Available online at www.sciencedirect.com
ScienceDirect
Journal of Power Sources 161 (2006) 1369–1384
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

# Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs Part 2: Simultaneous state and parameter estimation

Gregory L. Plett*, [1]

*Department of Electrical and Computer Engineering, University of Colorado at Colorado Springs, 1420 Austin Bluffs Parkway, P.O. Box 7150, Colorado Springs, CO 80933-7150, United States*

Received 9 March 2006; received in revised form 26 May 2006; accepted 6 June 2006
Available online 25 July 2006

**Abstract**

We have previously described algorithms for a battery management system (BMS) that uses Kalman filtering (KF) techniques to estimate such quantities as: cell self-discharge rate, state-of-charge, nominal capacity, resistance, and others. Since the dynamics of electrochemical cells are not linear, we used a nonlinear extension to the original KF called the extended Kalman filter (EKF).

Now, we introduce an alternative nonlinear Kalman filtering technique known as "sigma-point Kalman filtering" (SPKF), which has some theoretical advantages that manifest themselves in more accurate predictions. The computational complexity of SPKF is of the same order as EKF, so the gains are made at little or no additional cost.

This paper is the second in a two-part series. The first paper explored the theoretical background to the Kalman filter, the extended Kalman filter, and the sigma-point Kalman filter. It explained why the SPKF is often superior to the EKF and applied SPKF to estimate the state of a third-generation prototype lithium-ion polymer battery (LiPB) cell in dynamic conditions, including the state-of-charge of the cell.

In this paper, we first investigate the use of the SPKF method to estimate battery parameters. A numerically efficient "square-root sigma-point Kalman filter" (SR-SPKF) is introduced for this purpose. Additionally, we discuss two SPKF-based methods for simultaneous estimation of both the quickly time-varying state and slowly time-varying parameters. Results are presented for a battery pack based on a fourth-generation prototype LiPB cell, and some limitations of the current approach, based on the probability density functions of estimation error, are also discussed.


*Keywords:* Battery management system; Hybrid electric vehicle; Sigma-point Kalman filter; State-of-charge; State-of-health; Lithium-ion polymer

## 1. Introduction

This paper applies results from the field of study known variously as *sequential probabilistic inference* or *optimal estimation theory* to advanced algorithms for a battery management system (BMS). This BMS is able to estimate battery state-of-charge (SOC), instantaneous available power, and parameters indicative of the battery state-of-health (SOH) such as power fade and capacity fade, and is able to adapt to changing cell characteristics over time as the cells in the battery pack age. The algorithms have been successfully implemented on a lithium-ion polymer battery (LiPB) pack for hybrid-electric-vehicle (HEV) application,[2] and we also expect them to work well for other battery chemistries and less demanding applications.

We have previously reported work using *extended Kalman filters* (EKF) to solve the HEV BMS algorithm requirements [1–6]. We have since explored a different form of Kalman filter-

* Tel.: +1 719 262 3468; fax: +1 719 262 3589.
*E-mail addresses:* glp@eas.uccs.edu, gplett@compactpower.com.
[1] Also consultant to Compact Power Inc., 707 County Line Rd., P.O. Box 509, Palmer Lake, CO 80133. United States. Tel.: +1 719 488 1600x134; fax: +1 719 487 9485.

[2] HEV is a particularly good benchmark for these algorithms since it imposes demanding requirements on a pack of limited capacity, resulting in cell electrochemistries that are most often far from equilibrium. Therefore, advanced methods must be used to estimate SOC, SOH, and instantaneous power in order to safely, efficiently and aggressively exploit the pack capabilities.

0378-7753/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2006.06.004

ing called *sigma-point Kalman filters* (SPKF), and have found them to have several important advantages. We introduce SPKF here in a two-part series, of which this is the second part. The companion to this paper [7] introduces the SPKF and applies it to estimating the state of an LiPB-based HEV battery cell, particularly its SOC. In this paper, we build on the introduction to first investigate the use of the SPKF method to estimate battery parameters. A numerically efficient "square-root sigma-point Kalman filter" (SR-SPKF) is described for this purpose. Additionally, we discuss two SPKF-based methods for simultaneous estimation of both the quickly time-varying state and slowly time-varying parameters. Applications to the HEV BMS algorithm requirements are outlined, example results given using fourth-generation prototype LiPB cells, and conclusions made.

We note before continuing that there are conflicting objectives in making this paper both self-contained and simultaneously reducing redundant material with respect to the first paper in this series. We have perhaps erred on the side of conciseness in several areas. For a more in-depth discussion of sequential probabilistic inference and the base-line SPKF algorithm, the reader is referred to Ref. [7].

## 2. Summary of sequential probabilistic inference

Very generally, any causal dynamic system (e.g., a battery cell) generates its outputs as some function of its past and present inputs. Often, we can define a *state vector* for the system whose values together summarize the effect of all past inputs. Present system output is a function of present input and present state only; past input values need not be stored. The system's *parameter vector* comprises all quasi-static numeric quantities that describe how the system state evolves and how the system output may be computed. The state-vector quantities change on a relatively rapid time scale, and the parameter-vector quantities change on a relatively long time scale (or, not at all).

We assume that the electrochemical cell under consideration may be modeled using a discrete-time state–space model of the form

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1) \tag{1}$$

$$y_k = h(x_k, u_k, v_k, k). \tag{2}$$

Here, $x_k \in \mathbb{R}^n$ is the system state vector at time index $k$, and Eq. (1) is called the "state equation" or "process equation" and captures the evolving system dynamics. The known/deterministic input to the system is $u_k \in \mathbb{R}^p$, and $w_k \in \mathbb{R}^n$ is stochastic "process noise" or "disturbance" that models some unmeasured input which affects the state of the system. The output of the system is $y_k \in \mathbb{R}^m$, computed by the "output equation" (2) as a function of the states, input, and $v_k \in \mathbb{R}^m$, which models "sensor noise" that affects the measurement of the system output in a memoryless way, but does not affect the system state. $f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)$ is a (possibly nonlinear) state transition function and $g(x_k, u_k, v_k, k)$ is a (possibly nonlinear) measurement function.

In the companion to this paper we introduced the relevant theory describing the optimal solution to state estimation – based on a body of knowledge known as "sequential probabilistic inference" – and showed how various approximations and assumptions resulted in the Kalman filter, the extended Kalman filter, and the sigma-point Kalman filter [7]. Here, we summarize the key points required to understand parameter estimation and simultaneous state and parameter estimation. For greater detail, the reader should consult Ref. [7].

Table 1
Summary of the general sequential probabilistic inference solution

General state–space model
$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)$
$y_k = h(x_k, u_k, v_k, k)$,
where $w_k$ and $v_k$ are independent, Gaussian noise processes of covariance matrices $\Sigma_w$ and $\Sigma_v$, respectively

Definitions: let
$\tilde{x}_k^- = x_k - \hat{x}_k^-, \quad \tilde{y}_k = y_k - \hat{y}_k$

Initialization: for $k = 0$, set
$\hat{x}_0^+ = \mathbb{E}[x_0]$
$\Sigma_{\tilde{x},0}^+ = \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^{\mathrm{T}}]$

Computation: for $k = 1, 2, \ldots$ compute
State estimate time update: $\hat{x}_k^- = \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)|\mathbb{Y}_{k-1}]$
Error covariance time update: $\Sigma_{\tilde{x},k}^- = \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^{\mathrm{T}}]$
Output estimate: $\hat{y}_k = \mathbb{E}[h(x_k, u_k, v_k, k) \mid \mathbb{Y}_{k-1}]$
Estimator gain matrix: $L_k = \mathbb{E}[(\tilde{x}_k^-)(\tilde{y}_k)^{\mathrm{T}}](\mathbb{E}[(\tilde{y}_k)(\tilde{y}_k)^{\mathrm{T}}])^{-1}$
State estimate measurement update: $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$
Error covariance measurement update: $\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$

The general solution from which all the KF variants are derived comprises two major stages per measurement interval. First, it *predicts* the present state (the "time update") given all prior information; second, it *corrects* the prediction using the current measurement (the "measurement update"). The KFs not only estimate the state, but also the state error covariance matrix to provide an ongoing uncertainty estimate on the state estimation error.

Table 1 summarizes the general solution. In the notation we use, the decoration "circumflex" indicates an estimated quantity (e.g., $\hat{x}$ indicates an estimate of the true quantity $x$). A superscript "$-$" indicates an *a priori* estimate (i.e., a prediction of a quantity's present value based on past data) and a superscript "$+$" indicates an *a posteriori* estimate (e.g., $\hat{x}_k^+$ is the estimate of true quantity $x$ at time index $k$ based on all measurements taken up to and including time $k$). The decoration "tilde" indicates the error of an estimated quantity.

$$\mathbb{Y}_k = \{y_k, y_{k-1}, \ldots, y_0\}.$$

The symbol $\Sigma_{xy} = \mathbb{E}[xy^{\mathrm{T}}]$ indicates the auto- or cross-correlation of the variables in its subscript. (Note that often these variables are zero-mean, so the correlations are identical to covariances.) Also, for brevity of notation, we often use $\Sigma_x$ to indicate the same quantity as $\Sigma_{xx}$.

The algorithm first initializes the filter and iteratively executes six steps each measurement interval. First, the present state value is predicted using prior data. Second, the covariance of this state estimate error is updated. Third, the present cell output is predicted. Fourth, the Kalman gain matrix is computed. Fifth, the actual cell output is measured and compared to the

Table 2
Weighting constants for two sigma-point methods

| | $\gamma$ | $\alpha_0^{(m)}$ | $\alpha_k^{(m)}$ | $\alpha_0^{(c)}$ | $\alpha_k^{(c)}$ |
|---|---|---|---|---|---|
| UKF | $\sqrt{L+\lambda}$ | $\frac{\lambda}{L+\lambda}$ | $\frac{1}{2(L+\lambda)}$ | $\frac{\lambda}{L+\lambda}+(1-\alpha^2+\beta)$ | $\frac{1}{2(L+\lambda)}$ |
| CDKF | $h$ | $\frac{h^2-L}{h^2}$ | $\frac{1}{2h^2}$ | $\frac{h^2-L}{h^2}$ | $\frac{1}{2h^2}$ |

$\lambda=\alpha^2(L+\kappa)-L$ is a scaling parameter, with $(10^{-2}\leq\alpha\leq 1)$. Note that this $\alpha$ is different from $\alpha^{(m)}$ and $\alpha^{(c)}$. $\kappa$ is either 0 or $3-L$. $\beta$ incorporates prior information. For Gaussian RVs, $\beta=2$. $h$ may take any positive value. For Gaussian RVs, $h=\sqrt{3}$.

estimate. The output estimate error is weighted by the Kalman gain and used to update the state estimate. Finally, the state estimate error covariance is updated. The output of the filter at each time instant is the state estimate from step 5 and the error covariance from step 6 (used to compute error bounds on the state estimate).

## 3. Summary of sigma-point Kalman filters

For nonlinear systems, a closed-form solution or even an algorithm to exactly implement the general probabilistic inference solution in Table 1 is not available. The EKF is one approach to approximating the solution using a first-order linearization of the system dynamics, which is based on some questionable assumptions. Sigma-point Kalman filters are an alternate approach to generalizing the Kalman filter to state estimation for nonlinear systems. SPKFs rely on numeric approximations rather than analytic approximations of the EKF. Each time step, a set of points (*sigma points*) is chosen so that the (possibly weighted) mean and covariance of the points exactly matches the mean and covariance of the *a priori* random variable. These points are then passed through the nonlinear function, resulting in a transformed cloud of points. The *a posteriori* mean and covariance that are sought are then approximated by the mean and covariance of this cloud. Note that the sigma points comprise a fixed small number of vectors that are calculated deterministically—not like the Monte Carlo or particle filter methods. This method is used to approximate state and output estimates, and their covariances.

