

Desarrollo de software de control para sistema BMS embarcado en vehículo eléctrico mediante el empleo de herramientas de diseño basado en modelos

Ane Gelbentzu^{1 2}, Jon Romero¹, Alfredo Rubio², Eneko Otaola², Elena Trancho², Edorta Ibarra¹

¹Universidad del País Vasco (UPV/EHU), Ingeniero Torres Quevedo 1, 48013 Bilbao, España

²TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, España

Abstract—Los sistemas electrónicos de gestión de baterías (BMS, *Battery Management System*, en inglés) son fundamentales en aplicaciones avanzadas que requieren elevados niveles de seguridad, como por ejemplo sistemas de generación de energía eléctrica renovables y vehículos eléctricos. En particular, la fiabilidad del software de control del BMS resulta de extrema importancia. En este trabajo se emplea el diseño basado en modelos (MBD, *Model Based Design*, en inglés), de forma que se facilita el desarrollo de un software fiable y orientado al cumplimiento de niveles de seguridad automotrices ASIL (*Automotive Safety Integrity Level*, en inglés). Se describen los principales bloques funcionales desarrollados y se presentan resultados de simulación y experimentales que demuestran la correcta implementación de los mismos.

Index Terms—Sistemas de almacenamiento de energía, baterías de Litio-Ion, BMS, EKF, MBD

I. INTRODUCCIÓN

En la actualidad, los sistemas de almacenamiento de energía eléctrica basados en baterías electroquímicas son tecnologías relevantes para múltiples aplicaciones, tales como sistemas de generación de energía renovables [1] y transporte electrificado [2], [3]. Aunque existen varias tecnologías de baterías, las de Ión de Litio (Li-ion) son las más populares debido a su alta densidad de energía, alta eficiencia coulombica, ciclo de vida prolongado, bajo ratio de auto-descarga, y por estar libres de efectos de memoria [3].

Cuando varias celdas de Li-ion se agrupan, formando un pack de baterías, y éstas se emplean en aplicaciones avanzadas, es necesario disponer de un sistema electrónico de gestión (BMS, *Battery Management System*, en inglés) que monitorice el estado de carga (SoC, *State of Charge*, en inglés) [4] y el estado de salud (SoH, *State of Health*, en inglés) [1], [4], se ocupe de la conexión/desconexión de estas con otros subsistemas mediante contactores, gestione el proceso de carga, controle el equilibrado de las celdas y supervise los distintos parámetros del sistema (tensiones, corrientes y temperaturas), actuando en consecuencia ante situaciones de alarma y llevando al sistema, cuando sea necesario, a modo seguro [5].

Para poder lograr los niveles de seguridad y fiabilidad requeridos por una determinada aplicación – por ejemplo, niveles ASIL (*Automotive Safety Integrity Level*, en inglés) C o D en el contexto de la norma ISO 26262 aplicada a vehículos eléctricos [6] – es indispensable que el software del sistema BMS cumpla con unos estándares funcionales muy

estrictos. En este contexto, este trabajo desarrolla el software de control para un sistema BMS automotriz mediante la utilización de herramientas de diseño basado en modelos (MBD, *Model Based Design*, en inglés). Con el paradigma MBD, siguiendo el ciclo-en-V para el testeo del sistema de control, es posible reducir significativamente el tiempo de desarrollo [7]–[9]. Por definición, la aproximación MDB utiliza un modelo del sistema como un ejecutable de especificaciones durante el desarrollo. Esta forma de trabajo soporta el diseño y la simulación, tanto a nivel de sistema como de componente, permitiendo un testeo y una verificación continua durante todo el proceso [10].

A continuación, se detallan los aspectos de desarrollo e implementación más importantes del software BMS creado por los autores mediante el paradigma MBD. En particular, se describen la máquina de estados del sistema y los distintos procesos y secuencias que intervienen en el control. Se presta especial atención a los algoritmos de estimación de SoC y SoH implementados. Finalmente, se presentan resultados de simulación y experimentales que demuestran el correcto funcionamiento de la propuesta.

II. DESCRIPCIÓN DEL SISTEMA DE BATERÍAS

II-A. Esquema general del sistema eléctrico

La Fig. 1 muestra el circuito eléctrico equivalente del sistema bajo estudio. Cuenta con un pack de baterías en configuración 2P12S (2 celdas en paralelo, 12 en serie). Cada celda tiene una tensión nominal de 4,1 V para un SoC del 80%. Por lo tanto, el pack completo tiene una tensión nominal aproximada de 50 V. La capacidad por celda es de 8 A/h.

