

Towards Scalable Monitoring and Maintenance of Rechargeable Batteries

Aaron Mills, Joseph Zambreno

Electrical and Computer Engineering, Iowa State University, Ames, Iowa, USA

Email: {ajmills, zambreno}@iastate.edu

Abstract—Current research on State-of-Charge (SOC) tracking for rechargeable batteries focuses primarily on analyzing batteries consisting of a single cell, or otherwise treat a set of series-connected cells as a homogeneous unit. Yet, as the number of series-connected cells per battery increase, so does the challenge of ensuring safe and efficient operation over a potentially long period of deployment. Cell-level energy balancing is commonly proposed as a means to address the effects of cell property mismatch. However, no comprehensive solution exists addressing the need to maintain SOC accuracy over the full life of a large battery, while also managing the energy imbalance which develops between cells. If poorly managed, this imbalance can reduce usable capacity and lifespan. This paper proposes an integrated solution to these various issues by tracking SOC on a per-cell basis and applying SOC to a cell-balancing application. The effectiveness is demonstrated using a custom test platform.

I. INTRODUCTION

Batteries using high energy density cells such as NiMH or Li-ion require some degree of automated management to meet the goals of maximal *capacity availability*, maximal *longevity*, and maximal *safety* and *fault tolerance*. A significant research challenge in meeting these goals using automated systems is that the difficulty tends to increase as a function of the number of cells within a battery due to a well-known phenomenon: the mismatch of constituent cell properties [1]. Initially, each cell in a string of cells is manufactured with slightly different parameters, resulting in an overall pack capacity that is constrained by the weakest cell in order to ensure a safe operational range. Furthermore, over time, cell properties diverge as a product of their environment and usage pattern [2, 3]. The general effect is that the accessible battery capacity declines more quickly than the rating provided by the cell manufacturer after original testing at the factory. If one could maintain each cell's state at roughly the same level, the behavior of the pack as a whole will much more closely resemble that of an individual cell.

To guide the actions of any battery management system, typically a simple, static model is derived from the cell datasheet at system design time as a means to estimate battery State of Charge (SOC) and State of Health (SOH). We use these terms to indicate a 0% to 100% fuel-gauge of the energy contained in the cell, and the current capacity of the cell (based on its age), respectively. Two (not mutually-exclusive) means to determine SOC are Coulomb Counting and model-based approaches, of which here only the Kalman Filter is considered. Broader coverage of these topics appears in [4].

A. Coulomb Counting

Coulomb-counting is by far the most popular method to estimate SOC due to its conceptual simplicity and minimal

hardware requirements. Typically it involves periodically taking a measurement of the current flowing into or out of the system. In software these measurements are summed over time, thereby approximating the integration of instantaneous current over time. If one knows the initial cell capacity in amp-hours, one need only divide the summation value at that instant by the initial capacity to estimate SOC.

This method is easy to implement but suffers from several major problems. Errors in the assumed initial capacity, errors due to sensor offset or nonlinearity, and errors due to discrete-time approximation will never be corrected due to the open-loop nature of the mechanism. Differences in cell parameters due to manufacturing variation, and temperature differences among cells are additional complications which cannot easily be accounted for [5, 6].

This method is quite useful for checking other methods over shorter periods of duration. However, if we want to design a cell-balancing system which can remain effective over years and under unpredictable conditions without corrective maintenance, this is an unacceptable margin of error. Over time, a system basing its operations on such input may in fact accelerate the degradation of the cells that it was designed to maintain due to its inability to correctly derive the current state of the cells.

B. Model-based SOC Tracking

A number of papers have been published which apply Kalman filters to estimate state of charge [7, 8, 6]. However, the current body of research has several limitations.

- 1) To our knowledge, these previous approaches do not elaborate on how their techniques might extend from a single cell model to a large battery pack.
- 2) Papers published so far do not adequately consider the impact of variation of parameters within a pack. For example [7] measures the SOC of the pack as a whole, which is only a trustworthy metric when assuming the SOC of each internal cell is equal.
- 3) There is limited research on the effects of realistic measurement and system noise. Some adaptive techniques have appeared such as [7].
- 4) There is also limited research on the effect of system sample rate and jitter.

This paper is primarily concerned with addressing items 1 and 2. We show how SOC can be tracked for each cell individually in a four-cell battery, and how these results can be used in the context of cell balancing.