Specifically, if the input random vector $x$ has dimension $L$, mean $\bar{x}$, and covariance $\Sigma_{\tilde{x}}$, then $p+1=2L+1$ sigma points are generated as the set

$$\mathcal{X}=\{\bar{x},\ \bar{x}+\gamma\sqrt{\Sigma_{\tilde{x}}},\ \bar{x}-\gamma\sqrt{\Sigma_{\tilde{x}}}\},$$

with elements of $\mathcal{X}$ indexed from 0 to $p$, and where the matrix square-root $R=\sqrt{\Sigma}$ computes a result such that $\Sigma=RR^{\mathrm{T}}$. Usually, the efficient *Cholesky decomposition* [8,9] is used, resulting in lower-triangular $R$. The reader can verify that the weighted mean and covariance of $\mathcal{X}$ equal the original mean and covariance of random vector $x$ for a specific set of $\{\gamma,\alpha^{(m)},\alpha^{(c)}\}$ if we define the weighted mean as $\bar{x}=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{X}_i$, the weighted covariance as $\Sigma_{\tilde{x}}=\sum_{i=0}^{p}\alpha_i^{(c)}(\mathcal{X}_i-\bar{x})(\mathcal{X}_i-\bar{x})^{\mathrm{T}}$, $\mathcal{X}_i$ as the $i$th element of $\mathcal{X}$, and both $\alpha_i^{(m)}$ and $\alpha_i^{(c)}$ as real scalars with the necessary (but not sufficient) conditions that $\sum_{i=0}^{p}\alpha_i^{(m)}=1$ and $\sum_{i=0}^{p}\alpha_i^{(c)}=1$. The various sigma-point methods differ only in the choices taken for these weighting constants. Values for the two most common methods – the *unscented Kalman filter* (UKF) [10–15] and the *central difference Kalman filter* (CDKF) [16–18] – are summarized in Table 2. In this work, we use the CDKF parameters.

Table 3
Summary of the nonlinear sigma-point Kalman filter

Nonlinear state–space model

$x_k=f(x_{k-1},u_{k-1},w_{k-1},k-1)$

$y_k=h(x_k,u_k,v_k,k)$

where $w_k$ and $v_k$ are independent, Gaussian noise processes of covariance matrices $\Sigma_w$ and $\Sigma_v$, respectively

Definitions: let

$x_k^a=[x_k^{\mathrm{T}},\ w_k^{\mathrm{T}},\ v_k^{\mathrm{T}}]^{\mathrm{T}},\quad \mathcal{X}_k^a=[(\mathcal{X}_k^x)^{\mathrm{T}},\ (\mathcal{X}_k^w)^{\mathrm{T}},\ (\mathcal{X}_k^v)^{\mathrm{T}}]^{\mathrm{T}},\quad p=2\times\dim(x_k^a)$

Initialization: for $k=0$, set

$\hat{x}_0^+=\mathbb{E}[x_0]$ $\qquad$ $\hat{x}_0^{a,+}=\mathbb{E}[x_0^a]=[(\hat{x}_0^+)^{\mathrm{T}},\ \bar{w},\ \bar{v}]^{\mathrm{T}}$

$\Sigma_{\tilde{x},0}^+=\mathbb{E}[(x_0-\hat{x}_0^+)(x_0-\hat{x}_0^+)^{\mathrm{T}}]$ $\qquad$ $\Sigma_{\tilde{x},0}^{a,+}=\mathbb{E}[(x_0^a-\hat{x}_0^{a,+})(x_0^a-\hat{x}_0^{a,+})^{\mathrm{T}}]=\mathrm{diag}(\Sigma_{\tilde{x},0}^+,\Sigma_w,\Sigma_v)$

Computation: for $k=1,2,\ldots$ compute

State estimate time update

$\mathcal{X}_{k-1}^{a,+}=\{\hat{x}_{k-1}^{a,+},\ \hat{x}_{k-1}^{a,+}+\gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}},\ \hat{x}_{k-1}^{a,+}-\gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}\}$

$\mathcal{X}_{k,i}^{x,-}=f(\mathcal{X}_{k-1,i}^{x,+},u_{k-1},\mathcal{X}_{k-1,i}^{w,+},k-1)$

$\hat{x}_k^-=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{X}_{k,i}^{x,-}$

Error covariance time update

$\Sigma_{\tilde{x},k}^-=\sum_{i=0}^{p}\alpha_i^{(c)}(\mathcal{X}_{k,i}^{x,-}-\hat{x}_k^-)(\mathcal{X}_{k,i}^{x,-}-\hat{x}_k^-)^{\mathrm{T}}$

Output estimate

$\mathcal{Y}_{k,i}=h(\mathcal{X}_{k,i}^{x,-},u_k,\mathcal{X}_{k-1,i}^{v,+},k)$

$\hat{y}_k=\sum_{i=0}^{p}\alpha_i^{(m)}\mathcal{Y}_{k,i}$

Estimator gain matrix

$\Sigma_{\tilde{y},k}=\sum_{i=0}^{p}\alpha_i^{(c)}(\mathcal{Y}_{k,i}-\hat{y}_k)(\mathcal{Y}_{k,i}-\hat{y}_k)^{\mathrm{T}}$

$\Sigma_{\tilde{x}\tilde{y},k}^-=\sum_{i=0}^{p}\alpha_i^{(c)}(\mathcal{X}_{k,i}^{x,-}-\hat{x}_k^-)(\mathcal{Y}_{k,i}-\hat{y}_k)^{\mathrm{T}}$

$L_k=\Sigma_{\tilde{x}\tilde{y},k}^-\Sigma_{\tilde{y},k}^{-1}$

State estimate measurement update

$\hat{x}_k^+=\hat{x}_k^-+L_k(y_k-\hat{y}_k)$

Error covariance measurement update

$\Sigma_{\tilde{x},k}^+=\Sigma_{\tilde{x},k}^--L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}}$

To use SPKF in an estimation problem, we first define an augmented random vector $x^a$ that combines the randomness of the state, process noise, and sensor noise. This augmented vector is used in the estimation process. The math behind the SPKF is described in Ref. [7] and the overall solution is summarized in Table 3.

## 4. Computationally efficient square-root sigma-point Kalman filters

Sigma-point Kalman filters may be used directly for state estimation and we have shown that they produce better state estimates and much better covariance estimates than EKF [7]. The computational complexity is $\mathcal{O}(L^3)$, which is of equivalent complexity to EKF state estimation, where $L$ is the dimension of the augmented state. They may also be used for parameter estimation, as will be described in the sequel, but the computational complexity remains $\mathcal{O}(L^3)$, whereas the corresponding EKF method has complexity $\mathcal{O}(L^2)$.

The bottleneck in the SPKF algorithm is the computation of the matrix square-root $S_k S_k^{\mathrm{T}} = \Sigma_k$ each time step, which has computational complexity $\mathcal{O}(L^3/6)$ using a Cholesky factorization. A variant of the SPKF, to which we will refer as the square-root SPKF, propagates $S_k$ directly without needing to re-factor each time step [15,19,20]. This approach has several advantages: there are improved numeric properties as the covariances are guaranteed to be positive semi-definite, and although the state estimation update is still $\mathcal{O}(L^3)$, the parameter update may now be done in $\mathcal{O}(L^2)$. Therefore, for the same computational complexity of EKF, better results are obtained.

Three important linear-algebra techniques are required to implement SR-SPKF [8,9]:

- *QR decomposition*: The QR decomposition algorithm computes two factors $Q \in \mathbb{R}^{N\times N}$ and $R \in \mathbb{R}^{L\times N}$ for a matrix $A \in \mathbb{R}^{L\times N}$ such that $A = QR$, $Q$ is orthogonal, $R$ is upper-triangular, and $N \geq L$. The property of the QR factorization that is important here is that $R$ is related to the Cholesky factor we wish to find. Specifically, if $\tilde{R} \in \mathbb{R}^{L\times L}$ is the upper-triangular portion of $R$, then $\tilde{R}^{\mathrm{T}}$ is the Cholesky factor of $\Sigma = A^{\mathrm{T}}A$. That is, if $\tilde{R} = \mathrm{qr}(A^{\mathrm{T}})^{\mathrm{T}}$, where $\mathrm{qr}(\cdot)$ performs the QR decomposition and returns the upper-triangular portion of $R$ only, then $\tilde{R} = \mathrm{chol}(AA^{\mathrm{T}})$.[3]

  This information may be used by noticing that step 2 of the SPKF method computes

  $$\Sigma^-_{\tilde{x},k} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}^{x,-}_{k,i} - \hat{x}_k^-)(\mathcal{X}^{x,-}_{k,i} - \hat{x}^x_{k,i})^{\mathrm{T}},$$

  which may also be written as $\Sigma^-_{\tilde{x},k} = AA^{\mathrm{T}}$, where $A = [\sqrt{\alpha_i^{(\mathrm{c})}}(\mathcal{X}^{x,-}_{k,(0:p)} - \hat{x}_k^-)]$. A similar computation is performed in step 4. Rather than computing $AA^{\mathrm{T}}$ and then later computing the Cholesky factor thereof, we can instead compute the QR decomposition of $A^{\mathrm{T}}$, and propagate the $\tilde{R}$ component. The computational complexity of the QR decomposition is $\mathcal{O}(NL^2)$, whereas the complexity of the Cholesky factor is $\mathcal{O}(L^3/6)$ plus $\mathcal{O}(NL^2)$ to first compute $AA^{\mathrm{T}}$.
- *Cholesky downdating*: Note that the previous method fails whenever $\alpha_i^{(\mathrm{c})}$ is negative, because the square-root involved will not have a real value. In particular, $\alpha_0^{(\mathrm{c})}$ is frequently negative, and the final step of SPKF computes $\Sigma^+_{\tilde{x},k} = \Sigma^-_{\tilde{x},k} - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$. Using the latter case as an example, we cannot simply append a column to $A$ with value $\sqrt{-\Sigma_{\tilde{y},k}} L_k$ because of the negative sign inside the square-root. The solution to this problem is the Cholesky downdating procedure, which takes as input the Cholesky factor to $AA^{\mathrm{T}}$ and $\sqrt{\Sigma_{\tilde{y},k}} L_k$, and computes the updated factor for $AA^{\mathrm{T}} - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$. If $\Sigma_{\tilde{y},k}$ is not a scalar, downdating is done using each of its columns. This algorithm is only $\mathcal{O}(L^2)$.
- *Backsubstitution*: Finally, the solution to the estimator gain matrix computation is $L_k \Sigma_{\tilde{y},k} = \Sigma^-_{\tilde{x}\tilde{y},k}$, which may be written as $L_k = \Sigma^-_{\tilde{x}\tilde{y},k}(S_{\tilde{y},k} S^{\mathrm{T}}_{\tilde{y},k})^{-1}$, or alternately as $L_k(S_{\tilde{y},k} S^{\mathrm{T}}_{\tilde{y},k}) = \Sigma^-_{\tilde{x}\tilde{y},k}$. If $y_k$ is not a scalar, this equation may often be computed most efficiently via back-substitution in two steps. First, $(z)S^{\mathrm{T}}_{\tilde{y},k} = \Sigma^-_{\tilde{x}\tilde{y},k}$ is found, and then $L_k S_{\tilde{y},k} = z$ is solved. Since $S_{\tilde{y},k}$ is already triangular, no matrix inversion need be done. Backsubstitution has complexity $\mathcal{O}(N^2/2)$.

To use SR-SPKF in an estimation problem, we again define an augmented random vector $x^a$ that combines the randomness of the state, process noise, and sensor noise. This augmented vector is used in the estimation process as described below.

*SR-SPKF step 1: state estimate time update*. As with SPKF, each measurement interval, the state estimate time update is computed by first forming the augmented *a posteriori* state estimate vector for the previous time interval: $\hat{x}^{a,+}_{k-1} = [(\hat{x}^+_{k-1})^{\mathrm{T}}, \bar{w}, \bar{v}]^{\mathrm{T}}$. With SR-SPKF, the square-root augmented *a posteriori* covariance estimate is computed: $S^{a,+}_{\tilde{x},k-1} = \mathrm{diag}(S^+_{\tilde{x},k-1}, S_w, S_v)$. These factors are used to generate the $p+1$ sigma points: $\mathcal{X}^{a,+}_{k-1} = \{\hat{x}^{a,+}_{k-1}, \hat{x}^{a,+}_{k-1} + \gamma S^{a,+}_{\tilde{x},k-1}, \hat{x}^{a,+}_{k-1} - \gamma S^{a,+}_{\tilde{x},k-1}\}$. From the augmented sigma points, the $p+1$ vectors comprising the state portion $\mathcal{X}^{x,+}_{k-1}$ and the $p+1$ vectors comprising the process noise portion $\mathcal{X}^{w,+}_{k-1}$ are extracted. The process equation is evaluated using all pairs of $\mathcal{X}^{x,+}_{k-1,i}$ and $\mathcal{X}^{w,+}_{k-1,i}$ (where the subscript $i$ denotes that the $i$th element is being extracted from the original set), yielding the *a priori* sigma points $\mathcal{X}^{x,-}_{k,i}$ for time step $k$. Finally, the *a priori* state estimate is computed as $\hat{x}_k^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}^{x,-}_{k,i}$.

