Available online at www.sciencedirect.com

ScienceDirect

Journal of Power Sources 161 (2006) 1356–1368

JOURNAL OF POWER SOURCES

www.elsevier.com/locate/jpowsour

Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs
Part 1: Introduction and state 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, USA

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 (SOC), nominal capacity, resistance, and others. Since the dynamics of electrochemical cells are not linear, we used a non-linear extension to the original KF called the extended Kalman filter (EKF).

We were able to achieve very good estimates of SOC and other states and parameters using EKF. However, some applications e.g., that of the battery-management-system (BMS) of a hybrid-electric-vehicle (HEV) can require even more accurate estimates than these. To see how to improve on EKF, we must examine the mathematical foundation of that algorithm in more detail than we presented in the prior work to discover the assumptions that are made in its derivation. Since these suppositions are not met exactly in BMS application, we explore an alternative non-linear Kalman filtering techniques 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.

The SPKF method as applied to BMS algorithms is presented here in a series of two papers. This first paper is devoted primarily to deriving the EKF and SPKF algorithms using the framework of sequential probabilistic inference. This is done to show that the two algorithms, which at first may look quite different, are actually very similar in most respects; also, we discover why we might expect the SPKF to outperform EKF in non-linear estimation applications. Results are presented for a battery pack based on a third-generation prototype LiPB cell, and compared with prior results using EKF. As expected, SPKF outperforms EKF, both in its estimate of SOC and in its estimate of the error bounds thereof. The second paper presents some more advanced algorithms for simultaneous state and parameter estimation, and gives results for a fourth-generation prototype LiPB cell.

Keywords: Battery management system; Hybrid electric vehicle; Extended Kalman filter; Sigma-Point Kalman filter; State of charge; State of health

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) for hybrid-electric vehicle (HEV) application. 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 an increase in cell resistance (i.e., 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, and we also expect them to work well for other battery chemistries.

A hybrid-electric vehicle is one with both an internal-combustion engine and an electric motor. Both may be coupled directly to the power train—resulting in a "parallel hybrid"

* Tel. +1 719 262 3468; fax: +1 719 262 3589.
E-mail addresses: glp@eas.uccs.edu, gplett@compactpower.com.

[1] He is also consultant to Compact Power Inc., 707 County Line Rd., P.O. Box 509, Palmer Lake, CO 80133, USA. Tel. +1 719 488 1600 134; fax: +1 719 487 9485.

0378-7753/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2006.06.003

configuration—where the motor provides boost energy to supplement the engine, and acts as a generator when coasting, braking, or when the engine can supply extra power to charge the battery pack. Alternately, the engine may be used exclusively to drive a generator that charges the battery pack; the motor is then coupled directly to the power train—resulting in a "series hybrid" configuration. The series configuration promises greater potential efficiency, at the cost of a larger required battery pack. At the time of the writing of this paper, the only HEVs marketed in the US are parallel hybrid systems and require a battery pack of fairly modest size. Even so, and because demanding requirements on a pack of limited capacity result in cell electrochemistries that are often far from equilibrium, advanced methods must be used to estimate SOC, SOH, and instantaneous power in order to safely, efficiently and aggressively exploit the pack capabilities.

The methods we use to estimate these numeric quantities are based on model-based-estimation theory. Two separate components are needed to implement model-based estimation: (1) a model of cell dynamics, and (2) an algorithm that uses that model. The general approach is to use measured model inputs to predict measurable model outputs using present values of model state and parameters. Any difference between the model output and the measured system output can be attributed to errors in the measurements, states, parameters, or the model framework itself. An algorithm uses this difference signal to intelligently update its estimate of the model state and/or parameters. In this work, the *model* of cell dynamics is the "enhanced self-correcting" (ESC) model, introduced elsewhere [3,5] and briefly reviewed in Section 7.2 here. However, the model is not the focus of this work, but rather the *algorithms* that use that model, which comprise the family of Kalman filters.[2]

Kalman filters are an intelligent—and sometimes optimal—means for estimating the present value of the time-varying "state" of a dynamic system. By modeling our battery system to include the wanted unknown quantities in its state description, we may use a Kalman filter to estimate their values. An additional benefit of the Kalman filter is that it automatically provides dynamic error-bounds on these estimates as well. We exploit this fact to give aggressive performance from our battery pack, without fear of causing damage by overcharge or overdischarge.