The rest of the paper is organized as follows. In Section II we describe our choice of cell model, our test platform,

testing methodology, and how to characterize the two major components of a cell’s behavior. In Section III we discuss the Kalman Filter approach to SOC monitoring, develop a complete system model, and test the real-world performance of the filter. In Section IV we demonstrate the use of SOC monitoring in a cell balancing application. In Section V we present some overall analysis of the work, and finally in Section VI we discuss future goals.

II. CELL MODEL AND PARAMETER EXTRACTION

The second-order cell model shown in Fig. 1 is well-rationalized in [6]. This model strikes a good balance between model complexity and model accuracy. The elements $R_1, R_2, C_1,$ and C_2 determine the transient response of the cell, while R_0 is primarily responsible for the “voltage sag” of a cell under load. This element has a value which changes as the cell ages and thus is often used as a means to determine the cell’s health—for example, in [7]. Here R_0 is assumed constant. Also note that although V_{oc} is shown as a simple voltage source it is actually a complex non-linear function.

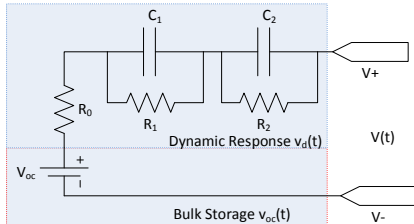


Fig. 1: Cell Model with Second-Order Dynamic Response

Once a cell model is identified, some specific parameters for that model need to be extracted for the cells under consideration. We treat this as two separate tasks: one for the dynamic response, which responds to cell load, and one for determining the SOC-OCV (open circuit voltage) curve for the bulk energy storage of the cell. This is independent of load but depends on factors such as ambient temperature and the number of times the cell has been cycled.

A. Test Platform

The test platform illustrated in Fig. 2 was developed to facilitate parameter extraction, test the performance of the Kalman filter, and test battery balancing algorithms. Voltage sense lines between each cell are not shown to maintain clarity. The major components are listed below.

- 1) Linear Technology’s LTC6803, which is a 12-channel SPI-controlled battery monitor. The device also includes outputs for controlling one switch per cell.
- 2) A current source and a current sink to emulate cell charging and discharging as in the context of a real application.
- 3) A serial-based command interface for controlling the internal board signals and reading sensor data. This interface allows the board to be integrated with Matlab or other PC scripts and controlled in real-time—for example individual discharge resistors can be activated or deactivated remotely.

For this platform we use just a single LTC6803 which allows monitoring and balancing of up to 12 cells. However, it is also possible to daisy chain the LTC6803 devices for a highly scalable solution. In the application, current-mode signaling is used between the bottom device and all upper devices rather than CMOS signaling. Additionally, each upper-level device is referenced to the top of the cell just below it, which limits the common-mode voltage that any channel input sees. Commands and data passes through each device from and to the main controller like a long shift register. This configuration allows one to manage a potentially unlimited number of cells.

B. Method

In order to identify the system, we pulse-tested the cells as proposed by the Hybrid Pulse Power Characterization (HPPC) scheme, described in [9] and illustrated in Fig. 3. This consists of a 18 second discharge pulse, followed by a brief rest period, followed by a 10 second charge pulse. The battery starts fully charged, and this sequence is repeated for every 10% SOC discharged (discharging at $C/1$ between sequences), and after a 1 hour rest. The magnitude of the pulses are dependent on the maximum values specified by the manufacture. The only deviation from the HPPC scheme here is that we use $C/1$ for discharging and $0.75C$ for charging since we are not interested in testing maximum power.

The advantage of using the HPPC scheme is that we can use the resultant data to find both the SOC-OCV curve (by checking V_{oc} during the rest periods) and also identify the dynamic system behavior (though the high-current charge-discharge pulses).