*SR-SPKF step 2: error covariance time update.* Using the *a priori* sigma points from step 1, the square-root *a priori* covariance estimate is computed as

$$S^-_{\tilde{x},k} = \mathrm{qr}\left\{\left[\sqrt{\alpha_i^{(\mathrm{c})}}\left(\mathcal{X}^{x,-}_{k,(0:p)} - \hat{x}_k^-\right)^{\mathrm{T}}\right]\right\}^{\mathrm{T}}.$$

*SR-SPKF step 3: estimate system output $y_k$*. The system output is estimated by evaluating the model output equation using

[3] Some implementational care is advised as the QR decomposition and Cholesky factorizations are not defined identically by all numeric packages. For example, chol() or qr() may return the transpose of the desired result.

Table 4
Summary of the square-root sigma-point Kalman filter for state estimation

| | |
|---|---|
| Nonlinear state–space model | |
| $x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)$ | |
| $y_k = h(x_k, u_k, v_k, k)$ | |
| where $w_k$ and $v_k$ are independent, zero-mean, Gaussian noise processes of covariance matrices $\Sigma_w$ and $\Sigma_v$, respectively | |
| Definitions: let | |
| $x_k^a = [x_k^{\mathrm{T}},\ w_k^{\mathrm{T}},\ v_k^{\mathrm{T}}]^{\mathrm{T}}, \quad \mathcal{X}_k^a = [(\mathcal{X}_k^x)^{\mathrm{T}},\ (\mathcal{X}_k^w)^{\mathrm{T}},\ (\mathcal{X}_k^v)^{\mathrm{T}}]^{\mathrm{T}}, \quad p = 2 \times \dim(x_k^a)$ | |
| Initialization: for $k = 0$, set | |
| $\hat{x}_0^+ = \mathbb{E}[x_0]$ | $\hat{x}_0^{a,+} = \mathbb{E}[x_0^a] = [(\hat{x}_0^+)^{\mathrm{T}},\ 0,\ 0]^{\mathrm{T}}$ |
| $S_{\tilde{x},0}^+ = \mathrm{chol}\{\mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^{\mathrm{T}}]\}$ | $S_{\tilde{x},0}^{a,+} = \mathrm{chol}\{\mathbb{E}[(x_0^a - \hat{x}_0^{a,+})(x_0^a - \hat{x}_0^{a,+})^{\mathrm{T}}]\} = \mathrm{diag}(S_{\tilde{x},0}^+, S_w, S_v)$ |
| Computation: for $k = 1, 2, \ldots$ compute | |
| State estimate time update | $\mathcal{X}_{k-1}^{a,+} = \{\hat{x}_{k-1}^{a,+},\ \hat{x}_{k-1}^{a,+} + \gamma S_{\tilde{x},k-1}^{a,+},\ \hat{x}_{k-1}^{a,+} - \gamma S_{\tilde{x},k-1}^{a,+}\}$<br>$\mathcal{X}_{k,i}^{x,-} = f(\mathcal{X}_{k-1,i}^{x,+}, u_{k-1}, \mathcal{X}_{k-1,i}^{w,+}, k-1)$<br>$\hat{x}_k^- = \sum_{i=0}^p \alpha_i^{(\mathrm{m})} \mathcal{X}_{k,i}^{x,-}$ |
| Error covariance time update | $S_{\tilde{x},k}^- = \mathrm{qr}\{[\sqrt{\alpha_i^{(\mathrm{c})}}(\mathcal{X}_{k,(0:p)}^{x,-} - \hat{x}_k^-)^{\mathrm{T}}]\}^{\mathrm{T}}$ |
| Output estimate | $\mathcal{Y}_{k,i} = h(\mathcal{X}_{k,i}^{x,-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, k)$<br>$\hat{y}_k = \sum_{i=0}^p \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}$ |
| Estimator gain matrix | $S_{\tilde{y},k} = \mathrm{qr}\{[\sqrt{\alpha_i^{(\mathrm{c})}}(\mathcal{Y}_{k,(0:p)} - \hat{y}_k)^{\mathrm{T}}]\}^{\mathrm{T}}$<br>$\Sigma_{\tilde{x}\tilde{y},k}^- = \sum_{i=0}^p \alpha_i^{(\mathrm{c})}(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$<br>$L_k = \Sigma_{\tilde{x}\tilde{y},k}^-(S_{\tilde{y},k}^{\mathrm{T}} S_{\tilde{y},k})^{-1}$ (solved by backsubstitution) |
| State estimate measurement update | $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$ |
| Error covariance measurement update | $S_{\tilde{x},k}^+ = \mathrm{downdate}\{S_{\tilde{x},k}^-, L_k S_{\tilde{y},k}\}$ |

the sigma points describing the spread in the state and noise vectors. First, we compute the points $\mathcal{Y}_{k,i} = h(\mathcal{X}_{k,i}^{x,-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, k)$. The output estimate is then $\hat{y}_k = \sum_{i=0}^p \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}$.

*SR-SPKF step 4: estimator gain matrix $L_k$.* To compute the estimator gain matrix, we must first compute the required covariance and square-root covariance matrices.

$$S_{\tilde{y},k} = \mathrm{qr}\left\{\left[\sqrt{\alpha_i^{(\mathrm{c})}}(\mathcal{Y}_{k,(0:p)} - \hat{y}_k)^{\mathrm{T}}\right]\right\}^{\mathrm{T}}$$

$$\Sigma_{\tilde{x}\tilde{y},k}^- = \sum_{i=0}^p \alpha_i^{(\mathrm{c})}(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)(\mathcal{Y}_{k,i} - \hat{y}_k).$$

Then, we simply compute $L_k = \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}$, solved by backsubstitution.

*SR-SPKF step 6: error covariance measurement update.* The final step computes the square-root form of $\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$ as $S_{\tilde{x},k}^+ = \mathrm{downdate}\left\{S_{\tilde{x},k}^-, L_k S_{\tilde{y},k}\right\}$. The SR-SPKF solution for state estimation is summarized in Table 4, and results for both SPKF and SR-SPKF for SOC estimation are given in Section 8.3.

## 5. Parameter estimation using SR-SPKF

The various methods for state estimation presented so far have assumed a known system model in the form of Eqs. (1) and (2). These equations will generally involve numeric values in their computations. Some of these values might be intrinsic constants, but others might be factors determined by the electrochemistry or construction of a particular cell. We refer to these latter factors as the "parameters" of the cell model.

To use the enhanced self-correcting (ESC) cell model as an example (cf. Section 7), the parameters comprise the following: the Coulombic efficiency $\eta$, the total capacity $C$, the filter poles $\alpha_1, \ldots, \alpha_{n_f}$, the filter weighting factors $g_1, \ldots, g_{n_{f-1}}$, the cell discharge and charge resistances $R^+$ and $R^-$, the hysteresis rate constant $\gamma$, and the maximum level of hysteresis $M$. Combined, they are

$$\theta = [\eta, C, \alpha_1, \ldots, \alpha_{n_f}, g_1, \ldots, g_{n_{f-1}}, R^+, R^-, \gamma, M]^{\mathrm{T}}.$$

We have previously shown how to estimate a system's state given a known model and noisy measurements. We now show how to estimate a system's parameters given a known state and clean measurements. We assume that there is a true value for $\theta$ that describes the cell under consideration, and wish to adapt an estimate $\hat{\theta}$ to converge to the true value. We begin by proposing a state–space model for the "dynamics" of the system parameters.

$$\theta_k = \theta_{k-1} + r_{k-1}$$
$$d_k = h(x_k, u_k, \bar{v}_k, \theta_k, k) + e_k.$$

The first equation states that the parameters are essentially constant, but that they may change slowly over time by some driving process, modeled by a process $r_k$ of small fictitious "noise". The output equation for the state–space model of true parameter dynamics is the estimate of cell output based on the previous state, the measured inputs, and parameter estimates, added to some estimation error $e_k$.

With this system defined, we can apply SR-SPKF to estimate the parameters. The SR-SPKF solution for parameter estimation is summarized in Table 5. The dominant differences between state and parameter estimation are: in parameter estimation the noises are assumed to contribute linearly to the state and measurement equations, and an approximate process noise update is performed (see [15] for details).

Table 5
Summary of the square-root sigma-point Kalman filter for parameter estimation

Nonlinear state–space model

$\theta_k = \theta_{k-1} + r_{k-1}$

$d_k = h(x_k, u_k, \bar{v}_k, \theta_k, k) + e_k$

where $r_k$ and $e_k$ are independent, zero-mean, Gaussian noise processes of covariance matrices $\Sigma_r$ and $\Sigma_e$, respectively

Definition: let

$$D_{r,k-1} = -\mathrm{diag}\{S^+_{\tilde{\theta},k-1}\} + \sqrt{\mathrm{diag}\{S^+_{\tilde{\theta},k-1}\}^2 + \mathrm{diag}\{\Sigma_{r,k-1}\}} \quad p = 2 \times \dim(\theta_k)$$

Initialization: for $k = 0$, set

$$\hat{\theta}^+_0 = \mathbb{E}[\theta_0] \quad S^+_{\tilde{\theta},0} = \mathrm{chol}\left\{\mathbb{E}[(\theta_0 - \hat{\theta}^+_0)(\theta_0 - \hat{\theta}^+_0)^{\mathrm{T}}]\right\}$$

Computation: for $k = 1, 2, \ldots$ compute

| | |
|---|---|
| Parameter estimate time update | $\hat{\theta}^-_k = \hat{\theta}^+_{k-1}$ |
| Error covariance time update | $S^-_{\tilde{\theta},k} = S^+_{\tilde{\theta},k-1} + D_{r,k-1}$ |
| Output estimate | $\mathcal{W}_k = \{\hat{\theta}^-_k, \hat{\theta}^-_k + \gamma S^-_{\tilde{\theta},k}, \hat{\theta}^-_k - \gamma S^-_{\tilde{\theta},k}\}$<br>$\mathcal{D}_{k,i} = h(x_k, u_k, \bar{v}_k, \mathcal{W}_{k,i}, k)$<br>$\hat{d}_k = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{D}_{k,i}$ |
| Estimator gain matrix | $S_{\tilde{d},k} = \mathrm{qr}\{[\sqrt{\alpha_i^{(\mathrm{c})}}(\mathcal{D}_{k,(0:p)} - \hat{d}_k) \quad \sqrt{\Sigma_e}]^{\mathrm{T}}\}^{\mathrm{T}}$<br>$\Sigma^-_{\tilde{\theta}\tilde{d},k} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})}(\mathcal{W}_{k,i} - \hat{\theta}^-_k)(\mathcal{D}_{k,i} - \hat{d}_k)^{\mathrm{T}}$<br>$L_k = \Sigma^-_{\tilde{\theta}\tilde{d},k}(S^{\mathrm{T}}_{\tilde{d},k} S_{\tilde{d},k})^{-1}$ (solved by backsubstitution) |
| Parameter estimate measurement update | $\hat{\theta}^+_k = \hat{\theta}^-_k + L_k(d_k - \hat{d}_k)$ |
| Error covariance measurement update | $S^+_{\tilde{\theta},k} = \mathrm{downdate}\{S^-_{\tilde{\theta},k}, L_k S_{\tilde{d},k}\}$ |

## 6. Joint and dual sigma-point filtering

We have now shown how to estimate the state of a system given a known model and noisy measurements and how to estimate the parameters of the system given a known state and clean measurements. We proceed to give two methods whereby one can simultaneously estimate both the state and parameters of a system given noisy measurements.

The various methods of state estimation presented so far have assumed a constant system model. However, when applying these procedures to estimate battery SOC, for example, we encounter a possible source of error: not all cells are created equal. Most of the research reported here has been conducted using prototype high-power LiPB cells, constructed by hand. In these cells, there is a great deal of variability in resistance, capacity, time constants, and so forth. Even when mass-produced, however, there is some cell-to-cell variation, which only increases as the cells age, both in accumulated cycles and in calendar life.

