Health and Performance Diagnostics in Li-ion Batteries with Pulse-Injection-Aided Machine Learning

Alan G. Li\textsuperscript{a}, Weizhong Wang\textsuperscript{a}, Alan C. West\textsuperscript{b}, Matthias Preindl\textsuperscript{a,*}

\textsuperscript{a}Department of Electrical Engineering, Columbia University in the City of New York, 500 W. 120th St., Mudd 1310, New York, NY, 10027, USA
\textsuperscript{b}Department of Chemical Engineering, Columbia University in the City of New York, 500 W. 120th St., Mudd 801, New York, NY, 10027, USA

Abstract

Performance metric diagnostics of lithium-ion batteries are important for electric vehicles. A novel diagnostics method during vehicle charging is proposed using a feedforward neural network and the battery voltage response to a current pulse perturbation, hence the name ‘pulse-injection-aided machine learning’ (PIAML). Performance metrics are quantified using state of health and state of power, representing capacity and power fade. Data is collected for lithium-ion battery cells at various states and pulsing scenarios, resulting in 5,184 unique voltage responses for evaluating the technique. PIAML is shown to estimate states of health and power with high fidelity, and can also be used to initialize the state of charge. In the best-case, average trial error is 0.0057 for state of health estimation, 0.0069 for power, and 0.0072 for charge. Neither charging history nor battery parameters are required, and diagnostics can be performed in less than 3 minutes. Results show that PIAML is a high-accuracy general-purpose technique with potential for wider applications.

Keywords: Battery management systems, Feedforward neural networks, Lithium batteries, State estimation

1. Introduction

Understanding lithium-ion battery (LIB) performance is crucial for proper control of the LIB systems. For electric vehicle (EV) applications especially, accurate knowledge of the states of the battery pack can improve performance and stability [1]. There are 3 key states used to gauge performance: state of charge (SoC), state of health (SoH), and state of power (SoP). The cell SoC varies directly with use, and can change significantly over short time periods. Over long time periods, cell degradation causes capacity and power fade – quantified using SoH and SoP. These two performance metrics directly affect the optimal operation of LIB systems [2]. Since battery states are not directly measurable, accurate diagnostics in the battery management system (BMS) are important for reducing degradation, increasing remaining useful life, and understanding the economic value of battery cells [3].

State estimation methods are broadly divided into two categories: model-based and data-driven. Model-based methods compute or measure parameters for electrochemical or regression models using laboratory data. Model-based approaches often have a trade-off between accuracy and computational speed [4]. They rely on battery models to faithfully reconstruct SoC or SoH based on the model parameters. Not only do these models face challenges in parametrization, but they are often used for estimating individual states for specific battery chemistries and thus lack generality [5]. Still, model-based methods remain popular.

Several model-based methods are used for both SoC and SoH estimation. Open-circuit voltage (OCV) charging curves are used in [6, 7]. Incremental capacity analysis (ICA) is related to OCV and used in [8, 9]. Estimation with OCV charging curves and ICA curves requires long data acquisition periods, which is usually not feasible on-line.
Coulomb counting is another simple method to approximate the charge passed out of the battery. In laboratory conditions, this method can accurately estimate SoH [10]. In real-time applications, error accumulates over time, which renders the method unusable. To address this, Kalman and particle filters apply error-correcting feedback systems to achieve high performance [11, 12, 13]. Kalman filters were also used for joint estimation of states [14], which has the potential to streamline BMS operations and increase accuracy by eliminating interdependencies. These filters can be complex and require long computation times, especially when more accurate battery models are used. State-specific methods include battery charge equalization models for SoC [15], and LIB frequency response techniques [16, 17] and physics-based models (PBM) for SoH [18, 19]. While accurate SoC estimation is fast, SoH methods tend to be time-consuming. The LIB frequency response is slow to acquire, and PBMs represent a challenging parameter identification problem that may not be widely applicable for all battery chemistries.

SoP, unlike SoC and SoH, is not as widely researched [20, 21]. It can be calculated using characteristic mapping, which simply defines SoP in terms of other states. More advanced SoP methods use equivalent circuit models [22, 23], particle filters [24], polarization voltage models [25], objective function minimization [26], and fuzzy look-up table [27].