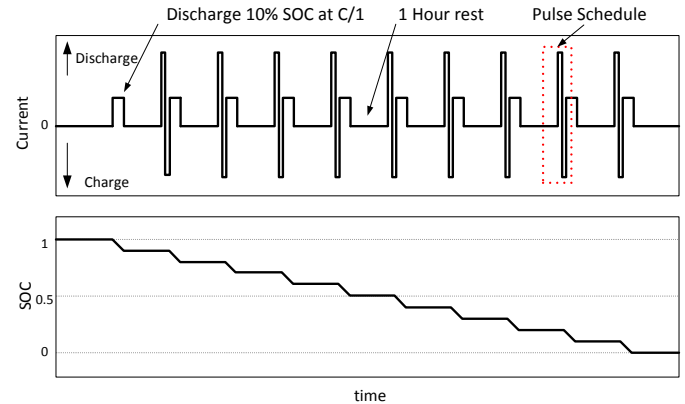
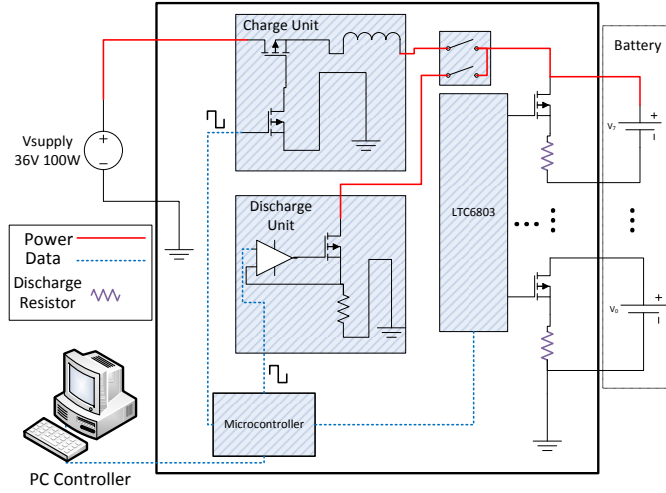


Fig. 3: Cell Characterization Current Profile (Pulse Heights Not To Scale)

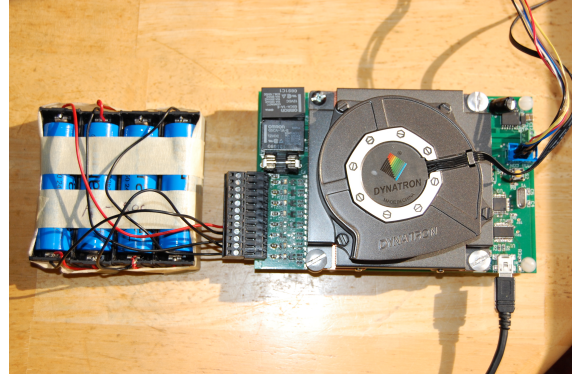
C. Characterizing Dynamic Response

Fortunately, it is not necessary to explicitly determine individual component values from our model in Fig. 1. If we think of the cell as a simple system with one input (the current which is loading it or charging it) and one output (the external terminal voltage), then by using system-identification tools, one can determine the transfer function from collected data.

Here we use the Autoregressive Exogenous (ARX) method of system identification, popularized in [10], to find a transfer function for the part of our cell model involving transient behavior. The general ARX model is shown, with y representing the output and u representing the input.



(a) Test Platform Architecture



(b) Four 18650 2.4Ah Li-Ion Cells Attached to Hardware

Fig. 2: Data Collection Hardware

$$y(t) + a_1(t-1) + \dots + a_{n_a}y(t-n_a) = b_0u(t) + b_1u(t-1) + \dots + b_{n_b}u(t-n_b) + e(t) \quad (1)$$

For our second-order model, since $i(t)$ is our input and $V_d(t)$ is our output, we get the following:

$$(1 + a_1q^{-1} + a_2q^{-2})V_d(t) = (b_1q^{-1} + b_2q^{-2}) \cdot i(t) + e(t) \quad (2)$$

The algorithm for finding system-specific constants a_1, a_2, b_1 , and b_2 given input and output data will not be covered in detail here, but tools to find them are included in many math packages such as Matlab.

1) *Results:* The ARX algorithm is applied to the mean of the four cell voltage responses. This ensures we can generate a “typical case” system model rather than one that is specific to an individual cell. The extracted values for our cells are $a_1 = -0.8835, a_2 = -0.007928, b_1 = 0.02707$, and $b_2 = -0.02219$. The results can be expressed in the form of a standard transfer function:

$$G(s) = \frac{0.02707 - 0.02219s}{s^2 - 0.8835s - 0.007928} \quad (3)$$

The ARX model fit is evaluated using a dataset separate from the training data. The input is a simple, repetitive charge-discharge cycle, shown in Fig. 7b, and the ARX model is used to run a simulation using only this input. A portion is highlighted in Fig. 4a (with DC component removed for clarity). The plot in Fig. 4b shows an error typically less than 1.5mV with noise spikes where the input transitions sharply.

