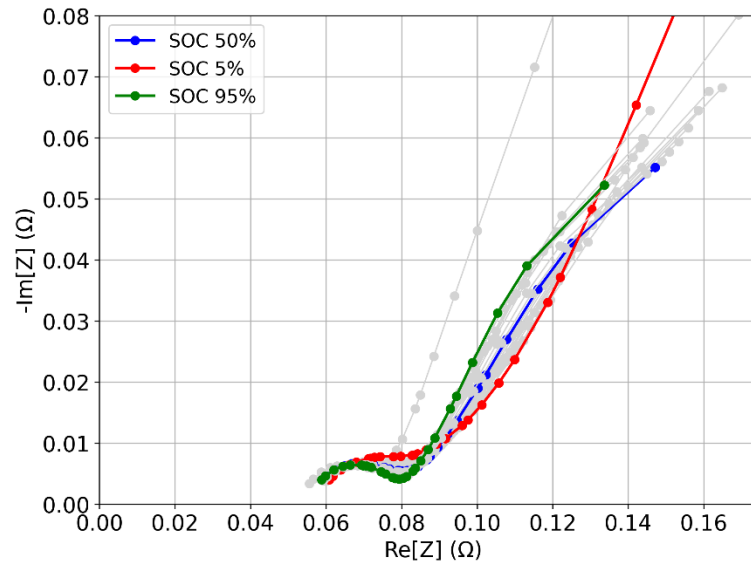
*Nota.* Adaptado de Paccha-Herrera et al. (2020).

## Implementación de modelos basados en aprendizaje automático

Los modelos basados en aprendizaje automático responden a los experimentos de EIS tomados de Mustafa et al. (2024), cuyas curvas de Nyquist se muestran en la Figura 11.

### Figura 11

*Diagrama de Nyquist con base en experimentos de EIS*



*Nota.* Las curvas de EIS fueron obtenidas desde 5% SOC hasta 100% SOC, con intervalos de 5% SOC. En la gráfica se resaltan algunas curvas para mejor interpretabilidad.

La estimación del estado de carga (SOC) basada en mediciones de EIS se aplicó ajustando los parámetros del circuito equivalente de la Figura 1 con las curvas de EIS y el voltaje terminal de la batería (Ver un ejemplo de algunos parámetros en la Tabla 1). Posteriormente, estos parámetros se utilizan para entrenar algoritmos RF y KNN para predecir el estado de carga (SOC) de la batería. El estado de carga (SOC) de la batería se predice para cada medición de EIS. Por lo tanto, también se requieren los parámetros del ECM para cada medición de SOC.

El algoritmo de RF se implementó utilizando 100 estimadores (árboles). Por otro lado, el método KNN se desarrolló con un número de vecinos más cercanos  $K = 2$ . Además, se

empleó una validación de k-fold en ambos casos, con 5 divisiones (*splits*) y un estado aleatorio (*random state*) de 42.

**Tabla 1**

*Parámetros del circuito equivalente ajustados a las curvas de Nyquist para SOC 0.6*

Parámetro	Valor
$C_{dl}$ (F)	0.1
$L$ (H)	$2 \times 10^{-6}$
$n$	0.5
$Q_{dl}$ ( $\Omega^{-1}s^n$ )	$1.03 \times 10^{-4}$
$R_{ct}$ ( $\Omega$ )	0.1
$R_e$ ( $\Omega$ )	$5.9283 \times 10^{-2}$

### Implementación del filtro de Kalman linealizado (LKF)

Para el enfoque de este trabajo, el vector de estado se puede formular de la siguiente manera:

$$x_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad (23)$$

donde  $x_{1,k}$  representa la resistencia interna de la batería y  $x_{2,k}$  es el SOC en el tiempo  $k$ .

El modelo bayesiano del KF necesita una ecuación de medida, la cual fue adoptada de Pola et al. (2015) y representa el voltaje  $v_0$  linealizado alrededor del punto  $x_0 = [x_{1,0} \ x_{2,0}]^T$  e  $i_0$ :

$$v_0 = v_L + (v_{OC} - v_L)e^{\gamma(x_{2,0}-1)} + \alpha v_L + (x_{2,0} - 1) + (1 - \alpha)v_L \left( e^{-\beta} - e^{-\beta\sqrt{x_{2,0}}} \right) - i_0 x_{1,0}, \quad (24)$$

en donde la corriente de descarga  $i$  es el parámetro de entrada en el modelo y  $v_{OC}$  es el voltaje de circuito abierto cuando el SOC es 100%. Los parámetros  $\alpha$ ,  $\beta$ ,  $\gamma$  y  $v_L$  se estiman fuera de línea.

Las matrices de las ecuaciones (8) y (9) están dadas por:

$$F_k = \begin{bmatrix} 1 & 0 \\ -\frac{\Delta t}{E_{crit}} i_0 \frac{\partial v_0}{\partial x_1} & 1 - \frac{\Delta t}{E_{crit}} i_0 \frac{\partial v_0}{\partial x_2} \end{bmatrix}, \quad (25)$$

$$H_k = \begin{bmatrix} \frac{\partial v_0}{\partial x_1} & \frac{\partial v_0}{\partial x_2} \end{bmatrix}, \quad (26)$$

Donde  $E_{crit}$  es la energía crítica, la cual representa la energía total que proporciona la batería.

### Implementación del filtro de partículas (PF)

Para el enfoque del PF primero se establece el modelo de transición de estado dado por:

$$x_1(k+1) = x_1(k) + \omega_1(k), \quad (27)$$

$$x_2(k+1) = x_2(k) - v(k)i(k)\Delta t E_{crit}^{-1} + \omega_2(k), \quad (28)$$

donde  $x_1$  es la resistencia interna de la batería y  $x_2$  es el SOC. Al igual que el LKF, el PF necesita una ecuación de medición, la cual también está dada en términos del voltaje y se escribe en forma discreta mediante:

$$v(k) = v_L + (v_{OC} - v_L)e^{\gamma(x_2(k)-1)} + \alpha v_L + (x_2(k) - 1) \\ + (1 - \alpha)v_L \left( e^{-\beta} - e^{-\beta\sqrt{x_2(k)}} \right) - i(k)x_1(k) + \eta(k), \quad (29)$$

En la formulación del PF se incluye tanto el ruido de proceso ( $\omega_1$  y  $\omega_2$ ) como el ruido de medición  $\eta$ .

Para tratar con parámetros desconocidos o variables en el tiempo, se utiliza la evolución artificial. Esta técnica estima estados y parámetros juntamente con el PF. Este método se implementa extendiendo la dimensión del vector de espacio de estados, donde el parámetro  $\theta$  se incluye como un estado adicional en forma de un modelo de caminata aleatoria (*random walk*) variable en el tiempo. En el presente enfoque se incluyó la resistencia interna de la batería  $x_1$  como el nuevo parámetro. Como resultado, el modelo de proceso extendido puede representarse mediante las siguientes ecuaciones dinámicas (Hong et al., 2020):

$$x_k = f_k(x_{k-1}, \theta_{k-1}, u_{k-1}, \omega_{k-1}), \quad (30)$$

$$\theta_k = \theta_{k-1} + \eta_{k-1}, \quad (28)$$

### 2.1. Contexto de la investigación

Esta investigación es de tipo teórica que no requirió realizarse en un lugar físico o geográfico en específico.

### 2.2. Diseño y alcance de la investigación

A pesar de que se desarrollaron modelos teóricos, los modelos bayesianos utilizaron los datos de experimentos presentados por Paccha-Herrera et al. (2020), mientras que para los métodos de aprendizaje automático se empleó el dataset desarrollado por Mustafa et al. (2024). En cuanto al alcance de la presente investigación, esta es de tipo descriptivo, ya que se describe y detalla la implementación de los algoritmos y luego se comparan sus respectivas métricas de desempeño.

### 2.3. Tipo y métodos de investigación

El tipo de investigación es cuantitativa, ya que se busca comparar el SOC real versus el SOC predicho y de esta manera obtener las respectivas métricas.

El método de investigación tiene un enfoque Hipotético-Deductivo, puesto que, partiendo de la formulación de los distintos métodos y evaluación de su comportamiento, se puede deducir su desempeño mediante métricas.

## 2.4. Población y muestra

La muestra en este caso de estudio corresponde a los dos tipos de baterías empleados: la celda LCO tipo 26650 con 4 Ah para el enfoque bayesiano de donde se extrajeron los datos de corriente y voltaje aplicados durante descarga, y la celda cilíndrica de 600 mAh para los métodos basados en aprendizaje automático, de donde se recuperaron los datos de EIS.

## 2.5. Técnicas e instrumentos de recolección de datos

Para el enfoque bayesiano, los datos de corriente y voltaje del ciclo HWFET durante la descarga fueron adquiridos empleando un dispositivo tipo carga programable BK Precision 8500 (ver Figura 12). Para la carga de la celda de 4 Ah se utilizó un cargador tipo iCharger 208B siguiendo el protocolo corriente constante - voltaje constante (CC-CV). Durante la carga se aplicó una corriente de 2 Ah. Las pruebas fueron realizadas a temperatura ambiente.

