35th International Electric Vehicle Symposium and Exhibition (EVS35)
Oslo, Norway, June 11-15, 2022

Process for sorting or binning cells via a machine-learning method to optimize battery-pack layouts

Gregory L. Plett[1], Steve Weiss[2]
[1]*Department of Electrical and Computer Engineering, University of Colorado Colorado Springs, Colorado Springs, CO 80918, United States,* gplett@uccs.edu,
[2]*Xilectric, Inc., 151 Martine St., Suite 125-1, Fall River, MA 02723,* steve.weiss@xilectric.com

Summary

Cells are generally sorted or binned according to capacity and ac resistance before incorporation into a battery pack. For applications where high-current pulses are demanded, this may not be the optimal approach; instead, cells should additionally be sorted by high-current dc resistance. This paper proposes a machine-learning method that uses low-current measurements to predict high-current dc resistance and demonstrates that a pack layout created by sorting cells according to high-current resistance yields lower RMS current deviation than does a random layout. The optimized layout is expected to extend pack life.

1 Abstract

Every lithium-ion cell produced by a manufacturing process has somewhat random characteristics [1]. Because of this, cells are generally sorted or binned before assembling them into battery packs [2–4]. Properly composed packs can lower the balancing activity required and can even extend life [5].
Presently, battery packs are built from cells that are matched in many cases simply by capacity Q and ac resistance R_0. This paper proposes that matching in this way is not the best approach for applications ranging from stop/start hybrids to performance-model electric vehicles requiring high-power pulses. Instead, we propose matching 30-s high-current dc pulse resistances R_{30}^{high}, which may result in much different pack composition than by matching R_0 and Q. Further, outlier cells can be detected and discarded/recycled that would have been missed when screening cells based only on R_0 and Q.
In principle, high-current pulses could be applied to the cells after manufacture and formation to compute R_{30}^{high} for matching purposes, but this requires multiple parallel high-power machines which is expensive; further, it stresses the cells needlessly. Instead, we propose a method that uses simple low-current measurements that can be made quickly. From these data, we extract a low-current R_{30}^{low} (which can be very different from R_{30}^{high} due to cell nonlinearities and dynamic time constants), and cell EIS impedances Z_{30}^{low} at the fundamental frequency and a number of its odd harmonics.
The method that is proposed uses simple machine-learning (ML) tools to form a linear combination of these measurements to estimate R_{30}^{high}. The outcome is a very accurate estimate of R_{30}^{high} as well as a new method for outlier/defect detection that can expose problems that are invisible when screening based only on R_0 and Q. It also informs a pack-layout method that minimizes current variation among cells in a battery pack. To the best of our knowledge, there is no other published work that proposes an ML method that can be used by a cell-manufacturing end-of-line (EOL) tester to predict a high-current resistance and then to use that prediction to propose an optimized battery-pack layout.

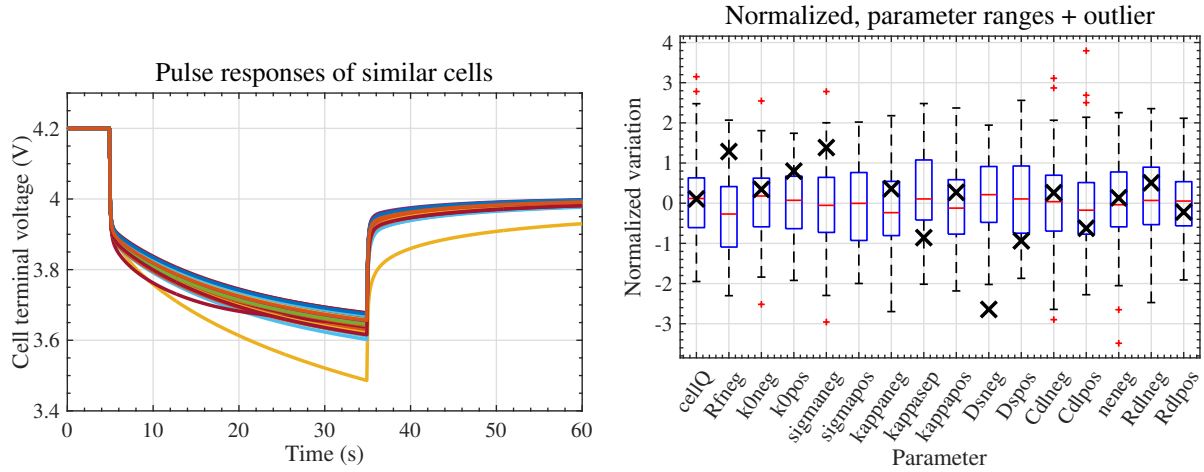


Figure 1: (a) High-current discharge-pulse responses from 100 simulated cells; (b) Characteristics of outlier cell compared with parameter ranges of all cells.

2 Simulations to create a machine-learning data set

ML tools depend on a set of "training" data that are used to optimize the parameters of the estimator, as well as a distinct "validation" or "testing" set of data that are used to determine whether the estimator is providing acceptable performance. For the work presented here, COMSOL was used to simulate a physics-based Doyle–Fuller–Newman (DFN) model of a lithium-ion battery cell [6–8] using somewhat randomized parameter values. The baseline cell had parameter values taken from Table 6.1 in [8]; notably, it has a nominal capacity of $20.5\,\mathrm{Ah}$ and a high-frequency resistance R_0 of around $1.05\,\mathrm{m\Omega}$. Each cell's actual capacity, solid and electrolyte conductivities, solid and electrolyte diffusivities, reaction-rate constants, and double-layer parameter values were all slightly randomized around nominal values, and simulations for a total of 100 randomized cells were run.