D. SOC-OCV Curve

In order to determine the SOC-OCV curve for our four cells, all cells are initially charged to 4.1V (100% SOC) and balanced to ± 5 mV. Then the same scheme described in Section II-B is followed. The cells are considered fully depleted at 3.1V. SOC is tracked by using classic Coulomb counting and assuming nominal 2.4Ah capacity. The result is a curve with 10 data points, and simple linear interpolation is used to estimate between data points.

III. KALMAN FILTER

At the core of the Kalman filter is a state-space model for the system in question. Our system fits fairly well into the traditional state-space model, which is shown below. Vector x_k is the state vector, u_k is a system disturbance, w_k is the variance-covariance matrix for (assumed white Gaussian) system noise, and v_k is the variance-covariance matrix for (assumed white Gaussian) measurement noise [11].

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

$$y_k = C_k x_k + D_k u_k + v_k \quad (5)$$

The one non-linear component for our system is the SOC-OCV curve. An Extended Kalman Filter (EKF) is a Kalman Filter which includes non-linear terms—typically the notation is modified by absorbing matrices A and B into function $g(k)$ and C and D into function $h(k)$.

$$x_{k+1} = f(x_k, u_k) + w_k \quad (6)$$

$$y_k = g(x_k, u_k) + v_k \quad (7)$$

A. System Equations

With the general state-space model in mind we develop our system equations. Our model consists of three states: the current SOC, the current value for V_d , and the previous value for V_d (as it is a second-order ARX model). We can write this in vector form:

$$x = \begin{pmatrix} SOC(k) \\ V_d(k) \\ V_d(k-1) \end{pmatrix} \quad (8)$$

The recursive computation for SOC shown below is similar to that from the Coulomb Counting method: η_i is the cell’s Coulombic Efficiency (we use 1), Δt is our simulation timestep (ie sensor sample period), and C_k is the nominal cell capacity. Each of these are treated as constants. Finally $i(k)$ is the battery current.

$$SOC(k+1) = SOC(k) + \frac{\eta_i \Delta t}{C_k} i(k+1) + w_s(k+1) \quad (9)$$

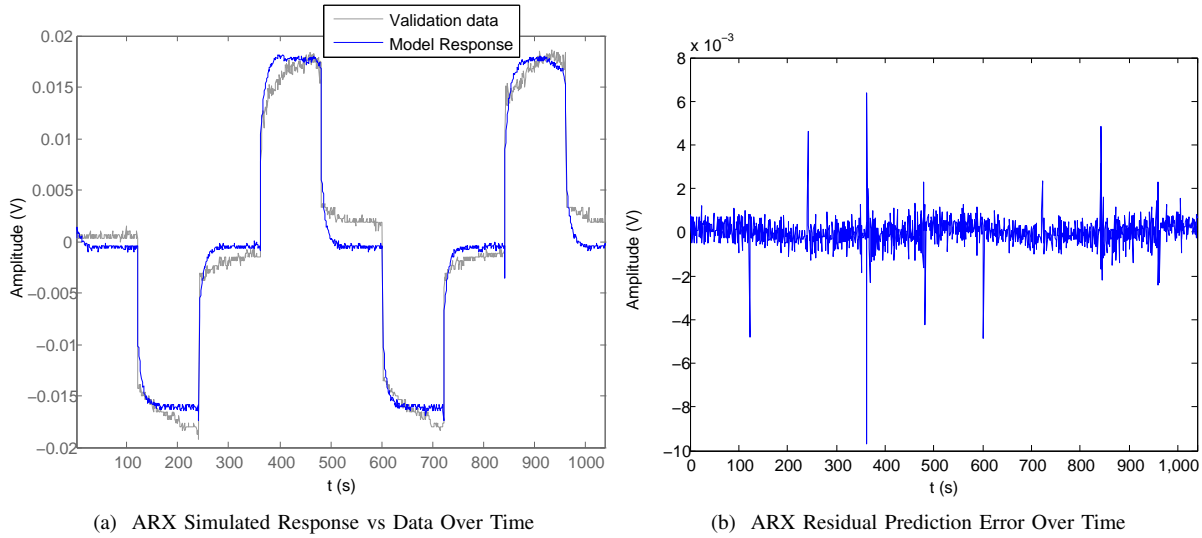


Fig. 4: ARX Model Validation

The dynamic part of the cell model is described next. Note that this scheme requires us to store the previous value of the loading current i :

$$V_d(k+1) = -a_1 V_d(k) - a_2 V_d(k-1) + b_1 i(k) + b_2 i(k-1) + w_{v_d}(k) \quad (10)$$

