Estimating State of Charge and State of Health of Rechargable Batteries on a Per-Cell Basis

Aaron Mills, Joseph Zambreno
Iowa State University, Ames, Iowa, USA
{ajmills,zambreno}@iastate.edu

***Abstract*—Much of current research on State-of-Charge (SOC) and State-of-Health (SOH) tracking for rechargeable batteries such as Li-ion focuses primarily on analyzing single cells, or otherwise treat a set of series-connected cells as a homogeneous unit. Since no two cells have precisely the same properties, for applications involving large batteries this can severely limit the accuracy and utility of the approach. In this paper we develop an model-driven approach using a Dual Unscented Kalman Filter to allow a Battery Monitoring System (BMS) to monitor in real time both SOC and SOH of each cell in a battery. A BMS is an example of a Cyber-Physical System (CPS) which requires deep understanding of the behavior of the physical system (i.e., the battery) in order to obtain reliability in demanding applications. In particular, since the SOH of a cell changes extremely slowly compared to SOC, our dual filter operates on two timescales to improve SOH tracking. We show that the use of the Unscented Kalman Filter instead of the more common Extended Kalman Filter simplifies the development of the system model equations in the multiscale case. We also show how a single "average" cell model can be used to accurately estimate SOH for different cells and cells of different ages.**

I. Introduction

The State of Charge (SOC) of a rechargeable chemical cell is a proportional to the amount of energy available. It is commonly expressed as a real value from 0 to 1, or as a percent from 0% to 100%, where 0% indicates that the cell cannot be safely discharged any further, and 100% indicates it cannot be charged any further. It is a unitless, normalized metric making it ideal for use as a "fuel gauge" for all kinds of battery technologies. A closely related concept is State of Health (SOH), which is related to the age of a cell. It usually is an estimation of either cell impedance or capacity. For example, a cell could be fully charged (100% SOC), but have severely degraded runtime (low SOH).

Due to the increasing demands of battery-powered applications, much research has been done on developing the architecture of Cyber-Physical systems (i.e. BMS) which monitor and maintain rechargeable batteries [1, 2]. As for determining SOC and SOH, a number of papers have been published which apply a Kalman Filter (KF) to rechargeable cells such as Li-ion or LiPo[3, 4, 5, 6]. This method has been shown capable of very accurate cell SOC estimation in real time.

However, an often-overlooked aspect of SOC estimation is that accurately estimating the SOC for an entire *battery*, even with a highly accurate cell model, cannot be done by simply considering the terminal voltage of that battery. This approach assumes that the constituent cells are always at the same SOC, and also have the same physical cell attributes, such as capacity or internal resistance. Research on model-building has shown this is not the case [7, 8]. This approach is especially ineffective when one wants to accurately predict available power, as over- or under-estimates near the top or bottom of the SOC range could result in some cells being forced outside their safe operational ranges.

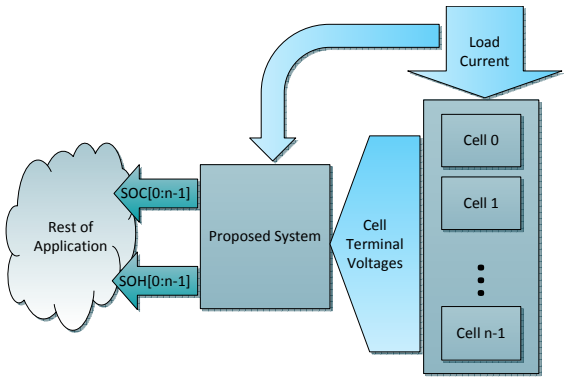


Fig. 1: System Overview. The system samples the voltage on each series-connected cell, and the current which is loading the string, and produces real time SOC and SOH estimates for each cell.

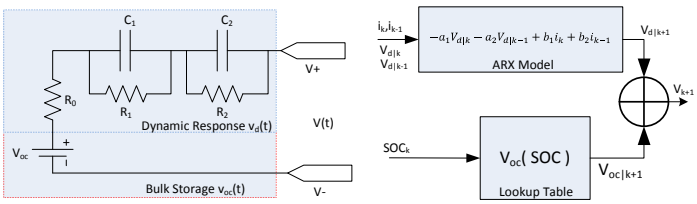
The most straightforward method to estimate SOC/SOH for individual cells in a battery is to develop a model of an "average" cell, and deploy one or more KF which use this model to track the states of a set of cells. This allows research on single-cell methods to directly translate to a multi-cell approach. A black-box view of the proposed system context is shown in Fig. 1. The per-cell information can then be fed to battery protection logic, cell balancing logic, or summarized for instrumentation. The use of a KF rather than a bare model significantly increases the noise resistance and adaptability of the system.