In contrast to model-based state estimation methods, data-driven state estimation may offer greater accuracy, speed, and generality. Data-driven techniques treat the battery cell as a black-box system. No electrochemical parameters are used. Rather, statistical techniques or machine learning are used to analyze data from cell processes. Typically data-driven techniques involve offline training before implementation in real applications. Offline training and data collection may require significant computing resources. After training, online estimation with data-driven methods uses relatively simple computations and achieves high accuracy [20].

Neural networks (NN) and support vector machines (SVM) are popular data-driven methods for multi-dimensional modelling [28]. Different architectures, such as feedforward NN (FNN), recurrent NN (RNN), ‘Transformer’ NN, or least-squares SVM, are optimized for different tasks [29, 30]. They are applied to SoC estimation [31, 32, 33], SoH estimation, [34, 35, 36, 37], and joint estimation [38, 39]. The performance of NN and SVM depends on the input data. A single time step of data may be straightforward to obtain, but may
not be able to yield accurate results. NN and SVM may also suffer from overfitting, and performance can vary depending on the quality of the training dataset. Fuzzy logic is used for SoC and SoP [40, 41, 42]. Fuzzy logic faces difficulties in generality and convergence time. Grey relational analysis (GRA) [43], and Gaussian process regression (GPR) [5, 44] are used for SoH. Both can be used for long-term predictions, but are computationally complex and thus face difficulty in real-time implementation. Particle-swarm optimization is used for SoC and SoH [45], but shows slow convergence.

1.1. Contributions and Outline

There is a need for a diagnostics technique that is general-purpose and highly accurate with short estimation time. This article thus proposes pulse-injection-aided machine learning (PIAML) for health and performance diagnostics, realized using a FNN. The scheduling framework for diagnostics is proposed for intermittent use during EV charging periods. Training is performed with thousands of voltage responses to a variety of current pulse perturbations of varying amplitudes and lengths. It is shown that PIAML achieves accurate and robust predictions of SoC, SoH, and SoP, with flexibility over the pulse shape.

An overview of the proposed application area, theoretical principles, and offline development for PIAML is given in Section 2. Verification results of PIAML of pulse scheduling are presented and discussed in Section 3. The article is summarized and future work is described in Section 4.

2. Pulse-Injection-Aided Machine Learning

2.1. Application in Electric Vehicle Diagnostics

PIAML is well-suited for EV diagnostics. It has been shown that daily charging profiles are highly predictable based on user profiles [46]. PIAML can exploit this predictability to become a reliable and fast daily diagnostics tool. SoH and SoP, unlike SoC, do not need to be continuously tracked, so it is sufficient to schedule the pulse during low-intensity applications like EV charging.

A diagram of the proposed pulse scheduling framework is shown in Figure 1. Once the EV is plugged into the charger, the pulse is applied to the battery pack. The voltage response of each cell is measured and input to PIAML diagnostics, which computes the performance metrics within seconds. This can be used to track capacity fade and power fade. PIAML can also provide a high-fidelity reference value for SoC, which can be used to periodically initialize SoC estimation methods such as coulomb counting or Kalman filters. After the pulse is injected, the BMS allows normal charging to resume. Since charging profiles often last for several hours [47], any disruption from PIAML perturbation is negligible.

2.2. Principles of pulse perturbation

Pulsing LIB cells is known to encode significant amounts of information about the internal cell processes [48, 49, 50]. Weppner et al. [48] introduced in 1977 the pulse-based galvanostatic intermittent
In PIAML, the cell is perturbed with a current pulse. The response is then fed to a FNN to predict target outputs such as LIB states. Fig. 2 compares model-based methods with PIAML. There are interdependencies between model-based estimators, but PIAML estimators can make predictions independently.