**Figura 12**

*Dispositivo tipo carga programable*



*Nota.* Recuperado de <https://www.bkprecision.com/products/dc-electronic-loads/8510B>

Para el enfoque de aprendizaje de máquinas, los autores del dataset experimental de EIS emplearon un equipo tipo Hioki BT4560.

Los algoritmos se implementaron en un computador portátil con procesador i7, 32 GB de memoria RAM utilizando la plataforma Jupyter Notebook con el lenguaje Python.

## **2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información**

Por una parte, el dispositivo BK Precision 8500 se caracteriza por poseer una resolución que va desde 0.1 mA a 1 mA, pudiendo manejar corrientes de hasta 30 A con un voltaje mínimo de operación de 0.14 V. La temperatura de operación es de 0 a 40 °C.

En cuanto al dispositivo para la prueba de EIS, el equipo Hioki BT4560 mide parámetros como resistencia, reactancia, impedancia y ángulo de fase. Su frecuencia es de 0.01 Hz hasta 1050 Hz, con rangos de medición de 30000 mΩ, 100000 mΩ y 100000 mΩ. Además, posee una corriente de medición desde 50 mA rms y voltaje de hasta 5 V. La precisión de la medida de impedancia es de  $\pm 0.4\%$ .

Dado que los datos obtenidos han sido respaldados por publicaciones previas, se tiene la confiabilidad necesaria para el procesamiento de estos.

## CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

### 3.1 Métodos de estimación de estado de carga

En el presente estudio se han enfocado algunos de los métodos clásicos y también otros algoritmos con proyección de exploración para la estimación del estado de carga de baterías de iones de litio, que son muy usadas en diversos dispositivos eléctricos y electrónicos, así como en vehículos eléctricos.

Entre los métodos directos, entre los más sencillos destaca el conteo de Columb, pero como desventaja, sufre de acumulación de errores y requiere una tasa adecuada de muestreo. Otro método directo es el de voltaje de circuito abierto, que no es adecuado para operación en línea (en tiempo real). Un método directo preciso está basado en la EIS, pero requiere equipos especiales y protocolos, pero, permite caracterizar la electroquímica de la celda.

Un método clásico constituye el filtro de Kalman, que presenta diversas variantes y en estudio se ha adoptado el filtro de Kalman linealizado, que puede trabajar con sistemas no lineales. Otra herramienta para abordar problemas no lineales es el filtro de partículas. Ambos métodos son enfoques bayesianos que requieren una ecuación de medición para actualizar la estimación.

Por otra parte, se disponen de modelos basados en circuitos eléctricos equivalentes, que son bastante empleados en BMS pueden ser incorporados en otros métodos. Estos métodos necesitan un ajuste correcto de parámetros.

Hoy en día, con el avance de la inteligencia artificial se proponen métodos basados en aprendizaje de máquinas y redes neuronales. De esta manera, en este estudio se enfocan dos métodos bajo dicho enfoque: bosques aleatorios y K-vecinos más cercanos. Los métodos basados en IA pueden ofrecer alta precisión, pero necesitan un entrenamiento

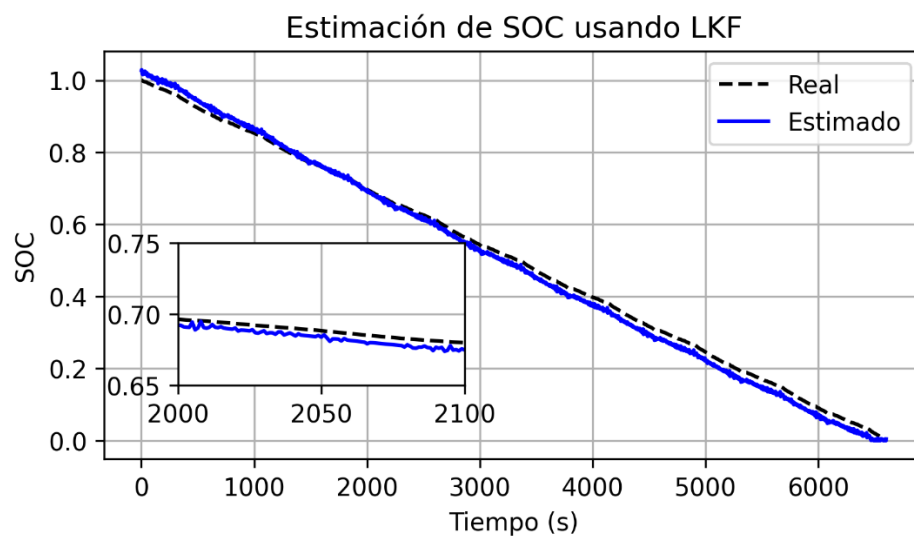
adecuado y, además, también se proponen métodos híbridos que combinen algoritmos de IA.

### 3.1 Resultados de modelos implementados

Las Figuras 13, 14, 15 y 16 presentan la estimación del SOC obtenida mediante el filtro de Kalman linealizado, filtro de partículas, bosque aleatorio K-vecinos más cercanos, respectivamente. Los dos últimos enfoques presentan una calidad de ajuste menor debido a que requieren más datos de entrenamiento. Además, los hiperparámetros del modelo podrían implementarse mediante otras técnicas que pueden incluir modelos híbridos. El algoritmo PF presenta un rendimiento robusto gracias a su capacidad para capturar el comportamiento no lineal y realizar estimaciones en línea simultáneamente. Además, los algoritmos bayesianos son más complejos de implementar y su robustez depende de diversos factores, como una adecuada ecuación del proceso, el ruido del proceso y la medición, etc. El ajuste y calibración de estos modelos puede requerir de paciencia.