In Section II we introduce our cell model, and validate simulation results with measurement data. In Section III we develop a framework for estimating both SOC and SOH (i.e. capacity) using a multiscale modification of the standard Unscented Kalman Filter (UKF). Note that this multi-scale approach is not application-specific; it can be used generally when a very slowly changing system parameter needs to be tracked. In Section IV we give a general analysis of our work, and finally conclude the paper in Section V.

II. Cell Model

The model we employ is identical to that described in [3], and cell data is collected in the same way. As depicted in Fig. 2, the model consists of two top-level components. The first is V_{oc}, which is a nonlinear function of SOC describing resting (unloaded) voltage of the cell. The second is our second-order approximation to the dynamic cell response, V_d. This signal, whose magnitude and direction is determined by the loading current i, is superimposed on V_{oc} to create an accurate representation of cell behavior.

As before we apply the Autoregressive Exogenous (ARX) system identification approach to find a linear approximation of the dynamic behavior, resulting in a compact state-space model.



(a) Lumped Circuit (b) Discrete-Time Approximation

Fig. 2: Second-Order Cell Model: model parameters are extracted from data collected during cell cycling.

A. State-Space Equations

In the interest of space, the state-space representation will only be summarized here. More detail appears in [3] and [6]. The ARX constants a_1,a_2,b_1, and b_2 are extracted from cell cycling data; η_i is the cell's Coulombic Efficiency (we use 1.0); Δt is our simulation timestep (i.e. sensor sample period); and C_k is the nominal cell capacity. Finally i_k is the measured battery current. As shown below, $\boldsymbol{F}(...)$ is our state update expression and $\boldsymbol{H}(x)$ is our output (measurement) expression.

$$\boldsymbol{F}(x_{k+1}, i) = \begin{pmatrix} SOC_{k+1} \\ V_{d|k+1} \\ V_{d|k} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} SOC_k \\ V_{d|k} \\ V_{d|k-1} \end{pmatrix} + \begin{pmatrix} \frac{\eta_i \Delta t}{C_k} & 0 & 0 \\ 0 & b_1 & -b_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_{k+1} \\ i_k \\ i_{k-1} \end{pmatrix} \quad (1)$$

$$\boldsymbol{H}(x) = V_{oc}(SOC_k) + V_{d|k} \quad (2)$$

In brief, SOC is updated using the standard Coulomb counting expression, shown below. This expression is central to many SOC algorithms.

$$SOC_{k+1} = SOC_k + \frac{\eta_i \Delta t}{C_k} i_{k+1} \quad (3)$$

B. Modeling Refinements

A property of Li-ion cells which impacts modeling accuracy is referred to as *open-circuit voltage hysteresis* [9]. This is a phenomenon whereby the cell voltage will ultimately settle to a slightly different value depending on the direction of the current which previously disturbed it . The magnitude of the hysteresis varies with SOC, but based on our measurements typically falls in the range of 10mv to 20mv at room temperature. In the flat region of the V_{oc} curve, this can represent an SOC estimation error of several percent.

This issue was not addressed in [3]. In [5] this was addressed explicitly by adding additional model states, at the expense of computational complexity. Here we address the issue *implicitly* by selecting the V_{oc} curve as the *mean* between a charge and discharge curve. This idea is shown in Fig. 3. When the ARX coefficients are fit to the data using this new curve, Vd will converge to a value above our mean OCV for charging and below it for discharging. Thus, hysteresis is implicitly encoded into the ARX coefficients. This assumes hysteresis is constant over all SOC, but as our results will show, the impact of this particular error is minimal.

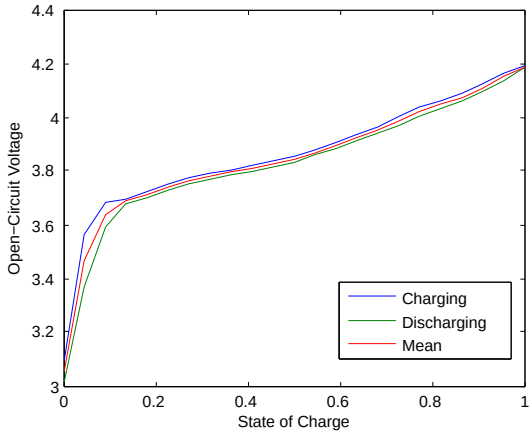


Fig. 3: Hysteresis in Li-ion OCV-SOC relationship.