It has been shown that the voltage responses to randomized pulses with a RNN can predict SoH [51]. The input pulse in [51], however, simulates a drive cycle, which may hinder RNN training due to random variation. Drive cycles may not be the optimal pulse shape. Additionally, PIAML was shown to yield accurate results for SoC in [32], but the pulse shape is difficult to use in real systems and PIAML is only verified for ‘snapshot’ applications at 25°C. In [35] and [36], partial-charging curves and machine learning are used for SoH diagnostics. While this minimizes disruption in cell operation, partial-charging curves are unideal for observing electrochemical overpotentials in the cell that could provide insight into internal cell processes [4]. In contrast, PIAML uses current pulse perturbation to obtain a voltage response that encodes more information in a much shorter period of time. Hence PIAML’s applicability to diagnostics beyond SoH [35, 36].

This article therefore investigates many different pulses at various temperatures, which increases the validity of the results and demonstrates robustness. Pulses are defined by their shape, amplitude, bias, and length. Pulse shape affects the frequency

---

Table 1: List of amplitude-bias ratios and constituent amplitudes and biases investigated

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-40</td>
<td>0.03</td>
<td>3</td>
</tr>
<tr>
<td>-30</td>
<td>0.03</td>
<td>1</td>
</tr>
<tr>
<td>-20</td>
<td>0.3</td>
<td>3</td>
</tr>
<tr>
<td>-10</td>
<td>0.3</td>
<td>1</td>
</tr>
<tr>
<td>∞</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

titration technique, now widely used for overpotential diagnostics [49]. More recently Weng et al. [50] used pulses to accurately characterize LIB cell lifetimes. There is no study that has fully explored the use of pulses as a general-purpose diagnostics signal.

---

Figure 2: Comparison of model-based estimation and proposed PIAML estimation framework

Figure 3: Flowchart representing offline tasks and online estimation process for PIAML
Figure 4: Photographs of experimental setup and diagram of cycling procedure for 0-bias pulses, showing (a) Cycler and temperature chamber, (b) Cells inside chamber, (c) Data collection cycling procedure diagram, (d) Sample of cycling voltage and current, and (e) Detail of pulse train.

Figure 5: Input and output data, showing (a) Current pulses from cycler labelled by ABR, where positive cycler current indicates charging, and (b) voltage responses to ABR = ∞dB pulse.
content of the pulse which, following spectroscopy principles, determines the amount of encoded information [52, 53]. Rectangle pulses are examined here, corresponding to constant current charge or discharge. A variety of pulse amplitudes and biases were tested to examine the effects of pulse shape on PIAML performance. We label pulses by the amplitude-bias ratio (ABR). Reducing the ABR allows the effects of ‘noise’ to be considered. ABR [dB] is given by

$$\text{ABR} = 20 \log \frac{I_A}{I_B}$$

(1)

where $I_A = \{0.03, 0.3, 3\} \text{ A}$ is the amplitude of the injected pulse current and $I_B = \{0, 1, 3\} \text{ A}$ is the bias current, resulting in the ABR values listed in Table 1. The overall input current to the cells $i(t)$ over time $t$ is the superposition of the constant bias $I_B$ with the pulse shape $i_p$,

$$i(t) = I_A i_p(t) + I_B$$

(2)

The bias acts as a source of noise which obfuscates $i_p$. Hysteresis from an incompletely rested cell could also act as a time-varying bias, reducing the ABR. A full evaluation of the effects of pre-pulse rest time is out of the scope of this study.

2.3. Offline development and online operation

Several offline tasks must be performed before PIAML can be implemented in a real system. A comparison of the offline and online workflow is shown in Fig. 3. Offline, data collection and processing are required to generate the input pulses and target outputs for training the FNN. Laboratory techniques are used to obtain the target states, such as coulomb-counting and equivalent circuit modelling. These methods are time consuming and infeasible for real-time use. Photographs of the experimental setup are shown in Fig. 4. Cycling procedure for $\infty$ ABR pulses is illustrated in Figs. 4c-4e. The perturbation pulses shown in Fig. 5a are used to obtain the voltage-time response of the cells, exemplified in Fig. 5b.