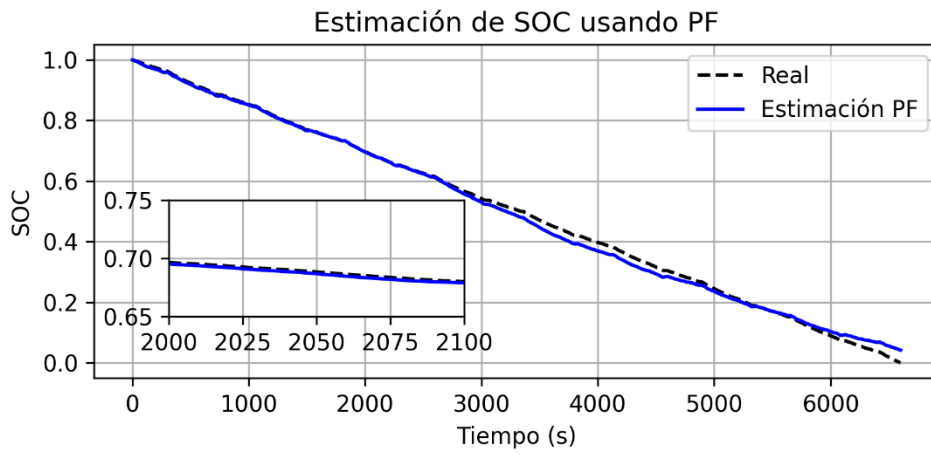
**Figura 13**

*Estimación de SOC mediante filtro de Kalman linealizado (LKF)*



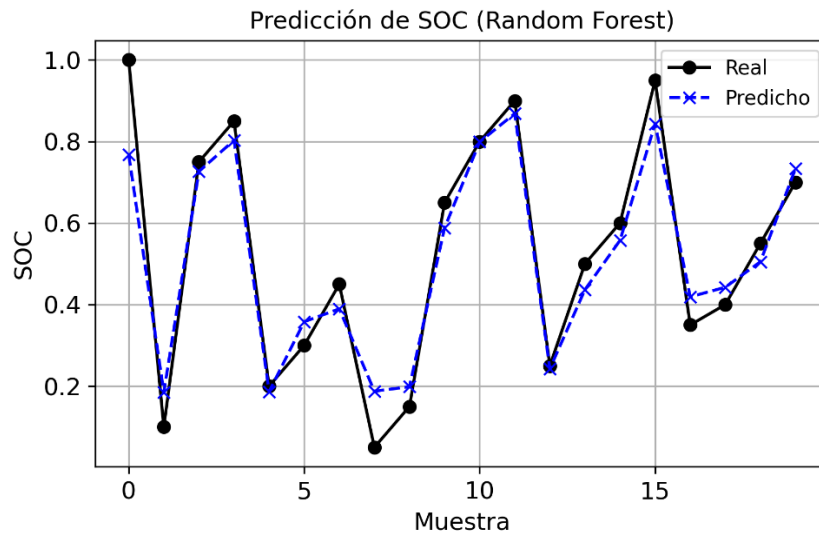
**Figura 14**

*Estimación de SOC mediante Filtro de Partículas (PF)*



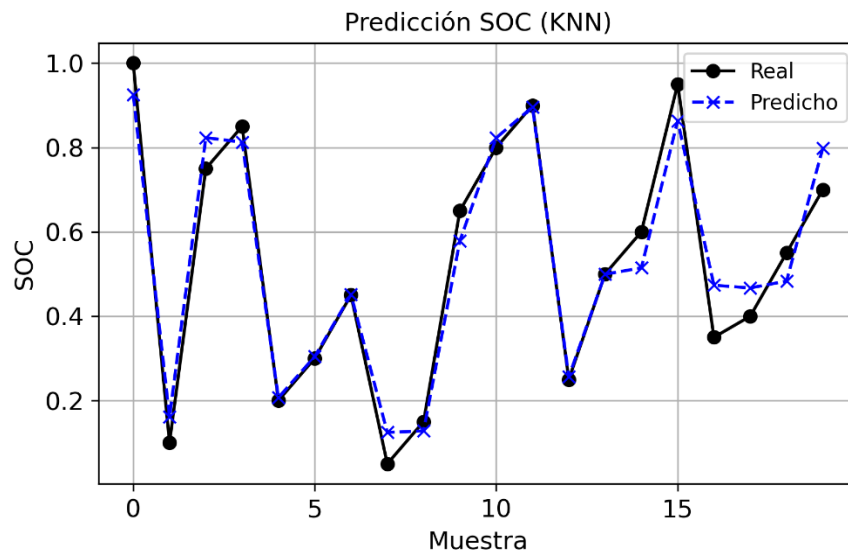
**Figura 15**

*Estimación de SOC mediante bosques aleatorios (RF)*



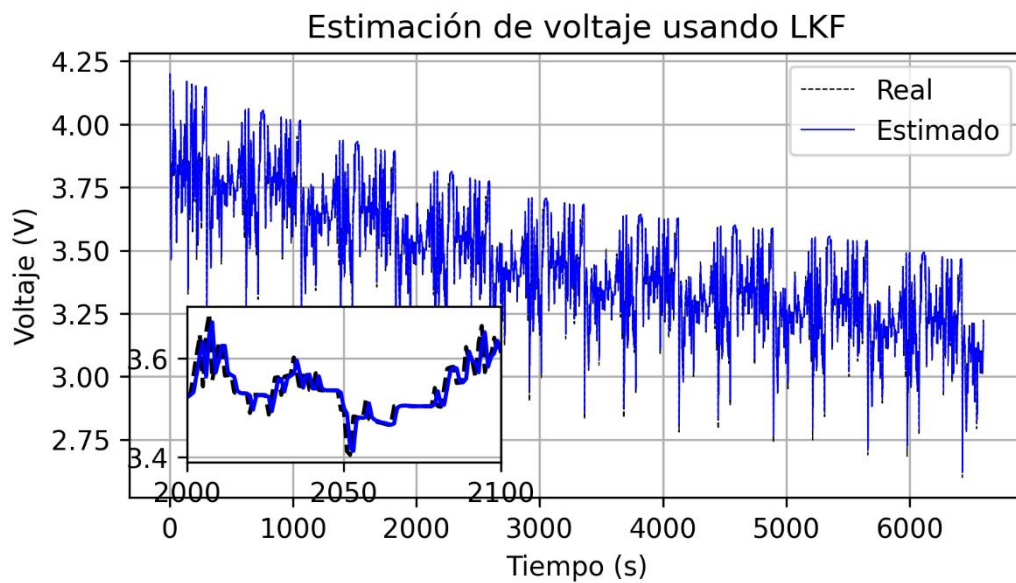
**Figura 16**

*Estimación de SOC mediante K-vecinos más cercanos (KNN)*



**Figura 17**

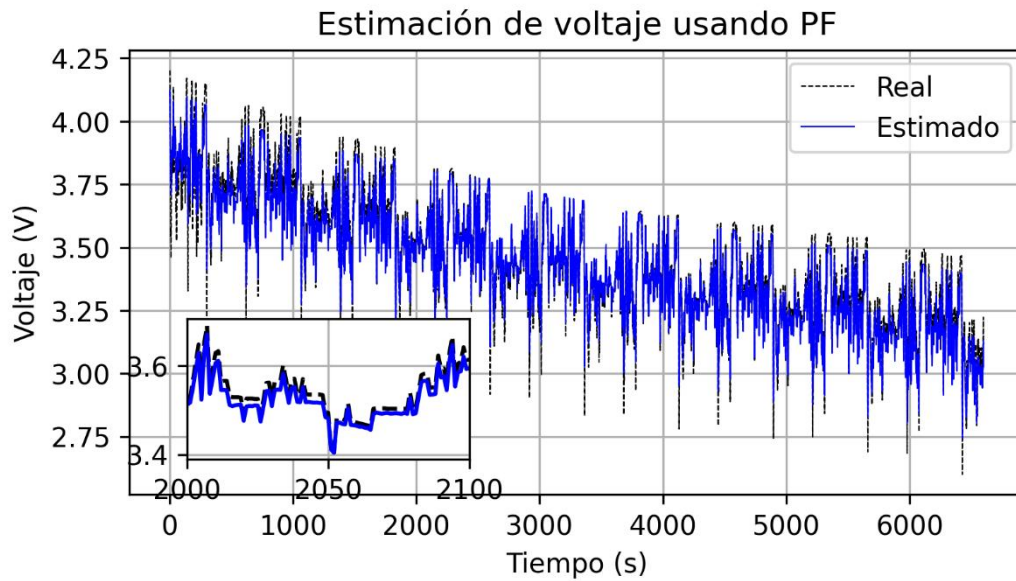
*Voltaje estimado mediante filtro de Kalman*



Además, las Figuras 18 y 19 muestran la estimación del voltaje mediante la aplicación del Filtro de Kalman y el Filtro de Partículas, respectivamente. El PF presenta ligeramente un mejor ajuste al comparar el voltaje real y el estimado.

**Figura 18**

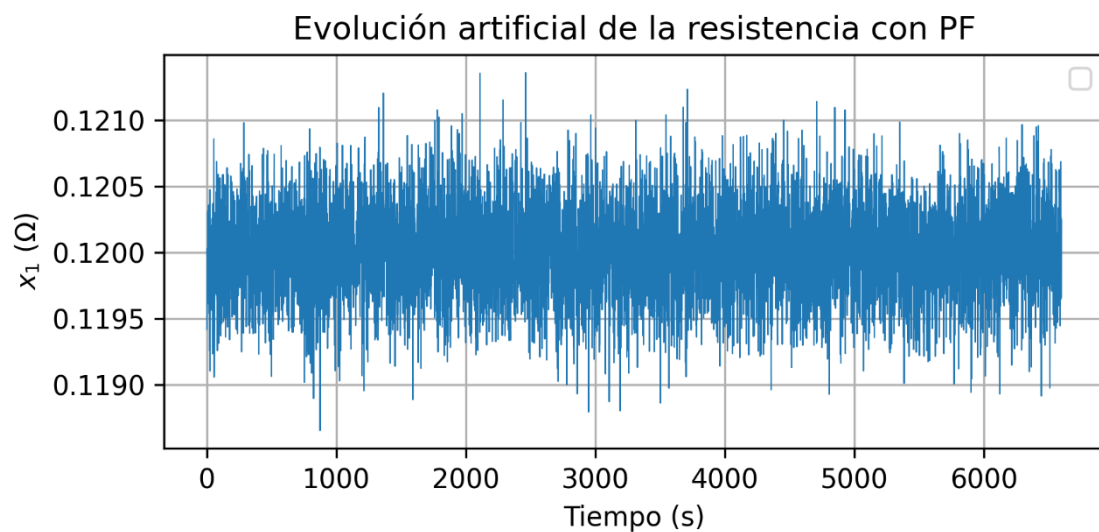
*Voltaje estimado mediante Filtro de Partículas*



Una ventaja de utilizar la evolución artificial de la resistencia de la batería es que su valor inicial evoluciona y se adapta continuamente al modelo PF, tal como se muestra en la Figura 19.

**Figura 19**

*Resistencia interna de la batería estimada mediante Filtro de Partículas*



Se realizó una comparación general de los modelos desarrollados en términos de las métricas RMSE y  $R^2$ , como se muestra en la Tabla 2. En general, los enfoques bayesianos alcanzan un mejor rendimiento que los métodos basados en aprendizaje automático mediante EIS. Debido a su naturaleza, RF y KNN solo predicen valores puntuales de SOC donde se realizaron experimentos de EIS, por lo que para una mejor estimación se requiere gran cantidad de experimentos.

PF presenta las mejores métricas debido a su estructura no lineal, que captura mejor las variaciones de SOC. LKF presenta un rendimiento ligeramente inferior a PF.

**Tabla 2**

*Comparativa de métricas entre los distintos algoritmos*

<b>Algoritmo</b>	<b>RMSE</b>	<b><math>R^2</math></b>
LKF	0.0153	0.9927
PF	0.0164	0.9968
RF	0.0788	0.9252
KNN	0.0618	0.9540

El análisis comparativo de los métodos de estimación del SOC para baterías de iones de litio ha presentado las principales ventajas y limitaciones de cada enfoque. El filtro de Kalman (KF) muestra un rendimiento fiable, pero presenta una ligera sensibilidad a las imprecisiones del modelo y al ruido, lo que podría generar errores durante largos periodos de funcionamiento. El PF, mostró una ligera robustez al gestionar no linealidades e incertidumbres, lo que lo hace especialmente adecuado para la estimación del estado de carga (SOC) en tiempo real en entornos dinámicos. Sin embargo, su complejidad computacional sigue siendo un reto, que requiere implementaciones optimizadas para su aplicación práctica en sistemas de gestión BMS. En el presente enfoque, la formulación del PF ecuación de proceso y ecuación de medición y los parámetros de la batería se adoptaron del trabajo de Pola et al. (2015). Dichos autores lograron una alta precisión en la estimación del SOC mediante la estimación de parámetros fuera de línea para la ecuación de proceso. Otra técnica que podría considerarse para capturar el

comportamiento no lineal es la EKF, utilizada en vehículos eléctricos para ofrecer una estimación del SOC en tiempo real en diferentes condiciones operativas, lo que mejora el rendimiento del BMS (Oloyede et al., 2024). Otras condiciones operativas y parámetros que deben explorarse incluyen otros ciclos de conducción, la temperatura, los efectos del envejecimiento de la batería, el frenado regenerativo y probar otras ecuaciones de proceso en el modelo.