El pack de baterías se conecta al resto de subsistemas mediante cuatro pares de relés electromecánicos (Fig. 1):

- Contactores de precarga $S_{pch,j}$, $j \in \{1, 2\}$. Realiza la precarga de los condensadores del *drive* y del cargador a través de la resistencia de precarga R_{pch} .
- Contactores principales $S_{p,j}$ del terminal positivo.
- Contactores $S_{chg,j}$ en la rama negativa para la conexión y desconexión del cargador.
- Contactores $S_{load,j}$ en la rama negativa para la conexión y desconexión de la carga o inversor de tracción.

La configuración de contactores con dos relés colocados en sentido opuesto permite su apertura segura en caso de falta en función del sentido de la corriente [11]. El sistema también

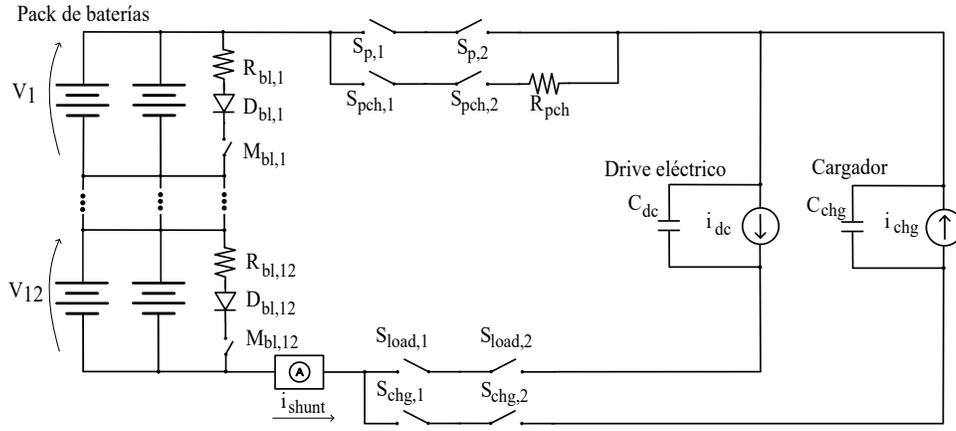


Fig. 1: Esquema general del circuito eléctrico equivalente del sistema bajo estudio, con un pack de baterías en configuración 2P12S.

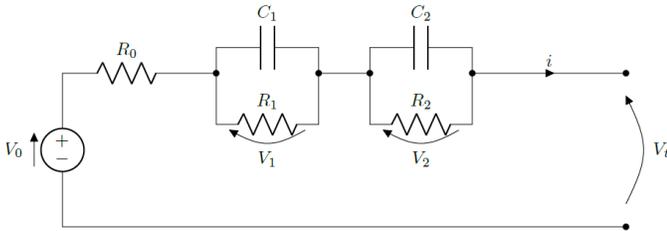


Fig. 2: Circuito eléctrico equivalente para una celda de batería Litio-Ión.

dispone de un circuito de equilibrado pasivo [5], [12]. La energía de las celdas con un valor de tensión que supere un determinado umbral se descargan a través de las resistencias $R_{bl,j}$, al activar los correspondientes MOSFETs $M_{bl,j}$. Los diodos $D_{bl,j}$ garantizan el sentido de la corriente de descarga.

II-B. Modelo equivalente de una celda de Li-ion

Para desarrollar el modelo anteriormente descrito como ejecutable de especificaciones, se han utilizado las librerías *Simscape Electrical* y *Simscape Battery* de Matlab/Simulink. Esta última permite modelar cada celda de Li-ion mediante un circuito equivalente Randle de segundo orden (Fig. 2) [13], donde $V_0(SoC, T)$ es la tensión de circuito abierto (OCV, *Open Circuit Voltage*, en inglés), $R_0(SoC, T)$ es la resistencia interna y $R_j(SoC, T)$ y $C_j(SoC, T)$ son las resistencias y capacidades de los circuitos tanque, con $j = \{1, 2\}$. Estos parámetros y sus relaciones se han extraído experimentalmente y se han almacenado en formato de tabla (LUT, *Look-up Table*, en inglés), siguiendo la metodología presentada en [14].

La dinámica de las tensiones de los circuitos tanque para un instante discreto k se define como:

$$V_j(k+1) = e^{-\frac{T_s}{\tau_j}} \cdot V_j(k) + R_j \cdot (1 - e^{-\frac{T_s}{\tau_j}}) \cdot i(k), \quad (1)$$

donde $\tau_j = R_j C_j$ y $j = \{1, 2\}$.

