

Integration of Hardware and Software for Battery Hardware-In-the-Loop Towards Battery Artificial Intelligence

Saehong Park, Scott Moura, Kyoungtae Lee* *Member, IEEE*

Abstract—This paper demonstrates a novel, compact-sized hardware-in-the-loop system, and its verification using machine learning and artificial intelligence features in battery controls. Conventionally, a battery management system involves algorithm development for battery modeling, estimation, and control. These tasks are typically validated by running the battery tester open-loop, i.e., the tester equipment executes the pre-defined experimental protocols line by line. Additional equipment is required to make the testing closed-loop, but the integration is typically not straightforward. To improve flexibility and accessibility for battery management, this work proposes a low-cost highly reliable closed-loop charger and discharger. We first focus on the electronic circuit design for battery testing systems to maximize the applied current accuracy and precision. After functional verification, we further investigate applications for closed-loop battery management systems. In particular, we extend the proposed architecture into the learning-based control design, which is a feedback controller. We utilize reinforcement learning techniques to highlight the benefits of closed-loop controls. As an example, we compare this learning-based control strategy with a conventional battery charging control. The experimental results demonstrate that the proposed experimental design is able to handle the learning-based controller and achieve a more reliable and safer charging protocol driven by artificial intelligence.

Index Terms—Advanced battery management system, Hardware-in-the-Loop, Programmable current controller, Battery status monitoring, Deep reinforcement learning

I. INTRODUCTION

Electrified transportation requires real-time monitoring and controls for large-scale battery systems. This requires real-time feedback control in order to ensure safety and performance. Especially, there have been enormous efforts in the area of modeling, estimation and controls for the battery management system (BMS).

The proper mathematical battery modeling is essential for battery control research. Battery models can be categorized into two groups: equivalent circuit models (ECMs) and electrochemical models. ECMs model the input-output behavior using circuit elements, such as resistors and capacitors. However, ECMs do not capture the physical phenomena inside the battery, including lithium transport, solid-electrolyte interphase dynamics, and degradation mechanisms. On the other hand, electrochemical models directly incorporate diffusion, intercalation, and electrochemical kinetics from physics perspectives, and formulate fundamental mathematical structures. Furthermore, physics-based models provide insights about how batteries degrade in various formations, such as solid

electrolyte interphase (SEI) layer growth, Lithium plating, and mechanical stresses [1], [2], [3].

The electrochemical parameter estimation is an essential procedure to enhance the electrochemical model, as these parameters reveal the material properties, such as state-of-charge (SOC), state-of-health (SOH), and state-of-power (SOP). A non-destructive parameter estimation requires the design of experiments that various input current profiles are applied to the battery for maximizing parameter identifiability, and then matched the output response with a specific parameter set [4]. In literature, the sensitivity of electrochemical parameters is analytically derived based on a physics-based model for parameter estimation [5]. The authors in [6] identify a subset of the physics-based model using a short duration of pulse profile. Bayesian optimization is applied to quantify the statistical parameter estimates as well as automated process in [7]. After key electrochemical parameters are identified, internal states in the battery system can be estimated by designing a state estimator [8].

Accurate battery modeling and its parameter estimation improve the battery control quality, leading to better lifespan and performance [9]. Several studies have been conducted to study the health-aware rapid battery charging technique for electrochemical models. A constrained feedback control strategy via reference governor is implemented to prevent battery degradation mechanisms using internal state estimation for battery charging in [10]. The authors in [11] formulate a minimum-time charging problem with health-related constraints and use nonlinear model predictive control (MPC). Similarly, the authors in [12] apply stochastic MPC to charge the battery packs under the presence of parameter uncertainties. The authors in [13] propose ultra-fast charging method for urban air mobility by using the asymmetric temperature modulation method. The authors in [14] design a wireless charging system for electric vehicles to provide constant power transfer.

In recent years, machine learning and artificial intelligence (ML & AI) have attracted attentions in a various way. The authors in [15] propose a electrochemical-thermal-neural-network model as a combination of physics-based and data-driven models. For battery capacity estimation, the authors in [16] apply a deep long short-term memory (LSTM) neural network for cells under diverse working conditions. Prognostic and diagnostic for battery management system widely utilize machine learning techniques to predict battery health or critical points [17], [18], [19]. However, the learning performance is limited by lab protocols, as the ML models rely on the data collected from pre-defined current profiles. Finding an optimal current profile that maximizes the performance is a challenging task, and therefore, automated exploration using

Saehong Park and Scott Moura are with the Energy, Controls and Applications Lab (eCAL) at the University of California, Berkeley, CA 94720, USA (E-mail: {sspark,smoura}@berkeley.edu).

Kyoungtae Lee is with the Berkeley Sensor & Actuator Center at the University of California, Berkeley, E-mail: ktlee@berkeley.edu.

* Corresponding Author: Kyoungtae Lee (ktlee@berkeley.edu).

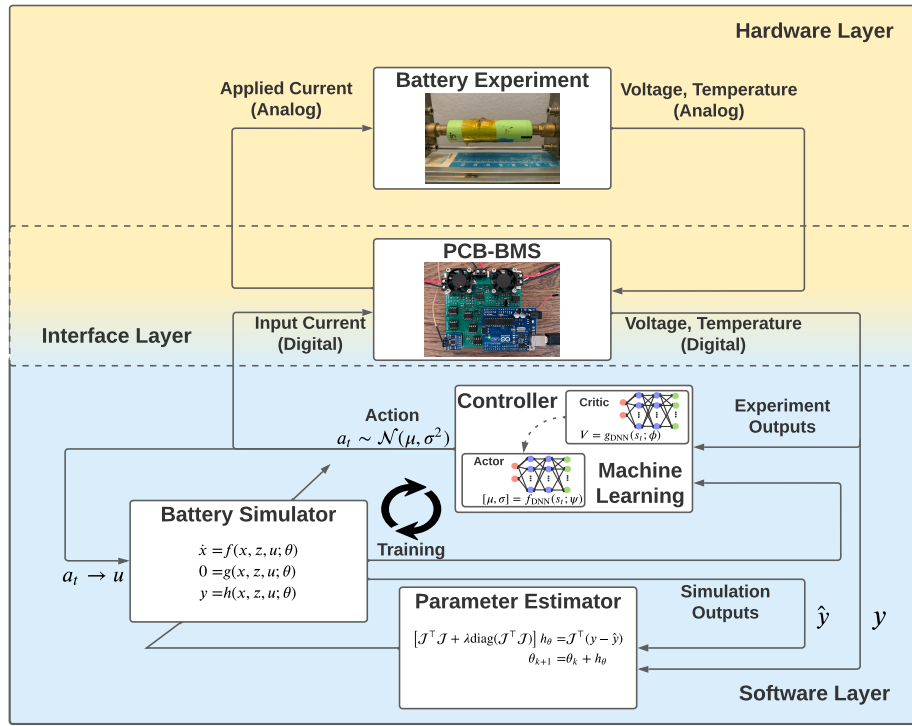


Figure 1: Proposed advanced battery management system architecture.

an AI is critical. This drives a high need for a hardware-in-the-loop system (HIL) with real-time current control and battery monitoring.

Real-time HIL is essential for an advanced battery management system. The controller can set the battery current in real-time, which is especially critical for deep neural network (DNN) based algorithms, as the battery current is determined by battery output measurements at previous time points. However, most conventional battery testers run open-loop, and lack real-time control. They run based on a pre-determined battery current profile that the user provides. External devices are required to make the tester system closed-loop, but integration is not trivial and requires additional expenses and learning curves. More importantly, the battery testers are bulky and expensive, precluding simultaneously testing many cells (i.e. more than 20) and BMS implementation in a product.

Real-time HIL for an advanced battery management must 1) monitor the battery status, such as temperature and voltage, 2) control the current accurately with a wide dynamic range in real-time, and 3) be a portable size for expansion. However, no previous works have satisfied all these requirements. Many works focused on battery voltage and temperature monitoring [20], [21], [22]. A pulse width modulation (PWM) based chopper strategy was used for charging, but high-frequency noise is inevitable, and thus it lacks a precise current control [23]. A power operation amplifier (OPAMP) and proportional-integral-derivative (PID) control-based CC-CV charger was proposed, but settling time, current resolution, and noise information are not presented [24]. A switching mode power supply-based battery testing system was introduced, but the system is open-loop (i.e. it does not have a negative feedback loop to mitigate noise and interference), and thus it lacks a

stable current control [25]. Therefore, we need a real-time HIL platform that enables accurate charging and discharging current control, as well as battery status monitoring.

Here, we present an integration of hardware and software for an advanced battery management system as shown in Figure 1. A printed circuit board (PCB) is designed to interface between the hardware and software for battery management, called PCB-BMS. The PCB-BMS receives the desired current input from the user in the software layer and applies the exact amount of current for charging or discharging. The PCB-BMS also measures battery voltage and temperature in real-time. The current inputs and outputs from the battery are stored in the digital (software) layer and used for modeling, estimation, and controls. We highlight this integrated system for identifying the battery model and training the parameterized controller (agent) to allow a health-aware battery fast-charging output feedback control.

We summarize our novel contributions as follows. Firstly, the hardware design for the battery charging and discharging control scheme is proposed. Specifically, we implement -5 A to +5 A battery current control within ≤ 10 mA accuracy by exploiting a negative feedback loop in the hardware. Accurate and precise real-time current control with output measurements allows users to easily apply a learning-based control strategy where an electrochemical-thermal model is used to train the agent to design a feedback controller for battery fast charging. To the best of the authors' knowledge, this is the first hardware implementation of a deep neural network-based fast-charging controller. The learning-based battery charging controller achieves reliable battery charging with minimal temperature rise, compared to the conventional CC-CV charging that overheats the battery. This work further provides insights

into not only the charging controller design using a deep neural network but also the importance of combining a physics-based model with parameter adaptation for training the AI agent. In this work, we define battery artificial intelligence terminology as designing a learning-based controller to the battery. Note that although we chose a DNN for hardware verification, this platform can be used to test and apply other battery management algorithms.