Some of the critical parameters, such as cell resistance and capacity, directly limit the pack performance through "power fade" and "capacity fade", so are indicators of the battery state-of-health. It is important to be able to estimate these and other parameters to: (1) maintain an accurate model for SOC estimation, (2) understand the present battery state-of-health, and (3) predict remaining service life.

Keeping in mind the previous discussion on estimating SOC, it is apparent that the quantities descriptive of the present battery pack condition exist on two time scales. Some change rapidly, such as SOC, which can traverse its entire range within minutes. Others may change very slowly, such as cell capacity, which might change as little as 20% in a decade or more of regular use. The quantities that tend to change quickly comprise the *state* of the system, and the quantities that tend to change slowly comprise the *time-varying parameters* of the system.

Any of the KF methods used to estimate SOC might be adapted to concurrently estimate both the state and the slowly time-varying cell parameters by combining the cell model state vector with the model parameters and simultaneously estimating the values of this augmented state vector. This method is called *joint estimation*. It has the disadvantages of large matrix operations due to the high dimensionality of the resulting augmented model and potentially poor numeric conditioning due to the vastly different time scales of the states (including parameters) in the augmented state vector. However, it is quite straightforward to implement. We first combine the state and parameter vectors to form an augmented state such that the dynamics may be represented by

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, u_{k-1}, \theta_{k-1}, w_{k-1}, k-1) \\ \theta_{k-1} + r_{k-1} \end{bmatrix}$$

$$y_k = h(x_k, u_k, v_k, k).$$

Note that to simplify notation, we will refer to the vector comprising both the present state and the present parameters as $\mathbb{X}_k$, the vector comprising the present process noise and present parameter noise as $\mathbb{W}_k$, and the equation combining the dynamics of the state and the dynamics of the parameters as $\mathcal{F}$. This allows us to write

$$\mathbb{X}_k = \mathcal{F}(\mathbb{X}_{k-1}, u_{k-1}, \mathbb{W}_{k-1}, k-1)$$

$$y_k = h(\mathbb{X}_k, u_k, v_k, k).$$

With the augmented model of the system state dynamics and parameter dynamics defined, we apply the SPKF method. Table 6 gives a listing of the steps, which correspond directly to the state estimation SPKF steps, except with larger matrix operations.

The second method, *dual estimation*, also uses a Kalman filtering approach to estimate both state and parameter values, but

Table 6
Summary of the nonlinear sigma-point Kalman filter for joint estimation

State–space model

$$\begin{bmatrix} x_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} f(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}, k-1) \\ \theta_{k-1} + r_{k-1} \end{bmatrix} \text{ or } \begin{matrix} \mathbb{X}_k = \mathcal{F}(\mathbb{X}_{k-1}, u_{k-1}, \mathbb{W}_{k-1}, k-1) \\ y_k = h(\mathbb{X}_k, u_k, v_k, k) \end{matrix}$$

$y_k = h(x_k, u_k, v_k, \theta_k, k)$

where $w_k$, $r_k$, and $v_k$ are independent, Gaussian noise processes of covariance matrices $\Sigma_w$, $\Sigma_r$, and $\Sigma_v$, respectively. For brevity, we let $\mathbb{X}_k = [x_k^{\mathrm{T}},\ \theta_k^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbb{W}_k = [w_k^{\mathrm{T}},\ r_k^{\mathrm{T}}]^{\mathrm{T}}$ and $\Sigma_{\mathbb{W}} = \mathrm{diag}(\Sigma_w, \Sigma_r)$

Definitions: let

$$\mathbb{X}_k^a = [\mathbb{X}_k^{\mathrm{T}},\ \mathbb{W}_k^{\mathrm{T}},\ v_k^{\mathrm{T}}]^{\mathrm{T}}, \quad \mathcal{X}_k^a = [(\mathcal{X}_k^{\mathbb{X}})^{\mathrm{T}},\ (\mathcal{X}_k^{\mathbb{W}})^{\mathrm{T}},\ (\mathcal{X}_k^{v})^{\mathrm{T}}]^{\mathrm{T}}, \quad p = 2 \times \dim(\mathbb{X}_k^a)$$

Initialization: for $k = 0$, set

$$\hat{\mathbb{X}}_0^+ = \mathbb{E}[\mathbb{X}_0] \qquad \hat{\mathbb{X}}_0^{a,+} = \mathbb{E}[\mathbb{X}_0^a] = [(\hat{\mathbb{X}}_0^+)^{\mathrm{T}},\ \bar{\mathbb{W}},\ \bar{v}]^{\mathrm{T}}$$

$$\Sigma_{\tilde{\mathbb{X}},0}^+ = \mathbb{E}[(\mathbb{X}_0 - \hat{\mathbb{X}}_0^+)(\mathbb{X}_0 - \hat{\mathbb{X}}_0^+)^{\mathrm{T}}] \qquad \Sigma_{\tilde{\mathbb{X}},0}^{a,+} = \mathbb{E}[(\mathbb{X}_0^a - \hat{\mathbb{X}}_0^{a,+})(\mathbb{X}_0^a - \hat{\mathbb{X}}_0^{a,+})^{\mathrm{T}}] = \mathrm{diag}(\Sigma_{\tilde{\mathbb{X}},0}^+, \Sigma_{\mathbb{W}}, \Sigma_v)$$

Computation: for $k = 1, 2, \ldots$ compute

State estimate time update

$$\mathcal{X}_{k-1}^{a,+} = \left\{ \hat{\mathbb{X}}_{k-1}^{a,+},\ \hat{\mathbb{X}}_{k-1}^{a,+} + \gamma\sqrt{\Sigma_{\tilde{\mathbb{X}},k-1}^{a,+}},\ \hat{\mathbb{X}}_{k-1}^{a,+} - \gamma\sqrt{\Sigma_{\tilde{\mathbb{X}},k-1}^{a,+}} \right\}$$

$$\mathcal{X}_{k,i}^{\mathbb{X},-} = \mathcal{F}(\mathcal{X}_{k-1,i}^{\mathbb{X},+}, u_{k-1}, \mathcal{X}_{k-1,i}^{\mathbb{W},+}, k-1)$$

$$\hat{\mathbb{X}}_k^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}_{k,i}^{\mathbb{X},-}$$

Error covariance time update

$$\Sigma_{\tilde{\mathbb{X}},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_{k,i}^{\mathbb{X},-} - \hat{\mathbb{X}}_k^-)(\mathcal{X}_{k,i}^{\mathbb{X},-} - \hat{\mathbb{X}}_k^-)^{\mathrm{T}}$$

Output estimate

$$\mathcal{Y}_{k,i} = h(\mathcal{X}_{k,i}^{\mathbb{X},-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, k)$$

$$\hat{y}_k = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}$$

Estimator gain matrix

$$\Sigma_{\tilde{y},k} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{Y}_{k,i} - \hat{y}_k)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$$

$$\Sigma_{\tilde{\mathbb{X}}\tilde{y},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_{k,i}^{\mathbb{X},-} - \hat{\mathbb{X}}_k^-)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$$

$$L_k = \Sigma_{\tilde{\mathbb{X}}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}$$

State estimate measurement update

$$\hat{\mathbb{X}}_k^+ = \hat{\mathbb{X}}_k^- + L_k(y_k - \hat{y}_k)$$

Error covariance measurement update

$$\Sigma_{\tilde{\mathbb{X}},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$$

uses separate Kalman filters for state estimation and parameter estimation. The computational complexity is smaller and the matrix operations may be numerically better conditioned. However, by decoupling state from parameters, any cross-correlations between changes are lost, leading to potentially poorer accuracy.

The mathematical model of cell dynamics again explicitly includes the parameters as the vector $\theta_k$:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}, k-1)$$
$$y_k = h(x_k, u_k, v_k, \theta_{k-1}, k).$$

Non-time-varying numeric values required by the model may be embedded within $f(\cdot)$ and $h(\cdot)$, and are not included in $\theta_k$. We also slightly revise the mathematical model of parameter dynamics to explicitly include the effect of the state equation.

$$\theta_k = \theta_{k-1} + r_{k-1}$$
$$d_k = h(f(x_{k-1}, u_{k-1}, \bar{w}_{k-1}, \theta_{k-1}, k-1),\ u_k, \bar{v}_k, \theta_{k-1}, k) + e_k.$$

With these two systems defined, we can apply the standard procedure of dual extended Kalman filtering [6,21], or generalize the procedure to other forms of Kalman filtering. Dual sigma-point Kalman filtering is outlined in Table 7, and comprises two carefully integrated SPKFs. The algorithm is initialized with the best guess of the true parameters $\hat{\theta}_0^+ = \mathbb{E}[\theta_0]$, and with the best guess of the cell state $\hat{x}_0^+ = \mathbb{E}[x_0]$. The estimation error covariance matrices are also initialized. The algorithm may be adapted to use an SR-SPKF for the state- and/or the parameter-estimation filter in a straightforward manner.

The dual sigma-point Kalman filter can be viewed by drawing a block diagram, as in Fig. 1. We see that the process essentially comprises two sigma-point Kalman filters running in parallel – one adapting the state and one adapting parameters – with some information exchange between the filters.

### 6.1. Convergence

The dual/joint extended Kalman filters will adapt $\hat{x}_k$ and $\hat{\theta}_k$ so that the model input-output relationship matches the cell input-output data as closely as possible. There is no built-in guarantee that the state or parameters of the model converge to anything with physical meaning. We take special steps to ensure that this occurs.

A very crude cell model may be used, combined with the dual/joint SPKF, to ensure convergence of the SOC state. Specif-

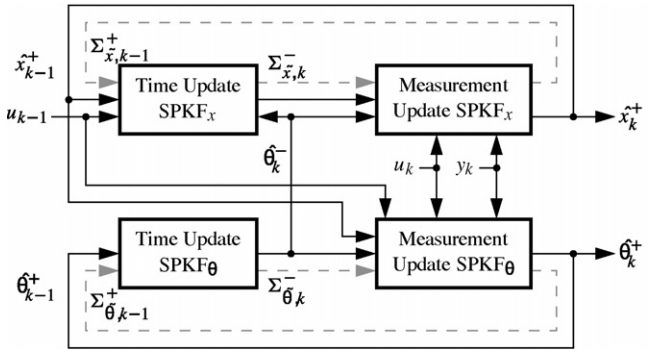


Fig. 1. Diagram of dual estimation method. Solid lines represent state- and parameter-vector signal flow, and dashed gray lines represent error covariance matrix signal flow.