III. DESCRIPCIÓN GENERAL DEL SOFTWARE BMS

III-A. Máquina de estados principal

Siguiendo el paradigma MBD, la máquina de estados principal se ha desarrollado mediante la *toolbox Stateflow* de

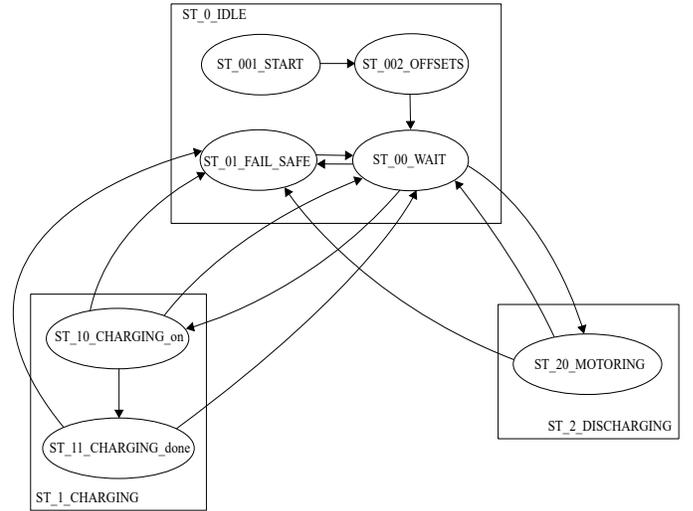


Fig. 3: Diagrama de la máquina de estados principal.

Matlab/Simulink. Esta se compone de tres estados (Fig. 3): *idle* (inactivo), *charging* (modo de carga) y *discharging* (modo de tracción). En el estado *idle* se distinguen los subestados de calibración automática de sensores, espera y modo seguro (*fail safe*). A este último estado se accede cuando se detecta una alarma crítica y se realiza una apertura de contactores para preservar la integridad del sistema hasta que el fallo se resuelva. Nótese que, en la configuración de máquina de estados propuesta, no está permitido pasar del estado *charging* a *discharging*, y viceversa, sin pasar por el estado *idle*, aumentando así el nivel de seguridad.

III-B. Secuencias

El software de control realiza las tareas necesarias para pasar de un estado a otro mediante secuencias. Estas son máquinas de estado secundarias que permiten, de manera organizada, realizar una serie de acciones. Algunas de las tareas de las que se encargan dichas secuencias son: la apertura y cierre de los contactores, el equilibrado de celdas, e incluso la calibración de varios parámetros iniciales del sistema.

A modo de ejemplo, la Fig. 4 muestra la secuencia que

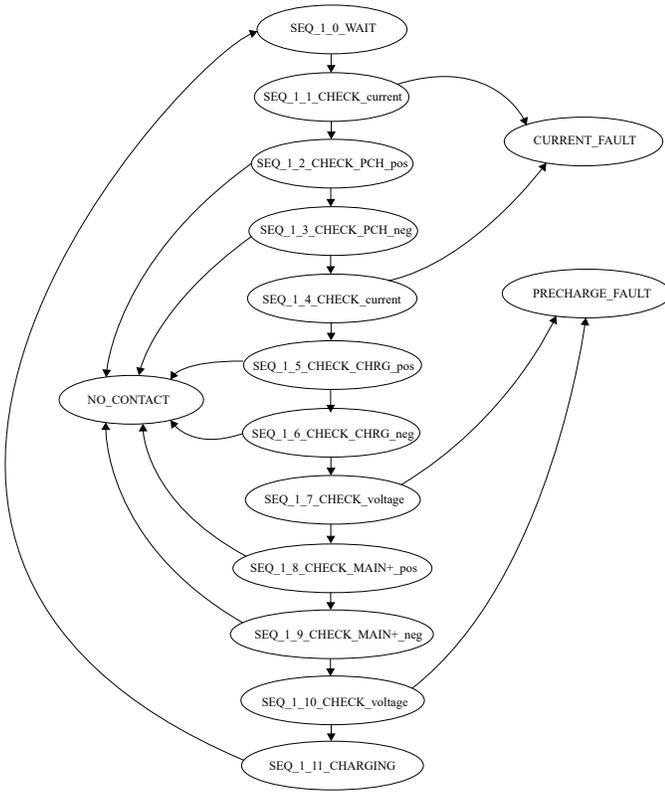


Fig. 4: Diagrama de flujo de la secuencia de cambio del subestado de espera al estado de carga.

permite pasar del subestado de espera al estado de carga. Cuenta con 15 estados, donde se realizan varias comprobaciones, garantizando que la transición se realiza de forma segura. En caso de falta, la secuencia se interrumpe y el sistema salta al modo seguro (*fail safe*).

III-C. Seguridad y gestión de alarmas

En una BMS la seguridad es fundamental. Por ello, se ha realizado un sistema de gestión de alarmas con bloques estándar de Matlab/Simulink. Las alarmas se dividen en dos niveles de seguridad, representadas mediante dos colores: rojo y ámbar. Las representadas en rojo implican que el sistema detenga cualquier secuencia o función que esté ejecutando y que pase al modo seguro. Se monitorizan variables eléctricas (tensiones y corrientes), estado de conectores y contactores y variables térmicas (temperaturas de celda y de PCB del BMS). En varios casos, los umbrales de alarma son variables, en función del modo de operación (*charging* o *discharging*). Las alarmas se registran y son reseteables por software, lo que permite rearmar el sistema.