This paper is organized as follows. Section II introduces the hardware design. Section III presents static and dynamic measurement results. Section IV briefly reviews an electrochemical-thermal model, its parameter estimation via sensitivity analysis, and learning-based control scheme for battery management system. In Section V, we conduct the battery charging experiments using the proposed hardware architecture, and provide a use case for the learning-based control. We summarize our efforts and provide future direction in Section VII.

II. BATTERY HARDWARE-IN-THE-LOOP SYSTEM DESIGN

A. Specifications

Design requirements of the portable on-line hardware-in-the-loop system for advanced battery management and testing are listed below. Note that the specifications described below are chosen for demonstration of the prototype. Thus, they can be changed depending on the target application by simply modifying circuit elements such as resistors, capacitors, and transistors without changing the overall system architecture. Design trade-offs and passive element value calculations are described in the system analysis sections (see Section II-B).

- **Software-based on-line battery monitoring and control** To give a maximum battery management flexibility to the user, this system minimizes the role of the hardware. The user controls the battery current with almost no latency (less than 10 ms). The battery terminal voltage, battery current, battery temperature, and printed circuit board (PCB) temperature information are transferred to the user in real-time. This way, almost every feature can be implemented in the software, including CC-CV charging/discharging and safety monitoring. In addition, as demonstrated in the results section, minimal latency design makes it possible to set the battery current in real-time by a learning algorithm almost instantaneously.
- **Accurate battery charging and discharging current control with a wide dynamic range** In this work, we aim for -5 A to +5 A battery current control within ≤ 10 mA accuracy. Then, the standard deviation of the battery current fluctuation when the system is in a steady state should be less than 10 mA. The battery current varies due to temperature drift, Nyquist-Johnson thermal noise, and circuit non-idealities (non-linear response, voltage offsets, and gain uncertainty). This drives the need for using a negative feedback loop, providing a stable control using high gain amplifiers.
- **High temporal resolution battery current control with minimal rise time and damping** Fast and low latency

current control is essential to minimize unwanted distortion in the desired current waveform. In this work, 1 second temporal resolution for current control is achieved with 95 % settling time less than 10 ms. 0.02 second control is also possible by sacrificing current uncertainty due to the finite settling time. Stabilizing the negative feedback loop by analyzing in the frequency domain is critical.

- **Battery status monitoring** Ambient and battery temperature, as well as battery terminal voltage, V_{battery} , are monitored in real-time. This provides additional dimensions of information to the user, allowing for better battery control. V_{battery} is quantized with 1.2 mV resolution, which can be further improved by using a better analog to digital converter (ADC). A Type-K thermocouple temperature sensor attached on the battery provides real-time battery temperature information. In addition, a thermistor measures the PCB temperature for safety monitoring with 10 mV/°C, leading to a 0.12 °C resolution.
- **Portable size for scalability** Conventional battery testing relies on using bulky external equipment, precluding testing many batteries (>20) at the same time. This work integrates all the necessary circuitry within a small form factor ($10 \times 10 \text{ cm}^2$) similar to the battery size, suitable for a large-scale battery testing and research. In addition, the system can be utilized in a product for real-time battery monitoring due to the small form factor.

B. System Design

1) *Charging*: Figure 2 illustrates the schematic diagram of the proposed system. The charging system (see charging control area in Figure 2) consists of a variable gain instrumentation amplifier (IA), 5 mΩ current sense resistor, $R_{\text{sense_c}}$, operational amplifier (OPAMP), and high power P-type metal-oxide-silicon field effect transistor (PMOS), M_c . These components form a negative feedback loop to accurately and linearly control the battery current, I_{battery} , using the input voltage, $V_{\text{ctrl_c}}$. The negative feedback can be checked by going around the loop. Suppose I_{battery} increases due to noise. Then the input voltage difference to the IA, $V_{\text{ctrl_c}} - V_{\text{battery}}$, increases, leading to an increase in the IA output voltage, $V_{\text{IAout_c}}$. Then, the OPAMP output voltage, $V_{\text{OPout_c}}$ increases. The current flowing through the PMOS decreases as the gate voltage, $V_{\text{OPout_c}}$ increases. As a result, I_{battery} decreases, eventually canceling out the initial current increase due to noise. Theoretically, the negative feedback loop rejects fluctuations on any node in the loop except that on the input, $V_{\text{ctrl_c}}$.

The DC loop-gain (LG) of the system can be computed as:

$$LG = A_{\text{DC_IA}} \times A_{\text{DC_OPAMP}} \times g_{m_c} \times R_{\text{sense_c}}, \quad (1)$$

where LG is the loop gain, $A_{\text{DC_IA}}$ and $A_{\text{DC_OPAMP}}$ are the DC gains of the IA and OPAMP, respectively, g_{m_c} is the transconductance ($\partial I / \partial V$) of the PMOS transistor (M_c), and $R_{\text{sense_c}}$ is the current sensing resistance. The loop gain needs to be large enough to minimize a DC error. The DC gain of

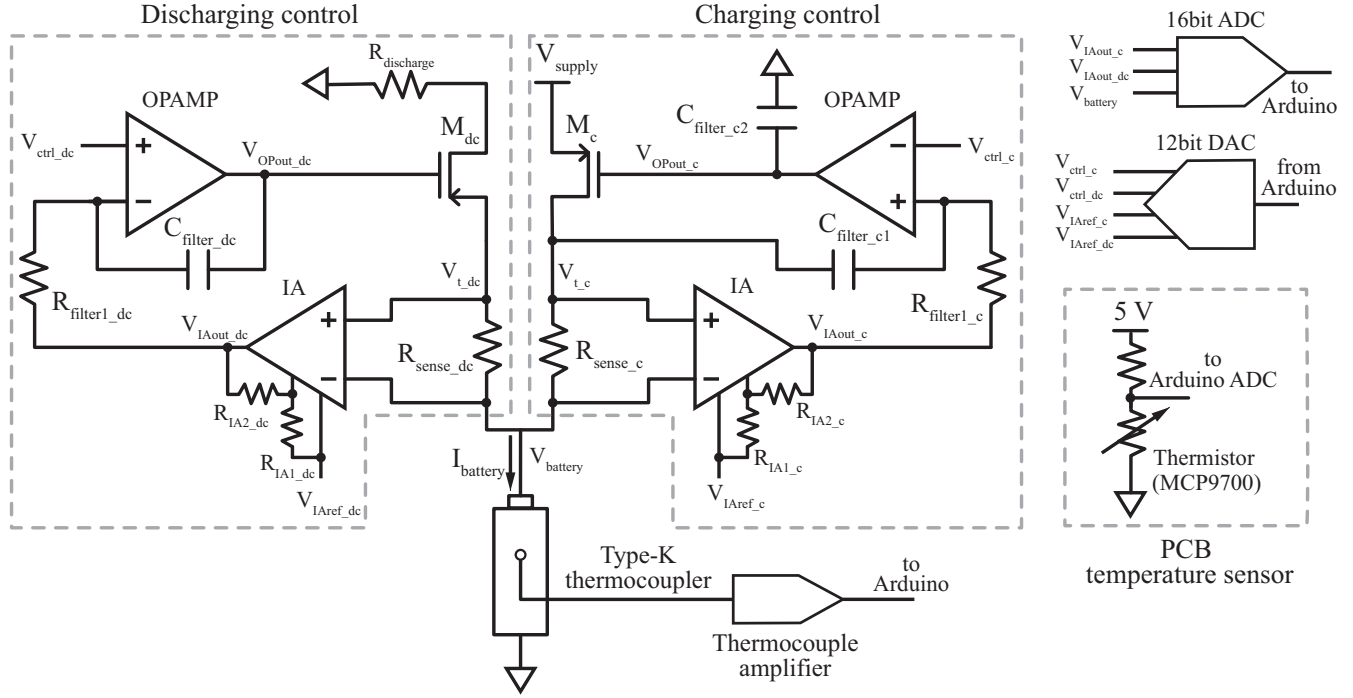


Figure 2: System schematic diagram.

the IA can be set by two resistors, R_{IA1_c} and R_{IA2_c} , as shown below.

$$A_{DC_IA} = 1 + \frac{R_{IA2_c}}{R_{IA1_c}} \quad (2)$$

In this work, R_{IA1_c} and R_{IA2_c} are 909Ω and $31.6k\Omega$, respectively, leading to $A_{DC_IA} = 35.8$.

If the negative feedback loop is functional, the loop tries to keep the positive and negative nodes of the input summing node the same. Therefore, because the loop gain is much greater than 1, the overall closed-loop gain is

$$\frac{\partial I_{battery}}{\partial V_{ctrl_c}} = \frac{A_{DC_OPAMP} \times g_{m_c}}{1 + A_{DC_IA} \times A_{DC_OPAMP} \times g_{m_c} \times R_{sense}} \approx \frac{1}{A_{DC_IA} R_{sense}} \approx 5.59. \quad (3)$$

For example, if V_{ctrl_c} decreases by 1 V, $I_{battery}$ decreases by 5.59 A. The relationship between V_{ctrl_c} and $I_{battery}$ is highly proportional due to the large loop gain. Note that depending on the application, the closed-loop gain can be set to another value by choosing different values for R_{IA1_c} , R_{IA2_c} , and R_{sense_c} . However, if A_{DC_IA} is too high, the loop becomes unstable.

The output voltage of the IA can be calculated as:

$$V_{IAout_c} = V_{IAref_c} + (V_{t_c} - V_{battery}) \left(1 + \frac{R_{IA2_c}}{R_{IA1_c}} \right). \quad (4)$$

To control $I_{battery}$ from 0 to 5 A with 10 mA resolution, V_{ctrl_c} should be controlled with $10mA/5.59 \approx 1.79mV$ resolution. Therefore, 12 bit resolution for V_{ctrl_c} control is enough. Finally, $I_{battery}$ is determined as below.

$$I_{battery} = \frac{V_{ctrl_c} - V_{IAref_c}}{A_{DC_IA} R_{sense}} = 5.59(V_{ctrl_c} - 0.1) \quad (5)$$

In this work, V_{IAref_c} is set to 0.1 V. Therefore, the maximum value of V_{ctrl_c} is $5A/5.59 + 0.1 \approx 0.99V$.