We have previously reported work using *extended Kalman filters* (EKF) to solve the BMS algorithm requirements [1–6]. We have since explored a different form of Kalman filtering called *sigma-point Kalman filters* (SPKF), and have found them to have several important advantages to be outlined herein.

We present the base-line SPKF and some variants, along with testing results and analysis, in a two-part series of papers. This first paper primarily derives the equations that govern optimal Gaussian sequential probabilistic inference, and then shows how these equations apply to standard Kalman filtering, extended Kalman filtering, and sigma-point Kalman filtering. The SPFK algorithm is exercised on the same data as presented in [5] to compare with EKF on a third-generation prototype LiPB HEV cell. In all cases, the SPKF outperforms the EKF, both in its ability to estimate SOC, and in its estimates of the error bounds thereof.

The second paper in this series [7] explores some more advanced algorithms: square-root sigma-point Kalman filtering, parameter estimation, and simultaneous state and parameter estimation using the joint and dual methods. Since it has already been established in this first paper that SPKF outperforms EKF, we no longer use data from the (now obsolete) third-generation cells, but present testing data from fourth-generation LiPB high-power HEV cells.

While there is necessarily some redundancy in presenting this work in a two-paper series, we felt that this approach had certain advantages to a single monolithic manuscript. First, each part is self-contained, with the first paper primarily serving to show the theoretical benefits of SPKF and to demonstrate these with examples and the second paper primarily serving to show how the SPKF might be used in actual application; secondly, the results of the first paper are found for the third-generation cell, and are directly comparable to previously reported results, while the results of the second paper are found for a fourth-generation cell only—since we have already established that the SPKF outperforms EKF, there is no need to rehash old data with the SPKF and burden this work with an excess of figures and tables—rather, we can lay the foundation for future work that will be reported for these cells.

2. 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).

For some applications, we desire to estimate the state- or parameter-vector quantities in real time, as the system operates. *Probabilistic inference* and *optimal estimation theory* are names given to the field of study that concerns estimating these hidden variables in an optimal and consistent fashion, given noisy or incomplete observations. For example, in this paper we will be concerned with estimating the state of an electrochemical cell and the parameters for a mathematical model describing the cell dynamics. SOC is one element of the state vector of particular concern, and factors such as resistance increase and power fade are elements from the parameter vector of concern to help estimate SOH. Observations are available to us at sampling points and include: cell current i_k, cell terminal voltage y_k, and cell temperature T_k, where the subscript k indicates that the measurement is taken at the kth sampling point.

[2] These algorithms may be applied to any cell model in the correct format, and hence may be used to estimate the state and parameters of cells with differing electrochemistries and physical configurations.

In the following, we will 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". The state equation captures the evolving system dynamics. System stability, dynamic controllability and sensitivity to disturbance may all be determined from this equation. 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 non-linear) state transition function and $g(x_k, u_k, v_k, k)$ is a (possibly non-linear) measurement function.

With this model structure, the evolution of unobserved states and observed measurements may be visualized as shown in Fig. 1. The conditional probability $p(x_k|x_{k-1})$ indicates that the new state is a function of not only the deterministic input u_{k-1}, but also the stochastic input w_{k-1}, so that the unobserved state variables do not form a deterministic sequence. Similarly, the conditional probability density function $p(y_k|x_k)$ indicates that the observed output is not a deterministic function of the state, due to the stochastic input v_k.