Por su parte, las alarmas de tipo ámbar únicamente indican avisos, y se dan en situaciones donde algún parámetro sobrepase un umbral que no represente un riesgo inmediato para el funcionamiento y la integridad del sistema. Esta información se puede enviar a otras unidades de control (ECU, *Electronic Control Unit*, en inglés) del vehículo para que, por ejemplo, el sistema de tracción limite la demanda de potencia.

IV. ALGORITMOS PARA LA ESTIMACIÓN DEL SOC Y SOH

IV-A. Estimación SoC mediante cuenta de Coulomb

La forma más simple para estimar el SoC consiste en integrar la corriente i a lo largo del tiempo. En su forma discreta, despreciando el término de eficiencia coulombica η , el SoC en el instante k se calcula como:

$$SoC(k) = SoC(k-1) + \frac{i(k)}{Q} T_s, \quad (2)$$

donde Q es la carga de la celda/batería y T_s es el periodo de muestreo del sistema de control.

Se trata de una solución simple y con una carga computacional baja. Por ello, es una técnica ampliamente utilizada en la industria [5]. Sin embargo, depende fuertemente de la determinación de las condiciones iniciales [$SoC(0)$] y acumula el error asociado a la integración de la corriente. Por ello, se trata de una técnica aplicable a sistemas en los que, con cierta periodicidad, puede eliminarse dicho error al llegarse periódicamente a condiciones en las que es posible determinar el SoC, de manera precisa, por otros métodos (por ej. a través de la tensión OCV) al alcanzar las condiciones de finalización de carga o descarga [15].

IV-B. Algoritmo SoC basado en EKF para celda Li-ion

Considerando las desventajas de la cuenta de Coulomb, en la literatura se han investigado diversas técnicas avanzadas de estimación basadas en redes neuronales [16], lógica difusa [17], modelos [17] y filtros extendidos de Kalman (EKF, *Extended Kalman Filter*, en inglés) [13], [17]–[19], entre otros. Estos últimos son de gran interés, ya que se caracterizan por lograr una gran precisión en la estimación con una carga computacional relativamente moderada.

El algoritmo EKF se ejecuta en varias fases o pasos. En la primera de ellas se debe inicializar el vector $\hat{x}(0|0)$, donde $x = [SoC, V_1, V_2, R_0, 1/C_{bat}]^T$ es el vector de estados del filtro de Kalman, y $1/C_{bat}$ es la inversa de la capacidad de la batería. También se debe inicializar la matriz de covarianza $\hat{P}(0|0)$ de 5×5 . Los elementos de la diagonal de P representan la covarianza de los errores asociados a los elementos de x . Por ejemplo, P_{11} es la covarianza asociada al error en la estimación del SoC .

Posteriormente, la fase de predicción se realiza a priori:

$$\hat{x}(k+1|k) = F \cdot \hat{x}(k|k) + G \cdot i, \quad (3)$$

donde F es:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & -\frac{i \cdot T_s}{3600} \\ 0 & e^{-\frac{T_s}{\tau_1}} & 0 & 0 & 0 \\ 0 & 0 & e^{-\frac{T_s}{\tau_2}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

y G es la matriz de control, la cual se deriva del circuito eléctrico equivalente (Fig. 2):

$$G = \begin{bmatrix} 0 & R_1(1 - e^{-\frac{T_s}{\tau_1}}) & R_2(1 - e^{-\frac{T_s}{\tau_2}}) & 0 & 0 \end{bmatrix}^T. \quad (5)$$

A continuación se determina el error de la estimación. Para ello, en primer lugar se calcula su covarianza:

$$\hat{P}(k+1|k) = F(k) \cdot \hat{P}(k, k) \cdot F^T + Q, \quad (6)$$

donde la matriz Q representa la covarianza en el ruido de los estados (los errores del modelo o desviaciones de parámetros no se pueden medir, por lo que su valor se puede ajustar mediante herramientas de optimización, por ejemplo). Una vez realizado lo anterior, se debe corregir la predicción calculada. Para ello, se determina la ganancia de Kalman K :

$$K(k+1) = \hat{P}(k+1|k) \cdot H^T(k) \cdot [H(k) \cdot \hat{P}(k+1|k) \cdot H^T(k) + R]^{-1}, \quad (7)$$

donde H es el vector de observación y R es la covarianza del ruido de medidas (al igual que Q , esta tampoco se puede medir y se debe estimar). H se emplea para determinar las tensiones entre bornas V_i :

$$H = \left[\frac{\partial V_0}{\partial SOC} \quad -1 \quad -1 \quad i \quad 0 \right], \quad (8)$$

donde

$$h(x, i) = V_0(SOC, T) - i \cdot R_0 - V_1 - V_2. \quad (9)$$

Una vez calculada K , es posible realizar una estimación a posteriori en (10). A partir de (9) se calcula la tensión entre bornas, y ésta se compara con la tensión medida físicamente en la batería.