Exploiting the negative feedback loop, $I_{battery}$ is not affected by $V_{battery}$ or power supply voltage, $V_{supply} = 4.5V$, as long as $V_{battery}$ does not exceed $V_{supply} - 0.1V$. Lastly, it is noteworthy that V_{supply} is preferred to be similar to $V_{battery}$, ideally $V_{supply} \approx \max(V_{battery}) + 0.1V$ to minimize waste in power and heat dissipation from the M_c PMOS transistor, as the power dissipation in the PMOS transistor is $V_{supply}I_{battery} - V_{t_c}I_{battery}$.

Ensuring feedback stability is essential, especially when there are more than one low frequency poles in the loop. In our circuit configuration, the IA has a dominant pole at A_{DC_IA}/f_{unity_IA} , where f_{unity_IA} is a unity gain frequency of the IA. Similarly, the dominant pole of OPAMP is at $A_{DC_OPAMP}/f_{unity_OPAMP}$, where f_{unity_OPAMP} is a unity gain frequency of the OPAMP. To ensure feedback stability, there should be only one dominant pole in the loop-gain. Since the dominant pole of the IA depends on its gain configuration and is at much higher frequency than that of the OPAMP (5,600 Hz and 0.8 Hz for the IA and OPAMP, respectively), we decided to decrease the OPAMP pole frequency by placing $R_{filter1_c}$ and C_{filter_c1} in a low pass filter configuration. This guarantees stability for both small and large signal step responses.

2) *Discharging*: Discharging control uses the same circuit components used for charging, an IA, OPAMP, and PMOS transistor. R_{IA1_dc} and R_{IA2_dc} are 909Ω and $110k\Omega$, respectively, resulting in $A_{DC_IA} = 122$. Similar to the charging case, the closed-loop gain for discharging is

$$\frac{\partial I_{battery}}{\partial V_{ctrl_dc}} \approx -\frac{1}{A_{DC_IA} R_{sense_dc}} \approx -1.64. \quad (6)$$

Unlike the charging circuit, the closed-loop gain is negative. That is, $I_{battery}$ decreases as V_{ctrl_dc} increases.

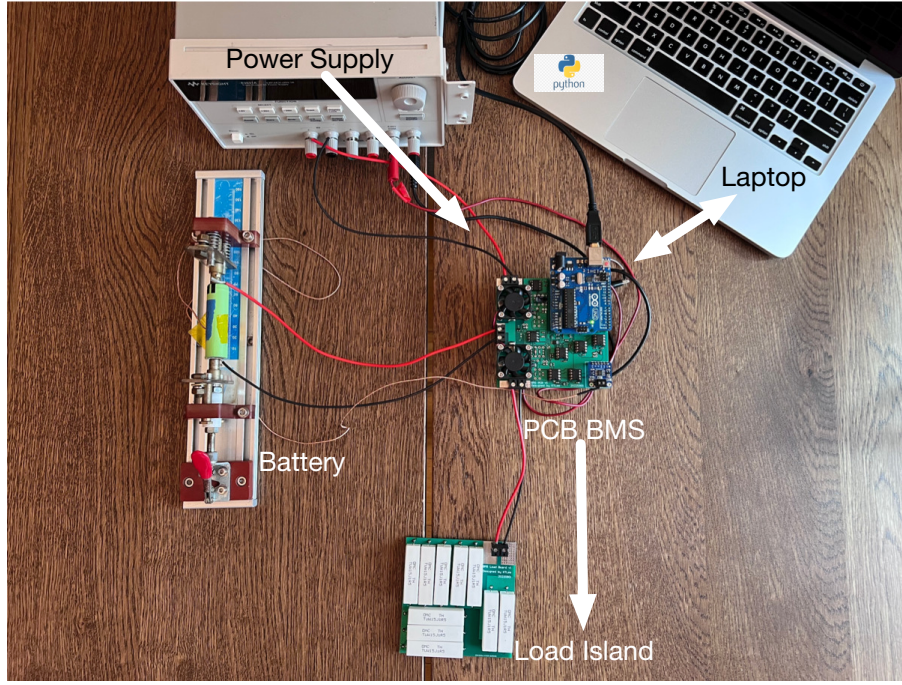


Figure 3: Advanced battery management system setup photo.

According to the datasheet of the IA (LTC2053), the input common range is limited as follow.

$$0V < (V_{t_dc} - V_{battery}) + V_{IAref_dc} < 5 - 1.3V. \quad (7)$$

Due to this limit, V_{IAref_dc} is set to be 3.5 V. Therefore, $I_{battery}$ can be determined by V_{ctrl_dc} as shown below.

$$I_{battery} = \frac{V_{ctrl_dc} - V_{IAref_dc}}{A_{DC_IA} R_{sense_dc}} = \frac{V_{ctrl_dc} - 3.5}{1.64}. \quad (8)$$

Note that negative $I_{battery}$ refers to discharging. For example, $I_{battery} = -1$ A means 1 A of current is being drained out from the battery.

The voltage across the discharging resistor, $R_{discharge}$, is $I_{battery} R_{discharge}$, which varies from 0 V to $5R_{discharge}$. Therefore, $5R_{discharge}$ must be smaller than $V_{battery}$ to ensure 5 A discharging. However, if $R_{discharge}$ is too small, most power is dissipated from the PMOS transistor (M_{dc}), generating unwanted heat on the PCB. As a result, we used 0.6Ω for $R_{discharge}$.

Similar to the charging circuit, R_{filter_dc} and C_{filter_dc} are selected for feedback stability. The low pass filter architecture for discharging is slightly different from that for charging because the impedance that PMOS transistors encounter is different.

3) *Data Conversion and Temperature Monitoring:* Battery voltage ($V_{battery}$) and IA output voltages (V_{IAout_c} and V_{IAout_dc}) are sampled by a 16-bit analog to digital converter (ADC, ADS1115, Texas Instruments) with a voltage resolution of 0.19 mV. Four 12-bit digital to analog converters (DAC, MCP4921, Microchip) set the reference voltages for the IAs (V_{IAref_c} and V_{IAref_dc}) and control voltages (V_{ctrl_c} and V_{ctrl_dc}). The Arduino (Arduino Uno Rev. 3, Arduino) controls the ADC and DACs and conveys the data to a laptop via USB. The

battery temperature is monitored in real time using a Type-K thermocouple with a thermocouple amplifier (MAX31855, Maxim). The Type-K thermocouple is directly attached to the surface of the cell can. Lastly, a thermistor (MCP9700, Microchip) is placed on the PCB near the PMOS transistors to measure the PCB surface temperature for safety monitoring.

4) *Hardware and Software Integration:* A python-based program is developed to interface between the Arduino and user script (i.e. the algorithm). A python multiprocessing module is used to segregate the role of the software. One processor (i.e. Arduino handler) continuously gathers measured data from the Arduino, and relays the data to the user script every 1 second. It also receives the desired $I_{battery}$ value from the user script, and sends it to the Arduino every 1 second. The user script communicates with the Arduino handler via pipes. This strategy ensures that the Arduino handler is running concurrently from the user script. Therefore, data and communication are not lost even when the user script uses most resources. In addition, the proposed software architecture can be easily integrated with GPU or cloud-based cluster computing service.

III. BATTERY HARDWARE-IN-THE-LOOP SYSTEM VERIFICATION

Figure 3 displays a picture of the overall system. The power supply provides 4.5 V for charging the battery. All the circuit components except $R_{discharge}$ are integrated in a PCB. The PCB dimensions are $10 \times 10 \text{ cm}^2$. Two $3 \times 3 \text{ cm}^2$ fans mitigate the PCB temperature rise due to heat dissipation from the PMOS transistors. A separate PCB (the load island in Figure 3) includes ten 1.5Ω cement resistors, resulting in $R_{discharge} = 0.6 \Omega$. Note that the load island can be replaced by other components (i.e. electric motor or lights) as long

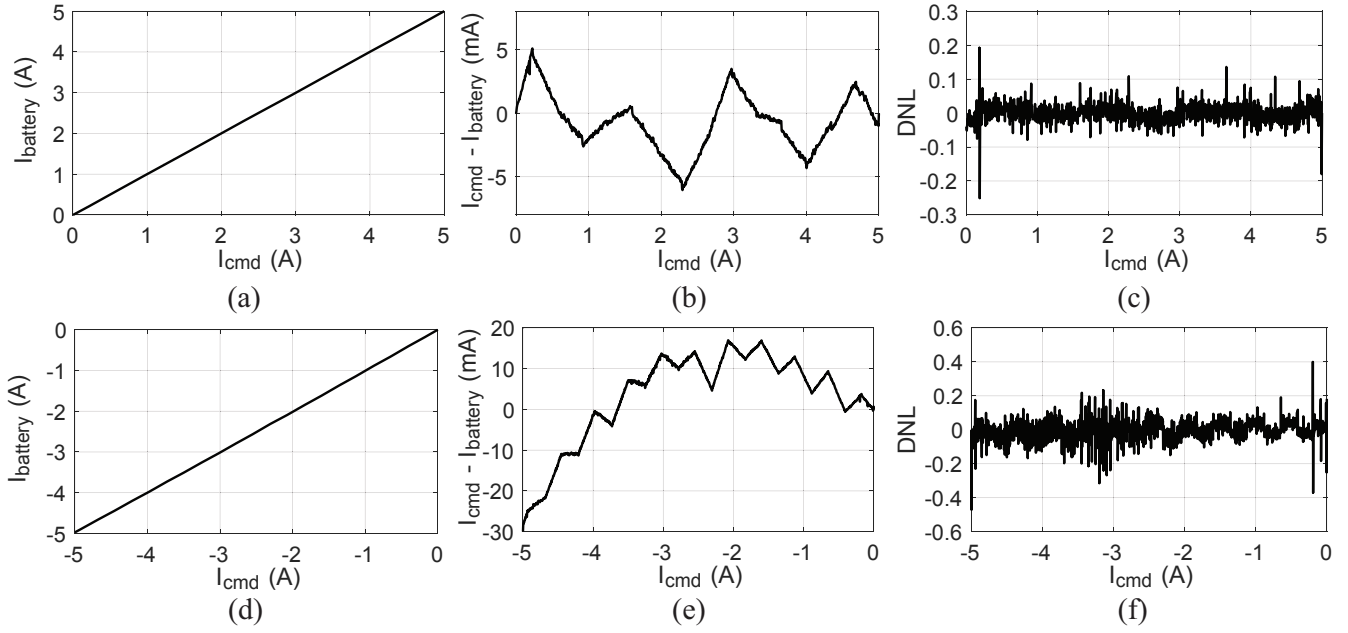


Figure 4: Static measurement results.

as the voltage does not exceed $V_{\text{battery}} - 0.1$ V. The Arduino board relays the data between the laptop and PCB. Lastly, a python-based program running on the laptop controls the PCB. It is noteworthy that although it is not demonstrated due to the current supply limit of the power supply that we use, the system can theoretically handle I_{battery} higher than 5 A by using a better power supply.