Each simulation began with a $5\,\mathrm{s}$ rest, followed by a 10C discharge pulse ($205\,\mathrm{A}$, based on nominal capacity) for $30\,\mathrm{s}$, followed by a $25\,\mathrm{s}$ rest. Voltage traces that resulted from these simulations are presented in Fig. 1(a). There is a clearly observable range of responses, including one significant outlier. Nothing intentional was done to force this outlier; it simply resulted from the randomized parameter values. Fig. 1(b) shows a box plot of the parameter ranges of the cells, indicating with "×" the parameters of the outlier cell for reference; the parameter that contributed most to this cell being an outlier was a low solid-diffusivity value in the negative electrode. Note that while the outlier cell is very evident from its response to a high-current pulse, it is not evident from its response to a low-current pulse. Detection of the outlier is the topic of Sect. 4.

From these high-current simulations, we can compute the true value we desire to estimate using ML:

$$R_{30}^{\mathrm{high}} = \frac{4.2\,\mathrm{V} - v_{\mathrm{min}}}{205\,\mathrm{A}} \times 1000\,[\mathrm{m\Omega}],$$

where v_{min} is the minimum pulse-response voltage, located at the end of the 30-s discharge pulse.

3 Machine learning approach

ML methods are automated ways to use a training data set to find the tunable parameters of an equation that estimates some output of interest. Our proposed approach uses a very simple form of ML, linear regression.

Using R_0: We begin by selecting the inputs to the estimator, which are called the "features". The first feature data set considered using the combination of a constant input "1" and the value of R_0 measured during the formation process. So, we can say that

$$\phi = \begin{bmatrix} 1 & R_0 \end{bmatrix}.$$

The model for estimating R_{30}^{high} using this feature vector is

$$\widehat{R}_{30}^{\text{high}} = \phi x,$$

where $\widehat{R}_{30}^{\text{high}}$ is an estimate of R_{30}^{high} and x is a 2×1 vector of parameter values to be determined by the ML method.

Data from the first 80 cells formed the training data set and data from the remaining 20 cells formed the testing data set (the outlier was an element of the training data set). The training data were organized as

$$\Phi = \begin{bmatrix} 1 & R_0^{(1)} \\ \vdots & \vdots \\ 1 & R_0^{(80)} \end{bmatrix},$$

where the superscript on R_0 denotes the cell number. The computed outputs of the training set were

$$Y = \begin{bmatrix} R_{30}^{\text{high},(1)} \\ \vdots \\ R_{30}^{\text{high},(80)} \end{bmatrix}.$$

Then, the optimized vector of parameter values was computed as

$$x = (\Phi^T \Phi)^{-1} \Phi^T Y,$$

which is the standard least-squares solution. Then, estimates of the high-current resistance for the validation set were found via

$$\widehat{R}_{30}^{\text{high},(\text{N})} = \begin{bmatrix} 1 & R_0^{(N)} \end{bmatrix} x$$

for $81 \leq N \leq 100$.

Using R_0 and Q: The same method is applied using different feature vectors ϕ. The next feature vector investigated included a constant, R_0, and cell capacity Q. In this case,

$$\phi = \begin{bmatrix} 1 & R_0 & Q \end{bmatrix}$$

and the remaining steps are the same, except that x is now 3×1 in size.

Using R_0, R_{30}, and Q: The next feature vector also included the cell's low-current R_{30}, defined as

$$R_{30}^{\text{low}} = \frac{4.2\,\text{V} - v_{\min}}{2.05\,\text{A}} \times 1000\,[\text{m}\Omega].$$

That is, the method for computing R_{30}^{low} is the same as the method for computing R_{30}^{high}, except that the test run to determine R_{30}^{low} used a C/10 rate instead of a 10C rate (it used the same cell parameters, but a different input-current magnitude). The revised feature vector was

$$\phi = \begin{bmatrix} 1 & R_0 & R_{30}^{\text{low}} & Q \end{bmatrix}.$$

Using R_0, R_{30}, Z_{30}, and Q: The next feature vector included EIS-determined cell impedance at 1, 3, 5, 7, 9, and 11 times the fundamental frequency of 1/60 Hz. Since impedance is a complex number at any frequency and since we desire to use real-valued regression, we separate the impedances into real and imaginary parts. Therefore, the feature vector was

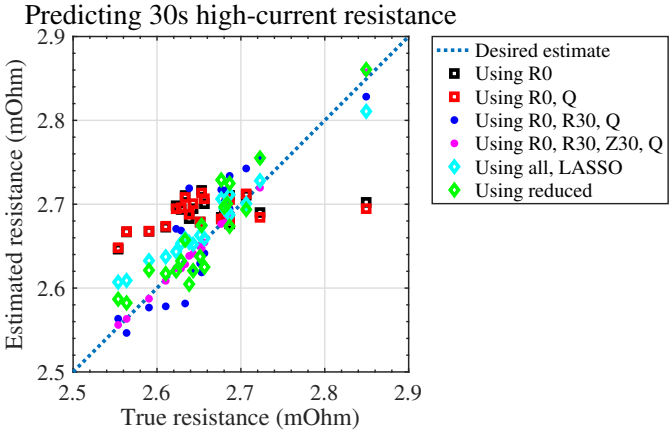
$$\phi = \begin{bmatrix} 1 & R_0 & R_{30}^{\text{low}} & \text{real}(Z_1) \cdots \text{real}(Z_{11}) & \text{imag}(Z_1) \cdots \text{imag}(Z_{11}) & Q \end{bmatrix}.$$

Using all with LASSO: The feature vector has now grown to 16 elements in size. This is not excessively large but it would be interesting to see if it could be trimmed. The LASSO method modifies the regression technique used to determine x such that it penalizes having too many components in x [9]. It does this by setting values in x to zero if they have only a small predictive component.

We applied the LASSO method to the data set and discovered that three elements of the previous 16-length feature vector were most significant: these were R_{30}^{low}, $\text{imag}(Z_1)$, and Q. It is very interesting that the high-frequency resistance R_0—currently used for cell sorting—was not one of these important features! Results presented shortly for the LASSO method used only these three features.