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1) \cdot [V_i(k) - h(\hat{x}(k|k), i)]. \quad (10)$$

Para terminar con el proceso, se estima P a posteriori:

$$\hat{P}(k+1|k+1) = [I - K(k+1) \cdot H] \cdot \hat{P}(k+1|k). \quad (11)$$

IV-C. Propuesta para la estimación del SoC general del pack de baterías

En este trabajo se propone ejecutar en paralelo un algoritmo por cuenta de Coulomb (apartado IV-A) y otro basado en EKF (apartado IV-B), de forma que la información obtenida por ambas se envíe, de forma complementaria, a la ECU (*Electronic Control Unit*, en inglés) central del vehículo para que esta las procese.

En relación al algoritmo EKF, en un pack de baterías con un elevado número de celdas no es práctico utilizar un bloque EKF para cada una de las celdas que la componen, ya que la carga computacional resultante puede ser excesiva. Por ejemplo, en una configuración 2P12S se requerirían 12 estimadores EKF ejecutándose simultáneamente. Por ello, en este trabajo se propone utilizar un único EKF que se alimenta con el valor medio de las tensiones de las celdas en serie:

$$V_{eq} = \langle V_i \rangle = \frac{1}{S} \sum_{i=0}^S V_s. \quad (12)$$

Por su parte, se propone que el estado individual de las celdas y sus correspondientes desequilibrios se estimen de forma simplificada. A partir de los valores promedio estimados por el algoritmo EKF principal se determinan las variaciones por celda mediante la implementación varios algoritmos *delta filtering* en paralelo [20].

IV-D. Estimación del SoH a partir del EKF

Es también importante conocer el estado de salud o SoH del sistema para su mantenimiento predictivo. En este contexto, el algoritmo EKF desarrollado permite la estimación de otra serie de parámetros internos de interés, además del SoC, tales como la resistencia interna R_0 y la inversa de la capacidad $1/C_{bat}$ (véase el vector \hat{x} definido en la sección IV-B).

Por lo general, a medida que la batería se degrada, los procesos de envejecimiento dan como resultado tanto una pérdida de capacidad de almacenamiento de energía como un aumento del valor de la resistencia interna R_0 , estando la evolución de ambos parámetros correlacionada [21]. Por ello, es posible utilizar la siguiente definición, dependiente de R_0 , para determinar el SoH del pack de baterías:

$$SoH(k) = \frac{R_{EOL}(T) - R_0(k)}{R_{EOL}(T) - R_{new}(T)}, \quad (13)$$

donde R_0 es el valor de la resistencia interna estimado por el algoritmo EKF y $R_{new}(T)$ y $R_{EOL}(T)$ son los valores de resistencia interna (en función de la temperatura) esperados cuando las celdas son nuevas o han llegado al final de su ciclo de vida (EOL, *End of Life*, en inglés), respectivamente.

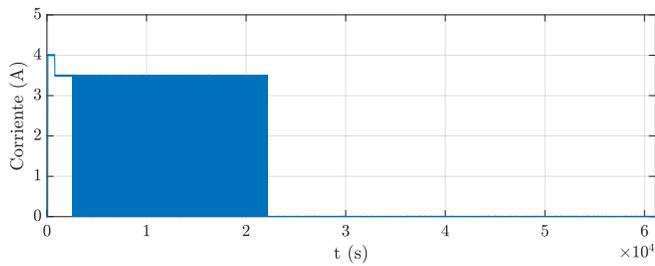
Alternativamente, es posible utilizar un indicador SoH basado en el valor estimado de C_{bat} , comparándolo con valores nominales, de EOL o técnicas más avanzadas [5], [22]. Finalmente, cabe destacar que también es posible ejecutar ambos cálculos del SoH en paralelo (a partir de R_0 y C_{bat}) y disponer de información complementaria que ayude a una determinación aún más precisa.

V. RESULTADOS DE SIMULACIÓN Y EXPERIMENTALES

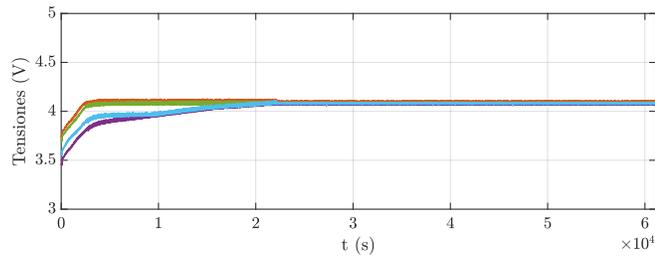
V-A. Verificación funcional del sistema BMS

Siguiendo el paradigma MBD, se ha realizado una extensiva campaña de ensayos para la verificación funcional del sistema en Matlab/Simulink, considerando todos los casos de interés (cambios de estado, situaciones de alarma, procesos de carga/descarga, etc.). El algoritmo de control desarrollado se ha simulado contra el modelo de planta (como ejecutable de especificaciones). Tanto el modelo de planta como el sistema de control se ejecutan con un periodo de 200 ms. Se ha incluido ruido bayesiano en las medidas de tensión y corriente para simular unas condiciones de operación más realistas.