The goal of probabilistic inference is to create an estimate of the system state given all observations $\mathbb{Y}_k = \{y_0, y_1, \ldots, y_k\}$. A frequently used estimator is the conditional mean

$$\hat{x}_k = \mathbb{E}[x_k|\mathbb{Y}_k] = \int_{R_{x_k}} x_k p(x_k|\mathbb{Y}_k)\,\mathrm{d}x_k,$$

where R_{x_k} is the set comprising the range of possible x_k, and $\mathbb{E}[\cdot]$ is the statistical expectation operator. The optimal solution to this problem computes the posterior probability density $p(x_k|\mathbb{Y}_k)$ recursively with two steps per iteration [8]. The first step computes probabilities for predicting x_k given all past observations

$$p(x_k|\mathbb{Y}_{k-1}) = \int_{R_{x_{k-1}}} p(x_k|x_{k-1}) p(x_{k-1}|\mathbb{Y}_{k-1})\,\mathrm{d}x_{k-1}$$

and the second step updates the prediction via

$$p(x_k|\mathbb{Y}_k) = \frac{p(y_k|x_k)p(x_k|\mathbb{Y}_{k-1})}{p(y_k|\mathbb{Y}_{k-1})}$$

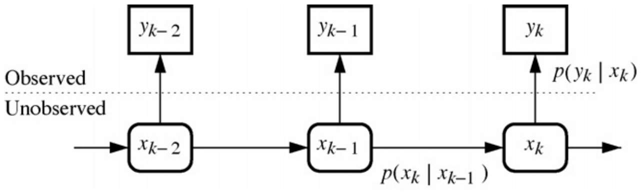

Fig. 1. Sequential probabilistic inference.

which is a simple application of Bayes' rule and the assumption that the present observation y_k is conditionally independent of previous measurements given the present state x_k. The relevant probabilities may be computed as

$$p(y_k|\mathbb{Y}_{k-1}) = \int_{R_{x_k}} p(y_k|x_k)p(x_k|\mathbb{Y}_{k-1})\,\mathrm{d}x_k$$

$$p(x_k|x_{k-1}) = \sum_{\{w: x_k = f(x_{k-1}, u_{k-1}, w, k-1)\}} p(w)$$

$$p(y_k|x_k) = \sum_{\{v: y_k = h(x_k, u_k, v, k)\}} p(v).$$

Although this is the optimal solution, finding a formula solving the multi-dimensional integrals in a closed-form is intractable for most real-world systems. For applications that justify the computational expense, the integrals may be closely approximated using Monte Carlo methods such as particle filters [9–12,8]. However, for battery packs containing tens or hundreds of cells, and hence perhaps hundreds or thousands of states, the present economics do not make this option feasible.

A simplified solution to these equations may be obtained if we are willing to make the assumption that all probability densities are Gaussian—this is the basis of the original Kalman filter, the extended Kalman filter, and the sigma-point Kalman filters to be discussed. Then, rather than having to propagate the entire density function through time, we need only to evaluate the conditional mean and covariance of the state (and parameters, perhaps) once each sampling interval. It can be shown (cf. Appendix A) that the recursion becomes:

$$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k) = \hat{x}_k^- + L_k\tilde{y}_k \tag{3}$$

$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}}, \tag{4}$$

where the superscript T is the matrix/vector transpose operator, and

$$\hat{x}_k^+ = \mathbb{E}[x_k|\mathbb{Y}_k] \tag{5}$$

$$\hat{x}_k^- = \mathbb{E}[x_k|\mathbb{Y}_{k-1}] \tag{6}$$

$$\hat{y}_k = \mathbb{E}[y_k|\mathbb{Y}_{k-1}] \tag{7}$$

$$\Sigma_{\tilde{x},k}^- = \mathbb{E}[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^{\mathrm{T}}] = \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^{\mathrm{T}}] \tag{8}$$

$$\Sigma_{\tilde{x},k}{}^+ = \mathbb{E}[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^{\mathrm{T}}] = \mathbb{E}[(\tilde{x}_k^+)(\tilde{x}_k^+)^{\mathrm{T}}] \tag{9}$$

$$\Sigma_{\tilde{y},k} = \mathbb{E}[(y_k - \hat{y}_k)(y_k - \hat{y}_k)^{\mathrm{T}}] = \mathbb{E}[(\tilde{y}_k)(\tilde{y}_k)^{\mathrm{T}}] \tag{10}$$

$$L_k = \mathbb{E}[(x_k - \hat{x}_k^-)(y_k - \hat{y}_k)^{\mathrm{T}}]\Sigma_{\tilde{y},k}^{-1} = \Sigma_{\tilde{x}\tilde{y},k}^-\Sigma_{\tilde{y},k}^{-1}. \tag{11}$$

While this is a linear recursion, we have not directly assumed that the system model is linear. 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. 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 Σ_x to indicate the same quantity as Σ_{xx}.