The third state equation is fairly trivial:

$$V_d(k) = V_d(k) + w_{v_d}(k-1) \quad (11)$$

Finally the output expression is:

$$y(k) = V_{oc}(s_k) + V_d(k) + v(k) \quad (12)$$

Putting together Equations 9, 10, and 11, we develop the matrix form of the state-space model:

$$\begin{pmatrix} SOC(k+1) \\ V_d(k+1) \\ V_d(k) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} SOC(k) \\ V_d(k) \\ V_d(k-1) \end{pmatrix} + \begin{pmatrix} \frac{ndt}{C} & 0 & 0 \\ 0 & b_1 & -b_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i(k+1) \\ i(k) \\ i(k-1) \end{pmatrix} + \begin{pmatrix} w_s(k+1) \\ w_s(k) \\ w_s(k-1) \end{pmatrix} \quad (13)$$

For an EKF algorithm itself we must also define the following. These do not appear in the state-space model but are needed in the EKF algorithm.

$$\hat{A}_k = \left. \frac{\partial f(x_k, u_k)}{\partial x_k} \right|_{x_k = \hat{x}_k^+} \quad (14)$$

$$\hat{C}_k = \left. \frac{\partial g(x_k, u_k)}{\partial x_k} \right|_{x_k = \hat{x}_k^-} \quad (15)$$

Considering Equation 14 we note that our A matrix consists of a linear combination of parameters, so the result of the partial derivative is simply A . For Equation 15, the expression for $V_d(k)$ is also a linear combination of parameters, so we need compute the derivative of $V_{oc}(SOC_k)$. Since $SOC(k)$ consists of a set of measured data points, this task is equivalent to finding the derivative of a set of line segments. The initialization values for the various Kalman matrices are shown in Table I.

B. Kalman Algorithm

In terms of computation, the Kalman filter involves the repeated application of the following steps:

- 1) Estimate state
- 2) Calculate error covariance
- 3) Calculate Kalman gain
- 4) Update state estimate based on measurement
- 5) Update error covariance based on measurement

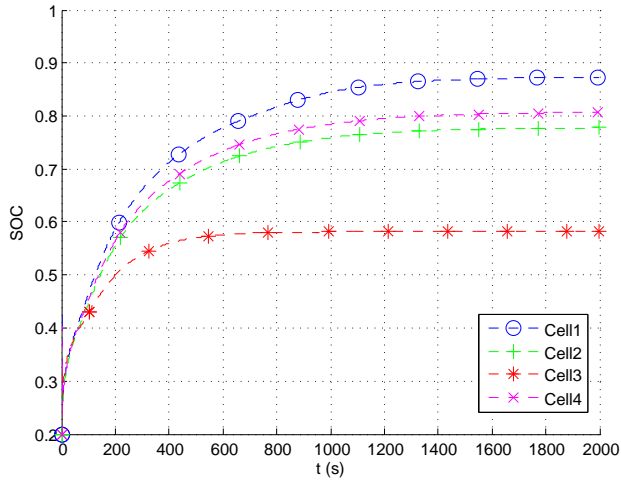
This process is shown more formally in lines 5 through 9 of Algorithm 1.