Por tratarse de una de las situaciones más relevantes, la Fig. 5 muestra una simulación correspondiente al proceso de carga, considerando una cierta variación en las condiciones iniciales del SoC de cada una de las celdas. El umbral de tensión de carga se ha fijado a $V_{th} = 4,1$ V (tensión nominal de las celdas con un SoC del 80%). La banda de tensión entre celdas objetivo para la finalización del proceso de carga se ha establecido con una tolerancia de 10 mV. La corriente de carga nominal se ha fijado a 4 A, y la corriente reducida (a aplicar durante el proceso de equilibrado) se ha fijado a 3.5 A [Fig. 5(b)]. El algoritmo desarrollado realiza la carga y el equilibrado de las tensiones por celda. Tal y como muestran los resultados, el proceso de carga se ha completado en 5 horas y 53 minutos. Al finalizar la carga, las tensiones de las 12 celdas se encuentran dentro de la banda de tolerancia previamente determinada.



(a) Corriente de carga (I_{shunt}).



(b) Tensiones por celda ($V_{cell,j}$, $j \in \{1, \dots, 12\}$).

Fig. 5: Proceso de carga con celdas inicialmente desequilibradas.

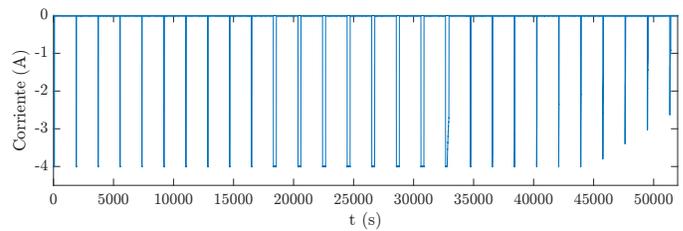
V-B. Estimador EKF

Nuevamente, siguiendo el procedimiento MBD, el algoritmo EKF desarrollado se ha verificado inicialmente en Matlab/Simulink para una única celda. Se ha aplicado un ciclo de descarga con pulsos de 4 A de duración variable [Fig. 6(a)]. En estas condiciones, la tensión de salida de la celda decrece [Fig. 6(b)] mientras que estimación del SoC sigue correctamente el valor real del modelo [Fig. 6(c)]. Para cuantificar la precisión de la estimación se ha determinado el error absoluto medio (MAE, *Mean Absolute Error*, en inglés) entre el SoC real y el estimado, y se ha logrado un valor del 0.3 %, lo cual se considera satisfactorio. Posteriormente y antes de realizar la validación experimental, se ha realizado una simulación PiL (*Processor in the Loop*). A partir de los modelos desarrollados y previamente validados en el entorno Matlab/Simulink, se ha generado código de forma automática. De este modo, se han minimizado posibles errores de codificación y se ha acelerado el proceso de desarrollo. El código se ha ejecutando en una tarjeta RD33771CDSTEVB de NXP. En modo PiL se han logrado resultados análogos a los mostrados por la Fig. 6.

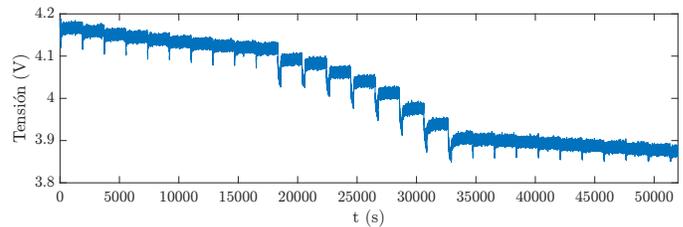
Finalmente, el algoritmo se ha validado experimentalmente utilizando el pack de baterías de la figura 7. La plataforma de evaluación para BMS de NXP utilizado cuenta con un AFE (*Analog Front End*, en inglés) MC33771C, además de la anteriormente mencionada tarjeta RD33771CDSTEVB. Tal y como se ha expuesto en el apartado IV-C, al trabajar con el pack completo, el EKF recibe la tensión promedio. Se han aplicado los ciclos de carga/descarga de la figura 8(a) y el algoritmo de estimación ha determinado, en cada momento, el SoC promedio del pack de baterías [Fig. 8(c)].