Table 7
Summary of the sigma-point Kalman filter for dual state and parameter estimation

Nonlinear state–space models

$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}, \theta_{k-1}, k-1)$ and $\theta_k = \theta_{k-1} + r_{k-1}$
$y_k = h(x_k, u_k, v_k, \theta_k, k)$ $\quad d_k = h(f(x_{k-1}, u_{k-1}, \bar{w}_{k-1}, \theta_{k-1}, k-1), u_k, \bar{v}_k, \theta_{k-1}, k) + e_k$

where $w_k$, $v_k$, $r_k$ and $e_k$ are independent, Gaussian noise processes of covariance matrices $\Sigma_w$, $\Sigma_v$, $\Sigma_r$ and $\Sigma_e$, respectively

Definitions

$x_k^a = [x_k^{\mathrm{T}},\ w_k^{\mathrm{T}},\ v_k^{\mathrm{T}}]^{\mathrm{T}},\ \mathcal{X}_k^a = [(\mathcal{X}_k^x)^{\mathrm{T}},\ (\mathcal{X}_k^w)^{\mathrm{T}},\ (\mathcal{X}_k^v)^{\mathrm{T}}]^{\mathrm{T}},\ p = 2 \times \dim(x_k^a)$

Initialization: for $k = 0$, set

$\hat{\theta}_0^+ = \mathbb{E}[\theta_0]$ $\quad \Sigma_{\tilde{\theta},0}^+ = \mathbb{E}[(\theta_0 - \hat{\theta}_0^+)(\theta_0 - \hat{\theta}_0^+)^{\mathrm{T}}]$

$\hat{x}_0^+ = \mathbb{E}[x_0]$ $\quad \hat{x}_0^{a,+} = \mathbb{E}[x_0^a] = [(\hat{x}_0^+)^{\mathrm{T}},\ \bar{w},\ \bar{v}]^{\mathrm{T}}$

$\Sigma_{\tilde{x},0}^+ = \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^{\mathrm{T}}]$ $\quad \Sigma_{\tilde{x},0}^{a,+} = \mathbb{E}[(x_0^a - \hat{x}_0^{a,+})(x_0^a - \hat{x}_0^{a,+})^{\mathrm{T}}] = \mathrm{diag}(\Sigma_{\tilde{x},0}^+, \Sigma_w, \Sigma_v)$

Computation: for $k = 1, 2, \ldots$ compute

Parameter estimate time update: $\hat{\theta}_k^- = \hat{\theta}_{k-1}^+$

Parameter covariance time update: $\Sigma_{\tilde{\theta},k}^- = \Sigma_{\tilde{\theta},k-1}^+ + \Sigma_r$

State estimate time update:
$$\mathcal{X}_{k-1}^{a,+} = \left\{ \hat{x}_{k-1}^{a,+},\ \hat{x}_{k-1}^{a,+} + \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}},\ \hat{x}_{k-1}^{a,+} - \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}} \right\}$$
$$\mathcal{X}_{k,i}^{x,-} = f(\mathcal{X}_{k-1,i}^{x,+}, u_{k-1}, \mathcal{X}_{k-1,i}^{w,+}, \hat{\theta}_k^-, k-1)$$
$$\hat{x}_k^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}_{k,i}^{x,-}$$

State covariance time update: $\Sigma_{\tilde{x},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)(\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)^{\mathrm{T}}$

Output estimate, parameter filter:
$$\mathcal{W}_k = \left\{ \hat{\theta}_k^-,\ \hat{\theta}_k^- + \gamma\sqrt{\Sigma_{\tilde{\theta},k}^-},\ \hat{\theta}_k^- - \gamma\sqrt{\Sigma_{\tilde{\theta},k}^-} \right\}$$
$$\mathcal{D}_{k,i} = h(f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}, \mathcal{W}_{k,i}, k-1), u_k, \bar{v}_k, \mathcal{W}_{k,i}, k)$$
$$\hat{d}_k = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{D}_{k,i}, \text{ or } \hat{d}_k = \mathcal{D}_{k,0}$$

Output estimate, state filter:
$$\mathcal{Y}_{k,i} = h(\mathcal{X}_{k,i}^{x,-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, \hat{\theta}_k^-, k)$$
$$\hat{y}_k = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}$$

State filter gain matrix:
$$\Sigma_{\tilde{y},k} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{Y}_{k,i} - \hat{y}_k)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$$
$$\Sigma_{\tilde{x}\tilde{y},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$$
$$L_k^x = \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}$$

Parameter filter gain matrix:
$$\Sigma_{\tilde{d},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{D}_{k,i} - \hat{d}_k)(\mathcal{D}_{k,i} - \hat{d}_k)^{\mathrm{T}}$$
$$\Sigma_{\tilde{\theta}\tilde{d},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{W}_{k,i} - \hat{\theta}_k^-)(\mathcal{D}_{k,i} - \hat{d}_k)^{\mathrm{T}}$$
$$L_k^\theta = \Sigma_{\tilde{\theta}\tilde{d},k}^- \Sigma_{\tilde{d},k}^{-1}$$

State estimate measurement update: $\hat{x}_k^+ = \hat{x}_k^- + L_k^x(y_k - \hat{y}_k)$

State covariance measurement update: $\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k^x \Sigma_{\tilde{y},k} (L_k^x)^{\mathrm{T}}$

Parameter estimate measurement update: $\hat{\theta}_k^+ = \hat{\theta}_k^- + L_k^\theta(y_k - \hat{d}_k)$

Parameter covariance measurement update: $\Sigma_{\tilde{\theta},k}^+ = \Sigma_{\tilde{\theta},k}^- - L_k^\theta \Sigma_{\tilde{d},k} (L_k^\theta)^{\mathrm{T}}$

ically,

$$y_k \approx \mathrm{OCV}(z_k) - Ri_k$$
$$\mathrm{OCV}(z_k) \approx y_k + Ri_k$$
$$\hat{z}_k = \mathrm{OCV}^{-1}(y_k + Ri_k).$$

By measuring the cell voltage under load, the cell current, and having knowledge of $R$ (perhaps through $\hat{\theta}$ from the dual/joint SPKF), and knowing the inverse OCV function for the cell chemistry, one can compute a noisy estimate of SOC, $\hat{z}_k$.

To combine this simple model with the dual/joint SPKF, the cell model being used (e.g., perhaps the ESC model in Section 7) has its output equation augmented with SOC:

$$h(x_k, u_k, v_k, \theta_k, k) = \begin{bmatrix} \mathrm{OCV}(z_k) - Ri_k + Gf_k + Mh_k \\ z_k \end{bmatrix}.$$

The joint/dual SPKF is run on this modified model, with the "measured" information in the measurement update being

$$\begin{bmatrix} y_k \\ \hat{z}_k \end{bmatrix}.$$

While the "noise" of $\hat{z}_k$ (short-term bias due to hysteresis effects and polarization filter voltages being ignored) prohibit it from being used as the primary estimator of SOC, its expected long-term behavior in a dynamic environment is accurate, and maintains the accuracy of the SOC state in the joint/dual SPKF.

## 7. The enhanced self-correcting cell model

In order to examine and compare performance of the proposed algorithms, we must first define a discrete-time state–space model of the form of (1) and (2) that applies to battery cells. Here, we briefly review the "enhanced self-correcting cell model" from Refs. [5,3]. This model includes effects due to open-circuit-voltage, internal resistance, voltage time constants, and hysteresis. For the purpose of example, we will later fit pa-

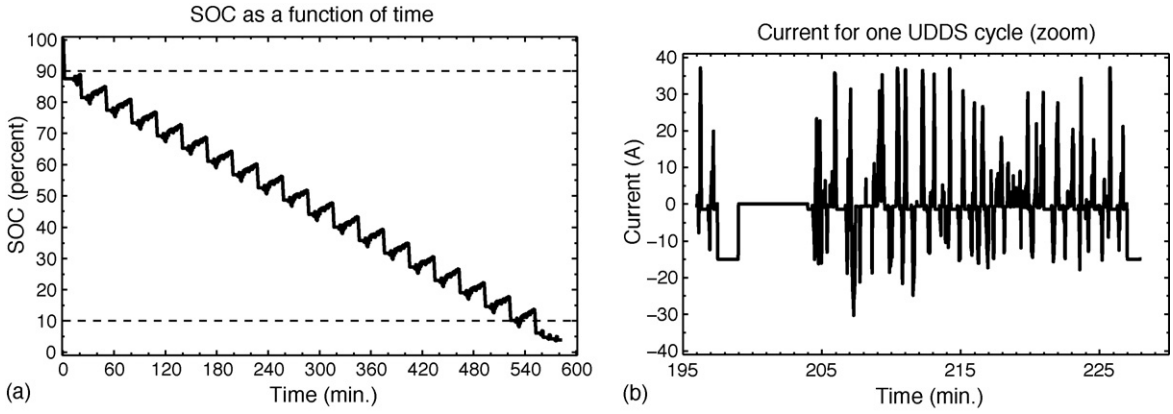


Fig. 2. Plots showing SOC vs. time and rate vs. time for UDDS cell tests. SOC is shown in (a) and rate for one UDDS cycle is shown in (b).

rameter values to this model structure to model the dynamics of high-power lithium-ion polymer battery cells, although the structure and methods presented here are general.

State-of-charge is captured by one state of the model. This equation is

$$z_k = z_{k-1} - \left(\frac{\eta_i \, \Delta T}{C}\right) i_{k-1},$$

where $\Delta T$ represents the inter-sample period (in seconds), and $C$ represents the cell capacity (in ampere-seconds).

The time constants of the cell voltage response are captured by several filter states. If we let there be $n_f$ time constants, then

$$f_k = A_f f_{k-1} + B_f i_{k-1}.$$

The matrix $A_f \in \mathbb{R}^{n_f \times n_f}$ may be a diagonal matrix with real-valued entries. If so, the system is stable if all entries have magnitude less than one. The vector $B_f \in \mathbb{R}^{n_f \times 1}$ may simply be set to $n_f$ "1"s. The value of $n_f$ and the entries in the $A_f$ matrix are chosen as part of the system identification procedure to best fit the model parameters to measured cell data.

The hysteresis level is captured by a single state

$$h_k = \exp\left(-\left|\frac{\eta_i i_{k-1} \gamma \, \Delta T}{C}\right|\right) h_{k-1} + \left(1 - \exp\left(-\left|\frac{\eta_i i_{k-1} \gamma \, \Delta T}{C}\right|\right)\right) \operatorname{sgn}(i_{k-1}),$$

where $\gamma$ is the hysteresis rate constant, again found by system identification.

The overall model state is

$$x_k = [f_k^{\mathrm{T}}, \; h_k, \; z_k]^{\mathrm{T}}.$$

The state equation for the model is formed by combining all of the individual equations, above.

The output equation that combines the state values to predict cell voltage is

$$y_k = \mathrm{OCV}(z_k) + G f_k - R i_k + M h_k,$$

where $G \in \mathbb{R}^{1 \times n_f}$ is a vector of constants that blend the time-constant states together in the output, $R$ the cell resistance (different values may be used for dis/charge), and $M$ is the maximum hysteresis level.

## 8. Application to battery management systems

### 8.1. Cell and cell test description

The cells used in this paper differ electrochemically from those reported in previous work. We refer to the older cells as GEN3 cells, and to the newer cells as G4 cells. The GEN3 cells are high-power (>20 C capable) 7.5Ah Mn spinel/graphite LiPB, and the G4 cells are very high-power (>30 C capable) 5Ah Mn spinel/blended-carbon LiPB, both reported in Ref. [22].

In order to compare the various Kalman filtering methods' abilities to estimate SOC and SOH, we gathered data from two prototype LiPB cells. One cell's data was used to tune cell model parameters, and the second cell's data was used in some tests to see how well the filters generalized to slightly different dynamics than expected. For the tests, we used a Tenney thermal chamber set at 25 °C and an Arbin BT2000 cell cycler. Each channel of the Arbin was capable of 20 A current, and 10 channels were connected in parallel to achieve currents of up to 200 A. The cycler's voltage measurement accuracy was ±5 mV and its current measurement accuracy was ±200 mA.

The cell test we use here comprised a sequence of 18 (full) "urban dynamometer driving schedule" (UDDS) cycles, separated by 15 A discharge pulses and 5-min rests, and spread over the 90–10% SOC range. The SOC as a function of time is plotted in Fig. 2(a), and rate as a function of time for one of the UDDS

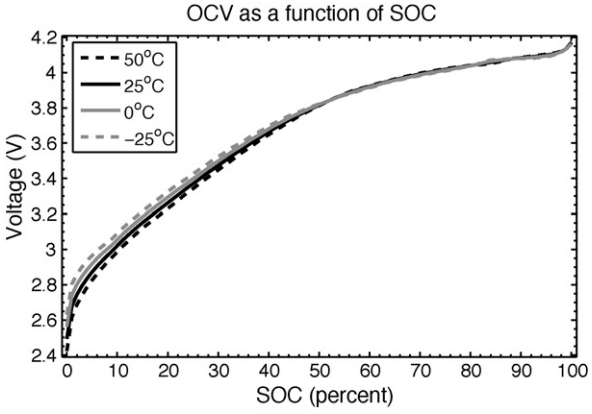


Fig. 3. Plot of open-circuit-voltage as a function of state-of-charge.