A. Static Measurement Results

Figure 4(a)-(c) depict the static measurement results for charging. The current command (I_{cmd} , the I_{battery} value that the user desires) is swept from 0 to 5 A with the finest step (~ 7 mA) and the actual I_{battery} is measured. As shown in Figure 4(a), they are highly linear with R^2 value greater than 0.9999. No saturation is observed. Since the system is essentially a digital (I_{cmd}) to analog (I_{battery}) converter, conventional metrics to evaluate a DAC can be used. Differential nonlinearity (DNL) measures the monotonicity of a DAC and adds additional quantization noise. The DNL is defined as below.

$$DNL(i) = \frac{I_{\text{battery}}(i+1) - I_{\text{battery}}(i)}{I_{\text{cmd}}(i+1) - I_{\text{cmd}}(i)} - 1, \quad (9)$$

where i indicates the i -th step. The DNL shows the slope of I_{battery} with respect to I_{cmd} at each step. The DNL value higher than -1 guarantees the monotonicity of the system. Figure 4(b) shows the DNL measurement result from 0 to 5 A. All DNL values are within ± 0.25 least significant bit (LSB), indicating that the system is highly accurate. The absolute current error, $I_{\text{cmd}} - I_{\text{battery}}$, measurement result at each step is shown in Figure 4(c). In every step, the absolute error is well within ± 6 mA, smaller than the finest resolution of the system for charging (7 mA).

Figure 4(d)-(e) show the static measurement results for discharging. Similar to charging, as shown in Figure 4(d), the system is highly linear with R^2 value greater than 0.9999 for discharging. The DNL values for discharging are within ± 0.43 LSB for all range (see Figure 4(e)). The absolute error exceeds ± 10 mA at certain range of I_{cmd} , but the relative error still remains within an acceptable range (i.e. 10 mA error at $I_{\text{cmd}} = -1$ A means 1 % relative error).

B. Dynamic Measurement Results

Figure 5(a) and (b) show the dynamic measurement results for charging and discharging, respectively. For charging, I_{cmd} is changed from 2.5 A (-2.5 A for discharging) to 0.5, 1, 1.5, 2, 3, 3.5, 4, 4.5, and 5 A (negative values for discharging) at $t = 0$ s, and I_{battery} is measured using an oscilloscope. The 95 % settling time (i.e. the time it takes for I_{battery} to reach 95 % of the I_{cmd}) for charging and discharging are less than 9.75 ms and 8.53 ms, respectively. If the temporal resolution is 1 sec (i.e. I_{cmd} changes every second), this ensures more than 99.6 % accuracy in charge to/from the battery ($Q = \int I dt$). Fast settling time enables even higher temporal resolution at the cost of charge uncertainty. For example, 0.02 second control (i.e. I_{cmd} changes every 20 ms) is possible, but the charge accuracy is 84.2 %.

C. V_{supply} and V_{battery} Dependency Measurement Results

Since the system senses and controls the current flowing through the current sensing resistors ($R_{\text{sense_c}}$ and $R_{\text{sense_dc}}$) using negative feedback loops, the system is not affected by variations in other parameters such as V_{supply} or V_{battery} . Figure 6(a) shows the I_{battery} variation for discharging when V_{battery} varies ($I_{\text{cmd}} = -1$ A). I_{battery} varies within ± 6.5 mA, which is smaller than the I_{cmd} resolution (7 mA). Similarly, Figure 6(b) depicts I_{battery} variation for charging when V_{battery}

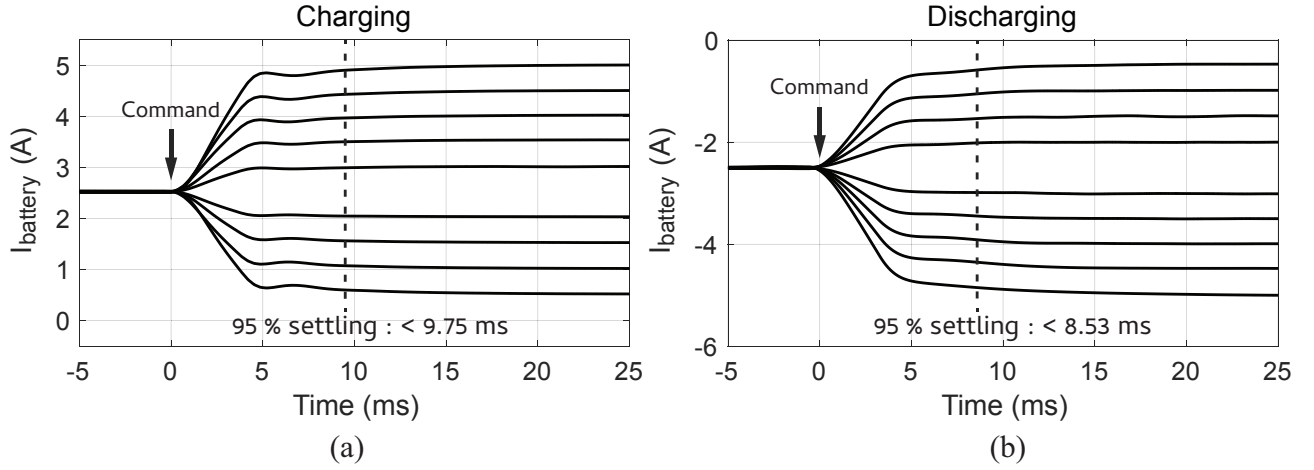


Figure 5: Dynamic measurement results.

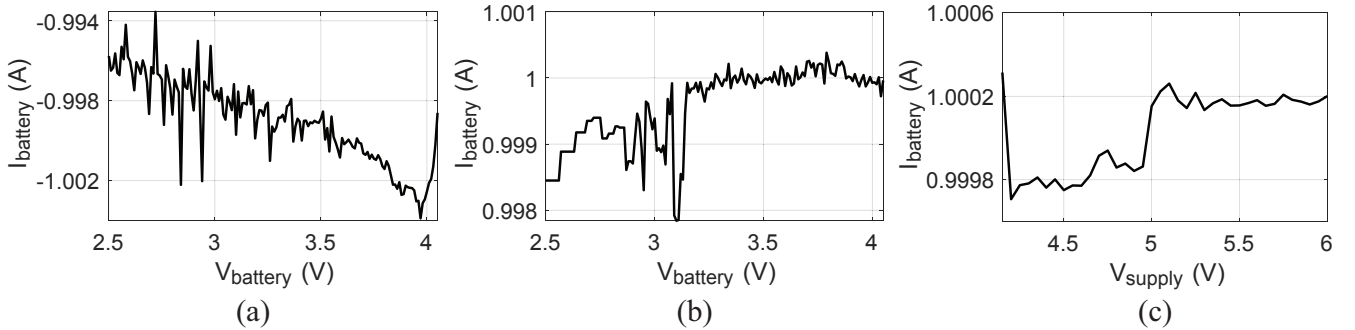


Figure 6: V_{supply} and V_{battery} dependency measurement results.

varies ($I_{\text{cmd}} = 1$ A), and the I_{battery} error is less than 2.1 mA. Lastly, I_{battery} variation for charging when V_{supply} varies ($I_{\text{cmd}} = 1$ A) is shown in Figure 6(c). The I_{battery} error is well within ± 0.3 mA, indicating that the system does not require high precision power supplies and makes the overall cost relatively cheap.

IV. BATTERY MANAGEMENT SYSTEM ALGORITHMS DESIGN

In this section, we briefly describe battery modeling, estimation, and controls perspectives. The design of the estimation & control algorithm is based on the dynamical system modeling to represent real battery behaviors. Specifically, a physics-based model is able to capture the internal electrochemical reaction mechanisms in granular detail, and is used for predicting the battery behavior. A non-invasive parameter estimation technique is used to reduce the gap between the system model and the battery experiment. Once the parameters are identified to represent the actual cell closely, then a learning-based control policy is trained via reinforcement learning for the purpose of minimum charging time while not violating the state constraints. The highlight of this proposed framework is enabling ‘learning-based control’ as a use case for the proposed closed-loop hardware-in-the-loop architecture.

A. Electrochemical-Thermal Model

Firstly, we will go through the mathematical battery model employed in this study. High-fidelity battery model can provide insights into battery cell design by assessing the influence of physical characteristics on battery performance. Based on the porous electrode theory, a mathematical model has been developed in which Li-ions are intercalated in spherical particles in the negative/positive electrodes. For instance, Li-ions are deintercalated from the positive electrode, dissolved in the electrolyte, and diffused to the negative electrode in charging process. We consider the Doyle-Fuller-Newman (DFN) model to predict the evolution of lithium concentration in the solid $c_s^\pm(x, r, t)$, lithium concentration in the electrolyte $c_e(x, t)$, solid electric potential $\phi_s^\pm(x, t)$, electrolyte electric potential $\phi_e(x, t)$, ionic current $i_e^\pm(x, t)$, molar ion fluxes $j_n^\pm(x, t)$, and thermal model to represent battery core, surface temperature $T_c(t)$, and $T_s(t)$. Discretization via finite element method is conducted to simulate the physics-based model in the x-direction including the negative electrode, separator, and positive electrode. The governing equations are written as:

$$\frac{\partial c_s^\pm}{\partial t}(x, r, t) = \frac{1}{r^2} \frac{\partial}{\partial r} \left[D_s^\pm r^2 \frac{\partial c_s^\pm}{\partial r}(x, r, t) \right], \quad (10)$$

$$\varepsilon_e^j \frac{\partial c_e^j}{\partial t}(x, t) = \frac{\partial}{\partial x} \left[D_e^{\text{eff}}(c_e^j) \frac{\partial c_e^j}{\partial x}(x, t) + \frac{1 - t_e^0}{F} i_e^j(x, t) \right], \quad (11)$$