Using reduced: Finally, we tried one more feature vector, somewhat enlarging the LASSO feature vector:

$$\phi = \begin{bmatrix} R_{30}^{\text{low}} & \text{real}(Z_1) & \text{real}(Z_3) & \text{imag}(Z_1) & \text{imag}(Z_3) & Q \end{bmatrix}.$$



Method uses:	RMSE (mΩ)
R_0	0.063441
R_0, Q	0.064713
R_0, R_{30}^{low}, Q	0.035586
R_0, R_{30}^{low}, Z_{30}, Q	**0.003457**
All, with LASSO	0.025869
Reduced set	0.025065

Figure 2: Validation results using the six estimators.

Results when estimating R_{30}^{high}

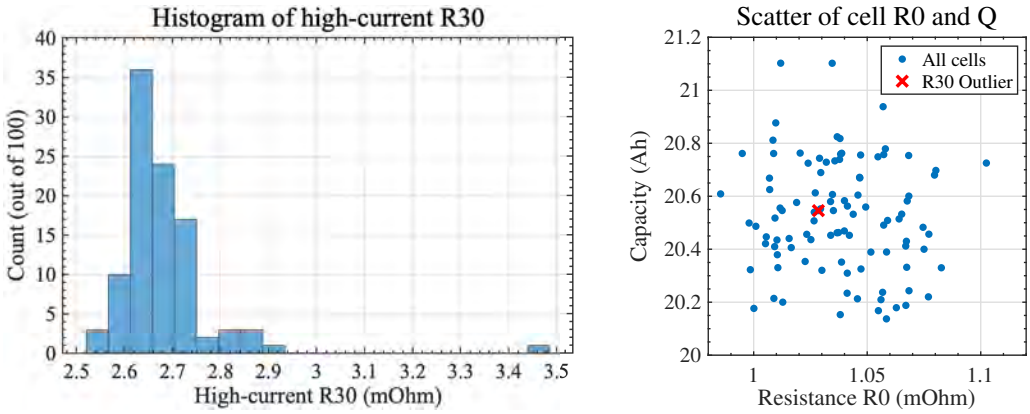
Six different estimators were trained using data from 80 cells and then tested on data from the remaining 20 cells. The objective is for $\widehat{R}_{30}^{\text{high}}$ to match R_{30}^{high} as closely as possible. Fig. 2 illustrates the validation results. An ideal outcome would be for all of the markers to fall directly on the blue dashed line where true and estimated resistances are the same.
Examining the figure, we see that estimates using only R_0 or using both R_0 and Q performed about the same. But, neither performed especially well. These results are indicative of the cell matching that would be performed today. All other matches are substantially better, with the estimates produced using R_0, R_{30}^{low}, Z_{30}, and Q being exceptionally good. Numeric results in terms of root-mean-squared estimation errors are tabulated in the table drawn in Fig. 2. If computation is a concern, the "using reduced" feature vector produces reasonable results using very few features.

4 Outlier detection

Fig. 1(a) clearly shows that one cell behaves differently from the others; it has a much higher 30-s resistance. In order to investigate this further, we plotted the histogram of all cell 30-s resistances, shown in Fig. 3(a). Here, we very clearly see the single outlier, having a 30-s resistance of $3.48\,\text{m}\Omega$. From the histogram, we note that any 30-s resistance above about $3\,\text{m}\Omega$ might be considered an outlier.
Can this outlier be detected using knowledge of only R_0 and Q, the typical information presently available during screening? Fig. 3(b) presents a scatter plot of Q versus R_0 for all 100 cells that were simulated. It would be convenient if the cell having excessive R_{30}^{high} were easily discernible from this plot, but it is not. The red "$\times$" symbol shows the cell having an excessive R_{30}^{high}, and it is right in the center of the scatter. Knowing only R_0 and Q, we cannot tell that this cell is an outlier.
Table 1 lists the estimates of $\widehat{R}_{30}^{\text{high}}$ for the outlier cell produced by the ML estimators using different feature sets at input. Comparing the estimates from this table to the true value, we see that only the last three methods have a reasonably close estimate. Further, comparing the estimates made using only R_0 and Q to the histogram in Fig. 3(a), we see that the (incorrect) estimates $\widehat{R}_{30}^{\text{high}}$ produced using only these features are well within the 0–$3\,\text{m}\Omega$ normal range predicted by the histogram, so screening would choose to "pass" this cell when it should be "failed". The estimate based on the feature vector that further uses R_{30}^{low} shows that the cell is borderline, but it would still probably pass the cell when the cell should be failed. Only the ML estimates made with feature vectors that incorporated Z_{30} correctly show that the cell should be failed.

5 Extending to the pack level

We would like to use the ML-predicted $\widehat{R}_{30}^{\text{high}}$ to sort or bin cells to compose an optimized battery-pack layout. To evaluate the performance of a proposed pack layout, we must be able to simulate battery packs



(a) (b)

Figure 3: Histogram of all high-current 30-s resistances. Scatter plot of Q versus R_0 to see whether outlier cell is easily discernible.

Table 1: Detecting the outlier resistance. Boldface entries correctly detect outlier.

Method	Estimate (mΩ)
True value of outlier resistance	3.4822
Using R_0	2.6801
Using R_0, Q	2.6805
Using R_0, R_{30}^{low}, Q	2.9739
Using R_0, R_{30}^{low}, Z_{30}, Q	**3.4765**
Using all, LASSO	**3.2002**
Using reduced	**3.3293**

comprising multiple cells wired in parallel and in series, where the characteristics (parameter values) of the cells are all somewhat different. To evaluate any solution to this question, we need to be able to simulate a high-current pack-level pulse discharge. There are two possible approaches to doing this:

1. We could simulate the battery pack in COMSOL. This would require somehow developing ways to couple multiple PDE-based cell models together to make a pack model. This approach was rejected for two reasons:
   (a) The development time might be enormous: we would need to develop new battery-pack PDE models in COMSOL, and we would need to validate these models somehow.
   (b) This approach would not be practical in an EOL cell tester since the physics-based model parameter values are unknown and since COMSOL simulations run slower than real time.
2. We could somehow create equivalent-circuit models (ECMs) of the cells and put those together to simulate a battery pack. This has some attractive features:
   (a) If we can find a way to make an ECM using only data that have already been collected, the model can be used to predict any load profile quite readily.
   (b) There are existing tools to simulate battery **packs** using ECMs of **cells** having different characteristics [10]. This enormously speeds development time.
   (c) ECMs run quickly: This approach is feasible for implementing in an EOL tester in real time.