C. Finding an Average Cell Model

Without additional consideration, this method only finds optimal parameters for a *specific* cell. However, in a production environment it is not practical to characterize each individual cell for large batteries, since this process can require from several hours to several days per cell. Since the properties of each cell are not identical, in order to minimize the *average* modeling error across a population of cells, we must identify an *average* cell model. This is a reasonable in the sense that the UKF (and EKF) typically assumes Gaussian noise overlaying our model expressions[10]; thus if the model is representative of the average case, the filter can adapt to the specific case. We demonstrate a first-effort approach to solving this problem by posing it as a minimization problem.

We start by defining $\hat{\boldsymbol{v}} = sim(p, u, n)$ as a function which simulates our model using the current profile vector u and the parameter vector p. This simulation is simply a recursive function which computes a sequence of terminal voltage estimates $\hat{\boldsymbol{v}}$ based on our state-space cell model, and runs for the number of samples n under our current profile. We can then compute the mean squared error between our estimated voltage response waveform $\hat{\boldsymbol{v}}$ and the measured voltage response of a specific cell k, $\boldsymbol{v}_k$. Finally, we develop a minimization problem whose objective function is a combination of the mean squared errors between our simulation and the measured voltage response of each of the cells in our sample pool.

In short, here our goal is to find a set of parameters p_{ave} which minimizes the average modeling error for all cells in a sample set, starting from the parameters extracted from a single cell. For this study, Matlab was used to perform the minimization task.

D. Model Simulation

The model is simulated using Matlab with only current as an input (i.e., open-loop) to determine its accuracy. As in [11], the current profile is derived from a series of Urban Dynamometer Driving Schedule (UDDS) test cycles [12], followed by a 5% SOC constant-current discharge, which overall depletes the cell from 100% SOC to 0%. The current is scaled to correspond to the nominal capacity of the cells.

In Table I the root-mean square error (RMSE) is computed with respect to voltage measurements made from real cells cycled with the same profile. We compare the RMSE against the ESC model[5], as well as several simpler models proposed

in that work. As anticipated, the modeling error lies approximately between that of the simpler models and the models which include additional states. As is done in [5], the RMS error is computed only over 5 to 95% SOC.

TABLE I: Modeling Error Comparison

Model	States	UDDS test RMSE (mV)
Combined	1	23.3
Simple	1	22.4
Zero-state hysteresis	1	22.3
One-state hysteresis	2	14.0
Proposed	3	13.0
ESC, n_f=2	4	7.2
ESC, n_f=4	6	4.2

Note that these results are for measurements taken at 25C. The dynamic portion of the model is a strong function of temperature; for a more complete model, temperature should also be considered an input.

III. Real-Time SOH Estimation

In addition to SOC, SOH is an important property of a rechargeable cell. In this section we show the logical developments leading to the multiscale UKF for SOH (capacity) estimation.

Although often simplified as constant, the Capacity of a cell changes very slowly with time. One cell cycle is defined as a complete discharge of a cell followed by a recharge, and is a common metric for describing its age. Our own measurements suggest a loss of around 0.1% per cycle. Although the rate is slow, a typical, modern Li-ion cell can be cycled more than 1000 times over its life. This will cause a measurable SOC divergence from the true value and decrease the accuracy of the filter. In Fig. 4, Equation 3 is evaluated over the current profile from Section II-D, assuming a *new* and an *aged* cell whose capacity was reduced by 10%.

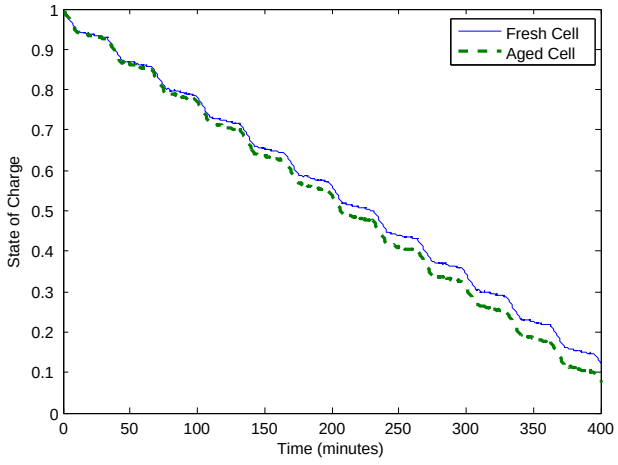


Fig. 4: Longterm SOC divergence due to incorrect capacity initialization.