$$C_c \frac{dT_c(t)}{dt} = \frac{T_s(t) - T_c(t)}{R_c} + \dot{Q}, \quad (12)$$

$$C_s \frac{dT_s(t)}{dt} = \frac{T_c(t) - T_s(t)}{R_c} + \frac{T_a(t) - T_s(t)}{R_u}, \quad (13)$$

where j indicates the negative electrode, separator, and positive electrode, denoted by $j \in \{-, \text{sep}, +\}$. According to [26], the heat generation inside the cylindrical cell, $\dot{Q}$, is defined as:

$$\begin{aligned} \dot{Q} = & I(t) [U^+(t) - U^-(t) - V(t)] \\ & - I(t) T(t) \frac{\partial}{\partial T} [U^+(t) - U^-(t)], \end{aligned} \quad (14)$$

and the algebraic equations of the model are given by

$$\sigma^{\text{eff}, \pm} \cdot \frac{\partial \phi_s^\pm}{\partial x}(x, t) = i_e^\pm(x, t) - I(t), \quad (15)$$

where $\phi_s^\pm$ represent the solid electric potentials and i_e is the ionic current in algebraic variables,

$$\sigma^{\text{eff}, \pm} \cdot \frac{\partial \phi_s^\pm}{\partial x}(x, t) = i_e^\pm(x, t) - I(t), \quad (16)$$

$$\begin{aligned} \kappa^{\text{eff}}(c_e) \cdot \frac{\partial \phi_e}{\partial x}(x, t) = & -i_e^\pm(x, t) + \kappa^{\text{eff}}(c_e) \cdot \frac{2RT}{F} (1 - t_c^0) \\ & \times \left(1 + \frac{d \ln f_{c/a}}{d \ln c_e}(x, t) \right) \frac{\partial \ln c_e}{\partial x}(x, t), \end{aligned} \quad (17)$$

where ϕ_e is the electrolyte electric potential across the x -direction. The ionic current is governed by the algebraic equation,

$$\frac{\partial i_e^\pm}{\partial x}(x, t) = a^\pm F j_n^\pm(x, t), \quad (18)$$

where $j_n^\pm$ are the molar ion flux in the anode and cathode, input to the spherical particles, and governed by following algebraic equation,

$$j_n^\pm(x, t) = \frac{1}{F} i_0^\pm(x, t) \left[e^{\frac{\alpha_a F}{RT} \eta^\pm(x, t)} - e^{-\frac{\alpha_c F}{RT} \eta^\pm(x, t)} \right], \quad (19)$$

this expression represents Butler-Volmer kinetics [26], the exchange current density, i_0 , is defined as:

$$i_0^\pm(x, t) = k^\pm [c_{ss}^\pm(x, t)]^{\alpha_c} [c_e(x, t) (c_{s, \max}^\pm - c_{ss}^\pm(x, t))]^{\alpha_a}, \quad (20)$$

where $c_{ss}^\pm$ stands for the Lithium concentration at surface expressed by:

$$c_{ss}^\pm(x, t) = c_s^\pm(x, R_s^\pm, t), \quad (21)$$

and overpotential is governed by following equation:

$$\eta^\pm(x, t) = \phi_s^\pm(x, t) - \phi_e(x, t) - U^\pm(c_{ss}^\pm(x, t)) - F R_f^\pm j_n^\pm(x, t). \quad (22)$$

The state-dependent parameters such as $D_e^{\text{eff}} = D_e(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$, $\sigma^{\text{eff}, j} = \sigma \cdot (\varepsilon_s^j + \varepsilon_f^j)^{\text{brug}}$, and $\kappa^{\text{eff}} = \kappa(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$ are the effective electrolyte diffusivity, effective solid conductivity, and effective electrolyte conductivity, respectively, given by the Bruggeman relationship. The details of boundary conditions for PDEs (10) – (11), ODEs (16), (17), and (18) can be found in the authors' previous work [4].

Note that thermal dynamics depend on the battery cell geometry, such as cylindrical or pouch. In this work, we use 18650 cylindrical battery cell using two thermal states, namely core temperature, T_c , and surface temperature, T_s . Regardless of cell geometry, electrochemical parameters, $D_s^\pm$, D_e , κ_e , and $k^\pm$, vary with temperature via the Arrhenius relationship:

$$\psi = \psi_{ref} \exp \left[\frac{E_\psi}{R} \left(\frac{1}{T_c} - \frac{1}{T_a} \right) \right], \quad (23)$$

where ψ represents a temperature-dependent parameter, E_ψ is the activation energy, and ψ_{ref} is the reference parameter value at room temperature. The model input is the applied current density $I(t)$ [A/m²], and the output is the voltage measured across the current collectors,

$$V(t) = \phi_s^+(0^+, t) - \phi_s^-(0^-, t). \quad (24)$$

After using suitable numerical methods, the electrochemical-thermal model can be simply expressed as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{z}, u; \boldsymbol{\theta}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (25)$$

$$\mathbf{0} = \mathbf{g}(\mathbf{x}, \mathbf{z}, u; \boldsymbol{\theta}), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \quad (26)$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{z}, u; \boldsymbol{\theta}), \quad (27)$$

where the states are $\mathbf{x} = [c_s^-, c_s^+, c_e, T_c, T_s]^\top \in \mathbb{R}^{n_x}$ and $\mathbf{z} = [\phi_s^-, \phi_s^+, i_e^-, i_e^+, \phi_e, j_n^-, j_p^+]^\top \in \mathbb{R}^{n_z}$, and parameters are denoted by $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$. The output variables are defined as $\mathbf{y} = [V(t), T_s(t)]^\top$ in (24), (13), and the input current, u , represents the applied current, $I(t)$ in (16).

B. Parameter Estimation

Sensitivity analysis is used to determine how the output of the model is affected by changes in parameter values [27]. Based on the nominal parameter values, local sensitivity analysis can be derived as the first-order partial derivatives of the system output with respect to the parameters. In this section, we briefly summarize how the local sensitivities in dynamical systems (25)-(27) are derived. With the sensitivity information, we can further extend to a parameter estimation framework via nonlinear least squares.

Define sensitivity variables as follows:

$$S_{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}}, \quad S_{\mathbf{z}} = \frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}, \quad S_{\mathbf{y}} = \frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}}, \quad (28)$$

where $S_{\mathbf{x}} \in \mathbb{R}^{n_x \times n_p}$, $S_{\mathbf{z}} \in \mathbb{R}^{n_z \times n_p}$, and $S_{\mathbf{y}} \in \mathbb{R}^{n_y \times n_p}$ are sensitivity vectors. The i, j matrix element is defined as the partial derivative of the i -th variable to the j -th parameter, e.g.

$$[S_{\mathbf{x}}]_{i,j}(t) = \frac{\partial x_i(t)}{\partial \theta_j}. \quad (29)$$

The evolution of the sensitivity variables is governed by the sensitivity differential algebraic equations (SDAEs), which can be derived following the procedure in [28]:

$$\dot{S}_{\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} S_{\mathbf{x}} + \frac{\partial f}{\partial \boldsymbol{\theta}} S_{\mathbf{z}} + \frac{\partial f}{\partial \boldsymbol{\theta}}, \quad S_{\mathbf{x}}(0) = S_{\mathbf{x}_0}, \quad (30)$$

$$0 = \frac{\partial g}{\partial \mathbf{x}} S_{\mathbf{x}} + \frac{\partial g}{\partial \mathbf{z}} S_{\mathbf{z}} + \frac{\partial g}{\partial \boldsymbol{\theta}}, \quad S_{\mathbf{z}}(0) = S_{\mathbf{z}_0}, \quad (31)$$

$$S_{\mathbf{y}} = \frac{\partial h}{\partial \mathbf{x}} S_{\mathbf{x}} + \frac{\partial h}{\partial \mathbf{z}} S_{\mathbf{z}} + \frac{\partial h}{\partial \boldsymbol{\theta}}. \quad (32)$$

The SDAEs provide a rigorous mathematical computation of the sensitivities compared to a perturbation method where sensitivities are obtained by perturbing each parameter slightly and calculating the output difference with respect to nominal parameters. Note that SDAEs are linear time-varying DAEs, which require computation of the Jacobian with respect to states and parameters at each time step. We utilize automatic differentiation tool since it provides more accurate, automated, and fast Jacobian calculations. In particular, the CasADi is adopted for efficient computing the derivatives [29]. In this work, the battery model DAEs (25) – (27) and the corresponding SDAEs (30) – (32) are simulated by running the IDAS integrator provided by SUNDIALS via the CasADi interface [30]. Subsequently, these sensitivity vectors are utilized to minimize the voltage/temperature prediction in a nonlinear least squares fashion, namely,

$$\min_{\hat{\theta}} \sum_{t=0}^{t_f} \left[y(t) - \hat{y}(t; \hat{\theta}) \right]^2. \quad (33)$$

To iteratively solve the optimization problem (33) we adopt the Levenberg-Marquardt (L-M) algorithm [31], [32]. In essence, L-M adaptively blends the parameter update scheme between the gradient descent method and the Gauss-Newton method based on

$$[\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})] \beta_{\hat{\theta}} = \mathbf{J}^T (y - \hat{y}), \quad (34)$$

where $\mathbf{J} = \partial \hat{y} / \partial \hat{\theta}$ is the local sensitivity of the output $\hat{y}$ at given local parameter $\hat{\theta}$, which is equivalent to the sensitivity vector computed from (30)-(32). The parameter, λ , trades off gradient descent update and Gauss-Newton update. Finally, the parameter estimates can be updated iteratively according to

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \beta_{\hat{\theta}}. \quad (35)$$

C. Fast Charging Control Problem

One of the critical tasks in the battery management system is to charge the battery as quickly as possible while suppressing the battery aging mechanisms. In this context, one could formulate a constrained control problem where the goal is to reach the final SOC in a minimum time without violating constraints. The control input is the applied current in the battery charger hardware, which is limited by,

$$-I_{\max} \leq I(t) \leq 0, \quad (36)$$

where the negative sign indicates charging mode during the battery operation. To restrict the various degradation mechanisms of the cell, we capture different level of constraints in the battery charging problem. Firstly, the battery cell temperature cannot surpass a certain limit, $T_{\max}$, such as:

$$T(t) \leq T_{\max}. \quad (37)$$

Note that the high temperature contributes to solid-electrolyte interphase (SEI) layer growth [9], which is a primary degradation mechanism.