The major development hurdle is to find a way to translate the data we have already collected—namely, R_0^{low}, Z_{30}, Q, and the voltage profile measured during the low-current 30 s pulse—to develop an ECM that can accurately predict a high-current pack discharge event. This is the first aspect we address.

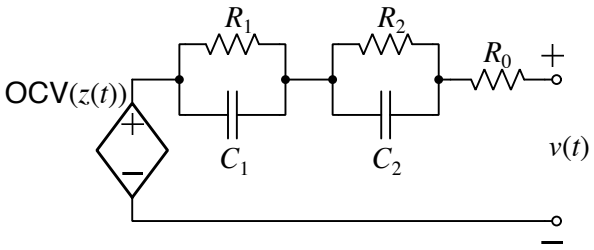


Figure 4: The form of equivalent-circuit model assumed here.

5.1 Making an ECM using available data

To make an ECM for every cell coming off the production line, we assume that we know the nominal OCV versus SOC relationship and that we know every cell's capacity Q. (We suspect that using a fixed nominal capacity Q_{nom} would be sufficient, however.) We also assume that we will use the previously described ML methods to make estimates of $\widehat{R}_{30}^{\mathrm{high}}$. The primary data set used to create the model is the current/voltage profile for the low-current 30 s discharge test.
The cell model that we assume is illustrated in Fig. 4 and has the following equations:

State of charge. State of charge at discrete-time index k is denoted as z_k. It evolves according to

$$z_{k+1} = z_k - i_k \Delta t / Q, \tag{1}$$

where i_k is the input current at time k, Δt is the *sample period*, and Q is the cell *total capacity*. SOC is unitless, so if i_k is measured in amperes and Δt is measured in seconds, then Q must be expressed in ampere-seconds.

Diffusion-resistor current. Current through the resistor R_j in the jth resistor–capacitor network at discrete-time index k is denoted as $i_{R_y,k}$ and evolves as

$$i_{R_j,k+1} = \exp\left(\frac{-\Delta t}{R_j C_j}\right) i_{R_j,k} + \left(1 - \exp\left(\frac{-\Delta t}{R_j C_j}\right)\right) i_k. \tag{2}$$

This term models the slow time constants of diffusion processes occurring within the cell.

Voltage. Voltage v_k at discrete-time index k is computed as

$$v_k = \mathrm{OCV}(z_k) - \sum_j R_j i_{R_j,k} - R_0 i_k, \tag{3}$$

where $\mathrm{OCV}(z_k)$ is the OCV as a function of SOC, and R_0 is the pure ohmic resistance of the cell.

To fit an ECM to a specific cell, we must somehow determine the values for R_0, R_k for $1 \le k \le 3$, and C_k for $1 \le k \le 3$, assuming that Q and the OCV relationship are already known. The values of these resistances and capacitances are found from the data collected during the low-current 30 s discharge test (one example of which is plotted as the solid blue line in Fig. 5):

- First, find SOC for all points in the test using Eq. (1).
- Then, find OCV for all points in the test using $\mathrm{OCV}(z_k)$.
- Isolate the portion of the test following the 30 s discharge when the cell is resting. Then, Eq. (2) is homogeneous (zero-input) and has geometrically (exponentially) decaying behaviors.
- Use an exponential-fitting function [11] to fit an exponential relationship to this voltage decay. This gives us the R–C time constants of the model.
- Simulate the entire profile (rest+discharge+rest) using these time constants to produce data for z_k, $i_{R_1,k}$, $i_{R_2,k}$, and $i_{R_3,k}$. Use these profiles in a linear regression to determine R_0, R_1, R_2, and R_3. This model fits the low-current measured data very well (see dashed blue line in Fig. 5).
- There is a problem when applying this low-current model to the high-current pulse. The red solid line in Fig. 5 displays the voltage response of the cell to a high-current pulse. The red dashed line shows the voltage response that the low-current ECM predicts for that test. The relaxation period is well modeled and the time constants are well modeled, but the overall high-current resistance is smaller than the low-current resistance and this is evident in the plot by a large voltage mismatch.

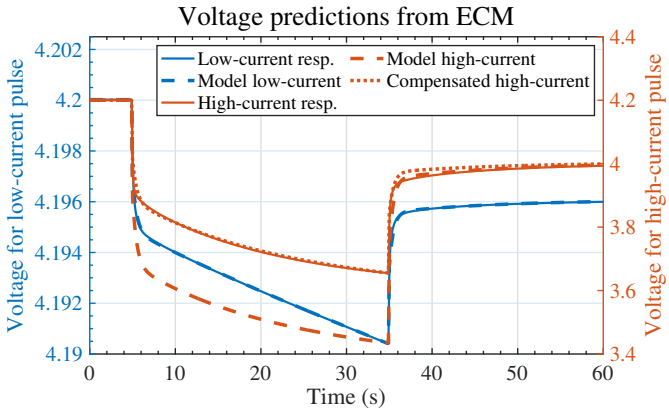


Figure 5: Example of steps to fitting ECM to data.