A. Dual Estimation

The Dual Kalman Filter is a commonly-proposed, generic architecture to simultaneously determine both system state and estimate one or more system parameters[1]. In short, one filter computes the current state x_k using the previous parameter estimate, while the other filter computes the parameter w using the previous state estimate. Thus, the state and parameter estimates evolve together over time.

[1]Also called weights. We use these terms interchangeably.

However, if we want to use the popular Extended Kalman Filter (EKF) in this "dual estimation" configuration, a significant complication arises. As in the standard EKF, the partial derivative of the state vector x_k must be determined analytically using state update and measurement equations $f(...)$ and $g(...)$.

$$\hat{A}_k^x = \left.\frac{\partial f(x_k, u_k)}{\partial x_k}\right|_{x_k=\hat{x}_k^+} \tag{4}$$

$$\hat{C}_k^x = \left.\frac{\partial g(x_k, u_k)}{\partial x_k}\right|_{x_k=\hat{x}_k^-} \tag{5}$$

For the Dual EKF we must additionally determine $\hat{C}_k$ for the weight filter. To do so, we need to find the *total* derivative, since x_k is a function of x_{k+1}, and both are a function of w. This derivative will be recursive in nature.

$$\hat{C}_k^w = \left.\frac{\mathrm{d}g(x_k, u_k)}{\partial w_k}\right|_{x_k=\hat{x}_k^-} \tag{6}$$

This is a potentially error-prone process which complicates system development–even more so for the multiscale framework that we wish to develop. Although not included due to space constraints, for the analytical solution for the EKF multiscale case, see [13].

A more recent alternative to the EKF, the UKF, belongs to the family of "Sigma Point" Kalman Filters. This approach does not rely on derivatives of the measurement or process functions. Instead it treats $\boldsymbol{x}$ as a random variable and applies an operation called the Unscented Transformation to estimate the KF statistics. Estimation accuracy is improved since the EKF in effect uses a first-order linear approximation of a nonlinear system to propagate state variables between iterations, whereas the UKF achieves at least a 2^{nd} order approximation. More details on the Unscented Transformation appear in [14].

B. Unscented Kalman Filter Algorithm

The key difference between the UKF and EKF is the calculation of a set of *sigma points* each iteration. The sigma points offer a reduced-complexity mechanism for estimating the mean and covariance of random variables which are propagated through a nonlinear function (such as our V_{oc} curve).

1) Compute Sigma Points
2) Estimate state
3) Calculate error covariance
4) Calculate Kalman gain
5) Update state estimate based on measurement
6) Update error covariance based on measurement

This process is shown more formally in lines 5 through 9 of Algorithm 1, and is based on equations in [15]. The process appears to be more complex than the EKF, but they actually are of the same computational complexity with respect to the size of the state vector [10].

The choice of constants α, β, and κ influence the behavior of the Unscented Transform [10]. The value of α has an effect on how spread out the sigma points will be around the mean of $\boldsymbol{x}$. Generally values in the range of 1×10^{-3} to 1×10^{-4} give good results. In general we have found that the behavior is not

Algorithm 1 UKF Algorithm, for the Additive Noise Case [15]

procedure INITIALIZE

$\hat{\boldsymbol{x}}_0^+ \leftarrow E[\boldsymbol{x}_0]$

$\boldsymbol{P}_0 \leftarrow E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_\mathbf{0}^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T]$

$\lambda \leftarrow \alpha^2(dim(\boldsymbol{x}) + \kappa) - dim(\boldsymbol{x})$ # Compute scaling constants

$\gamma \leftarrow \sqrt{dim(\boldsymbol{x}) + \lambda}$

$W_0^{(m)} \leftarrow \frac{\lambda}{dim(\boldsymbol{x})+\lambda}$ # Compute the Mean and Covariance weight vectors

$W_0^{(c)} \leftarrow \frac{\lambda}{dim(\boldsymbol{x})+\lambda} + 1 - \alpha^2 + \beta$

for $i = 1, ..., 2 \cdot dim(\boldsymbol{x})$ **do**

$W_i^{(m)} \leftarrow W_i^{(c)} \leftarrow \frac{1}{2(dim(\boldsymbol{x})+\lambda)}$

procedure UPDATE

$\boldsymbol{\chi}_{k-1} \leftarrow [\hat{\boldsymbol{x}}_{k-1}, \hat{\boldsymbol{x}}_{k-1} + \gamma\sqrt{\boldsymbol{P}_{k-1}}, \hat{\boldsymbol{x}}_{k-1} - \gamma\sqrt{\boldsymbol{P}_{k-1}}]$ # Calculate the Sigma Points