Six pulse shapes are extracted from the $\infty$ ABR pulse, as labelled in Fig. 5a. The $\infty$dB pulse shape represents the best-case scenario: a high ABR pulse applied to a cell rested for 1 hour. There are four individual portions named ‘Charge’, ‘Rest 1’, ‘Discharge’, and ‘Rest 2’, each lasting 60s or 30s. The two unipolar portions are named ‘Unipolar-charge’ and ‘Unipolar-discharge’ and last 1.5 min. The unipolar portions only require charge or discharge capability. They are easier to implement in real systems than the full bipolar pulse because a single current source or sink are required, and not both. Effects of rest period can also be examined using the portions. Unipolar-charge portions are applied after a 1 hour rest, after which the cell hysteresis voltage has dissipated. Unipolar-discharge portions, however, are applied after a 30 s rest period. This means that the hysteresis voltage remains significant, as can be seen in Fig. 4e.

Three cell states are considered as NN targets: SoC, SoH, and SoP. The cycling procedures create

Figure 6: Overview of state variation, showing (a) Variation of SoP against SoH, color-coded by SoC, for pulses with ABR = $\infty$dB, and (b) Aggregated capacity fade curves for 21 cells at various temperatures.
a wide range of cell states. Nominal SoC ranges from 0.05 to 0.9, while SoH ranges from 0.69 to 1. As suggested by Fig. 5b, degraded cells have higher impedance, which causes pulses applied at the same SoC to diverge over time. Evolution of SoP with SoC and SoH is shown in Fig. 6a. The variation of maximum capacity with cycle number for all cells at the various temperatures is shown in Fig. 6b. Note that this represents the vector of aggregated unique capacities at a given temperature, and not the degradation of any one cell.

NN training is a supervised learning process where the input voltage response is matched to target outputs. FNN are formed from several ‘hidden layers’ of interconnected nodes. Nodes within the input and hidden layers are linked by network weights, and generate outputs based on the activation function. During training, the FNN learns the optimal weights between nodes. In each training cycle, weights are readjusted based on the predicted output, the optimizer function, and the batch size of the training data after comparison with the target data.

Online, PIAML is implemented in the BMS using the trained FNN. A trained FNN is a matrix network of weights and connections learnt during the offline training process. The input voltage response and temperature are fed into the FNN, which then predicts the state. This can be performed quickly for networks with a small number of parameters [54]. PIAML learns the target state from the pulses, thus ‘bypassing’ traditional techniques. No further battery parameters are needed, making PIAML an attractive online alternative for diagnostics.

3. Verification Results and Discussion

Verification of the model accuracy and robustness is performed comparing the best-case scenario with results using a variety of pulse parameters. Mean absolute error (MAE) is used for comparison with other studies. PIAML performance for the best-case, varying ABR, pulse portions, and unipolar perturbation are shown in Fig. 7. For the best case in Fig. 7a, PIAML achieves average MAE of 0.0057 for SoH estimation, 0.0072 for SoC estimation, and 0.0069 for SoP estimation. Accuracy deteriorates in Fig. 7b as ABR decreases. This is especially pronounced for SoH and SoP estimation, though SoC estimation remains relatively accurate. The worst-case pulse shapes have extremely low ABR, yet the MAE remains below 0.10 for estimation of all states. This suggests that PIAML is resistant to increased ‘noise’ obfuscating the pulse signal. Individual portions, shown in Fig. 7c, yield higher error, with deterioration particularly notable for SoH and SoP. Results also suggest that different state information is encoded in different portions. Rest periods may encode more SoC information than charge or discharge, suggesting that a shorter pulse can be used. Variation in the MAE ranges may be due uncertainty on the target data. Coulombic efficiency and OCV, for example, are approximated from the cycling data, thus affecting the target values of SoC and SoP. Thus FNN estimation error may not only reflect FNN accuracy, but also data uncertainty.

For unipolar pulse shapes, it can be seen from Fig. 7d that unipolar charge and discharge pulses are comparable in performance to the full pulse. Unipolar-charge attains similar performance to the full pulse, but is half the length. This would facilitate its use in PIAML diagnostics during EV charging. Reducing the rest time from 1 hour to 30 s, as in the unipolar-discharge portion, only slightly reduces accuracy. This shows that PIAML is robust against non-ideal pulsing conditions. The success of unipolar perturbation suggests that the pulse amplitude, duration, and rest period can be further reduced.