Por otra parte, los enfoques basados en aprendizaje automático (RF y KNN) son menos prácticos que los enfoques bayesianos. RF y KNN predicen el SOC empleando mediciones EIS, pero estos experimentos requieren equipos complejos y no son adecuados para su implementación en línea. Además, los algoritmos de aprendizaje automático requieren el ajuste de distintos parámetros. Zhang et al. (2025) reportaron métricas similares a las mostradas en la Tabla 2 para la predicción del SOC mediante RF y KNN.

Los resultados indican que los enfoques basados en modelo bayesianos como KF y PF, siguen siendo esenciales debido a su interpretabilidad y aplicabilidad en tiempo real, mientras que las técnicas basadas en aprendizaje automático tienen potencial de perfeccionamiento. Los trabajos futuros podrían explorar metodologías híbridas que puedan aprovechar las capacidades de ambos enfoques, integrando técnicas de corrección en línea con modelos de aprendizaje profundo para lograr una estimación robusta y adaptativa del SOC.

## CONCLUSIONES

Existen muchos métodos de estimación de SOC en baterías de iones de litio, que pueden ser directos e indirectos. Entre los métodos directos más sencillos está el conteo de Coulomb, pero sufre de acumulación de errores. Los enfoques bayesianos como KF y PF son a menudo utilizados en distintos sistemas, mientras que las nuevas técnicas con enfoque de IA pueden ofrecer resultados prometedores, pero aún están en continuo desarrollo.

Este estudio ha demostrado que los métodos de estimación del estado de carga (SOC) varían significativamente en precisión, implementación computacional y adaptabilidad a los comportamientos no lineales de las baterías. El enfoque KF proporciona estimaciones fiables, pero requiere atención para abordar problemas de no linealidad y el ruido. El algoritmo PF presenta robustez en condiciones dinámicas. De esta manera, los enfoques bayesianos han presentado mejor desempeño en términos de las métricas de RMSE y  $R^2$  calculadas. Por otro lado, los algoritmos de aprendizaje automático aplicados en este estudio presentan un desempeño aceptable, pero requieren datos EIS especiales para el entrenamiento y no son capaces de predecir el comportamiento del SOC en línea.

Las investigaciones futuras deberían centrarse en la optimización de modelos híbridos de estimación del SOC que integren técnicas de filtrado tradicionales con otros enfoques de aprendizaje automático.

Además, se deben realizar esfuerzos para mejorar la eficiencia computacional de las aplicaciones en tiempo real y ampliar los conjuntos de datos de entrenamiento para mejorar la capacidad de generalización de los modelos basados en datos. La validación en condiciones reales de funcionamiento en vehículos eléctricos y sistemas de almacenamiento de energía es relevante para confirmar la aplicabilidad práctica y la fiabilidad de estos métodos de estimación y predicción.

## **RECOMENDACIONES**

Dado que existen un sinnúmero de métodos de estimación de SOC, es recomendable que los estudios se realicen empleando un solo tipo de batería para una mejor generalización de resultados.

La calibración de los modelos es esencial para su desempeño, por lo que se pueden ajustar los diversos parámetros con nuevos modelos o técnicas. Esto también puede incluir el probar distintos tipos de perfiles de corriente variables.

## REFERENCIAS

- Almaita, E., Alshkoor, S., Abdelsalam, E., & Almomani, F. (2022). State of charge estimation for a group of lithium-ion batteries using long short-term memory neural network. *Journal Of Energy Storage*, 52, 104761. <https://doi.org/10.1016/j.est.2022.104761>
- Arulampalam, Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions On Signal Processing*, 50(2), 174-188. <https://doi.org/10.1109/78.978374>
- Buchicchio, E., De Angelis, A., Santoni, F., Carbone, P., Bianconi, F., & Smeraldi, F. (2023). Battery SOC estimation from EIS data based on machine learning and equivalent circuit model. *Energy*, 283, 128461. <https://doi.org/10.1016/j.energy.2023.128461>
- Chen, Y., Li, R., Sun, Z., Zhao, L., & Guo, X. (2023). SOC estimation of retired lithium-ion batteries for electric vehicle with improved particle filter by H-infinity filter. *Energy Reports*, 9, 1937-1947. <https://doi.org/10.1016/j.egy.2023.01.018>
- Chen, G., Peng, W., & Yang, F. (2024). An LSTM-SA model for SOC estimation of lithium-ion batteries under various temperatures and aging levels. *Journal Of Energy Storage*, 84, 110906. <https://doi.org/10.1016/j.est.2024.110906>
- Cui, Z., Hu, W., Zhang, G., Zhang, Z., & Chen, Z. (2022). An extended Kalman filter based SOC estimation method for Li-ion battery. *Energy Reports*, 8, 81-87. <https://doi.org/10.1016/j.egy.2022.02.116>
- Demirci, O., Taskin, S., Schaltz, E., & Demirci, B. A. (2024). Review of battery state estimation methods for electric vehicles - Part I: SOC estimation. *Journal Of Energy Storage*, 87, 111435. <https://doi.org/10.1016/j.est.2024.111435>
- Diaz, C., Quintero, V., Perez, A., Jaramillo, F., Burgos-Mellado, C., Rozas, H., Orchard, M. E., Saez, D., & Cardenas, R. (2020). Particle-Filtering-Based Prognostics for the State of Maximum Power Available in Lithium-Ion Batteries at Electromobility Applications. *IEEE Transactions on Vehicular Technology*, 69(7), 7187-7200. <https://doi.org/10.1109/tvt.2020.2993949>

- DVSR, S., Badachi, C., & Green, R. C., II. (2023). A review on data-driven SOC estimation with Li-Ion batteries: Implementation methods & future aspirations. *Journal of Energy Storage*, 72, 108420. <https://doi.org/10.1016/j.est.2023.108420>
- Guerrab, O., & Lakhdari, F. (2025). Enhanced state-of-charge estimation of Li-ion batteries using an extended particle filter approach. *Electric Power Systems Research*, 248, 111932. <https://doi.org/10.1016/j.epsr.2025.111932>
- Hema, R., & Venkatarangan, M. J. (2024). Advancing sustainable development: Introducing a novel fast charging technique for Li-ion batteries with supercapacitor integration. *Computers & Electrical Engineering*, 120, 109810. <https://doi.org/10.1016/j.compeleceng.2024.109810>
- Hong, Z., Xu, L., & Chen, J. (2020). Artificial evolution based cost-reference particle filter for nonlinear state and parameter estimation in process systems with unknown noise statistics and model parameters. *Journal of The Taiwan Institute of Chemical Engineers*, 112, 377-387. <https://doi.org/10.1016/j.jtice.2020.04.009>
- Hossain, M., Haque, M., & Arif, M. (2022). Kalman filtering techniques for the online model parameters and state of charge estimation of the Li-ion batteries: A comparative analysis. *Journal of Energy Storage*, 51, 104174. <https://doi.org/10.1016/j.est.2022.104174>
- Ítalo, J. (2018, noviembre 8). *KNN (K-Nearest Neighbors) # 1: How it works*. Medium. <https://medium.com/data-science/knn-k-nearest-neighbors-1-a4707b24bd1d>
- Lazanas, A. C., & Prodromidis, M. I. (2023). Electrochemical Impedance Spectroscopy—A tutorial. *ACS Measurement Science Au*, 3(3), 162-193. <https://doi.org/10.1021/acsmeasuresciau.2c00070>
- Meddings, N., Heinrich, M., Overney, F., Lee, J., Ruiz, V., Napolitano, E., Seitz, S., Hinds, G., Raccichini, R., Gaberšček, M., & Park, J. (2020). Application of electrochemical impedance spectroscopy to commercial Li-ion cells: A review. *Journal of Power Sources*, 480, 228742. <https://doi.org/10.1016/j.jpowsour.2020.228742>