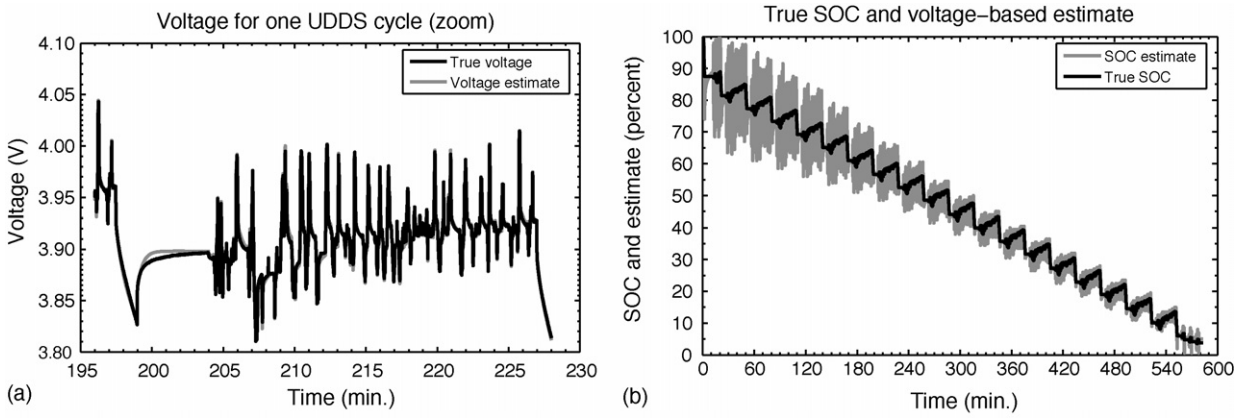


Fig. 4. Modeling of voltage and SOC: (a) ESC modeling of cell voltage vs. true cell voltage for one UDDS cycle and (b) instantaneous voltage-based (non-SPKF-based) SOC estimate plotted vs. true SOC.

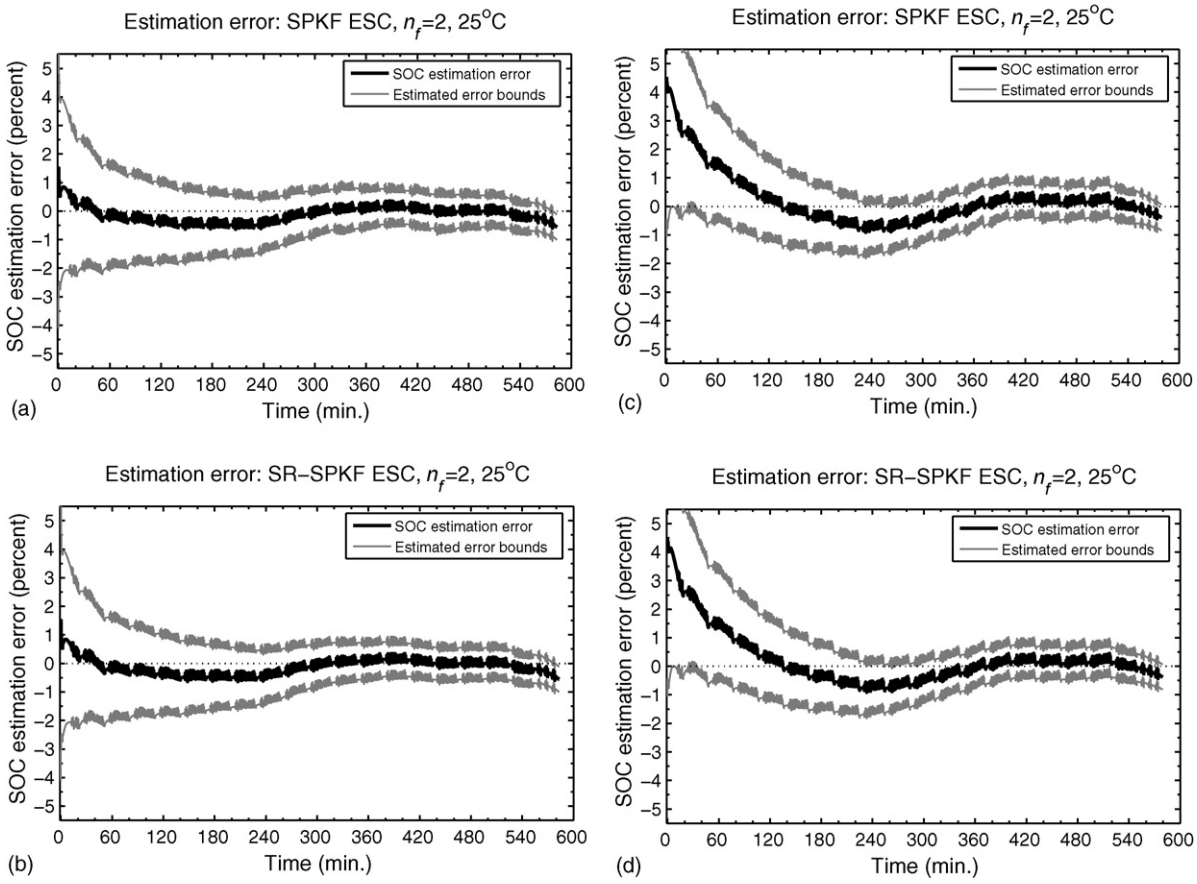


Fig. 5. SOC estimation error for SPKF vs. SR-SPKF with correct filter initialization: (a and b) use training data and (c and d) use generalization data.

Table 8
Comparison of SPKF vs. SR-SPKF in UDDS test results predicting SOC

| | Correctly initialized | | | Incorrectly initialized | | | |
|---|---|---|---|---|---|---|---|
| | RMS error (%) | Maximum error (%) | Bounds error (%) | RMS error (%) | Maximum error (%) | Bounds error (%) | Time to converge (s) |
| SPKF | 0.30 | 1.51 | 0.99 | 1.44 | 20.19 | 2.74 | 1138 |
| SR-SPKF | 0.30 | 1.51 | 0.98 | 1.44 | 20.19 | 2.68 | 1138 |
| SPKF generalize | 0.95 | 4.49 | 0.35 | 2.01 | 23.15 | 7.75 | 2816 |
| SR-SPKF generalize | 0.94 | 4.49 | 0.35 | 2.01 | 23.15 | 7.76 | 2816 |

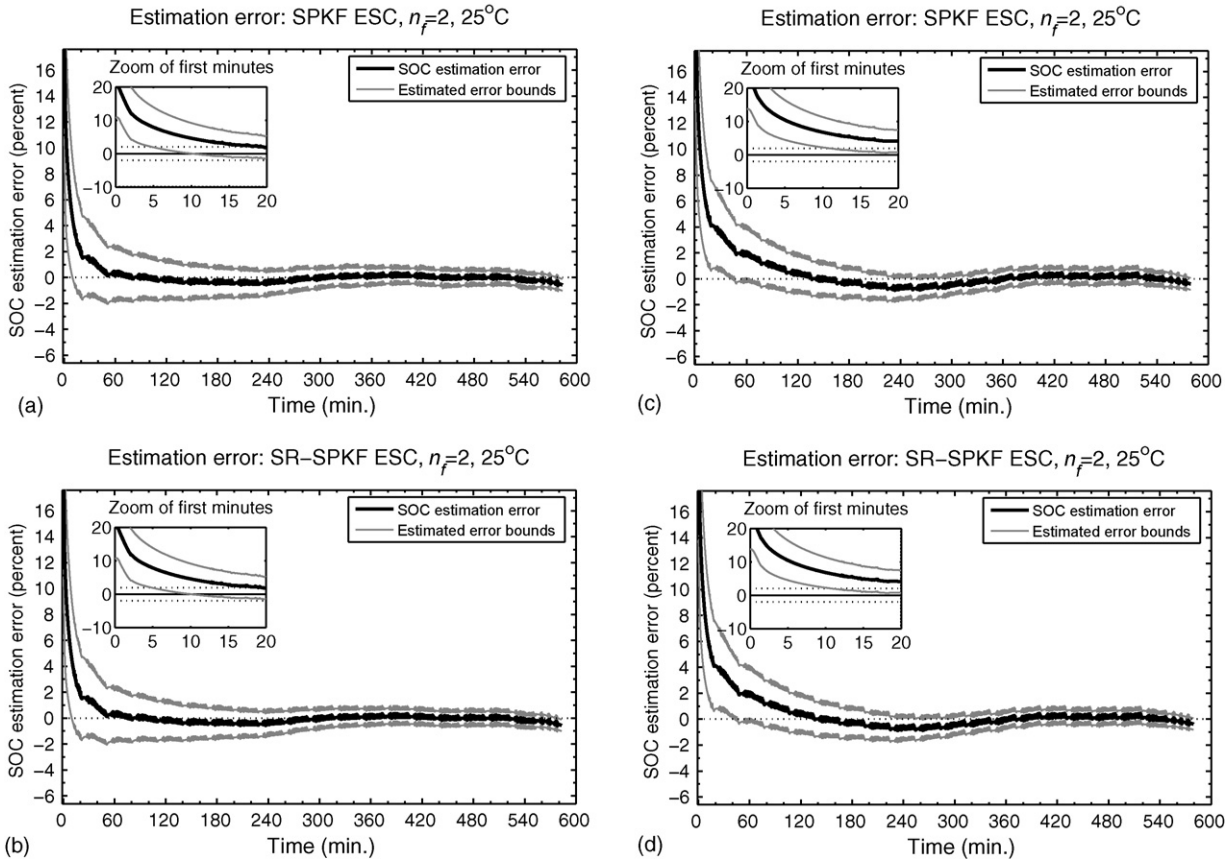


Fig. 6. SOC estimation error for SPKF vs. SR-SPKF with incorrect filter initialization: (a and b) use training data and (c and d) use generalization data.

Table 9
Comparison of joint SPKF vs. dual SPKF vs. dual SR-SPKF in UDDS test results predicting SOC

| | Correctly initialized | | | Incorrectly initialized | | | |
|---|---|---|---|---|---|---|---|
| | RMS error (%) | Maximum error (%) | Bounds error (%) | RMS error (%) | Maximum error (%) | Bounds error (%) | Time to converge (s) |
| Joint SPKF | 0.29 | 1.34 | 1.31 | 1.13 | 19.83 | 3.02 | 1134 |
| Dual SPKF | 0.32 | 1.33 | 0.21 | 1.35 | 19.83 | 2.15 | 1181 |
| Dual SR-SPKF | 0.27 | 1.33 | 1.05 | 1.26 | 19.83 | 2.74 | 1138 |
| Joint SPKF generalize | 0.93 | 4.40 | 0.21 | 2.11 | 22.94 | 8.02 | 2839 |
| Dual SPKF generalize | 1.18 | 5.43 | 10.75 | 2.02 | 22.87 | 12.30 | 3950 |
| Dual SR-SPKF generalize | 0.90 | 4.35 | 0.23 | 2.01 | 22.87 | 10.33 | 3369 |

cycles is plotted in Fig. 2(b). We see that SOC increases by about 5% during each UDDS cycle, but is brought down slightly less than 10% during each discharge between cycles. The entire operating range for these cells (10–90% SOC, delineated by dashed lines in the figure) is excited during the cell test.