Sample plots of the FNN training and validation loss in Fig. 8 show that overfitting did not occur. The individual predictions against cycle number are highly accurate, as shown in Fig. 9. In Fig. 9b, SoH decreases monotonically with cycle number, while SoC and SoP in Figs. 9a and 9c do not. This is because the SoC may vary from 0 to 1 at any point in the cycle, while capacity loss is irreversible. Since the SoP is a function of both SoC and SoH, it also tends downward.

PIAML is compared with other diagnostics methods in Table 2. Error refers to the average MAE, if more than one value is reported. Time is defined as the minimum time needed to collect input data for the estimator and process the data to generate a prediction, as reported in the study. It can be seen that PIAML strikes a balance between accuracy and speed, with high generality.

4. Conclusion

The diagnostics method PIAML was proposed for use during EV charging cycles. PIAML was shown to yield fast and accurate results for SoC,
Figure 7: PIAML results using different pulse parameters, showing (a) Best case MAE for pulse with ABR = ∞ dB, (b) MAE for pulses with varying ABR, (c) comparison of MAE between individual portions of ABR = ∞ dB pulse, and (d) comparison of MAE between unipolar portions of ABR = ∞ dB pulse.

Figure 8: NN training loss showing the decrease in PIAML training and validation error with epoch number for full-length pulse with ABR = ∞ dB.

Figure 9: Plots demonstrating PIAML accuracy on the unseen testing subset, using full-length pulse with ABR = ∞ dB and plotted against cycle number.
SoH, and SoP estimation using a FNN and the voltage response to a current pulse perturbation. Validation was performed for a wide range of cell states and pulse shapes. PIAML was demonstrated to be an accurate technique for health and performance diagnostics that remains effective even for pulse shapes with low ABR.

Future work includes investigation of more pulse shapes. Unipolar perturbation with smaller amplitudes and shorter durations could facilitate direct real-time state estimation. Pulses could be applied as superpositions over a discharge, as shown by the results for low ABR. These alterations could allow the BMS to directly inject a low-amplitude, high-frequency pulse using the balancing circuits during drive cycles. Additionally, the effects of prior usage or different aging techniques could be explore. PIAML may have applicability in lower levels of degradation, such as degradation modes [2]. Exploring the full potential of PIAML could allow for cheaper and more reliable cell diagnostics in future battery applications.

Methods

Data Collection

Data is collected using Samsung INR18650-30Q lithium nickel cobalt aluminum (NCA) oxide cells. NCA cells have desirable performance, but require additional safety considerations [55]. Cell characteristics are summarized in Table 3.

Cells are cycled using the Neware BTS4000 series 5V6A cycler at 3 temperatures, {5, 25, 40}°C, and at standard pressure. A total of 21 cells are cycled, with each ABR and temperature applied to multiple cells to reduce individuality effects. Voltage and cycler current are monitored at 10 Hz.

For each SoH, a capacity check is performed with a 0.1 C-rate constant current (CC) discharge from full. Cells were then recharged using CC and constant voltage procedures. After resting, pulses are applied at various SoC levels achieved using a 0.3 Ah discharge, until a 2.5V cut-off voltage is reached. For pulses with non-zero bias, the bias current acts as a continuous discharge, so pulses are instead injected at intervals of time corresponding a 0.05 change in SoC. For all cells, degradation is performed with 50 charge/discharge cycles at 1 C-rate, until the cell fails an ‘end-of-life’ test. In total, 5,184 pulses at unique combinations of SoC, SoH, SoP, and temperature are obtained for training and evaluation. All pulses were 1 min, except for the 3 min ∞ dB pulse.