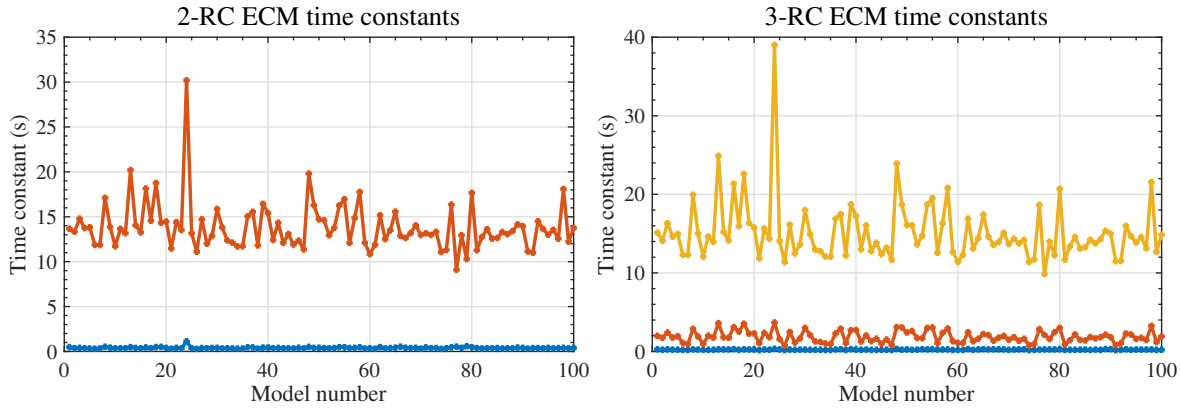


Figure 6: R–C time constants discovered for each of the 100 cells.

- However, we can estimate the high-current $\widehat{R}_{30}^{\text{high}}$ using our ML method (here we assume we do this perfectly) and compare to the ECM's low-current $\widehat{R}_{30}^{\text{low}}$. For high-current simulations, we replace R_0 by $R_0(\widehat{R}_{30}^{\text{high}}/\widehat{R}_{30}^{\text{low}})$ and we replace the other resistances R_n in the R–C circuits by $R_n(\widehat{R}_{30}^{\text{high}}/\widehat{R}_{30}^{\text{low}})$. Doing this produces the dotted red line in Fig. 5.

So, at this point we have created an ECM of a cell fresh off the production line using only measured data that were collected in a couple of minutes of testing. This model is valid only around the range for which the ML method was trained, but should be perfect for simulating a high-current pack discharge current at around $100\,\%$ SOC.
A comment regarding the results from this step. The R–C time constants found for each cell are plotted in Fig. 6. The left plot shows models having only two R–C pairs and the right plot shows models having three R–C pairs. It is very interesting that we can also see the outlier cell (#24) quite clearly in both of these plots. This cell has a much slower time constant than any of the other cells. This feature might also be useful when sorting cells but we have not investigated this possibility.

5.2 How to simulate a battery pack

Once we have created an ECM for every cell in the battery pack, it is relatively straightforward to simulate the performance of a battery pack containing multiple cells wired in series and/or in parallel [10, 12]. To do so, we must first notice something critical about the ECM equations. Some of the equations depend on the present value of input current and others depend only on past values of the input current.
Notice that present values of SOC and the diffusion-resistor currents depend only on past input-current values. For example, SOC at time $k+1$ depends only on input current at time k, and through recursion, input current at times prior to k. Notice that voltage depends on present and past values of input current.

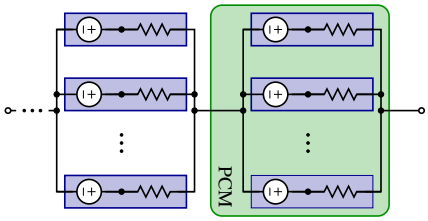


Figure 7: Schematic of a battery pack where modules comprise multiple cells wired in parallel.

Overall, we can write

$$v_k = \underbrace{\text{OCV}(z_k) - \text{diffusion}(i_{R_j,k})}_{\text{not a function of instantaneous current}} - R_0 i_k. \tag{4}$$

When we place multiple cells in parallel, their terminal voltages v_k must be equal according to Kirchhoff's voltage law. The total current passing through the battery pack must be split among parallel cells in order to enforce this law—in general, the split will not be equal among all cells because of differing R_0, differing z_k, and/or differing diffusion currents.

When pack current changes, the OCV and diffusion voltages cannot change instantly. What can change instantly is the voltage drop across R_0 in each cell. This information is key to unlocking a method to simulate cells connected in parallel.

To see how to simulate a battery pack comprising modules of parallel cells (PCMs), we refer to Fig. 7. In the figure, each blue rectangle delineates one battery cell. As discussed in conjunction with Eq. (4), each cell's voltage can be modeled as having a fixed part that does not depend on the present cell current, and a variable part that does depend on present cell current. Fig. 7 symbolizes the fixed part as a voltage source and the variable part as a resistance in each cell. Therefore the voltage source in the figure is not only OCV but also includes present diffusion voltages.

By Kirchhoff's voltage law, all terminal voltages of cells connected in parallel must be equal; by Kirchhoff's current law, the sum of currents through all cells connected in parallel must equal the total battery-pack current. We define current through cell j of a PCM at time k as $i_{j,k}$, its fixed voltage as $v_{fj,k}$, the PCM overall voltage as v_k, and the resistance of the jth cell as $R_{0,j}$. Then, Ohm's law applied across each cell's resistance gives cell current as

$$i_{j,k} = \frac{v_{fj,k} - v_k}{R_{0,j}}, \tag{5}$$

which could be computed if we were able to find v_k.

We arrive at the total battery-pack current by summing all parallel cell currents

$$i_k = \sum_{j=1}^{N_p} \frac{v_{fj,k}}{R_{0,j}} - v_k \sum_{j=1}^{N_p} \frac{1}{R_{0,j}}.$$

By rearranging, we can solve for the PCM voltage

$$v_k = \frac{\sum_{j=1}^{N_p} \frac{v_{fj,k}}{R_{0,j}} - i_k}{\sum_{j=1}^{N_p} \frac{1}{R_{0,j}}}. \tag{6}$$

To summarize, we first evaluate Eq. (6) to determine each PCM's terminal voltage. Then, we compute the individual cell currents using Eq. (5). Once we have the independent cell currents $i_{j,k}$, we can update the cell-model state associated with each cell. Pack voltage is computed by summing all PCM voltages.

6 Packs constructed using random layout

The battery-pack simulations we report here are based on forming different 7S3P packs from subsets of the 100 cells already described. In terms of possible permutations of these cells to build battery packs of size 21, there are "one hundred choose twenty-one" or about 2×10^{21} possible combinations. It is not feasible to simulate all of these combinations in an exhaustive way, so we generated representative results by generating battery packs by randomly selecting groups of 21 cells from the 100. Overall, we simulated 10,000 random combinations.