### 8.2. Fitting data to the enhanced self-correcting model

Data collected from the first cell was used to identify initial parameters for the ESC cell model. The goal is to have the cell model output resemble the cell terminal voltage under load as closely as possible, at all times, when the cell model input is equal to the cell current. Model fit was judged by comparing root-mean-squared (RMS) estimation error (estimation error equals cell voltage minus model voltage) over the portions of the cell tests where SOC was between 5 and 95%. Model error outside that SOC range was not considered as the HEV pack operation design limits are 10–90% SOC. Details for how the open-circuit-voltage curve was generated and how the model parameters were fit are described in Ref. [23]. In particular, the open-circuit-voltage as a function of state-of-charge for these cells is plotted in Fig. 3. Values were fit to the other ESC-model parameters, with very close agreement between the cell model voltage prediction and the cell true voltage. In this work, the model employs two low-pass filter states ($n_f = 2$), a nominal capacity of 5.0 Ah, and an inter-sample interval of $\Delta T = 1$ s.[4]

[4] We note here that prior work with the third-generation cells used $n_f = 4$, and that this work using fourth-generation cells uses $n_f = 2$. There are competing objectives here: to make the model as accurate as possible, and to make the filter as computationally efficient as possible. The minimum number of filter states that can result in a zero dc-gain is two, and we find that SPKF is enough superior

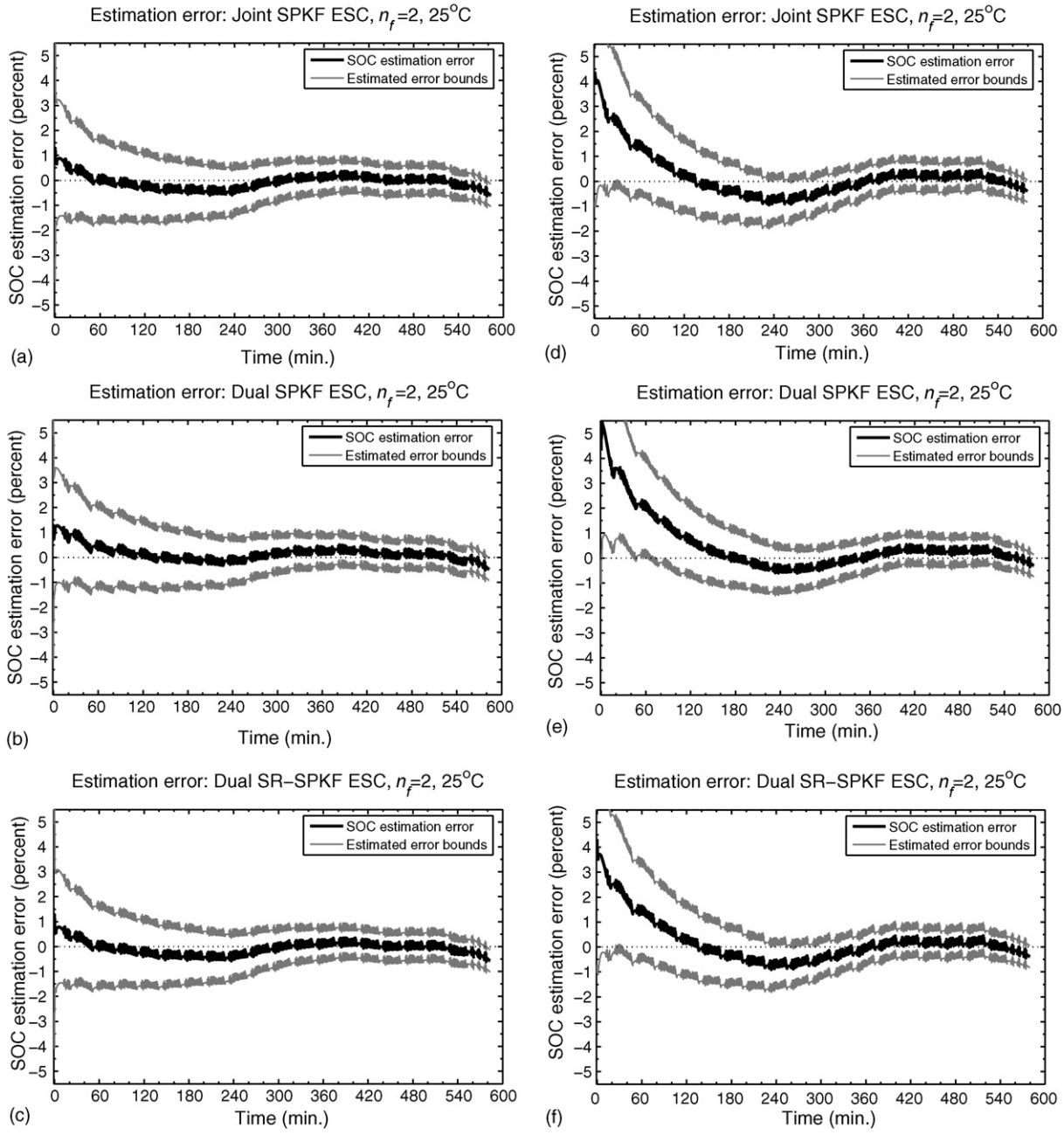


Fig. 7. SOC estimation error for joint SPKF vs. dual SPKF vs. dual SR-SPKF with correct initialization: (a–c) use training data and (d–f) use generalization data.

Some modeling results are shown in Fig. 4. In frame (a), the ESC cell voltage prediction is compared with test data. Very close agreement is observed especially during the dynamic part of the test. In frame (b), an instantaneous voltage-based SOC estimate, calculated as $\hat{z}_k = \mathrm{OCV}^{-1}(y_k + Ri_k)$ (cf. Section 6.1), is compared with true SOC.[5] This comparison is made to show: (1) that such an estimate is too noisy to be used as an estimate of SOC by itself (and therefore we need to use more advanced methods, such as SPKF), but (2) it yields *average* behavior in dynamic tests that *is* accurate, and so is useful to ensure convergence of the parameters in a dual or joint application. Note that there is no point in low-pass filtering this result, as the delay of such a filter would make the estimate useless. Rather, the tuning

to EKF that it gives acceptable estimates with a poorer model. Therefore, we have chosen to use $n_f = 2$ here.

[5] We calculate "true" SOC using Coulomb counting from the Arbin test equipment. Bias and noise in the Arbin current sensor will cause this value to drift from the ideal value, but over the relatively short duration of the tests and given the high accuracy of the Arbin sensors we feel that this is a reasonable approximation to true SOC. The cost of using such high-accuracy sensors in a production BMS is prohibitive, which is why we instead investigate using intelligent algorithms and less-expensive sensors.

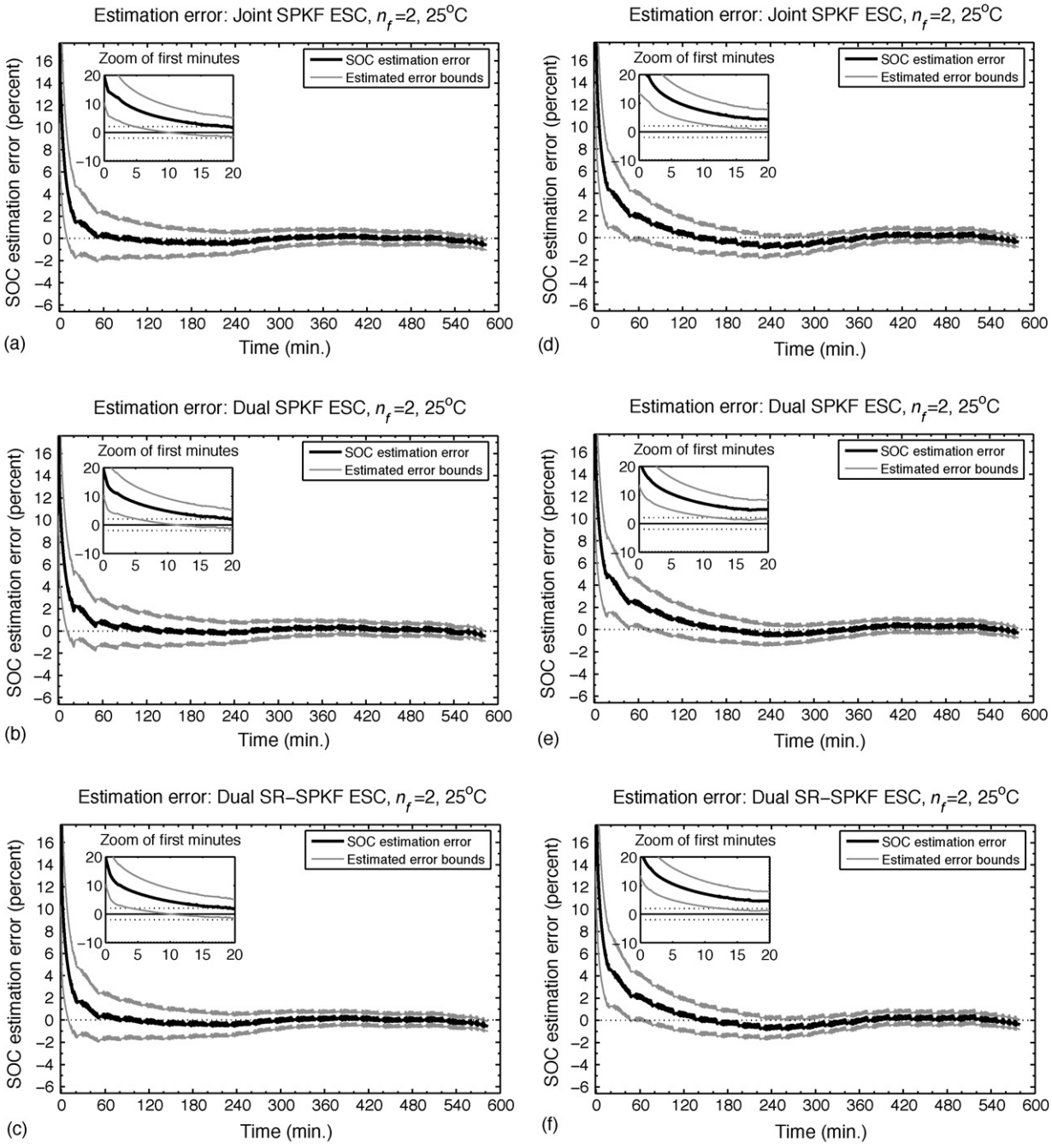


Fig. 8. SOC estimation error for joint SPKF vs. dual SPKF vs. dual SR-SPKF with incorrect initialization: (a–c) use training data and (d–f) use generalization data.

parameters of the SPKF are used to adjust the "belief" that the filter has in the accuracy of $\hat{z}_k$ when adapting its own internal state.

### 8.3. Examples of SPKF and SR-SPKF

In the companion to this paper, we showed SPKF SOC estimation results compared against EKF results for th GEN3 cell. The conclusion was that the SPKF was better in all cases, so we do not include EKF results here. Rather, we begin by presenting SPKF results for the G4 cell. For brevity, we include only room-temperature results here.

Fig. 5(a) shows SOC estimation error and predicted error bounds for the training cell, where the SPKF state was correctly initialized.[6] In particular, the SOC state of the SPKF was initialized to the correct value of 100%. Frame (b) shows the same result using SR-SPKF. The results are nearly identical, as expected. RMS SOC estimation error was 0.3% and maximum

[6] The error bounds are computed as plus/minus three times the square-root of the covariance matrix diagonal element corresponding to the SOC state. These are referred to as "three-sigma" bounds, and if all densities are Gaussian, the bounds should correctly encompass the true value of SOC 99.7% of the time. Section 8.5 discusses the validity of this assumption in more detail.

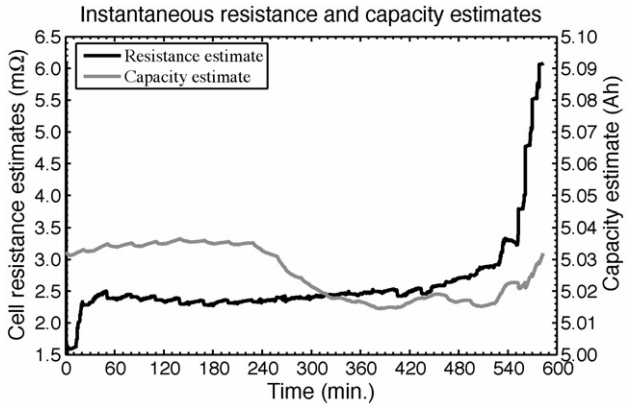


Fig. 9. Instantaneous resistance and capacity estimates by joint SPKF (dual filter results similar).

SOC error was 1.51%. The estimation error bounds correctly included the true SOC value for all but about 1% of the test run. The bounds were incorrect only at the very end of the test where true SOC was below 10% and out of the expected operating range of the cell—thus the bounds correctly included the true SOC for all SOCs of importance.

Frames (c) and (d) show the same results when the SPKF was applied to data from the second cell—the one not used to train the model parameters. We see slightly poorer performance, as would be expected. Note that post-test analysis indicates that this cell may have been overcharged to about 104% before the test began – based on OCV data – which explains why the SPKF gave an initial 4% error when initialized to 100%. The filter was still able to converge to the true SOC.

Fig. 6 shows results parallel to those of Fig. 5 for the case where the filter SOC state was intentionally initialized with an incorrect value (80% rather than 100%) to demonstrate the convergence properties of SPKF. We see relatively fast convergence to within 5% SOC error, about 20-min convergence to 2% error for the training cell and about 45-min convergence to 2% error for the generalization cell. Convergence time may be adjusted by varying $\Sigma_w$ and $\Sigma_v$, and could have been made faster than the results shown. The tradeoff would have been poorer SOC bounds estimate.

Table 8 summarizes the results of these tests. The "bounds error" column shows the percentage of time the SOC estimation bounds did not encompass the true SOC. Note that this was always at very low and high SOC – out of the operating range of the cell – so is not of concern here. The "time to converge" column for the generalization tests show how many iterations (seconds) were required for the SOC error to converge to less than 2%.