$\boldsymbol{\chi}^*_{k|k-1} \leftarrow \boldsymbol{F}(\boldsymbol{\chi}_{k-1}, \boldsymbol{u}_{k-1})$ # State Estimate Time Update

$\hat{\boldsymbol{x}}_k^- \leftarrow \sum_{i=1}^{2\cdot dim(\boldsymbol{x})} W_i^{(m)} \boldsymbol{\chi}^*_{k|k-1}$ # Error Covariance time-update

$\boldsymbol{P}_k^- \leftarrow \sum_{i=1}^{2\cdot dim(\boldsymbol{x})} W_i^{(c)} (\boldsymbol{\chi}^*_{i,k|k-1} - \hat{x}_k^-)(\boldsymbol{\chi}^*_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-)^T + \boldsymbol{Q}$

$\boldsymbol{\chi}_{k|k-1} \leftarrow [\boldsymbol{\chi}^*_{k|k-1}, \boldsymbol{\chi}^*_{0,k|k-1} + \gamma\sqrt{\boldsymbol{Q}}, \boldsymbol{\chi}^*_{0,k|k-1} - \gamma\sqrt{\boldsymbol{Q}}]$ # Augment Sigma Points

$\boldsymbol{Y}_{k|k-1} \leftarrow \boldsymbol{H}(\boldsymbol{\chi}_{k|k-1})$ # Compute Output Estimate

$\hat{\boldsymbol{y}}_k^- \leftarrow \sum_{i=1}^{2\cdot dim(\boldsymbol{x})} W_i^{(m)} \boldsymbol{Y}_{i,k|k-1}$

$\boldsymbol{P}_{\tilde{y}_k\tilde{y}_k} \leftarrow \sum_{i=1}^{2\cdot dim(\boldsymbol{x})} W_i^{(c)} (\boldsymbol{Y}_{i,k|k-1} - \hat{y}_k^-)(\boldsymbol{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-)^T + \boldsymbol{R}$ # Compute Estimator Gain Matrix

$\boldsymbol{P}_{x_k y_k} \leftarrow \sum_{i=1}^{2\cdot dim(\boldsymbol{x})} W_i^{(c)} (\boldsymbol{\chi}_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{Y}_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-)^T$

$\boldsymbol{K}_k \leftarrow \boldsymbol{P}_{x_k y_k} \boldsymbol{P}^{-1}_{\tilde{y}_k\tilde{y}_k}$ # Calculate Kalman Gain

$\hat{\boldsymbol{x}}_k^- \leftarrow \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)$ # Update state estimate based on measurement

$\boldsymbol{P}_k \leftarrow \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{P}_{\tilde{y}_k\tilde{y}_k} \boldsymbol{K}_k^T$ # Update error covariance based on measurement

especially sensitive to the value of α. The value of β is usually set to $3 - dim(x)$, where $dim(x)$ is the number of states in our state vector. Finally, κ is set to 2 for an assumed Gaussian distribution of our noise covariances. Thus we are generally only required to adjust α, if at all. For all work presented here we use $\alpha = 10^{-4}$,$\beta = 0$, and $\kappa = 2$.

C. UKF Extension For Multiple Time Scales

A cell's SOH changes extremely slowly compared to SOC, and is only weakly related to the cell's voltage within a typical sampling window. Thus, reliable estimation requires a very low noise floor. In [13] a multi-scale framework was proposed to address these issues. Although applied to the EKF in that work, we show the approach works equally well with the UKF, with the additional advantage of not needing to determine the recursive derivative equations (Section III-A).

In general, the multiscale KF is just an adaptation of the Dual Kalman Filter in which the state-estimating filter and the parameter-estimating filter are temporally decoupled so as to operate on different time scales. Our state filter, which estimates SOC, iterates at $\Delta t = 1$ second intervals. Henceforth for brevity we refer to it as our *micro-scale* filter. Now, our parameter (capacity) estimating filter iterates at at $L\Delta t$ second intervals, where L is a fixed epoch. We refer to it as our *macro-scale* filter. Effectively the constant L dictates how many *micro* iterations must pass before each *macro* iteration. This split-epoch iterative procedure is illustrated in Fig. 5.