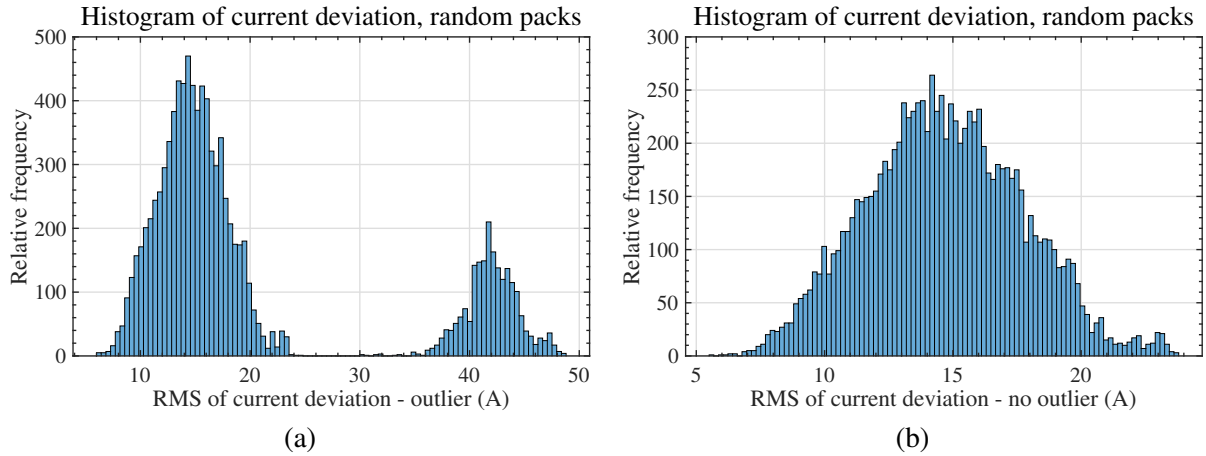


Figure 8: Histograms cell-current deviation for 10,000 pack simulations: in (a) the "outlier cell" was permitted to be in a pack configuration; in (b), it was not.

Packs were organized by placing three cells in series to make "parallel-cell modules" or PCMs. Then, seven PCMs were wired in series. The applied-current profile comprised 5 s of rest, 30 s of 615 A discharge current (a 10C rate), and 25 s of rest. The pack began with all cells at 100 % SOC.
Each simulation produced twenty-one profiles of current versus time (one per cell) and seven profiles of voltage versus time (one per PCM). A profile of "current deviation" versus time was computed as the maximum cell current minus the minimum cell current in every PCM—there were seven of these current-deviation profiles per simulation, one per PCM. As a summary statistic, the RMS value of these current deviations was computed and recorded.
Fig. 8(a) displays histograms of current and voltage deviation for the same 10,000 randomly organized battery packs (from the 100 fixed cells). We desire small current deviation to extend life according to Gogoana et al. [5]. We do not report voltage deviation here but we also desire small voltage deviation so that one PCM does not unduly limit performance of the overall battery pack (e.g., if one PCM discharges to a much lower voltage than the others).
In the figure, we see bimodal distribution of current. We wondered whether this might be caused because some packs include the "outlier cell" and other packs do not contain this outlier.[1] To test this conjecture, we simulated an additional 10,000 randomly organized battery packs from the 99 non-outlier cells. Histogram results of these simulations are shown in Fig. 8(b). The current-deviation distribution is no longer bimodal, showing that the culprit was the outlier cell. This cell should definitely not be used in a battery pack. The distribution is still fairly wide—can this be improved? We propose a simple cell-sorting method to see if its results are better.

7 Packs constructed using sorted layout

Using the same 100 cells, we considered the already-computed C/10 30 s pulse resistance R_{30}^{low} for every cell. We then sorted the cells in order, from lowest resistance to highest resistance. We created one pack with cells 1 to 21; the second pack with cells 2 to 22; the third with cells 3 to 23, and so forth. Overall, this gives $100 - 21 + 1 = 80$ distinct packs that we simulated with the same profile.
Fig. 9 shows results from these simulations. Since there are far fewer results, we can display all simulation outputs instead of only statistical results as was needed for the random battery packs.
The top-left frame shows current deviations for all simulated scenarios superimposed. There is no deviation for the first 5 s when the pack is resting. There is an abrupt deviation upon applying the discharge pulse, which quickly decreases for most packs but then tends to climb slightly over time until the pulse is removed. When the pack is allowed to rest, there is a further transient, which then decays toward zero.

[1] It is possible to derive the probability of choosing 21 non-outlier cells out of the 100 total cells—the answer turns out to be 79 %. It seems reasonable that about 79 % of the histogram area is in the lower-RMS deviation mode and that about 21 % is in the upper-RMS deviation mode.