- Miao, Y., & Gao, Z. (2022). Estimation for state of charge of lithium-ion batteries by adaptive fractional-order unscented Kalman filters. *Journal of Energy Storage*, 51, 104396. <https://doi.org/10.1016/j.est.2022.104396>
- Movassagh, K., Raihan, A., Balasingam, B., & Pattipati, K. (2021). A Critical Look at Coulomb Counting Approach for State of Charge Estimation in Batteries. *Energies*, 14(14), 4074. <https://doi.org/10.3390/en14144074>
- Mustafa, H., Bourelly, C., Vitelli, M., Milano, F., Molinara, M., & Ferrigno, L. (2024). SoC estimation on Li-ion batteries: A new EIS-based dataset for data-driven applications. *Data in Brief*, 57, 110947. <https://doi.org/10.1016/j.dib.2024.110947>
- Oloyede, M., Akpakwu, G., Myburgh, H., De Freitas, A., & Kunatsa, T. (2024). A Review on State-of-Charge Estimation Methods, Energy Storage Technologies and State-of-the-Art Simulators: Recent Developments and Challenges. *World Electric Vehicle Journal*, 15(9), 381. <https://doi.org/10.3390/wevj15090381>
- Paccha-Herrera, E., Calderón-Muñoz, W. R., Orchard, M., Jaramillo, F., & Medjaher, K. (2020). Thermal Modeling Approaches for a LiCoO<sub>2</sub> Lithium-ion Battery—A Comparative Study with Experimental Validation. *Batteries*, 6(3), 40. <https://doi.org/10.3390/batteries6030040>
- Paccha-Herrera, E., Jaramillo-Montoya, F., Calderón-Muñoz, W. R., Tapia-Peralta, D., Solórzano-Castillo, B., Gómez-Peña, J., & Paccha-Herrera, J. (2024). A particle filter-based approach for real-time temperature estimation in a lithium-ion battery module during the cooling-down process. *Journal of Energy Storage*, 94, 112413. <https://doi.org/10.1016/j.est.2024.112413>
- Pisani, M. A., García, D., & Valderrama, H. (2025). Review of State-of-Charge Estimation Methods for Electric Vehicle Applications. *World Electric Vehicle Journal*, 16(2), 87. <https://doi.org/10.3390/wevj16020087>
- Pola, D. A., Navarrete, H. F., Orchard, M. E., Rabie, R. S., Cerda, M. A., Olivares, B. E., Silva, J. F., Espinoza, P. A., & Perez, A. (2015). Particle-Filtering-Based Discharge Time Prognosis for Lithium-Ion Batteries With a Statistical

- Characterization of Use Profiles. *IEEE Transactions On Reliability*, 64(2), 710-720. <https://doi.org/10.1109/tr.2014.2385069>
- Ren, H., Jia, L., Yin, L., Dang, C., & Chen, Z. (2024). Research on battery pack SOC consistency based on the electric-thermal-fluid coupling model. *Journal of Energy Storage*, 97, 112924. <https://doi.org/10.1016/j.est.2024.112924>
- Tavakol-Moghaddam, Y., Boroushaki, M., & Astaneh, M. (2024). Reinforcement learning for battery energy management: A new balancing approach for Li-ion battery packs. *Results in Engineering*, 23, 102532. <https://doi.org/10.1016/j.rineng.2024.102532>
- Tabine, A., Laadissi, E. M., Mastouri, H., Elachhab, A., Bouzaid, S., & Hajjaji, A. (2025). A novel approach for accurate SOC estimation in Li-ion batteries in view of temperature variations. *Results in Engineering*, 25, 103962. <https://doi.org/10.1016/j.rineng.2025.103962>
- Thomas, H., & Weatherspoon, M. H. (2025). Capacity and State-of-Health Prediction of Lithium-Ion Batteries Using Reduced Equivalent Circuit Models. *Batteries*, 11(4), 162. <https://doi.org/10.3390/batteries11040162>
- Wang, Q., Wang, J., Zhao, P., Kang, J., Yan, F., & Du, C. (2017). Correlation between the model accuracy and model-based SOC estimation. *Electrochimica Acta*, 228, 146-159. <https://doi.org/10.1016/j.electacta.2017.01.057>
- Xu, C., Cleary, T., & Fathy, H. K. (2023). Improving Li-S Battery SOC Estimation using an SOC-Dependent Resistance Model. *IFAC-PapersOnLine*, 56(3), 439-444. <https://doi.org/10.1016/j.ifacol.2023.12.063>
- Zarchan, P., & Musoff, H. (2015). *Fundamentals of Kalman Filtering: A Practical Approach, 4th ed.* American Institute of Aeronautics and Astronautics, Inc.
- Zhang, X., Zhang, L., Wu, J., Bai, W., Dai, H., Lin, H., Zhang, F., & Yang, Y. (2025). SOC estimation of lithium-ion batteries using equivalent circuit model and Nyquist plots from EIS data: A machine learning approach. *Journal of Electroanalytical Chemistry*, 987, 119093. <https://doi.org/10.1016/j.jelechem.2025.119093>

Zou, C., Zhang, L., Hu, X., Wang, Z., Wik, T., & Pecht, M. (2018). A review of fractional-order techniques applied to lithium-ion batteries, lead-acid batteries, and supercapacitors. *Journal of Power Sources*, 390, 286-296.  
<https://doi.org/10.1016/j.jpowsour.2018.04.033>

## **ANEXOS**

### **CÓDIGOS DE PROGRAMACIÓN**

# Kalman\_18\_Febrero\_2026

February 18, 2026

## 1 Filtro de Kalman linealizado

```
[17]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from matplotlib.ticker import FormatStrFormatter

# Cargar datos
data = pd.read_excel("HW.xlsx") # Archivo con Current, Voltage, Time, SOC
I = data["Current"].values
y = data["Voltage"].values
t_max = len(y)

# Parámetros de batería
E_crit = 48000
V_L = 3.997
V_0 = 4.14
alpha = 0.15
beta = 17
gamma = 10.5
delta_T = 1

# Punto de operación
x_1o = 0.12 # Resistencia interna
x_2o = 0.5 # SOC
I_o = 2

# SOC real
GT_soc = data["SOC"] # Se usa columna SOC del archivo

# Inicialización matrices
nx = 2
Xk_ = np.zeros((nx, t_max))
Xk = np.zeros((nx, t_max))
Pk_ = np.zeros((nx, nx, t_max))
```

```

Pk = np.zeros((nx, nx, t_max))
Kk = np.zeros((nx, t_max))

# Condición inicial
Xk[:, 0] = [0.12, 1] # Resistencia SOC
Pk[:, :, 0] = np.array([[1e-2, 0], [0, 1e-2]])

# Funciones
def predict(A, B, Xk_prev, Uk_prev, Xo, Uo, Pk_prev, Q):
    Xk_pred = A @ (Xk_prev - Xo) + B * (Uk_prev - Uo)
    Xk_pred = np.ravel(Xk_pred) # Forzar vector 1D
    Pk_pred = A @ Pk_prev @ A.T + Q
    return Xk_pred, Pk_pred

def correct(Xk_pred, Xo, Uk, Uo, Zk, Zo, Pk_pred, H, G, R):
    Kk = Pk_pred @ H.T @ np.linalg.inv(H @ Pk_pred @ H.T + R)
    Kk = Kk.ravel() # Forzar vector 1D

    ek = Zk - (H @ (Xk_pred - Xo) + G * (Uk - Uo)) - Zo
    Xk_new = Xk_pred + Kk * ek

    I = np.eye(len(Pk_pred))
    Pk_new = (I - np.outer(Kk, H)) @ Pk_pred @ (I - np.outer(Kk, H)).T + np.
↳outer(Kk, R * Kk)

    # Saturación
    Xk_new[0] = max(0, Xk_new[0])
    Xk_new[1] = max(0, Xk_new[1])

    return Xk_new, Pk_new, Kk

# Derivadas parciales y matrices
v_o = V_L + (V_0 - V_L)*np.exp(gamma*(x_2o-1)) + alpha*V_L*(x_2o-1) \
    + (1-alpha)*V_L*(np.exp(-beta) - np.exp(-beta*np.sqrt(x_2o))) - I_o*x_1o

dv_x1 = -I_o
dv_x2 = gamma*(V_0 - V_L)*np.exp(gamma*(x_2o-1)) + alpha*V_L \
    + (1-alpha)*V_L*beta*np.exp(-beta*np.sqrt(x_2o))/(2*np.sqrt(x_2o))
dv_i = -x_1o

dx1_x1 = 1
dx1_x2 = 0
dx2_x1 = -E_crit**(-1)*delta_T*I_o*dv_x1
dx2_x2 = 1 - E_crit**(-1)*delta_T*I_o*dv_x2

A = np.array([[dx1_x1, dx1_x2],
              [dx2_x1, dx2_x2]])