An issue not addressed in [13] is that of improperly initialized SOC. Since the macro-scale filter relies on the SOC sampled at the beginning and end of epoch L, if the SOC

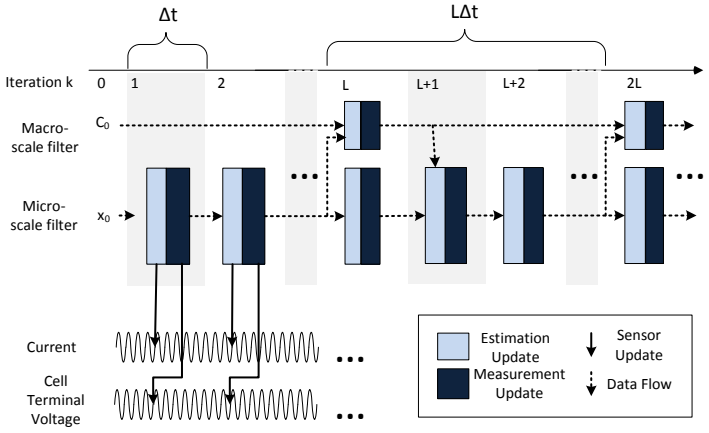


Fig. 5: Multi-scale Kalman Filtering: a standard UKF state-estimator (micro-scale filter) and parameter estimator (macro-scale filter) advance together in lockstep. The macro-scale filter is able to discern and correct for long-term deviations in SOC which are due to incorrect capacity in the micro-scale filter.

estimate has not yet converged to the "true" SOC, the first macro-scale capacity estimate may over- or under-shoot. To prevent this, the micro-scale filter can be allowed to iterate for two L-epochs before the macro-filter starts.

1) Micro-scale Filter Process: The micro-scale filter runs the standard UKF, shown in Algorithm 1, and uses the state-space expressions $\boldsymbol{F}(...)$ and $\boldsymbol{H}(...)$ described in Section II.

2) Macro-scale Filter Process: By constrast, the macro-scale expressions for $\boldsymbol{F}(...)$ and $\boldsymbol{H}(...)$ follow the standard template for a parameter-estimating KF. The single state vari-

able represents our capacity estimate, with an identity time-update function. Although the macro-scale filter also follows Algorithm 1, the computations reduces to scalar operations since there is only one state. The state evolves though an imagined process noise which must be characterized by variance $\boldsymbol{Q}$.

$$w_{k+1} = w_k \tag{7}$$

The measurement equation is a small adjustment to Equation 3: it sums the current over L time steps before adjusting the state value. Thus, it is a projection over L micro-filter updates. Notice for the macro-scale filter, our measurement is actually the output of the micro-scale filter. The measurement noise is characterized by variance $\boldsymbol{R}$.

$$SOC_{k+1} = SOC_k + \frac{\eta_i \Delta t}{C_k} \sum_{j=0}^{L-1} i_{k,j} \tag{8}$$

Note that the current is summed over $L\Delta t$ seconds, which serves to amplify the effect of any error in the micro-scale filter which is caused by an incorrect capacity estimate. At the end of an iteration, the macro-scale filter passes the new capacity estimate back to the micro-scale filter.

Finally, we use the "forgetting factor" approach for parameter estimation [10], in which $\boldsymbol{Q}$ is initially set to a fairly large value, and then updated each iteration in the following way:

$$Q_{k+1} = (\delta - 1)^{-1} \boldsymbol{P}_k \tag{9}$$

With this approach, the gain of the filter will decay approximately exponentially. This allows more rapid convergence than a fixed $\boldsymbol{Q}$ value, and as $\boldsymbol{Q}$ will never reach 0, neither will the estimation uncertainty covariance, which will prevent the estimate from getting stuck in local minima. A summary of parameter values is shown in Table III and Table II.

D. Capacity Estimation Results

Our Dual UKF algorithm was implemented in Matlab. Each run takes four arguments: the repeated UDDS current profile and its corresponding voltage waveform, the value of L, and the desired number of iterations. As output it produces SOC and SOH waveforms. Fig. 6 shows a portion of the input current, voltage waveform, and the SOC estimate produced by the micro-scale filter. Table III and Table II list all the algorithm values used to produce the results. Identical model parameters and UKF covariances are used for all tests.

1) Choice of L: The constant L has a significant impact on the convergence of the capacity estimate. When $L = 1$ we have a degenerate case which is equivalent to the standard Dual UKF. We found that the capacity estimate did not converge to a realistic value in this case, nor for small values of L in general. Fig. 7 shows the convergence behavior for several L values. Note that convergence takes up to 2 hours for large values of L; this is considered acceptable since it only happens once if estimates are periodically saved to non-volatile memory.

2) Adaptability: Using $L = 1200$, the ability for the system to estimate capacity was evaluated using data based on two scenarios: comparing a fresh cell and the *same* cell aged by 100 cycles (Fig. 8), and comparing three *different* cells (Fig. 9).