VI. CONCLUSIONES

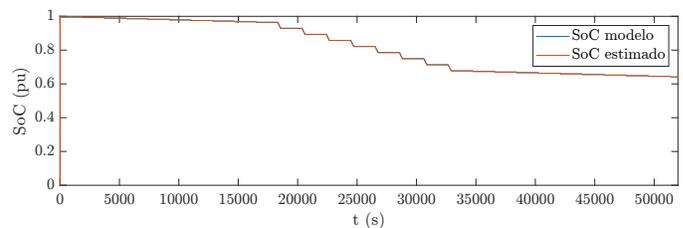
En este artículo se describen todos los elementos software requeridos para el desarrollo de una plataforma BMS aplicada



(a) Pulsos de corriente de descarga.



(b) Tensión en bornas de la celda.



(c) SoC obtenido en el modelo de simulación Simscape vs valor estimado.

Fig. 6: Resultados de simulación para algoritmo EKF con ciclo de descarga.

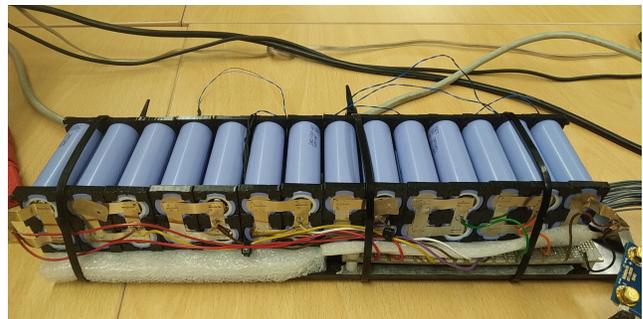
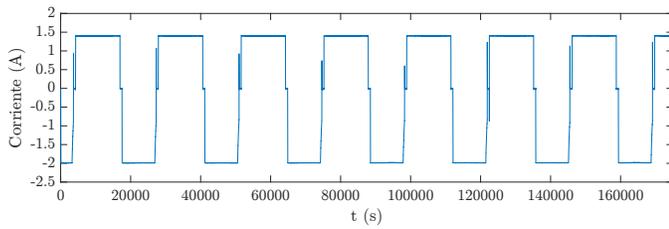


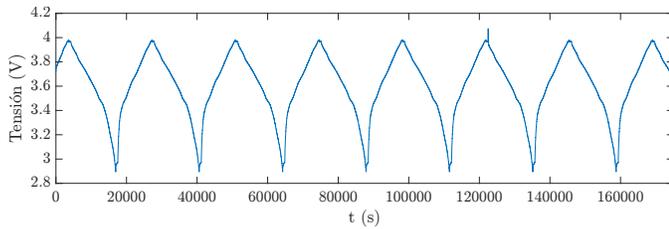
Fig. 7: Pack de baterías Li-ion utilizado para la validación experimental.

a vehículo eléctrico. Su complejidad es elevada, especialmente considerando el alto número de medidas y señales involucradas, alarmas, procesos, etc. En este escenario, el paradigma MBD ha demostrado sus fortalezas, ya que el testeo continuo de las distintas funcionalidades ha permitido una depuración temprana de errores. Por su parte, la generación automática de código ha evitado errores de codificación y ha permitido acelerar el proceso de desarrollo. Los resultados obtenidos (simulación y experimentales) muestran un comportamiento satisfactorio del firmware desarrollado.

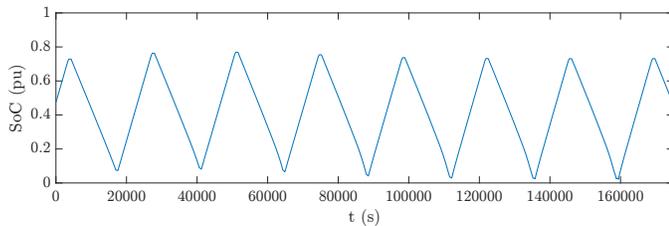
Esta plataforma y su software servirán, en un futuro, como punto de partida para el desarrollo de BMS industriales, así como de banco de pruebas para la propuesta de nuevas soluciones.



(a) Pulsos de corriente de carga/descarga.



(b) Tensión promedio en bornas del pack de baterías.



(c) SoC estimado por el algoritmo EKF.

Fig. 8: Resultados experimentales para algoritmo EKF con ciclos de carga/descarga.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Gobierno del País Vasco dentro de las ayudas para grupos del Sistema Universitario Vasco (IT1440-22), por el Aula Tecnalia de la Escuela de Ingeniería de Bilbao y por la Unión Europea Horizon Europe en el contexto del proyecto eBRT2023 (n° 101095882).