```

```

B = -delta_T * E_crit**(-1) * np.array([0, I_o * dv_i + v_o]) # Vector 1D

H = np.array([[dv_x1, dv_x2]])
G = dv_i

#Varianza (o matriz de covarianza) ruidos ecuación de estados y observacion
Q = np.array([[0.001, 0], [0, 0.0002]])
R = np.array([[0.0005]])

# Filtro de Kalman
for k in range(1, t_max):
    # Predicción
    X_delta, Pk[:, :, k] = predict(A, B, Xk[:, k-1], I[k-1],
                                   np.array([x_1o, x_2o]), I_o, Pk[:, :, k-1],
    ↪Q)
    Xk[:, k] = X_delta + np.array([x_1o, x_2o - delta_T * E_crit**(-1) * v_o *
    ↪I_o])

    # Corrección
    Xk[:, k], Pk[:, :, k], Kk[:, k] = correct(Xk[:, k], np.array([x_1o, x_2o]),
                                               I[k], I_o, y[k-1], v_o,
                                               Pk[:, :, k], H, G, R)

# Resultados
SOC_e = Xk[1, 1:]
y_out = (H @ (Xk[:, 1:] - np.array([x_1o, x_2o]).reshape(-1, 1)) + G*(I[1:] -
    ↪I_o) + v_o).ravel()

# Métricas
end_idx = min(4400, len(y_out))
t_rmse = slice(300, end_idx)

mse_voltage = mean_squared_error(y[t_rmse], y_out[t_rmse])
rmse_voltage = np.sqrt(mse_voltage)

mse_soc = mean_squared_error(GT_soc[t_rmse], SOC_e[t_rmse])
rmse_soc = np.sqrt(mse_soc)

r2_soc = r2_score(GT_soc[t_rmse], SOC_e[t_rmse]) # Nuevo: R² SOC

print(f"MSE voltaje: {mse_voltage:.6f} | RMSE voltaje: {rmse_voltage:.6f}")
print(f"MSE SOC: {mse_soc:.6f} | RMSE SOC: {rmse_soc:.6f} | R² SOC: {r2_soc:.
    ↪6f}")

# GRAFICAR SOC CON ZOOM
plt.figure(figsize=(6, 3))

```

```

plt.plot(data["Time"], GT_soc, label="Real", linestyle="dashed", color="black")
plt.plot(data["Time"][1:], SOC_e, label="Estimado", color="blue")
plt.xlabel("Tiempo (s)")
plt.ylabel("SOC")
plt.title("Estimación de SOC usando LKF")
plt.legend()
plt.grid()

# Zoom
mask_zoom = (data["Time"] >= 2000) & (data["Time"] <= 2100)
mask_zoom_soc = mask_zoom[1:] # Alinear con SOC_e
ax_inset = inset_axes(
    plt.gca(),
    width="35%", height="35%", loc='lower left',
    bbox_to_anchor=(0.075, 0.15, 1, 1),
    bbox_transform=plt.gca().transAxes
)

ax_inset.plot(data["Time"][mask_zoom], GT_soc[mask_zoom], 'k--')
ax_inset.plot(data["Time"][1:][mask_zoom_soc], SOC_e[mask_zoom_soc], 'b')
ax_inset.set_xlim(2000, 2100)
ax_inset.set_ylim(0.65, 0.75)
ax_inset.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))
ax_inset.grid(True)

plt.savefig("SOC_LKF_zoom.png", dpi=300, bbox_inches="tight")
plt.show()

# GRAFICAR VOLTAJE CON ZOOM
plt.figure(figsize=(6, 3))
plt.plot(data["Time"], y, label="Real", linestyle="dashed", color="black",
        ↪linewidth = 0.5)
plt.plot(data["Time"][1:], y_out, label="Estimado", color="blue",linewidth = 0.
        ↪5)
plt.xlabel("Tiempo (s)")
plt.ylabel("Voltaje (V)")
plt.title("Estimación de voltaje usando LKF")
plt.legend()
plt.grid()

ax_inset = inset_axes(
    plt.gca(),
    width="35%", height="35%", loc='lower left',
    bbox_to_anchor=(0.05, 0.05, 1, 1),
    bbox_transform=plt.gca().transAxes
)

```

```

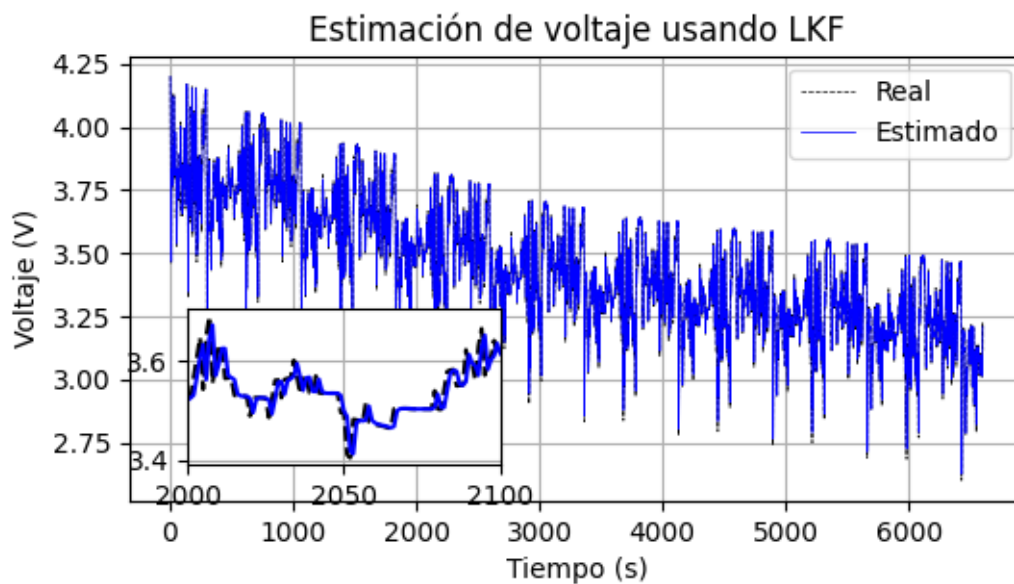
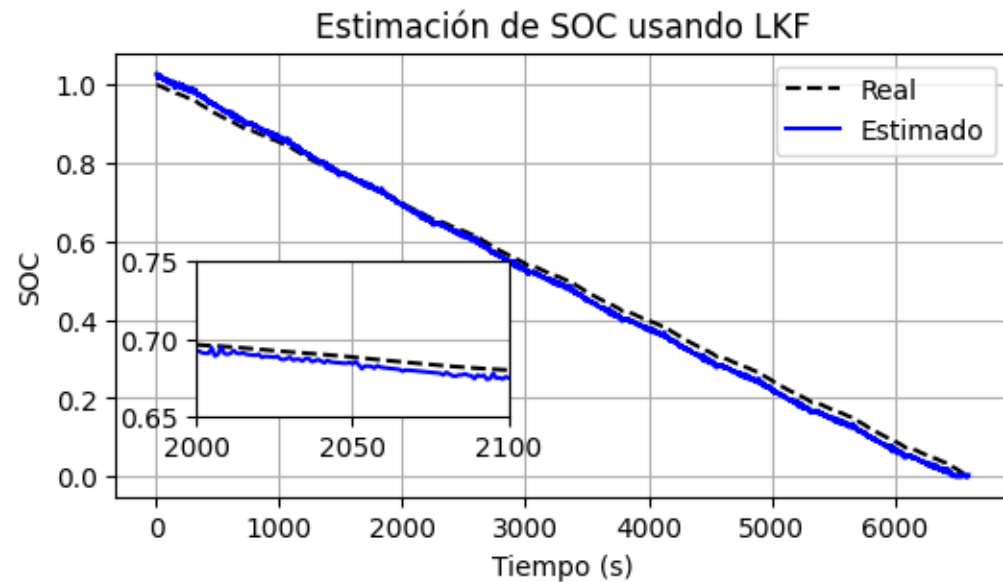
ax_inset.plot(data["Time"][mask_zoom], y[mask_zoom], 'k--')
ax_inset.plot(data["Time"][1:][mask_zoom_soc], y_out[mask_zoom_soc], 'b')
ax_inset.set_xlim(2000, 2100)
ax_inset.grid(True)

plt.savefig("Voltage_LKF_zoom.png", dpi=300, bbox_inches="tight")
plt.show()

```

MSE voltaje: 0.000051 | RMSE voltaje: 0.007115

MSE SOC: 0.000233 | RMSE SOC: 0.015274 |  $R^2$  SOC: 0.992717



# PF\_18\_Febrero\_2026

February 18, 2026

## 1 Filtro de partículas

```
[12]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from matplotlib.ticker import FormatStrFormatter

# Cargar datos
file_path = "HW.xlsx"
data = pd.read_excel(file_path)

# Parámetros de la batería =
R_internal = 0.12 # Ohm
N_particles = 300 # Partículas para el filtro

# Parámetros del modelo
alpha = 0.15
beta = 17
gamma = 10.5
v_0 = 4.14
v_L = 3.997
Ecritica = 46000 # Coulombs

# Parámetros de ruido
sigma_w1 = 0.001
sigma_w2 = 0.0001
sigma_eta = 0.001

# === Inicialización SOC ===
SOC_initial = data["SOC"].iloc[0]
particles = np.random.normal(SOC_initial, 0.02, N_particles)
particles = np.clip(particles, 0, 1)
weights = np.ones(N_particles) / N_particles

# Vector para almacenar resultados
```

```

estimated_SOC_corrected = []
estimated_voltage = []
R_dynamic_series = []

# Intervalo de tiempo
dt = np.diff(data["Time"], prepend=data["Time"].iloc[0])

# Función de resampling
def systematic_resample(weights):
    N = len(weights)
    positions = (np.arange(N) + np.random.uniform()) / N
    indexes = np.zeros(N, dtype=int)
    cumulative_sum = np.cumsum(weights)
    i, j = 0, 0
    while i < N:
        if positions[i] < cumulative_sum[j]:
            indexes[i] = j
            i += 1
        else:
            j += 1
    return indexes

# Filtro de partículas
for i in range(len(data)):
    I = data["Current"].iloc[i]
    V_meas = data["Voltage"].iloc[i]

    # Ruido
    w1 = np.random.normal(0, sigma_w1, N_particles)
    w2 = np.random.normal(0, sigma_w2, N_particles)

    # Evolución SOC
    particles = particles - V_meas * I * dt[i] * (1 / Ecritica) + w2
    particles = np.clip(particles, 0, 1)

    # Resistencia dinámica
    R_dynamic = R_internal * (1 + 0.05 * np.random.randn(N_particles)) + w1
    eta = np.random.normal(0, sigma_eta, N_particles)

    # Voltaje predicho
    V_pred = (
        v_L + (v_0 - v_L) * np.exp(gamma * (particles - 1))
        + alpha * v_L * (particles - 1)
        + (1 - alpha) * v_L * (np.exp(-beta) - np.exp(-beta * np.
↪sqrt(particles)))
        - I * R_dynamic + eta
    )