Taking the temperature impact into account, the following control problem can be formulated:

$$\begin{aligned} & \max_{I(t)} \quad -t_f \\ & \text{subject to} \\ & \quad \text{battery dynamics, (10) – (22),} \\ & \quad \text{input current constraint, (36),} \\ & \quad \text{temperature constraint, (37),} \\ & \quad \text{initial states, } V(t_0) = V_0, T(t_0) = T_0, \\ & \quad \text{SOC}(t_f) = \text{SOC}_{\text{ref}}, \end{aligned} \quad (38)$$

where $t_0 = 0$ and t_f are the initial and final time of the charging procedure, respectively, V_0 and T_0 are the initial values for voltage and temperature, respectively. $\text{SOC}(t)$ can be computed by coulomb counting method, and SOC_{ref} is the reference SOC at which the charging is considered to complete. When the full-order electrochemical-thermal model is considered, this becomes a large-scale optimization problem due to the hundreds number of states. We explored this challenge in the previous work [33], and the reinforcement learning (RL) approach provides the solution via approximated dynamic programming, where the value function is estimated from generated samples while improving the controller performance, called actor-critic approach. In this work, we highlight that the proposed HIL configuration enhances the data-driven solutions experimentally. Readers are referred to the authors' former work [33] for more details of RL in battery controls.

Note that the structure of the actor-critic networks remains consistent during training and experimental testing in this work. Deep neural network topologies with varying numbers of neurons are used to build the actor-critic networks [34]. Specifically, we adopt the same configuration as our previous work to make consistent results (i.e. two hidden layers with 20 - 20 neurons in the actor-network and two hidden layers with 100 - 75 neurons in the critic network). Training hyper-parameters for these networks are summarized in Table I. Readers are referred to equations (9) – (15) in [33] for detailed actor-critic description.

Variable	Description	Value
γ	Discount factor	0.99
η_{π}, η_Q	Learning rate of actor and critic network	$10^{-4}, 10^{-3}$
τ	Soft update rate of target networks	10^{-3}

Table I: hyper-parameter lists for actor-critic networks.

The RL requires to design appropriate reward function, which needs to represent the optimization problem (38). In this work, we integrate the objective and constraints into reward function, such as:

$$r_{t+1} = r_{\text{fast}} + r_{\text{safety}}(s_t, a_t), \quad (39)$$

where r_{fast} is an immediate penalty for each time step before reaching the reference SOC. The state constraint is also formulated by means of linear penalty functions at each time step [35] Notice that the RL does not require specific knowledge of the system, so-called *environment*, which implies constraint violation will necessarily be experienced by the agent during

the training. While the model-based RL can be examined to guarantee robust constraint satisfaction for safety-critical applications, such as autonomous driving [36], the actor-critic approach is allowed to exceed the constraints for learning purposes since the battery is a marginally stable system [37] and learning will not create unsafe instabilities. The key factor is that the simulation model should be identical to actual cell prior to experiments via parameter estimation, otherwise, the agent might experience un-explored state-action pairs during the testing. We formulate the following reward function using the available output measurements, namely,

$$r_{\text{safety}}(s_t, a_t) = r_{\text{temp}}(s_t, a_t) + r_{\text{volt}}(s_t, a_t), \quad (40)$$

where reward function, r_{temp} and r_{volt} , are computed at each time step by measuring the outputs,

$$r_{\text{temp}}(s_t, a_t) = \begin{cases} \rho_{\text{temp}}(T(t) - T_{\max}), & \text{if } T(t) \geq T_{\max} \\ 0, & \text{otherwise.} \end{cases} \quad (41)$$

$$r_{\text{volt}}(s_t, a_t) = \begin{cases} \rho_{\text{volt}}(V(t) - V_{\max}), & \text{if } V(t) \geq V_{\max} \\ 0, & \text{otherwise.} \end{cases} \quad (42)$$

The fast charging term, r_{fast} , and penalty function coefficients, r_{safety} , are tuned as $T_{\max} = 35^\circ\text{C}$, $r_{\text{fast}} = -0.1$, and $\rho_{\text{temp}} = -5$. The maximum allowable voltage is obtained as $V_{\max} = 4.2\text{V}$ according to the datasheet and the voltage constraint is set to $\rho_{\text{volt}} = -100$. The charging current is restricted by the hardware setup, such as $[0, 1.8\text{C}]$, where C is the C-rate related to the considered cell. The performance of RL agent in simulation is investigated in [38]. In this work, we demonstrate that the RL agent can be deployed to the real battery system as an output-feedback controller in the proposed HIL architecture. Notice that the proposed HIL architecture can handle any learning algorithm.

V. BATTERY MANAGEMENT SYSTEM ALGORITHMS VALIDATION

In this section, we demonstrate the functionality of the proposed HIL system, starting from validating the physics-based model, conducting parameter estimation, and applying learning-based control for battery fast-charging. The objective is to show how the hardware and software are integrated and validated for a fast-charging application. As previously mentioned, we count on the electrochemical-thermal model to represent real battery behavior discussed in Section IV-A. The electrochemical parameters of the battery model are adopted from the author's previous work [4], which was to identify the electrochemical parameters in a non-invasive fashion for a 18650 Lithium nickel cobalt aluminum oxide cathode and a graphite anode (NCA) battery. Figure 7 presents a comparative analysis between the simulated model outputs and the experimental measurements using the proposed battery hardware-in-the-loop system. The measured voltage is equivalent to V_{battery} in Section III and compared with the electrochemical-thermal model voltage output (24). The overall root mean square error (RMSE) achieves 14.9 mV for the pulse profile and 11.3 mV for the urban dynamometer driving schedule (UDDS)

Method	$T_{\max}$ Reach Time	$T_{\max}$ Violation Time	$T_{\max}$ Reach Time	Target SOC Reach Time
RL	Not reached	0 sec.	Not reached	2,186 sec.
PID	648 sec.	1,294 sec.	1,410 sec.	1,942 sec.

Table II: Summary of charging time [seconds] up to 80% SOC under the temperature constraint.

driving cycle profile. This voltage RMSE implies that there exists a model-mismatch between the physics-based model described in Section IV-A and the actual cell due to battery cell aging or manufacturing variation. This model-mismatch can be minimized by conducting the parameter estimation algorithm described in Section IV-B until the voltage difference is lower than the acceptable threshold.

It is desired to compare the difference between the input current command, I_{cmd} and the applied current input, I_{battery} as a validation purpose discussed in Section III, and the results are presented in the second subplot. Note that the applied current input, I_{battery} , is used because this is the actual current applied to the battery. Quantitatively, the proposed BMS achieves RMSE of 8.6 mA for pulse profile and 4.6 mA for driving cycle profile, which meets the design criteria for hardware development discussed in Section II.

Another output comparison between the experiment and the mathematical model is the surface temperature of the cell. We first simulate the model with the nominal thermal parameters, denoted as 'before thermal parameter estimation'. The RMSE between measurement and initial model output is 0.7848°C for pulse profile and 0.8019°C for UDDS profile. Even though the electrochemical parameters are identified and validated from previous work [4], one can notice that the thermal parameters, such as heat convection, R_u and conduction resistance, R_c remained as not-identified, which motivates us to identify them using the surface temperature measurement and model output (13) shown in Figure 7. As described in Section IV-B, these thermal parameters, namely, $\theta = [R_c, R_u]^\top$, are identified via the nonlinear least square method with sensitivity vectors of these parameters with respect to surface temperature measurement as an output discussed in Section IV-B. After the iterative process, the identified thermal parameters achieve 0.4713°C RMSE for pulse profile and 0.2053°C RMSE for UDDS profile denoted as 'after thermal parameter estimation'. To ensure safety during the battery testing, the PCB temperature is monitored by the hardware-in-the-loop in real time.

After validating the electrochemical-thermal model in terms of output measurements, such as output voltage and battery surface temperature, we train the actor-critic networks via reinforcement learning described in Section IV-C. To ensure safety and health during charging, the battery surface temperature constraint is set to $T_{\max} = 35^\circ\text{C}$ in (37). The conventional constant current constant voltage (CC-CV) is also implemented via PID control approach as a baseline. Figure 8 presents the comparison of the learning-based charging control with the conventional approach in terms of temperature constraint violation and time for reaching the target SOC. Table II summarizes this comparative study. The battery starts at 2.8 V and the SOC is simply computed by looking up open-