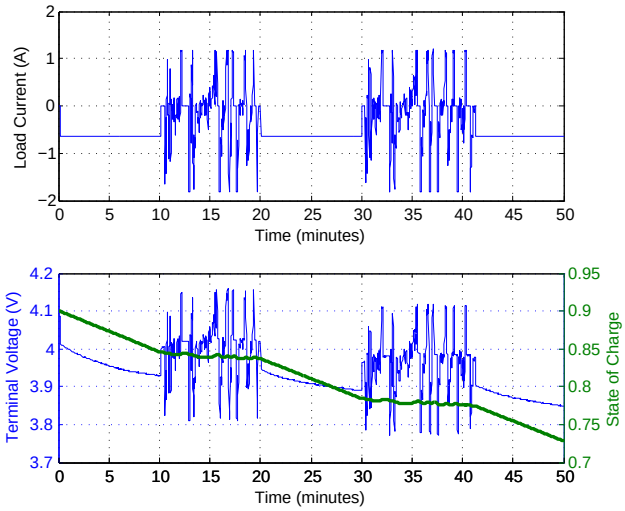


Fig. 6: Repeated UDDS current profile (upper plot), and computed terminal voltage & SOC (lower plot). The SOC is estimated by the micro-scale filter.

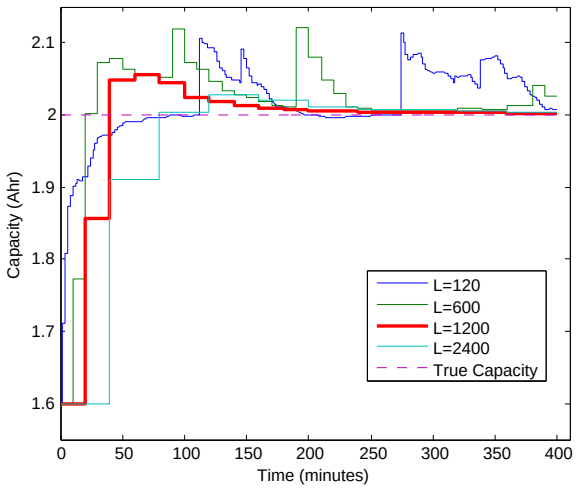


Fig. 7: Capacity convergence for various values of L. Larger values show improved immunity against estimation noise. The capacity is estimated by the macro-scale filter.

IV. Analysis

The procedure for applying the KF to any application is generally the same:

1) Select a model of the system
2) Identify model parameters from collected data
3) Tune KF covariances based on *a priori* knowledge about system behavior

The need to tune the various filter parameters (i.e., the measurement and process covariances) for good performance is one of the more significant barriers to applying the KF. It is also critical for parameter estimation. Usually this procedure is done by hand, using knowledge of expected system behavior.

Besides tuning, correct estimation convergence requires the model to have as low RMSE as possible. The capacity evolves as a proportion of the error between the expected SOC at the end of an epoch $L\Delta t$ and the actual SOC computed by the micro-scale filter over the same time period. If this error is mainly due to a model mismatch rather than incorrect Capacity, convergence behavior is non-optimal. Thus, future work requires further study on the impact of model selection.

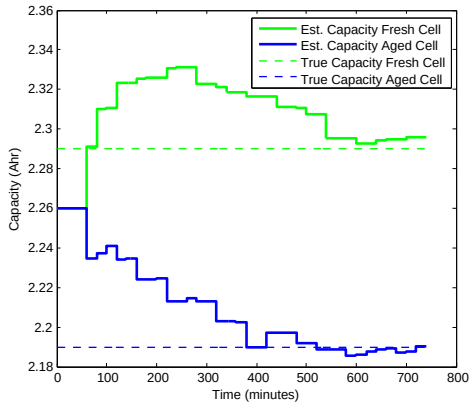


Fig. 8: Capacity Estimation: distinguishing fresh (L=1200,C=2.29) and aged (L=1200,C=2.19) cells.

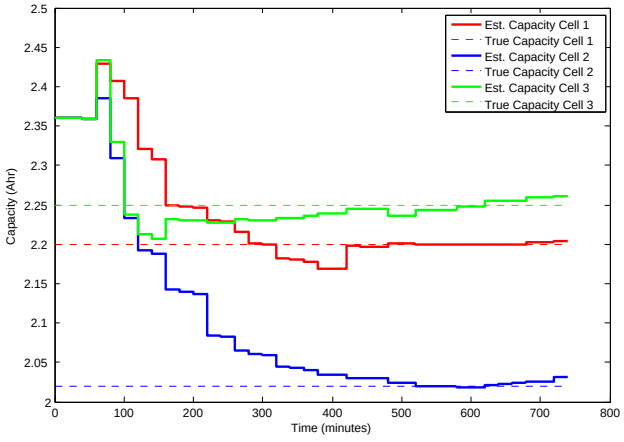


Fig. 9: Capacity Estimation: distinguishing different cells (L=1200).