In the following sections we will apply Eqs. (3)–(11), and approximations thereof, with different sets of assumptions to derive the Kalman filter, the extended Kalman filter, and sigma-point Kalman filters. All members of this family of filters will comply with a structured sequence of six steps per iteration, as outlined here.

General step 1: State estimate time update. Each measurement interval, the first step is to compute an updated prediction of the present value of x_k, based on *a priori* information and the system model. This is done using Eqs. (1) and (6) as

$$\hat{x}_k^- = \mathbb{E}[x_k|\mathbb{Y}_{k-1}] = \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)|\mathbb{Y}_{k-1}].$$

General step 2: Error covariance time update. The second step is to determine the predicted state-estimate error covariance matrix $\Sigma_{\tilde{x},k}^-$ based on *a priori* information and the system model.

We compute $\Sigma_{\tilde{x},k}^- = \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^{\mathrm{T}}]$ using Eq. (8), knowing that $\tilde{x}_k^- = x_k - \hat{x}_k^-$.

General step 3: Estimate system output y_k. The third step is to estimate the system's output using present *a priori* information and Eqs. (2) and (7)

$$\hat{y}_k = \mathbb{E}[y_k|\mathbb{Y}_{k-1}] = \mathbb{E}[h(x_k, u_k, v_k, k)|\mathbb{Y}_{k-1}.]$$

General step 4: Estimator gain matrix L_k. The fourth step is to compute the estimator gain matrix L_k by evaluating $L_k = \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}$.

General step 5: State estimate measurement update. The fifth step is to compute the *a posteriori* state estimate by updating the *a priori* estimate using the estimator gain and the output prediction error $y_k - \hat{y}_k$ using (3). There is no variation in this step in the different Kalman filter methods; implementational differences between Kalman approaches do manifest themselves in all other steps, however.

General step 6: Error covariance measurement update. The final step computes the *a posteriori* error covariance matrix using Eq. (4). The estimator output comprises the state estimate $\hat{x}_k^+$ and error covariance estimate $\Sigma_{\tilde{x},k}^+$. The estimator then waits until the next sample interval, updates *k*, and proceeds to step 1.

The general sequential probabilistic inference solution is summarized in Table 1. The following sections describe specific applications of this general framework. First, we derive the Kalman filter, and then the extended Kalman filter. This is done to give confidence that the sequential probabilistic inference approach does indeed result in the equations for these standard algorithms, to show the assumptions made when performing the derivations, and to give insight into how we might improve estimation for non-linear systems. The reader may wish to skip ahead to Section 5 on the first reading of this paper, and then return to Sections 3 and 4 for a greater depth of understanding.

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 Σ_w and Σ_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: $\hat{y}_k = \mathbb{E}[h(x_k, u_k, v_k, k)|\mathbb{Y}_{k-1}]$

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

3. Optimal application to linear systems: the Kalman filter

In this section, we take the general results from Section 2 and apply them to the specific case where the system dynamics are linear. Linear systems have the desirable property that all probability distributions do in fact remain Gaussian if the stochastic inputs are Gaussian, so the assumptions made in deriving the filter steps hold exactly. If the system dynamics are linear, then the Kalman filter (first presented in [13,14]) is the optimal minimum-mean-squared-error and maximum-likelihood estimator. The format of this section is to first introduce the form of a linear state-space model, and then to apply the six steps from Section 2 to this form to derive the linear Kalman filter equations.

The linear Kalman filter assumes that the system being modeled can be represented in the "state-space" form

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}$$

$$y_k = C_k x_k + D_k u_k + v_k.$$

The matrices $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{n\times p}$, $C_k \in \mathbb{R}^{m\times n}$ and $D_k \in \mathbb{R}^{m\times p}$ describe the dynamics of the system, and are possibly time varying. Also, both w_k and v_k are assumed to be mutually uncorrelated white Gaussian random processes, with zero mean and covariance matrices with known value:

$$\mathbb{E}[w_n w_k^{\mathrm{T}}] = \begin{cases} \Sigma_w, & n = k; \\ 0, & n \neq k. \end{cases} \qquad \mathbb{E}[v_n v_k^{\mathrm{T}}] = \begin{cases} \Sigma_v, & n = k; \\ 0, & n \neq k. \end{cases}$$

The assumptions on the noise processes w_k and v_k and on the linearity of system dynamics are rarely (never) met in practice, but the consensus of the literature and practice is that the method still works very well.