Offline data processing

Target outputs are calculated using traditional modelling techniques. SoC quantifies the remaining charge \( q \) in the cell relative to the maximum charge capacity of the cell \( Q_m \), calculated with

\[
\text{SoC} = \frac{q}{Q_m}
\]  

SoH is the normalized maximum capacity, given by

\[
\text{SoH} = \frac{Q_m}{Q_{m0}}
\]  

where \( Q_{m0} \) is the maximum capacity of the unaged cell. SoH decreases as the cell degrades. Reference values for \( Q_m \) and \( q \) can be obtained from coulomb counting, represented by

\[
q = \eta \int i(t) dt
\]
where $\eta$ is coulombic efficiency, assumed constant at 0.99, and $i(t)$ is the relevant cell current. SoP quantifies the peak power output of the cell, important for measuring power fade. It is affected by both SoC and SoH. SoP is defined using the peak discharge current $I_{pk}$,
\[
\text{SoP} = \frac{I_{pk}}{I_{pk0}}
\]
(6)
where $I_{pk0}$ is the peak discharge current of the cell at SoC=1 and SoH=1. Current $I_{pk}$ is a function of the SoC and SoH, defined with the model-based dynamic multi-parameter method [22],
\[
I_{pk} = \frac{V_{OC} - V_{\text{min}}}{\frac{\Delta V}{\Delta V_{OC}} + R_1(1 - e^{-\frac{\Delta t}{R_1 C_1}}) + R_0}
\]
(7)
where $V_{OC}$ is the open-circuit voltage, $V_{\text{min}} = 2.5V$ is the cut-off voltage, $\Delta t = 60s$ is the time horizon, and $R_0$, $R_1$, and $C_1$ are the state-varying first-order equivalent circuit model parameters. Experimental approaches for measuring $I_{pk}$ pose an inherent safety risk, meaning that $I_{pk}$ must be approximated using a battery model [22, 23, 26]. Adaptive characteristic maps may also be used [21].

**Neural Network Design**

The Keras module of Tensorflow [56] is used to build a simple sequential FNN. A different FNN is trained for estimating each state, and for each unique pulse shape. The FNN is a regression network. This means that the FNN output does not belong to a discrete set of states. Thus even though the target data has discrete values, the FNN will be able to perform state estimation for all values in the range 0 to 1. Voltage responses and the vector of temperature measurements are used as input data, and the target outputs are the battery states, calculated offline.

Training is configured to minimize the mean squared error. Cross-validation was performed by running 20 trials with randomly selected subsets of data from all cells and cycles for training, validation, and testing. Random allocations followed the distribution of 64% for training, 16% for validation, and 20% for testing. Training and validation data are used to optimize the FNN and prevent overfitting. The testing subset is not seen by the FNN during training, and is therefore presented to the FNN as a completely new dataset.

**Table 4: Shared hyperparameters amongst FNN models**

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hidden layers</td>
<td>2</td>
</tr>
<tr>
<td>Network weight constraint</td>
<td>5</td>
</tr>
<tr>
<td>Dropout rate</td>
<td>0.1</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Batch size</td>
<td>64</td>
</tr>
<tr>
<td>Training epochs</td>
<td>32000</td>
</tr>
<tr>
<td>Activation function</td>
<td>relu</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
</tbody>
</table>

A summary of the chosen hyperparameters is listed in Table 4. The same hyperparameters are used for estimation of each state, except the number of nodes per hidden layer. To choose the optimal node number for each state, repeated tests using varying node numbers was performed. The node number that yielded the lowest prediction error from the $\infty$ dB pulse was then selected for further evaluation. Training was performed offline with 32000 epochs using a TITAN Xp NVIDIA graphical processing unit. Online, predictions made with the trained FNN require minimal computation time (on the order of microseconds) and do not require any specific battery model.

**Acknowledgements**

We acknowledge NVIDIA Corporation for donating a Titan Xp GPU, Columbia University’s Shared Research Computing Facility project for computing resources (supported by NIH Research Facility Improvement grant 1G20RR030893-01), and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract C090171. We thank Robert C. Mohr for help on the experimental setup.

**References**


[34] J. Hong, Z. Wang, W. Chen, L. Wang, P. Lin, C. Qu, Online accurate state of health estimation for battery
systems on real world electric vehicles with variable driving conditions considered, Journal of Cleaner Production 294 (2021).


[45] X. Yang, Y. Chen, B. Li, D. Luo, Battery states online estimation based on exponential decay particle swarm optimization and proportional-integral observer with a hybrid battery model, Energy 191 (2020).