Algorithm 1 Kalman Filter

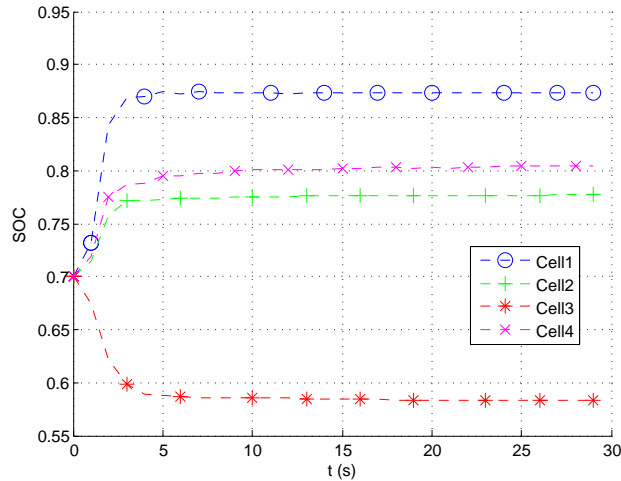
- 1: **procedure** INITIALIZE
 - 2: $\hat{x}_0^+ \leftarrow E[x_0]$
 - 3: $\Sigma_{\hat{x},0}^+ \leftarrow E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]$
 - 4: **procedure** UPDATE
 - 5: $\hat{x}_k \leftarrow f(\hat{x}_{k-1}^+, u_{k-1})$
 - 6: $\Sigma_{\hat{x},k}^- \leftarrow \hat{A}_{k-1} \Sigma_{\hat{x},k-1}^+ \hat{A}_{k-1}^T + \Sigma_w$
 - 7: $L_k \leftarrow \Sigma_{\hat{x},k}^- \hat{C}_k^T [\hat{C}_k \Sigma_{\hat{x},k}^- + \Sigma_v]^{-1}$
 - 8: $\hat{x}_k^+ \leftarrow \hat{x}_k^- + L_k [y_k - g(\hat{x}_k^-, u_k)]$
 - 9: $\Sigma_{\hat{x},k}^+ \leftarrow (I - L_k \hat{C}_k) \Sigma_{\hat{x},k}^-$
-

C. State Initialization

The system state will eventually settle on the true SOC value even when it is initialized poorly. However, if the sample rate is low, for example once per second, this can drastically increase the delay before the SOC measurement is ready for use by the system. As seen in Fig. 5a, when all cells are incorrectly initialized at 0.20 SOC, at 1s sampling, cell 1 takes almost 30 minutes to reach steady-state. The state initialization vector should thus be saved in non-volatile memory periodically and then recovered at system reset to minimize startup delay. As seen in Fig. 5b, even initializing the states to 0.70 reduces the time to steady-state to less than 20s. Conveniently, since the Kalman Error Covariance matrix is updated each iteration, it is also possible to determine when SOC has been acquired when the uncertainty reduces to a given threshold. The uncertainty (variance) for each state is



(a) Acquisition delay for SOC(0)=0.2



(b) Acquisition delay for SOC(0)=0.7

Fig. 5: Impact of SOC state initialization over time

contained in the diagonals of the matrix. The plot in Fig. 6 shows how the uncertainty reduces over the first 10 iterations (seconds) of the algorithm. The bounds represent two standard deviations. Unfortunately, it is possible to make the state “over confident” by underestimating the system noise.

D. Performance Analysis

In Matlab we track the SOC of each cell simultaneously by maintaining an array of structures, one per cell. The various fields are shown in Table I. The column “shared” indicates if a field is common among all cells or is single-instance.

As in Section II-C1 we test our system equations on a separate dataset. Fig. 7b shows the system impulse while Fig. 7a compares the SOC computed by the Kalman filter to a an SOC profile which is generated by simple Coulomb-counting (e.g. integrating current over time).

The maximum error for all cells was 0.0041 and the highest Mean Squared Error was 4.8291×10^{-6} .

IV. CELL BALANCING USING SOC

Compared to directly using terminal voltage, SOC is a superior control variable for input to for battery balancing, as by design, it ignores much of a cells dynamic behavior. Fig. 8