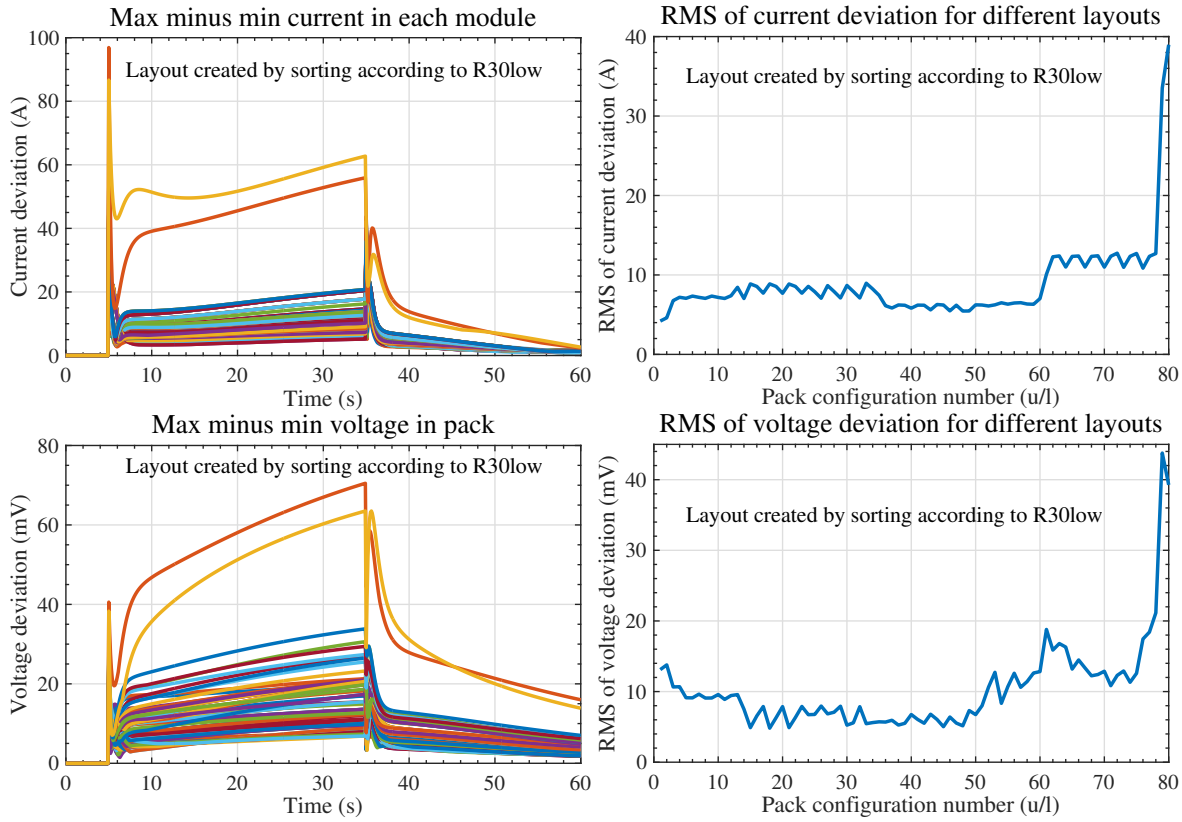


Figure 9: Sorting according to R_{30}^{low}.

We see that two of the 80 traces are visually distinct; these contain the second-highest R_{30}^{low} cell, which is the outlier (not readily detected using R_{30}^{low} only).
The top-right frame shows the RMS value of each of these traces (computed over the entire simulation time) plotted versus pack number. Most of the results have an RMS current deviation of less than 10 A, but a few are a little higher than that, and the final two (which include the outlier cell) are substantially higher. If we remove the outlier cell, all RMS currents are less than about 13 A. Comparing with Fig. 8, we see that this is an excellent result—the worst-case sorting result is much better than the average random result.
However, it is a little odd that the outlier cell is not the worst-case R_{30}^{low}. We recognize that the outlier cell might be better discovered by sorting according to R_{30}^{high}, which is the 30 s 10C pulse resistance instead of C/10 pulse resistance. Here, we used the actual value of R_{30}^{high}, but we have seen that it can be estimated well using the ML approach presented before.
Using the same 100 cells, we computed the 10C 30 s pulse resistance R_{30}^{high} for every cell. We then sorted the cells in order, from lowest resistance to highest resistance. We created one pack with cells 1 to 21; the second pack with cells 2 to 22; the third with cells 3 to 23, and so forth. Overall, this gives $100 - 21 + 1 = 80$ distinct packs that we simulated with the same profile.
Fig. 10 shows results from these simulations. These are organized in the same format as for Fig. 9. Some quick observations include:

- The outlier cell is now indeed the final cell, and is included in only one of these tests. It causes much higher current and voltage deviation in that test, so the cell should be discarded.
- As a result, only one of the current-deviation traces is visually much different from the others. If the outlier is removed, the current deviations are all very similar.
- RMS current deviation is now hovering mostly around 6 A, which is even better than when sorting according to R_{30}^{low}. Compared with the histogram for randomly generated packs, **all** of these packs are in close competition for the best.

While we have no proof that this method of laying out a battery pack is the "best" approach, it does produce high-quality results, where all packs that do not use the outlier cell have very similar cell-current