```

```

# Guardar voltaje estimado (promedio de partículas)
estimated_voltage.append(np.mean(V_pred))

# Verosimilitud
likelihood = np.exp(-((V_meas - V_pred) ** 2) / 0.0007)

# Actualizar pesos
weights *= likelihood
weights += 1e-300
weights /= np.sum(weights)

# Resampling si ESS bajo
ESS = 1.0 / np.sum(weights**2)
if ESS < N_particles / 2:
    indexes = systematic_resample(weights)
    particles = particles[indexes]
    weights = np.ones(N_particles) / N_particles

# Guardar SOC estimado
estimated_SOC_corrected.append(np.sum(particles * weights))

# guardar R_dynamic promedio (y opcionalmente su desviación)
R_dynamic_series.append(np.mean(R_dynamic))
# R_dynamic_std_series.append(np.std(R_dynamic))

# Guardar resultados en dataframe
data["SOC_estimado_correcto"] = estimated_SOC_corrected
data["Voltage_estimado"] = estimated_voltage
data["R_dynamic_prom"] = R_dynamic_series

# Métricas SOC
rmse_corrected = np.sqrt(mean_squared_error(data["SOC"],
↳data["SOC_estimado_correcto"]))
r2_corrected = r2_score(data["SOC"], data["SOC_estimado_correcto"])
mse_corrected = mean_squared_error(data["SOC"], data["SOC_estimado_correcto"])
print(f"RMSE SOC: {rmse_corrected:.6f}")
print(f"R² SOC: {r2_corrected:.6f}")
print(f"MSE SOC: {mse_corrected:.6f}")

# GRAFICAR SOC CON ZOOM
plt.figure(figsize=(6, 3))
plt.plot(data["Time"], data["SOC"], label="Real", linestyle="dashed",
↳color="black")

```

```

plt.plot(data["Time"], data["SOC_estimado_correcto"], label="Estimación PF",
         color="blue")
plt.xlabel("Tiempo (s)")
plt.ylabel("SOC")
plt.title("Estimación de SOC usando PF")
plt.legend()
plt.grid()

# Inset para zoom
ax_inset = inset_axes(
    plt.gca(),
    width="35%", height="35%", loc='lower left',
    bbox_to_anchor=(0.075, 0.15, 1, 1),
    bbox_transform=plt.gca().transAxes
)
mask_zoom = (data["Time"] >= 2000) & (data["Time"] <= 2100)
ax_inset.plot(data["Time"][mask_zoom], data["SOC"][mask_zoom], 'k--')
ax_inset.plot(data["Time"][mask_zoom],
              data["SOC_estimado_correcto"][mask_zoom], 'b')
ax_inset.set_xlim(2000, 2100)
ax_inset.set_ylim(0.65, 0.75)
ax_inset.yaxis.set_major_formatter(FormatStrFormatter('%.2f'))
ax_inset.grid(True)

plt.tight_layout()
plt.savefig("SOC_HW_FP_zoom.png", dpi=300, bbox_inches="tight")
plt.show()

# GRAFICAR VOLTAJE CON ZOOM
plt.figure(figsize=(6, 3))
plt.plot(data["Time"], data["Voltage"], label="Real", linestyle="dashed",
         color="black", linewidth = 0.5)
plt.plot(data["Time"], data["Voltage_estimado"], label="Estimado",
         color="blue", linewidth = 0.5)
plt.xlabel("Tiempo (s)")
plt.ylabel("Voltaje (V)")
#plt.ylim(2, 4.25)
plt.title("Estimación de voltaje usando PF")
plt.legend()
plt.grid()

# zoom en voltaje
ax_inset = inset_axes(
    plt.gca(),
    width="35%", height="35%", loc='lower left',
    bbox_to_anchor=(0.05, 0.05, 0.9, 0.9),

```

```

    bbox_transform=plt.gca().transAxes
)
mask_zoom = (data["Time"] >= 2000) & (data["Time"] <= 2100)
ax_inset.plot(data["Time"][mask_zoom], data["Voltage"][mask_zoom], 'k--')
ax_inset.plot(data["Time"][mask_zoom], data["Voltage_estimado"][mask_zoom], 'b')
ax_inset.set_xlim(2000, 2100)
ax_inset.grid(True)

plt.savefig("Voltage_HW_FP_zoom.png", dpi=300, bbox_inches="tight")
plt.show()

# GRAFICA evolución de Resistencia
plt.figure(figsize=(6, 3))
plt.plot(data["Time"], data["R_dynamic_prom"], linewidth = 0.5)
plt.xlabel("Tiempo (s)")
plt.ylabel("$x_1 \ (\Omega)$")
plt.title("Evolución artificial de la resistencia con PF")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("R_dynamic_evolution.png", dpi=300, bbox_inches="tight")
plt.show()

# (Opcional) Guardar resultados a CSV
# data.to_csv("PF_results_with_Rdynamic.csv", index=False)

```

RMSE SOC: 0.015236

R<sup>2</sup> SOC: 0.997209

MSE SOC: 0.000232

C:\Users\edwin\AppData\Local\Temp\ipykernel\_15944\3305551702.py:148:

UserWarning: This figure includes Axes that are not compatible with tight\_layout, so results might be incorrect.

```
plt.tight_layout()
```

# SOC\_voltage - 18feb

February 18, 2026

## 1 RANDOM FOREST

```
[8]: import os
import re
import numpy as np
import pandas as pd
from scipy.optimize import minimize
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt

# RUTA
folder = r"C:\Users\edwin\Desktop\SOC_11_measures\Model"
param_list = ["Re", "Cd1", "Qd1", "ndl", "Rct", "L", "Voltage"]

# FUNCIONES PARA AJUSTE DEL MODELO ECM
def Z_CPE(Q, p, omega):
    Q = max(Q, 1e-12)
    omega = np.where(omega == 0, 1e-6, omega)
    return 1 / (Q * (1j * omega) ** p)

def impedance_model(x, omega):
    Re, Cd1, Qd1, ndl, Rct, L = x
    Z_cd1 = 1 / (1j * omega * Cd1)
    Z_cpe = 1 / (Qd1 * (1j * omega) ** ndl)
    Z_parallel = 1 / (1 / Z_cpe + 1 / Rct)
    Z_total = Re + Z_cd1 + Z_parallel + 1j * omega * L
    return Z_total

def objective_function(x, omega, Z_data):
    Z_model = impedance_model(x, omega)
    return np.linalg.norm(np.concatenate([
        np.real(Z_data - Z_model),
        np.imag(Z_data - Z_model)
    ]))
```

```

def extract_ecm_params(frequencies, Z_vector):
    omega = 2 * np.pi * frequencies
    Re = np.real(Z_vector[-1])
    Rct = 0.1
    Cd1 = 1e-6
    Qd1 = 1e-4
    nd1 = 0.5
    L = 1e-8

    x0 = [Re, Cd1, Qd1, nd1, Rct, L]
    bounds = [
        (0, 0.1),
        (1e-12, 1e-1),
        (1e-12, 1e2),
        (0, 1),
        (1e-12, 0.1),
        (0, 1)
    ]

    result = minimize(objective_function, x0, args=(omega, Z_vector),
                      bounds=bounds, method='L-BFGS-B')
    return result.x

# EXTRAER DEL SOC DESDE EL NOMBRE DEL ARCHIVO
def extract_soc_from_filename(filename):
    match = re.search(r'SoC_(\d+)', filename)
    return int(match.group(1)) / 100 if match else None # Escala 0-1

# PROCESAMIENTO DE ARCHIVOS
files = [f for f in os.listdir(folder) if f.endswith(".csv")]
data = []

for file in files:
    soc = extract_soc_from_filename(file)
    if soc is None:
        continue
    path = os.path.join(folder, file)
    df = pd.read_csv(path)

    freq = df["Frequency(Hz)"].values
    Z = df["R(ohm)"].values + 1j * df["X(ohm)"].values
    params = extract_ecm_params(freq, Z)
    voltage_mean = df["V(V)"].mean()
    data.append([file, soc] + list(params) + [voltage_mean])

df_params = pd.DataFrame(data, columns=["File", "SOC"] + param_list)