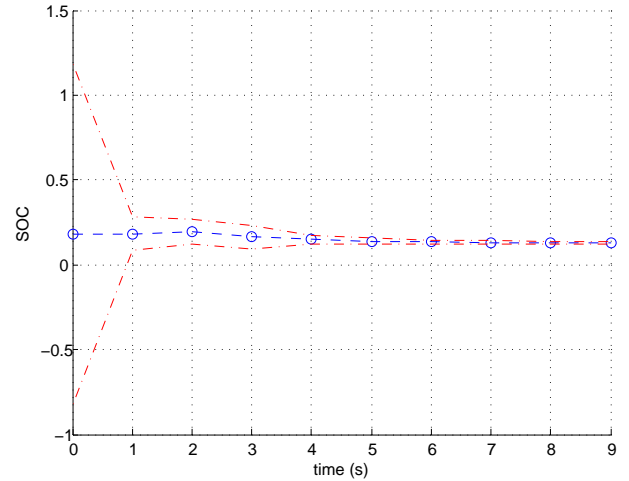


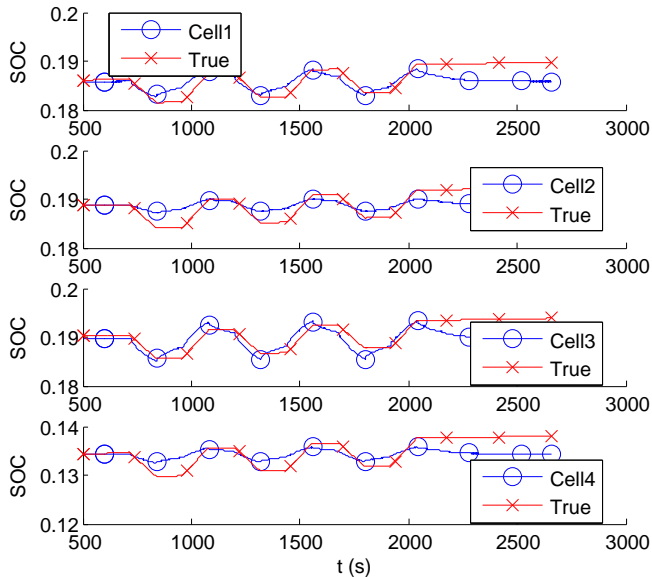
Fig. 6: Reduction in State Uncertainty Over Time

TABLE I: Kalman Filter Variables

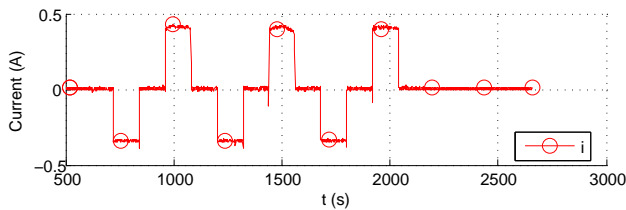
Field	Formal	Meaning	Shared	Initialization
A	A	From state-space model	Y	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 1 & 0 \end{pmatrix}$
B	B	From state-space model	Y	$\begin{pmatrix} \frac{ndt}{C} & 0 & 0 \\ 0 & b_1 & b_2 \\ 0 & 0 & 0 \end{pmatrix}$
C	C	From state-space model	Y	$\begin{pmatrix} Voc(SOC(0)) \\ 1 \\ 0 \end{pmatrix}$
D	D	From state-space model	Y	0
Q	Σ_w	System Noise Covariance	Y	$\begin{pmatrix} 1^{-6} & 0 & 0 \\ 0 & 1^{-3} & 0 \\ 0 & 0 & 1^{-3} \end{pmatrix}$
R	Σ_v	Measurement Noise Covariance	Y	$1 \cdot 10^{-4}$
P	$\Sigma_{\hat{x},k}^+$	Error Covariance	N	$\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{pmatrix}$
x	\hat{x}_k	System state	N	$\begin{pmatrix} E(SOC(0)) \\ 0 \\ 0 \end{pmatrix}$
y	y_k	System output (Estimated V(t))	N	N/A
z	z_k	System input (Measured V(t))	N	$E(V(0))$
u	u_k	System disturbance i(t)	Y	$\begin{pmatrix} i_0 \\ 0 \\ 0 \end{pmatrix}$

shows the initial 2000 seconds of the four initially unbalanced cells being brought into balance. The discharge resistor which can be placed in parallel with each cell by the controller has a value of 33Ω , which typically translates into a discharge current of around 150mA (around C/4).

The balancing algorithm is fairly simple: find the cell with the maximum SOC and activate its discharge resistor. The algorithm terminates when the difference between the maximum and minimum SOC is 0.005 (above our maximum