REFERENCES

- [1] B. Yang, Y. Qian, Q. Li, Q. Chen, J. Wu, E. Luo, R. Xie, R. Zheng, Y. Yan, S. Su, and J. Wang, "Critical summary and perspectives on state-of-health of lithium-ion battery," *Renewable and Sustainable Energy Reviews*, vol. 190, no. 114077, pp. 1–28, 2024.
- [2] J. Goodenough, "How we made the Li-ion rechargeable battery," *Nature Electronics*, vol. 1, p. 204, 2018.
- [3] J. Tian, Y. Fan, T. Pan, X. Zhang, J. Ying, and Q. Zhang, "A critical review on inconsistency mechanisms, evaluation methods and improvement features of lithium-ion battery energy storage systems," *Renewable and Sustainable Energy Reviews*, vol. 189, no. 113978, pp. 1–21, 2024.
- [4] L. Zhou, X. Lai, B. Li, Y. Yao, M. Yuan, J. Weng, and Y. Zheng, "State estimation models of lithium-ion batteries for battery management system: status, challenges, and future trends," *Batteries*, vol. 9, no. 131, pp. 1–23, 2023.
- [5] "User manual s-BMS, single boards: scalable battery management system for automotive, industrial and storage battery pack applications," Lithium Balance, Tech. Rep., 2016.
- [6] C. Lidstrom, M. Nyberg, C. Bondesson, and J. Westman, "Improved pattern for ISO 26262 ASIL decomposition with dependent requirements," in *Proc. of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2019, pp. 28–35.
- [7] J. Martins, S. Spataru, T. Kerekes, D. Sera, P. Douglass, G. Yang, and K. Moth, "Test platform for rapid prototyping of digital control for power electronic converters," in *Proc. of the IEEE Industrial Electronics Society Conference (IECON)*, 2019, pp. 2056–2061.

- [8] A. Rón, A. Tarrasó, A. Luna, and P. Rodríguez, "Analysis of rapid control prototyping performance for power conversion applications," in *Proc. of the IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020, pp. 3225–3230.
- [9] E. Ibarra, A. Arias, I. Martínez de Alegría, A. Otero-Olavarrieta, A. Matallana, and L. de Mallac, "Rapid control prototyping platform to regulate a high-current low-voltage DC/DC power converter prototype to feed superconducting electromagnets," in *Proc. of Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)*, 2023, pp. 1–6.
- [10] [Online]. Available: <https://es.mathworks.com/solutions/model-based-design.html>
- [11] [Online]. Available: https://www3.panasonic.biz/ac/e/control/relay/vehicle/ev_special/index.jsp
- [12] Z. Omariba, L. Zhang, and D. Sun, "Review of battery cell balancing methodologies for optimizing battery pack performance in electric vehicles," *IEEE Access*, vol. 7, pp. 129 335–129 352, 2017.
- [13] C. Taborelli and S. Onori, "State of charge estimation using extended Kalman filters for battery management system," in *Proc. of the IEEE International Electric Vehicle Conference (IEVC)*, 2014, pp. 1–8.
- [14] A. Rubio, J. Alzola, I. Marino, J. Rodríguez, J. Gondra, S. García, and S. Burusteta, "Caracterización y modelado de baterías de iones de litio," in *Proc. of Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)*, 2010, pp. 817–822.
- [15] A. Gismero, E. Schaltz, and D. Stroe, "Recursive state of charge and state of health estimation method for Lithium-Ion batteries based on Coulomb counting and open circuit voltage," *Energies*, vol. 13, no. 1811, pp. 1–11, 2020.
- [16] E. Chemali, P. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of Li-ion batteries using deep neural networks: a machine learning approach," *Journal of Power Sources*, vol. 400, pp. 242–255, 2018.
- [17] J. Rivera-Barrera, N. Muñoz Galeano, and O. Sarmiento-Maldonado, "SoC estimation for lithium-ion batteries: review and future challenges," *Electronics*, vol. 6, no. 102, pp. 1–33, 2017.
- [18] S. Yang, S. Zhou, Y. Hua, X. Zhou, X. Liu, Y. Pan, H. Ling, and B. Wu, "A parameter adaptive method for state of charge estimation of lithium-ion batteries with an improved extended Kalman filter," *Scientific Reports*, vol. 11, no. 5805, pp. 1–15, 2021.
- [19] D. Domenico, G. Fiengo, and A. Stefanopoulou, "Lithium-ion battery state of charge estimation with a Kalman filter based on an electrochemical model," in *Proc. of the IEEE International Conference on Control Applications*, 2008, pp. 702–707.
- [20] H. Dai, X. Wei, Z. Sun, J. Wang, and W. Gu, "Online cell SOC estimation of Li-ion battery packs using a dual time-scale Kalman filtering for EV applications," *Applied Energy*, vol. 95, pp. 227–237, 2012.
- [21] C. Van and D. Quang, "Estimation of SoH and internal resistances of Lithium ion battery based on LSTM network," *International Journal of Electrochemical Science*, vol. 18, no. 100166, pp. 1–9, 2023.
- [22] Z. Lin, Y. Cai, W. Liu, C. Bao, J. Shen, and Q. Liao, "Estimating the state of health of lithium-ion batteries based on a probability density function," *International Journal of Electrochemical Science*, vol. 18, no. 100137, pp. 1–9, 2023.