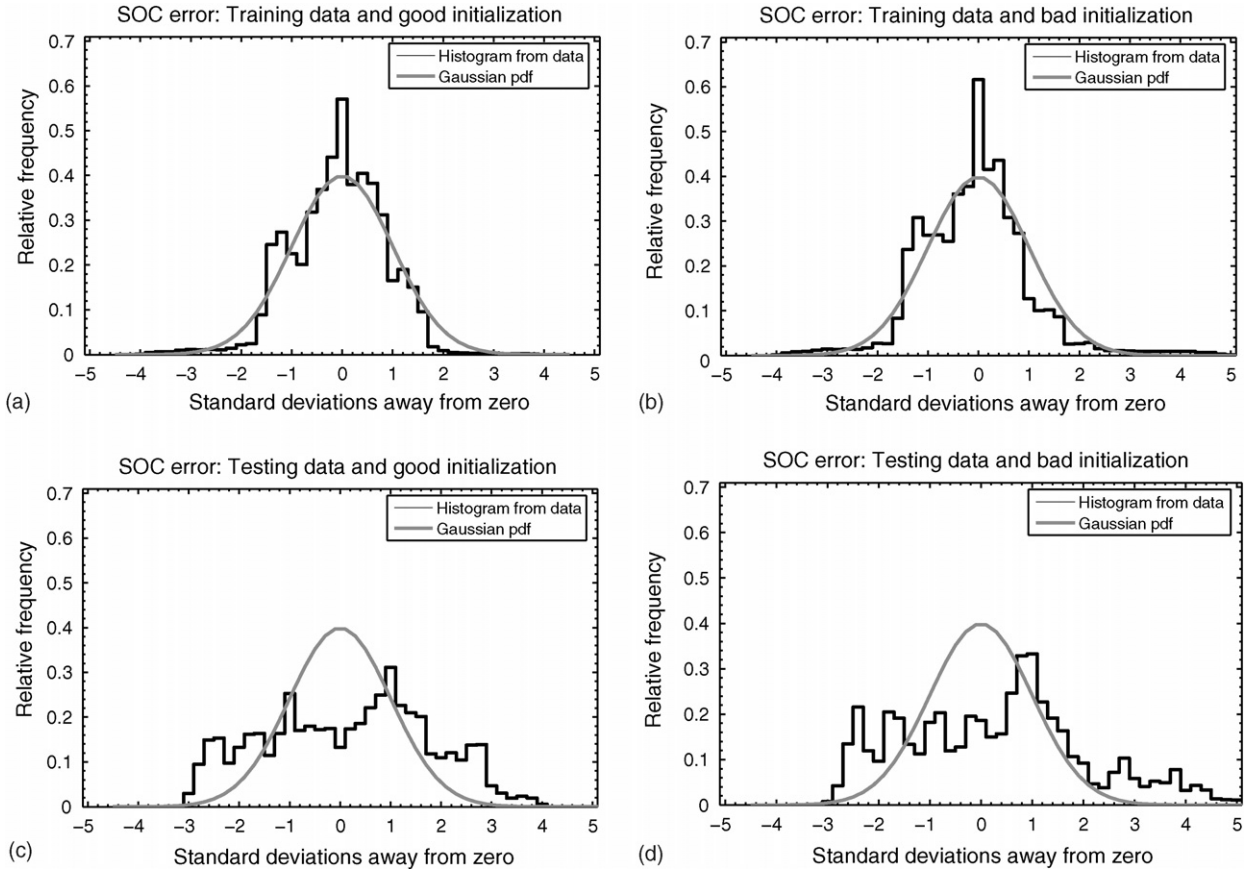


Fig. 10. Empirical probability–density functions of SOC estimation error normalized by standard deviation: (a) training data and correct initialization, (b) training data and incorrect initialization, (c) generalization data and correct initialization, and (d) generalization data and incorrect initialization. The standard-normal distribution is overlaid in each case.

### 8.4. Examples of joint and dual ID

We conducted tests to determine the SOC estimation performance of the joint/dual SPKF filters as well. Figs. 7 and 8 show results parallel to those of Figs. 5 and 6.

In Fig. 7, results are shown when the filter state was correctly initialized; in Fig. 8, the SOC state was intentionally initialized to an incorrect value (80% rather than 100%) to show convergence properties. In frames (a)–(c), results using the cell from which parameters were originally trained are shown; in frames (d)–(f), generalization results are shown. Note that the joint/dual filters adapt their parameters, so that over time they learn the dynamics of any cell they are modeling, and SOC estimation error improves.

Table 9 summarizes results for these tests. We note that they are very similar to the SOC results for the plain SPKF filter. Over a (much) longer test we would expect some improvement in the joint/dual results as the filters adapt to changing cell dynamics.

In particular, Fig. 9 shows the 1-Hz resistance and capacity parameters adapting over the course of the test. The change in resistance is valid—these cells have much higher resistance at low SOC than at moderate SOC. The variation in the capacity estimate between 5.02 and 5.04 Ah is a little less certain. It is likely that the capacity state is exhibiting some short-term variation to mask some other modeling error. We have observed that the long-term behavior of the capacity state is stable, however, and is able to track changes in the true cell capacity. The adaptation may be slowed down to minimize the masking effect shown here by lowering the fictitious noise covariance in $\Sigma_r$ corresponding to the capacity state.

### 8.5. Analysis of SOC estimation error

Before concluding this paper, we present some results analyzing the SOC estimation error. All of the Kalman filter variants make the assumption that the noises affecting the system of concern and the measurements made are Gaussian (normal) random variables, and that the state estimation errors are also Gaussian. In Fig. 10, we plot summary histograms of SOC estimation error accumulated through the various test runs using SPKF, SR-SPKF, joint SPKF, dual SPKF, and dual SR-SPKF. In every case, each instantaneous SOC estimation error was divided by its corresponding one-sigma error bound before computing the histogram, and then the histogram was normalized to have unit area to form an empirical probability density function (PDF) of the SOC estimation error. If all KF assumptions were being met, this distribution should match a unit-variance standard normal PDF. We see that this assumption is fairly close to being met when using the training data, but less well met when using the generalization data. In particular, we note that three-sigma error bounds (as reported elsewhere in this paper) are theoretically accurate 99.7% of the time, but we see from the histograms that four-sigma bounds would give a little more safety in an implementation. Also, since we now see that the KF assumptions are not being met, doubt is cast on how much better we might be able to do using KF techniques. Most likely, one would need to use a particle filter – which does not assume Gaussian RVs – to do better, but the added computational complexity might not be tolerable in a commercial application.

## 9. Conclusions

This paper concludes a two-part series discussing the application of sigma-point Kalman filters to battery management algorithms. In the first paper, we introduced the general probabilistic inference solution to optimal estimation, and derived the KF, EKF, and SPKF from this solution using different sets of assumptions. The SPKF was shown to be theoretically more precise than EKF; testing with real cell data supported this analysis.

This paper showed how SPKF could be very closely approximated by SR-SPKF, which gives speed advantages in the case of a linear state equation, such as when estimating parameters rather than states. Simultaneous state and parameter estimation was also introduced via joint and dual SPKF methods. Results were presented for a prototype LiPB HEV cell that demonstrate that these methods work very well. For example, typical SOC estimation errors of less than 1% are reported, both using training and testing data, with near perfect error bounds—these were in fact perfect over the 10–90% expected SOC operating range of the cell. Parameters such as cell resistance and SOC can also be simultaneously estimated.

The various KF methods are derived assuming that the probability density functions of sensor and process noises are Gaussian. We have seen by example that this assumption is not strictly adhered to in this application (nor can it be exactly). This limits the ability of the KF methods to estimate states and parameters, and indicates that different approaches might need to be taken if even greater accuracy is needed. However, the SPKF does very well even though this assumption is not met exactly.

For further reading, we have shown elsewhere that the state and parameters may be used to very precisely estimate dynamic available power, and to compute which cells must be equalized [6,24].

## Acknowledgements

This work was supported in part by Compact Power Inc. (CPI). The use of company facilities, and many enlightening discussions with Drs. Mohamed Alamgir, Bruce Johnson, Dan Rivers, and others are gratefully acknowledged.

I would also like to thank the reviewers for their helpful comments, which led to improvements in the clarity of this work with respect to what I originally wrote.

## References

[1] G. Plett, LiPB dynamic cell models for Kalman-filter SOC estimation, in: CD-ROM Proceedings of the 19th Electric Vehicle Symposium (EVS19), Busan, Korea, October 2002.

[2] G. Plett, Kalman-filter SOC estimation for LiPB HEV cells, in: CD-ROM Proceedings of the 19th Electric Vehicle Symposium (EVS19), Busan, Korea, October 2002.

[3] G. Plett, Advances in EKF SOC estimation for LiPB HEV battery packs, in: CD-ROM Proceedings of the 20th Electric Vehicle Symposium (EVS20), Long Beach, CA, November 2003.
[4] G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 1: background, J. Power Sources 134 (2) (2004) 252–261.
[5] G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 2: modeling and identification, J. Power Sources 134 (2) (2004) 262–276.
[6] G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 3: parameter estimation, J. Power Sources 134 (2) (2004) 277–292.
[7] G. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Introduction and state estimation, Int. J. Power Sources., 10.1016/j.jpowsour.2006.06.003.
[8] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C: The Art of Scientific Computing, second ed., Cambridge University Press, 1992.
[9] G. Stewart, Matrix Algorithms, Basic Decompositions, vol. I, SIAM, 1998.
[10] S. Julier, J. Uhlmann, H. Durrant-Whyte, A new approach for filtering nonlinear systems, in: Proceedings of the American Control Conference, 1995, pp. 1628–1632.
[11] S. Julier, J. Uhlmann, A General Method for Approximating Nonlinear Transformations of Probability Distributions, Technical Report, RRG, Department of Engineering Science, Oxford University, November 1996.
[12] S. Julier, J. Uhlmann, A new extension of the Kalman filter to nonlinear systems, in: Proceedings of the 1997 SPIE AeroSense Symposium, SPIE, Orlando, FL, April 21–24, 1997.
[13] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proc. IEEE 92 (3) (2004) 401–422.
[14] E. Wan, R. van der Merwe, The unscented Kalman filter for nonlinear estimation, in: Proceedings of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, 2000.
[15] E. Wan, R. van der Merwe, The unscented Kalman filter, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley Inter-Science, New York, 2001, pp. 221–282 (Chapter 7).
[16] M. Nørgaard, N. Poulsen, O. Ravn, Advances in Derivative-free State Estimation for Nonlinear Systems, Technical Report IMM-REP-1998-15, Department of Mathematical Modeling, Technical University of Denmark, 28 Lyngby, Denmark, April 2000.
[17] M. Nørgaard, N. Poulsen, O. Ravn, New developments in state estimation for nonlinear systems, Automatica 36 (11) (2000) 1627–1638.
[18] R. van der Merwe, E. Wan, Sigma-point Kalman filters for probabilistic inference in dynamic state–space models, in: Proceedings of the Workshop on Advances in Machine Learning, Montreal, June 2003, Available at http://choosh.ece.ogi.edu/spkf/spkf_files/WAML2003.pdf (Accessed 20 May 2004).
[19] R. van der Merwe, E. Wan, The square-root unscented Kalman filter for state and parameter estimation, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Salt Lake City, Utah, 2001.
[20] R. van der Merwe, E. Wan, Efficient derivative-free Kalman filters for online learning, in: Proceedings of the European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2001.
[21] E. Wan, A. Nelson, Dual extended Kalman filter methods, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley Inter-Science, New York, 2001, pp. 123–174.
[22] S. Kim, J. Yu, J. Namgoong, J. Kim, M. Kim, Progress in Li-ion polymer battery and pack system of LG Chem for transportation applications, in: CD-ROM Proceedings of the 20th Electric Vehicle Symposium (EVS20), Long Beach, CA, 2003.
[23] G. Plett, Results of temperature-dependent LiPB cell modeling for HEV SOC estimation, in: CD-ROM Proceedings of the 21st Electric Vehicle Symposium (EVS21), Monaco, April 2005.
[24] G. Plett, High-performance battery-pack power estimation using a dynamic cell model, IEEE Trans. Vehicular Technol. 53 (5) (2004) 1586–1593.