V. Conclusion

In this paper we have demonstrated a new approach to estimating the the SOC and SOH of a battery using a Multi-scale UKF approach. Compared to the EKF, the UKF eliminates the need to compute the recursive total derivative parameter estimation, which is challenging for the multi-scale case. The system's ability to track the state of different cells using the same general model was also demonstrated. The next step is an analysis of the supporting CPS architecture and the relationship between model sophistication and computational requirements.

Acknowledgements

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

TABLE II: Macro-scale Filter Variables

Symbol	Meaning	Initialization
δ	"Forgetting factor" for anealing	0.94
Q	System Noise Covariance	1^{-3}
R	Measurement Noise Covariance	1^{-3}
P_k	Error Covariance	1^{-3}
w_k	System state (capacity/SOH estimate)	$C_{nominal}$
y_k	System output (SOC projected over L time steps)	N/A
u_k	System input (SOC recieved from micro-scale filter)	i_0

TABLE III: Micro-scale Filter Variables

Symbol	Meaning	Initialization
A	From state-space model	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 1 & 0 \end{pmatrix}$
B	From state-space model	$\begin{pmatrix} \frac{\eta_i \Delta t}{C} & 0 & 0 \\ 0 & b_1 & b_2 \\ 0 & 0 & 0 \end{pmatrix}$
C	From state-space model	$\begin{pmatrix} V_{oc}(SOC(0)) \\ 1 \\ 0 \end{pmatrix}$
D	From state-space model	0
Q_k	System Noise Covariance	$\begin{pmatrix} 1^{-6} & 0 & 0 \\ 0 & 1^{-3} & 0 \\ 0 & 0 & 1^{-3} \end{pmatrix}$
R	Measurement Noise Covariance	$1{\cdot}10^{-4}$
P_k	Error Covariance	$\begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 0.25 \end{pmatrix}$
x_k	System state (SOC estimate)	$\begin{pmatrix} \mathbf{E}(SOC(0)) \\ 0 \\ 0 \end{pmatrix}$
y_k	System output (measured v(t))	N/A
u_k	System input i(t)	$\begin{pmatrix} i_0 \\ 0 \\ 0 \end{pmatrix}$

References

[1] H. Kim and K. Shin, "Efficient sensing matters a lot for large-scale batteries," in *2011 IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, April 2011, pp. 197–205.

[2] H. Kim and K. G. Shin, "Dependable, efficient, scalable architecture for management of large-scale batteries," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '10, pp. 178–187.

[3] A. Mills and J. Zambreno, "Towards scalable monitoring and maintenance of rechargeable batteries," in *2014 IEEE International Conference on Electro/Information Technology (EIT)*, June 2014, pp. 624–629.

[4] D. Haifeng, W. Xuezhe *et al.*, "State and parameter estimation of a HEV Li-ion battery pack using Adaptive Kalman Filter with a new SOC-OCV concept," in *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, vol. 2, 2009, pp. 375–380.

[5] 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.

[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. Shin, M. Poncino, E. Macii, and N. Chang, "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.

[8] M. Dubarry, N. Vuillaume, and B. Y. Liaw, "Origins and accommodation of cell variations in Li-ion battery pack modeling," *International Journal of Energy Research*, vol. 34, no. 2, pp. 216–231, 2010.

[9] M. A. Roscher, O. Bohlen, and J. Vetter, "Ocv hysteresis in Li-Ion batteries including two-phase transition materials," *International Journal of Electrochemistry*, 2011.

[10] E. A. Wan and R. van der Merwe, "The unscented kalman filter," in *Kalman filtering and neural networks*, S. S. Haykin, Ed., 2001, pp. 231–234.

[11] G. L. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. modeling and identification," *Journal of Power Sources*, vol. 134, no. 2, pp. 252 – 261, 2004.

[12] United States Environmental Protection Agency, "Urban dynamometer driving schedule (UDDS)," http://www.epa.gov/nvfel/testing/dynamometer.htm, 2013.

[13] C. Hu, B. D. Youn, T. Kim, and J. Chung, "Online estimation of lithium-ion battery state-of-charge and capacity with a multiscale filtering technique," in *Annual Conference of the Prognostics and Health Management Society*, 2011.

[14] E. Wan and R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium.*, 2000, pp. 153–158.

[15] R. Van Der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, Oregon Health & Science University, 2004.