(a) SOC Tracking for Four Series-Connected Cells



(b) System Impulse

Fig. 7: EKF Response Testing

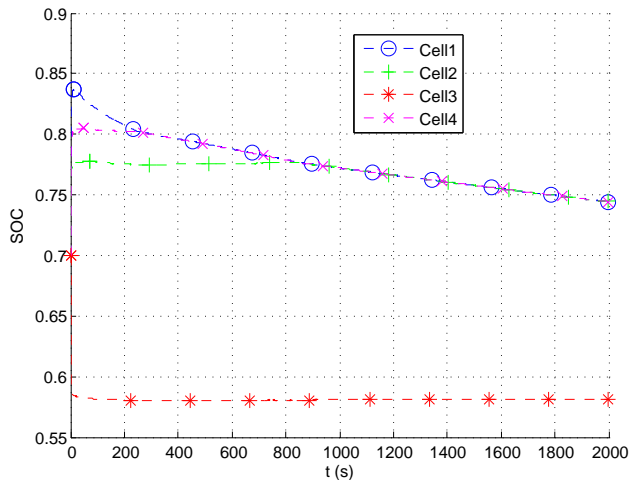


Fig. 8: Cell Balancing using SOC

expected error), at which point all discharge resistors are deactivated. Cell 1 and 4 are balanced after around 200s, these two cells were balanced with Cell 2 at 800s, and all cells were balanced by around 6000s. Balancing speed has an inverse relationship with the value of the discharge resistor, with the side-effect of higher heat generation at higher rates.

V. ANALYSIS

There are a few limitations with the process as described in this work. For one, the cells are assumed to have a fixed, equal

capacity and internal resistance R_0 . Not only do these two parameters vary from cell to cell, but also tend to decrease and increase, respectively, as a cell is repeatedly cycled. Second, temperature was excluded as an input. It is known that the SOC-OCV curve varies with temperature and has also been shown to contribute to cell imbalance [12].

VI. FUTURE WORK

The next major consideration for this project is how to develop a fully-embedded architecture. Hardware implementations of Kalman filters have the advantage of faster update time, lower update jitter, and lower power consumption than the equivalent microprocessor implementation [13].

The Kalman filter implementation is extended that proposed in [14]. This approach folds a typical 2-D systolic array onto a 1-D array to reduce resource requirements. There are more compact implementations, such as that proposed in [13], with the tradeoff of increased design complexity. A full treatment of this topic, including implementation details, is left to future work.

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation (NSF) under award CNS-1116810.

REFERENCES

- [1] D. Shin *et al.*, "A statistical model of cell-to-cell variation in Li-ion batteries for system-level design," in *IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2013, pp. 94–99.
- [2] V. Muenzel *et al.*, "Modeling reversible self-discharge in series-connected Li-ion battery cells," in *IEEE TENCON Spring Conference*, 2013, pp. 470–474.
- [3] C. Sen and N. Kar, "Battery pack modeling for the analysis of battery management system of a hybrid electric vehicle," in *IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2009, pp. 207–212.
- [4] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *ISRN Applied Mathematics*, no. 7, 2013.
- [5] K. S. Ng *et al.*, "Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries," *Applied Energy*, vol. 86, no. 9, pp. 1506 – 1511, 2009.
- [6] S. Yuan, H. Wu, and C. Yin, "State of charge estimation using the extended kalman filter for battery management systems based on the arx battery model," *Energies*, vol. 6, no. 1, pp. 444–470, 2013.
- [7] D. Haifeng, W. Xuezhe, and S. Zechang, "State and parameter estimation of a hev li-ion battery pack using adaptive kalman filter with a new soc-ocv concept," in *Measuring Technology and Mechatronics Automation, 2009. ICMTMA '09. International Conference on*, vol. 2, 2009, pp. 375–380.
- [8] G. L. Plett, "Extended kalman filtering for battery management systems of lipb-based {HEV} battery packs: Part 1. background," *Journal of Power Sources*, vol. 134, no. 2, pp. 252 – 261, 2004.
- [9] I. N. Engineering and E. Laboratory. (2001) Pngv battery test manual. [Online]. Available: http://avt.inel.gov/battery/pdf/pngv_manual_rev3b.pdf
- [10] E. J. Hannon, *Multiple time series*. New York: John Wiley and Sons, Inc, 1970.
- [11] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*. Virginia: American Institute of Aeronautics and Astronautics, Inc, 2009.
- [12] V. Muenzel *et al.*, "Modeling reversible self-discharge in series-connected li-ion battery cells," in *TENCON Spring Conference, 2013 IEEE*, April 2013, pp. 470–474.
- [13] A. Sudarsanam, "Analysis of field programmable gate array-based kalman filter architectures," in *All Graduate Theses and Dissertations*, 2010. [Online]. Available: <http://digitalcommons.usu.edu/etd/788>
- [14] A. Bigdeli *et al.*, "A new pipelined systolic array-based architecture for matrix inversion in fpgas with kalman filter case study," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 75–75, Jan. 2006.