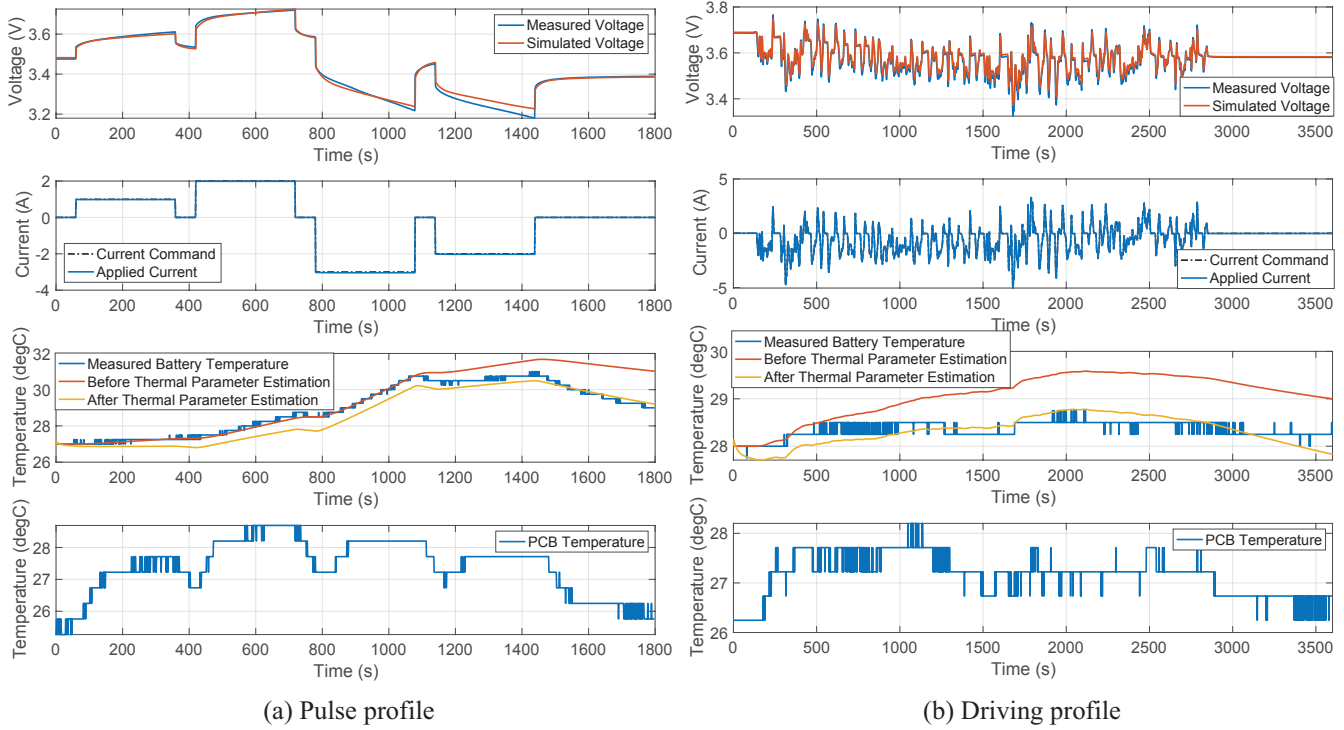


Figure 7: The comparison between the electrochemical model and battery experiment with the proposed BMS architecture.

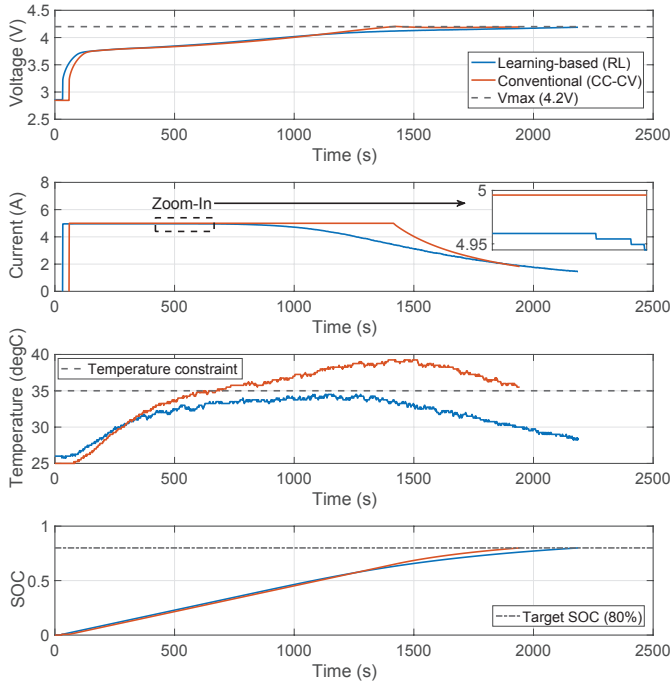


Figure 8: Comparison of NCA 18650 Li-ion battery charging controls between learning-based (reinforcement learning) and conventional approach (CC-CV via PID control).

circuit-voltage with coulomb counting. Note that the battery starts to charge with different idle time in order to present the voltage/current/temperature curves distinctly. The conventional CC-CV charging scheme is set to the maximum allowable charging current, 5A, and constant voltage, 4.2 V as $V_{\max}$. The CC-CV method reaches the $V_{\max}$ faster than the learning-based

control while the surface temperature exceeds the temperature constraint around 66% of charging time. The learning-based control, which is trained from a physics-based model in order not to violate the temperature constraint, achieves the charging task without violating the temperature constraint. To compare its charging performance, SOC is computed by adding the cumulative coulomb counting from its initial SOC. Due to the presence of temperature constraint, the learning-based controller takes 2,186 seconds which is longer than 1,942 seconds from the baseline, however, this difference can be compensated by not violating the constraint. It's remarkable to state that training the agent in simulation space perfectly works in experimental space by using a deep neural network output-feedback controller.

In this work, our objective is to demonstrate how the proposed battery HIL system can be utilized to apply the actual charging control from the trained agent. Note that the agent is trained for learning the best charging policy while not violating the temperature constraint throughout the learning process. To make the training process more reliable, one needs to make the environment close to the real world, which is the reason why we used the physics-based model and validated electrochemical parameters. Although there exists a model mismatch between the real battery and the physics-based model, we observe that such model mismatch does not impact the charging policy when it comes to comparing with the conventional approach as shown in Figure 8.

VI. DISCUSSION

A. Scalability

The proposed battery HIL system is inherently scaling-friendly to test many battery cells independently and simulta-

neously. Firstly, the PCB controls each battery independently by measuring the current through the shunt resistor and controlling it in a negative feedback loop. Therefore, several PCBs can be stacked up in parallel to test multiple batteries simultaneously. The cost per PCB is relatively cheap ($\sim \$100$ USD) compared to the existing battery tester channel, and the user can scale the system by simply stacking additional PCBs. Secondly, the proposed battery HIL system is able to connect multiple PCBs to a single PC through a USB hub. Theoretically, one USB port can handle up to 127 devices. Given the data rate each PCB requires (<0.1 Mbps), large-scale (~ 100) battery testing can be achieved using a USB port. Also, multiple Python programs can be run independently in separate threads, making the scaling straightforward. Lastly, multiple PCBs can share a power supply as long as the summation of charging currents for every battery at a given time point does not exceed the maximum allowable current value that the power supply can output. For example, a commercially available 1,000 A power supply supports 200 batteries charging at 5 A current simultaneously.

B. Current control accuracy

In the proposed system, I_{battery} is firstly converted into the voltage domain via the shunt resistor ($R_{\text{sense_c}}$ or $R_{\text{sense_dc}}$) and measured by sampling output voltage of the instrumentation amplifier (IA) ($V_{\text{IAout_c}}$ or $V_{\text{IAout_dc}}$). Since the input impedance of the ADC is ~ 22 Mohm which is much larger than the output impedance of the IA and resistive divider ($\ll 150$ kohm), I_{battery} measurements have high accuracy. In addition, the IA has the gain error of ~ 0.001 , ensuring minimal distortion during the signal amplification. Therefore, I_{battery} measurement results would be close to the actual battery current.

The proposed system accurately controls the battery current even with a temperature drift. The negative feedback suppresses any drift or noise (including temperature drift or 60 Hz noise) in the loop if the loop gain is large enough. However, there are errors that the negative feedback cannot mitigate, leading to directly or indirectly affecting the accuracy. Firstly, the negative feedback cannot suppress the error of the control voltage input, $V_{\text{ctrl_c}}$ or $V_{\text{ctrl_dc}}$. However, because they are supplied by 12-bit high precision DACs with only ± 0.2 differential nonlinearity (DNL) and ± 2 integral nonlinearity (INL), the worst case current error would be ~ 13 mA for charging and ~ 1.5 mA for discharging, respectively. In addition, resistance change due to the temperature drift affects the current measurement accuracy. However, the shunt resistors have a temperature coefficient of only $\pm 75\text{ppm}/^\circ\text{C}$. Therefore, the resistance will only change $0.18\text{ m}\Omega$ even with the temperature drift of 25°C , leading to the maximum current error of 9 mA at 5 A I_{battery} . To measure the temperature change, the thermistor is placed right next to the shunt resistors. As shown in Figure 7, the PCB temperature does not change more than 5°C , indicating that the temperature effect would be negligible.

C. Battery requirements

The proposed battery HIL system can be utilized to test any batteries, as long as the battery operating voltage is within the

(2V, 5.5V) range, suitable for most Li-ion batteries (cylindrical, prismatic, and pouch). Specifically, the system requires that the battery voltage should be within 0 $\sim$ 5.5 V during charging, and higher than $R_{\text{load}} \times I_{\text{battery}}$ during discharging. For example, in this study, the battery voltage should be higher than 3.0 V when discharging with the maximum current, 5 A, as R_{load} is set to 0.6 Ω . If the discharging current is small, the battery voltage can be less than 2.5 V. For instance, 1 A discharging current allows the battery voltage higher than 0.6 V. Note that although we chose 0.6 Ω for R_{load} to test a Li-ion battery for this work, the user can set R_{load} value accordingly depending on the application and battery operating voltage.

D. Hardware specifications

Table III summarizes the component ratings and specifications. Note that the discharging resistors were chosen for demonstration purposes, and they can be replaced to any electrical loads (e.g. motors or displays).

E. Software implementations

The proposed battery HIL system is running on a Python platform, which allows users to apply a variety of Python packages. Specifically, we exploit the PyTorch deep learning toolbox for implementing reinforcement learning algorithm in this work. The training process is performed on a Python running an Intel core i9-9900K CPU with a clock rate of 3.60 GHz, and a GPU, GeForce RTX 2080 Ti. The agent parameterized by deep neural network is trained by interacting with the OpenAI Gym environment. It is remarkable to note that the environment should represent the actual battery cell prior to training, and the training process needs to be executed in simulation for reliable experimental validation afterward.

VII. CONCLUSION

In this paper, we propose a compact, low-cost battery HIL system consisting of battery monitoring and control, digital communication, and a Python-based hardware controller. The proposed charging and discharging hardware design is rigorously verified by measuring the actual current to the battery on a millisecond time-scale. This also allows users to control the battery charging and discharging current for validating battery management algorithms.

As an application for the proposed system, we experimentally demonstrate a reinforcement learning algorithm using deep neural networks for fast-charging applications. A deep neural network-based controller is trained by a physics-based simulator with temperature constraints and deployed to the actual battery for validation. The experimental results show that the proposed learning-based control achieves safer battery charging control compared to the existing simple constant current constant voltage approach. With this hardware-in-the-loop, the user could validate any type of battery management algorithms including system identification, design of state estimator, connection to electrified transportation, and learning-based battery materials discovery.