```

```

# Variables de entrada (incluye voltaje)
X = df_params[param_list].values
y = df_params["SOC"].values # Escala 0-1

# VALIDACIÓN CRUZADA
kf = KFold(n_splits=5, shuffle=True, random_state=42)
y_real_all = []
y_pred_all = []

for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    y_real_all.extend(y_test)
    y_pred_all.extend(y_pred)

# MÉTRICAS
mae = mean_absolute_error(y_real_all, y_pred_all)
rmse = np.sqrt(mean_squared_error(y_real_all, y_pred_all))
r2 = r2_score(y_real_all, y_pred_all)

print("=== MÉTRICAS PROMEDIO K-FOLD (ESCALA 0-1) ===")
print(f"MAE : {mae:.4f}")
print(f"RMSE : {rmse:.4f}")
print(f"R2 : {r2:.4f}")

# GRÁFICA
plt.figure(figsize=(6, 4))
plt.plot(y_real_all, 'o-', label='Real', color='black')
plt.plot(y_pred_all, 'x--', label='Predicho', color='blue')
plt.xlabel("Muestra", fontsize=12)
plt.ylabel("SOC", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Predicción de SOC (Random Forest)")
plt.legend()
plt.grid(True)
plt.xticks(np.arange(0, len(y_real_all), 5))
plt.tight_layout()
plt.savefig("SOC_V_RF.png", dpi=300, bbox_inches='tight')
plt.show()

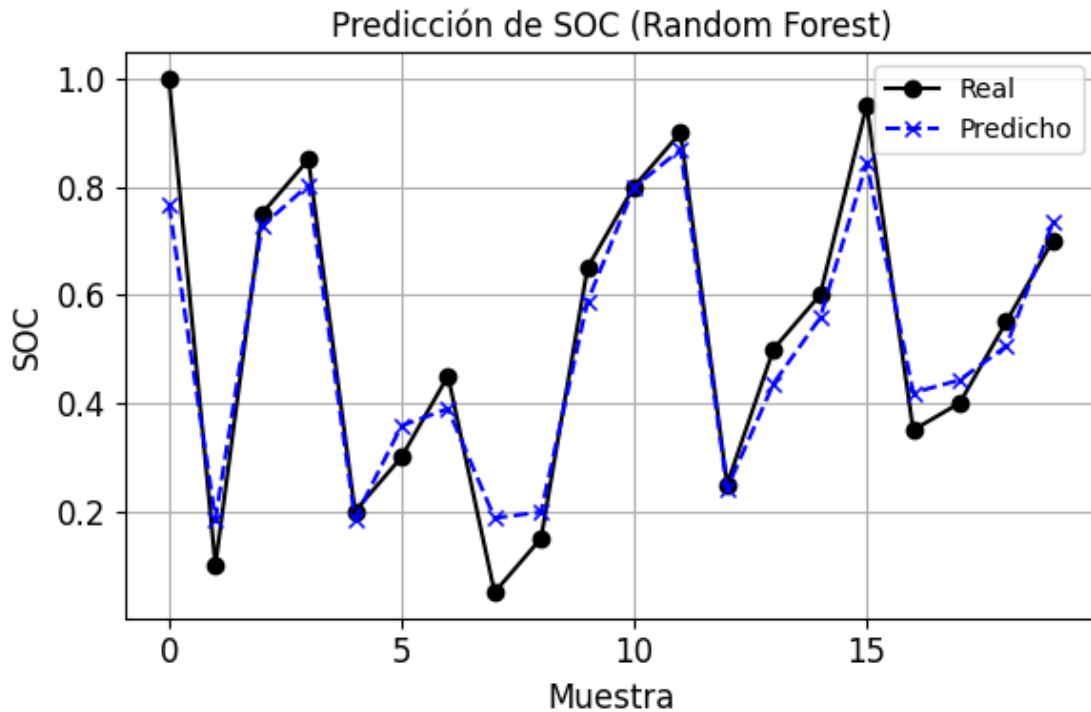
```

```

=== MÉTRICAS PROMEDIO K-FOLD (ESCALA 0-1) ===
MAE : 0.0605

```

RMSE : 0.0788  
R<sup>2</sup> : 0.9252



### 1.1 Extraer parámetros

```
[7]: df_params.loc[df_params["SOC"] == 0.6, ["Re", "Cd1", "Qd1", "ndl", "Rct", "L", "Voltage"]]
```

```
[7]:
```

	Re	Cd1	Qd1	ndl	Rct	L	Voltage
12	0.059283	0.1	0.000103	0.5	0.1	0.000002	3.293015

## 2 KNN

```
[9]: import os
import re
import numpy as np
import pandas as pd
from scipy.optimize import minimize
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import KFold
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

```

# RUTA
folder = r"C:\Users\edwin\Desktop\SOC_11_measures\Model" # Cambia esta ruta
↳según tus archivos
param_list = ["Re", "Cdl", "Qdl", "ndl", "Rct", "L", "Voltage"]

# FUNCIONES PARA AJUSTE DEL MODELO ECM
def Z_CPE(Q, p, omega):
    Q = max(Q, 1e-12)
    omega = np.where(omega == 0, 1e-6, omega)
    return 1 / (Q * (1j * omega) ** p)

def impedance_model(x, omega):
    Re, Cdl, Qdl, ndl, Rct, L = x
    Z_cdl = 1 / (1j * omega * Cdl)
    Z_cpe = 1 / (Qdl * (1j * omega) ** ndl)
    Z_parallel = 1 / (1 / Z_cpe + 1 / Rct)
    Z_total = Re + Z_cdl + Z_parallel + 1j * omega * L
    return Z_total

def objective_function(x, omega, Z_data):
    Z_model = impedance_model(x, omega)
    return np.linalg.norm(np.concatenate([
        np.real(Z_data - Z_model),
        np.imag(Z_data - Z_model)
    ]))

def extract_ecm_params(frequencies, Z_vector):
    omega = 2 * np.pi * frequencies

    # Inicialización usando criterio de alta frecuencia
    Re = np.real(Z_vector[-1]) # high-frequency intercept
    Rct = 0.01
    Cdl = 1e-8
    Qdl = 1e-5
    ndl = 0.5
    L = 1e-8

    x0 = [Re, Cdl, Qdl, ndl, Rct, L]
    bounds = [
        (0, 0.1), # Re
        (1e-12, 1e-1), # Cdl
        (1e-12, 1e3), # Qdl
        (0, 1), # ndl
        (0, 1), # Rct
        (0, 1) # L
    ]

```

```

result = minimize(objective_function, x0, args=(omega, Z_vector),
                  bounds=bounds, method='L-BFGS-B')
return result.x

# EXTRAER SOC DESDE EL NOMBRE DEL ARCHIVO
def extract_soc_from_filename(filename):
    match = re.search(r'SoC_(\d+)', filename)
    return int(match.group(1)) / 100 if match else None # Escala 0-1

# PROCESAMIENTO
files = [f for f in os.listdir(folder) if f.endswith(".csv")]
data = []

for file in files:
    soc = extract_soc_from_filename(file)
    if soc is None:
        continue
    path = os.path.join(folder, file)
    df = pd.read_csv(path)

    # Extraer frecuencias e impedancia
    freq = df["Frequency(Hz)"].values
    Z = df["R(ohm)"].values + 1j * df["X(ohm)"].values
    params = extract_ecm_params(freq, Z)

    # Calcular voltaje promedio
    voltage_mean = df["V(V)"].mean()

    # Agregar parámetros ECM y voltaje
    data.append([file, soc] + list(params) + [voltage_mean])

# Crear DataFrame
df_params = pd.DataFrame(data, columns=["File", "SOC"] + param_list)
X = df_params[param_list].values
y = df_params["SOC"].values # Escala 0-1

# VALIDACIÓN CRUZADA
kf = KFold(n_splits=5, shuffle=True, random_state=42)
y_real_all = []
y_pred_all = []

# Configurar modelo KNN
model_config = KNeighborsRegressor(n_neighbors=2, weights='distance')

for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

```

```

model = model_config
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

y_real_all.extend(y_test)
y_pred_all.extend(y_pred)

# MÉTRICAS
mae = mean_absolute_error(y_real_all, y_pred_all)
rmse = np.sqrt(mean_squared_error(y_real_all, y_pred_all))
r2 = r2_score(y_real_all, y_pred_all)

print("=== MÉTRICAS PROMEDIO K-FOLD (KNN + Voltaje, ESCALA 0-1) ===")
print(f"MAE : {mae:.4f}")
print(f"RMSE : {rmse:.4f}")
print(f"R2 : {r2:.4f}")

```

```

=== MÉTRICAS PROMEDIO K-FOLD (KNN + Voltaje, ESCALA 0-1) ===
MAE : 0.0494
RMSE : 0.0618
R2 : 0.9540

```

```

[10]: # GRÁFICA
plt.figure(figsize=(6, 4))
plt.plot(y_real_all, 'o-', label='Real', color='black')
plt.plot(y_pred_all, 'x--', label='Predicho', color='blue')
plt.xlabel("Muestra", fontsize=12)
plt.ylabel("SOC", fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Predicción SOC (KNN)")
plt.legend()
plt.grid(True)
plt.xticks(np.arange(0, len(y_real_all), 5))

plt.tight_layout()
plt.savefig("SOC_V_KNN.png", dpi=300, bbox_inches='tight')
plt.show()

```