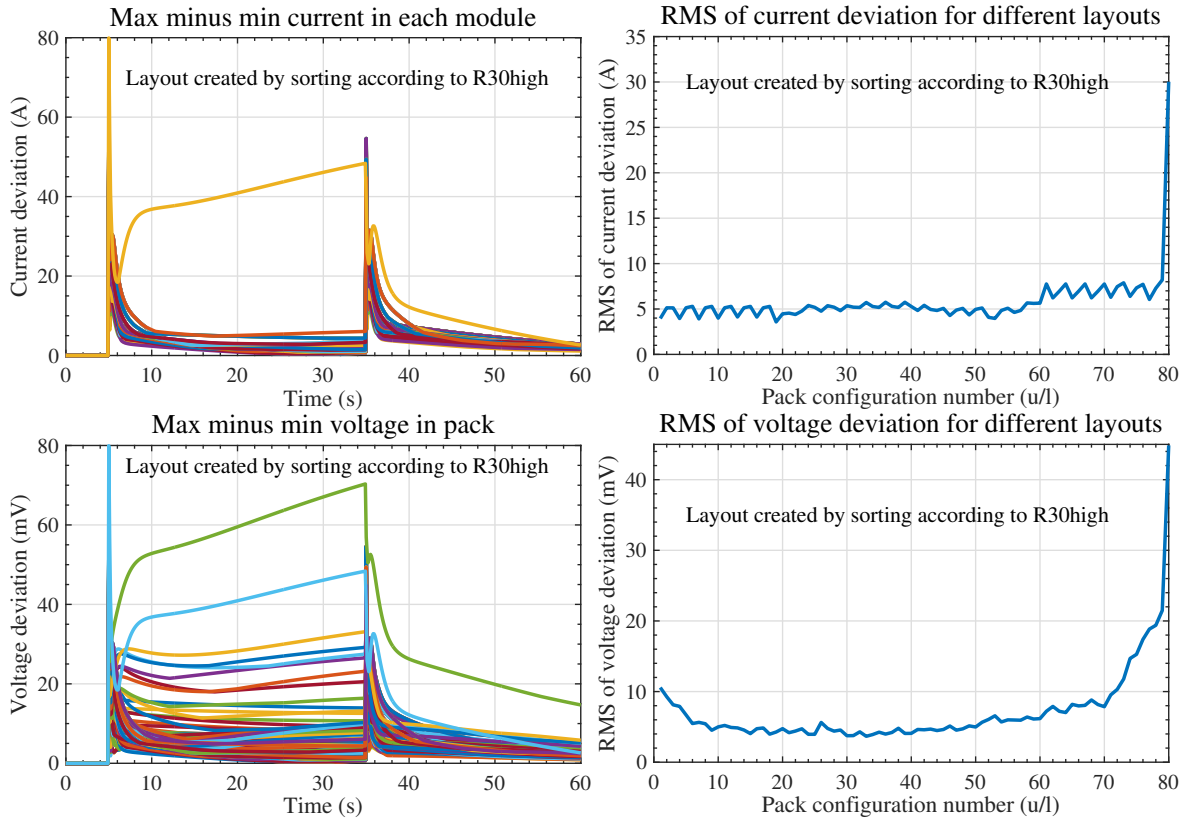


Figure 10: Sorting according to R_{30}^{high}.

and PCM-voltage profiles.

8 Conclusions

This paper has proposed an ML method that uses input features that can be measured quickly using low-current inputs to predict the high-current pulse resistance of a cell. This information can be used to detect outlier cells and to create pack layouts that (appear to) minimize current deviations in the battery pack.
Several general conclusions can be drawn:

- It is never good to include the outlier cell in a battery pack. The histograms from the randomly generated packs show this quite clearly, and the results from the sorted packs concur. The outlier cell should be discarded/recycled.
- Sorting by R_{30}^{low} produces much better results than for unsorted packs. This measurement can be made very quickly, so would not slow down the pack-construction line.
- Sorting by R_{30}^{high} produces the best results of anything that we attempted. This value must be estimated using the ML approach from values of R_{30}^{low} and EIS impedances that can also be measured rapidly by an EOL cell tester.
- According to Gogoana et al. [5], minimizing current variation will extend life; therefore, we believe that sorting by either R_{30}^{low} or R_{30}^{high} promises to extend life substantially. We expect that laying out a pack according to sorted resistances will even out heat generation, which is a desirable feature.

Acknowledgments

This material is based upon work supported by the U.S. Government under Contract No. W56HZV-20-C-0018.

References

[1] D. Shin, M. Poncino, E. Macii, and N. Chang, "A statistical model-based cell-to-cell variability management of li-ion battery pack," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 2, pp. 252–265, 2014.

[2] A. Abdellahi, S. Atalay, and A. Rajan, "Impact of cell variability on pack statistics for different vehicle segments," *Journal of Power Sources*, vol. 508, p. 230246, 2021.

[3] X. He, G. Zhang, X. Feng, L. Wang, G. Tian, and M. Ouyang, "A facile consistency screening approach to select cells with better performance consistency for commercial 18650 lithium ion cells," *Int. J. Electrochem. Sci*, vol. 12, no. 11, pp. 10 239–10 258, 2017.

[4] J. Kim, J. Shin, C. Chun, and B. Cho, "Stable configuration of a li-ion series battery pack based on a screening process for improved voltage/soc balancing," *IEEE Transactions on Power Electronics*, vol. 27, no. 1, pp. 411–424, 2011.

[5] R. Gogoana, M. B. Pinson, M. Z. Bazant, and S. E. Sarma, "Internal resistance matching for parallel-connected lithium-ion cells and impacts on battery pack cycle life," *Journal of Power Sources*, vol. 252, pp. 8–13, 2014.

[6] M. Doyle, T. F. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *J. Electrochemical Society*, vol. 140, no. 6, p. 1526, 1993.

[7] J. Newman and K. E. Thomas-Alyea, *Electrochemical systems*. John Wiley & Sons, 2012.

[8] G. L. Plett, *Battery Management Systems, Volume 1: Battery Modeling*. Artech House, 2015.

[9] R. Muthukrishnan and R. Rohini, "Lasso: A feature selection technique in predictive modeling for machine learning," in *2016 IEEE international conference on advances in computer applications (ICACA)*. IEEE, 2016, pp. 18–20.

[10] G. L. Plett, *Battery Management Systems, Volume 2: Equivalent-Circuit Methods*. Artech House, 2015.

[11] S. D. Foss, "A method of exponential curve fitting by numerical integration," *Biometrics*, pp. 815–821, 1970.

[12] G. L. Plett and M. J. Klein, "Simulating battery packs comprising parallel cell modules and series cell modules," in *Proc. of EVS*, 2009, pp. 1–17.

Authors

Dr. Plett received his Ph.D. in Electrical Engineering from Stanford University and is now Professor of Electrical and Computer Engineering at the University of Colorado Colorado Springs. His research focuses on topics in control systems as applied to the management of high-capacity battery systems, such as found in electric vehicles. Current research efforts include: physics-based reduced-order modeling of ideal lithium-ion dynamics and degradation mechanisms; nondestructive parameter estimation for physics-based models; estimation of cell electrochemical and degradation state; state-of-charge and state-of-health estimation; life-extending power-prediction methods.

Steve Weiss is the president of Xilectric. Xilectric develops technologies to improve the capacity retention of Li-ion batteries focusing on both material development and characterization techniques. Xilectric's Li-ion characterization techniques include tools for materials development and end-of-line tools for quality control in a manufacturing environment. Dr. Weiss received his Ph.D. in Chemical Engineering from the Massachusetts Institute of Technology where he was an NSF Fellow. Dr. Weiss graduated Summa Cum Laude with a B.S. degree in Chemical Engineering from UIUC where he was named to the Bronze Tablet.