Learning-based approach requires a training process to learn the control objective before use. Once the training is

Component	Model	Quantity	Size ($L \times W \times H \text{mm}^3$)	Operating temperature ($^{\circ}\text{C}$)	Operating voltage or operating power
Shunt resistor	CSS2H-3920-K-5L00F	2	$10 \times 5.2 \times 0.1$	$-55 \sim 170$	2 W
PMOS transistor	IPD042P03L3GATMA1	2	$6.6 \times 10.6 \times 2.3$	$-55 \sim 175$	150 W
Instrumentation amplifier	LTC2053-HMS8	2	$3.0 \times 4.9 \times 1.1$	$-40 \sim 125$	$0 \sim 5.5 \text{ V}$
OP amp	OPA2344	2	$8 \times 9.5 \times 4$	$-40 \sim 85$	$0 \sim 5.5 \text{ V}$
Discharging resistor	TUW151JR5E	10	$49 \times 13 \times 12$	Not specified	15 W
ADC	ADS1115	1	$5 \times 3 \times 1.1$	$-40 \sim 125$	$0 \sim 5.5 \text{ V}$
DAC	MCP4921	4	$8 \times 9.5 \times 4$	$-55 \sim 125$	$0 \sim 5.5 \text{ V}$
Thermistor	MCP9700AT-E/TT	1	$2.9 \times 2.4 \times 1$	$-40 \sim 150$	$0 \sim 5.5 \text{ V}$
Arduino	Arduino Uno Rev 3	1	$69 \times 53 \times 15$	$-40 \sim 85$	$0 \sim 5 \text{ V}$
Thermocouple amplifier	MAX31855	1	$5.1 \times 6.0 \times 2.5$	$-40 \sim 125$	$0 \sim 3.3 \text{ V}$
Thermocouple type-K	Adafruit	1	N/A	$-200 \sim 1350$	N/A

Table III: Hardware specifications.

done, the controller is fixed in most cases, determining the battery current based on the previous observations. This lacks an adaptive control, and thus ignores battery-to-battery and environmental variations. However, as shown in our previous work [38], the learning-based algorithm can be adaptively tuned during the actual operation. In this work, we believe that the proposed HIL system can bring adaptive battery control capability into the real world. It is highly envisioned that the proposed HIL system enables optimized battery-specific charging and discharging control to maximize the battery performance.

VIII. ACKNOWLEDGEMENT

The authors would like to thank Dr. Raejoon Jung for valuable discussions regarding the software architecture. This work is supported by the National Science Foundation under Grant No. 1847177.

REFERENCES

- [1] C. R. Birkil, M. R. Roberts, E. McTurk, P. G. Bruce, and D. A. Howey, "Degradation diagnostics for lithium ion cells," *Journal of Power Sources*, vol. 341, pp. 373–386, 2017.
- [2] A. Jana, A. S. Mitra, S. Das, W. C. Chueh, M. Z. Bazant, and R. E. García, "Physics-based, reduced order degradation model of lithium-ion batteries," *Journal of Power Sources*, vol. 545, p. 231900, 2022.
- [3] P. M. Attia, A. Bills, F. B. Planella, P. Dechent, G. Dos Reis, M. Dubarry, P. Gasper, R. Gilchrist, S. Greenbank, D. Howey *et al.*, "knees" in lithium-ion battery aging trajectories," *Journal of The Electrochemical Society*, vol. 169, no. 6, p. 060517, 2022.
- [4] S. Park, D. Kato, Z. Gima, R. Klein, and S. Moura, "Optimal experimental design for parameterization of an electrochemical lithium-ion battery model," *Journal of The Electrochemical Society*, vol. 165, no. 7, pp. A1309–A1323, 2018.
- [5] Q. Lai, S. Jangra, H. J. Ahn, G. Kim, W. T. Joe, and X. Lin, "Analytical derivation and analysis of parameter sensitivity for battery electrochemical dynamics," *Journal of Power Sources*, vol. 472, p. 228338, 2020.
- [6] D. Lu, M. S. Trimboli, G. Fan, R. Zhang, and G. L. Plett, "Nondestructive pulse testing to estimate a subset of physics-based-model parameter values for lithium-ion cells," *Journal of The Electrochemical Society*, vol. 168, no. 8, p. 080533, 2021.
- [7] Y. Kuhn, H. Wolf, A. Latz, and B. Horstmann, "Ep-bolfi: Measurement-noise-aware parameterization of continuum battery models from electrochemical measurements applied to full-cell gitt measurements," *arXiv preprint arXiv:2208.03289*, 2022.
- [8] D. Zhang, S. Park, L. D. Couto, V. Viswanathan, and S. J. Moura, "Beyond battery state of charge estimation: Observer for electrode-level state and cyclable lithium with electrolyte dynamics," *IEEE Transactions on Transportation Electrification*, 2022.
- [9] A. Tomaszewska, Z. Chu, X. Feng, S. O'Kane, X. Liu, J. Chen, C. Ji, E. Endler, R. Li, L. Liu *et al.*, "Lithium-ion battery fast charging: A review," *eTransportation*, vol. 1, p. 100011, 2019.
- [10] L. D. Couto, R. Romagnoli, S. Park, D. Zhang, S. J. Moura, M. Kinnaert, and E. Garone, "Faster and healthier charging of lithium-ion batteries via constrained feedback control," *IEEE Transactions on Control Systems Technology*, 2021.
- [11] R. Klein, N. A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, and A. Kojic, "Optimal charging strategies in lithium-ion battery," in *American Control Conference (ACC)*, 2011. IEEE, 2011, pp. 382–387.
- [12] A. Pozzi and D. M. Raimondo, "Stochastic model predictive control for optimal charging of electric vehicles battery packs," *Journal of Energy Storage*, vol. 55, p. 105332, 2022.
- [13] T. Liu, X.-G. Yang, S. Ge, Y. Leng, and C.-Y. Wang, "Ultrafast charging of energy-dense lithium-ion batteries for urban air mobility," *eTransportation*, vol. 7, p. 100103, 2021.
- [14] B. Shi, F. Yang, C. Hu, and M. Ouyang, "Modelling and improvement of oscillation problem in a double-sided lcc compensation network for electric vehicle wireless power transfer," *eTransportation*, vol. 8, p. 100108, 2021.
- [15] F. Feng, S. Teng, K. Liu, J. Xie, Y. Xie, B. Liu, and K. Li, "Co-estimation of lithium-ion battery state of charge and state of temperature based on a hybrid electrochemical-thermal-neural-network model," *Journal of Power Sources*, vol. 455, p. 227935, 2020.
- [16] W. Li, N. Sengupta, P. Dechent, D. Howey, A. Annaswamy, and D. U. Sauer, "Online capacity estimation of lithium-ion batteries with deep long short-term memory networks," *Journal of power sources*, vol. 482, p. 228863, 2021.
- [17] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggadakis *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.
- [18] X. Jia, C. Zhang, L. Wang, L. Zhang, and X. Zhou, "Early diagnosis of accelerated ageing for lithium-ion batteries with an integrated framework of ageing mechanisms and data-driven methods," *IEEE Transactions on Transportation Electrification*, 2022.
- [19] J. Tian, R. Xiong, W. Shen, J. Lu, and X.-G. Yang, "Deep neural network battery charging curve prediction using 30 points collected in 10 min," *Joule*, vol. 5, no. 6, pp. 1521–1534, 2021.
- [20] B. Tarhan, O. Yetik, and T. H. Karakoc, "Hybrid battery management system design for electric aircraft," *Energy*, vol. 234, p. 121227, 2021.
- [21] S. Kivrak, T. Özer, Y. Oğuz, and E. B. Erken, "Battery management system implementation with the passive control method using mosfet as a load," *Measurement and Control*, vol. 53, no. 1-2, pp. 205–213, 2020.
- [22] H.-C. Chen, S.-S. Li, S.-L. Wu, and C.-Y. Lee, "Design of a modular battery management system for electric motorcycle," *Energies*, vol. 14, p. 3532, 2021.
- [23] D. Pavković, M. Hrgetić, A. Komljenović, V. Smetko *et al.*, "Battery current and voltage control system design with charging application," in *2014 IEEE Conference on Control Applications (CCA)*. IEEE, 2014, pp. 1133–1138.
- [24] Q. Ouyang, G. Xu, H. Fang, and Z. Wang, "Fast charging control for battery packs with combined optimization of charger and equalizers," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 11, pp. 11 076–11 086, 2020.
- [25] Y. Wang, S.-Y. Kim, Y. Chen, H. Zhang, and S.-J. Park, "An smps-based lithium-ion battery test system for internal resistance measurement," *IEEE Transactions on Transportation Electrification*, 2022.
- [26] K. E. Thomas, J. Newman, and R. M. Darling, "Mathematical modeling of lithium batteries," in *Advances in lithium-ion batteries*. Springer, 2002, pp. 345–392.

- [27] H. Yue, M. Brown, F. He, J. Jia, and D. B. Kell, "Sensitivity analysis and robust experimental design of a signal transduction pathway system," *International Journal of Chemical Kinetics*, vol. 40, no. 11, pp. 730–741, 2008.
- [28] H. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.
- [29] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [30] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "Sundials: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.
- [31] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [32] S. Park, D. Zhang, R. Klein, and S. Moura, "Estimation of cyclable lithium for li-ion battery state-of-health monitoring," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3094–3101.
- [33] S. Park, A. Pozzi, M. Whitmeyer, H. Perez, A. Kandel, G. Kim, Y. Choi, W. T. Joe, D. M. Raimondo, and S. Moura, "A deep reinforcement learning framework for fast charging of li-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 2, pp. 2770–2784, 2022.
- [34] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [35] A. E. Smith, D. W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz, "Penalty functions," *Handbook of evolutionary computation*, vol. 97, no. 1, p. C5, 1995.
- [36] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *Advances in neural information processing systems*, vol. 30, 2017.
- [37] S. J. Moura, "Estimation and control of battery electrochemistry models: A tutorial," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 3906–3912.
- [38] S. Park, A. Pozzi, M. Whitmeyer, W. T. Joe, D. M. Raimondo, and S. Moura, "Reinforcement learning-based fast charging control strategy for li-ion batteries," *arXiv preprint arXiv:2002.02060*, 2020.