KF step 1: State estimate time update. Here, we compute

$$\begin{aligned}\hat{x}_k^- &= \mathbb{E}[A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}|\mathbb{Y}_{k-1}] \\ &= A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1},\end{aligned}$$

by the linearity of expectation, noting that w_{k-1} is zero-mean.

KF step 2: Error covariance time update. First, we note that the estimation error may be found by comparing $x_k =$

Table 2
Summary of the linear Kalman filter from reference [15]

Linear state-space model
$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}$
$y_k = C_k x_k + D_k u_k + v_k$,
where w_k and v_k are independent, zero-mean, Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively

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: $\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1}$
Error covariance time update: $\Sigma_{\tilde{x},k}^- = A_{k-1}\Sigma_{\tilde{x},k-1}^+ A_{k-1}^{\mathrm{T}} + \Sigma_w$
Output estimate: $\hat{y}_k = C_k\hat{x}_k^- + D_k u_k$
Estimator gain matrix: $L_k = \Sigma_{\tilde{x},k}^- C_k^{\mathrm{T}}[C_k\Sigma_{\tilde{x},k}^- C_k^{\mathrm{T}} + \Sigma_v]^{-1}$
State estimate measurement update: $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$
Error covariance measurement update: $\Sigma_{\tilde{x},k}^+ = (I - L_k C_k)\Sigma_{\tilde{x},k}^-$

$A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}$ with $\hat{x}_k^-$ as computed in step 1. We find that $\tilde{x}_k^- = A_{k-1}\tilde{x}_{k-1}^+ + w_{k-1}$. Therefore,

$$\begin{aligned}\Sigma_{\tilde{x}_k}^- &= \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^{\mathrm{T}}]\\ &= \mathbb{E}[(A_{k-1}\tilde{x}_{k-1}^+ + w_{k-1})(A_{k-1}\tilde{x}_{k-1}^+ + w_{k-1})^{\mathrm{T}}]\\ &= A_{k-1}\Sigma_{\tilde{x},k-1}^+ A_{k-1}^{\mathrm{T}} + \Sigma_w.\end{aligned}$$

The cross terms drop out of the final result since the white state noise w_{k-1} is not correlated with the state at time $k - 1$.

KF step 3: Estimate system output. We estimate the system output as

$$\hat{y}_k = \mathbb{E}[C_k x_k + D_k u_k + v_k|\mathbb{Y}_{k-1}] = C_k\hat{x}_k^- + D_k u_k,$$

since v_k is zero-mean.

KF step 4: Estimator (Kalman) gain matrix. To compute L_k, we first need to compute several covariance matrices. Since we know that $y_k = C_k x_k + D_k u_k + v_k$, it follows that $\tilde{y}_k = C_k\tilde{x}_k^- + v_k$ and

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

Again, the cross terms are zero since v_k is uncorrelated with the estimate $\tilde{x}_k^-$. Similarly,

$$\mathbb{E}[\tilde{x}_k^-\tilde{y}_k^{\mathrm{T}}] = \mathbb{E}[\tilde{x}_k^-(C_k\tilde{x}_k^- + v_k)^{\mathrm{T}}] = \Sigma_{\tilde{x},k}^- C_k^{\mathrm{T}}.$$

Combining,

$$L_k = \Sigma_{\tilde{x},k}^- C_k^{\mathrm{T}}[C_k\Sigma_{\tilde{x},k}^- C_k^{\mathrm{T}} + \Sigma_v]^{-1}.$$

KF step 6: Error covariance measurement update. Finally, we update the error covariance matrix.

$$\begin{aligned}\Sigma_{\tilde{x},k}^+ &= \Sigma_{\tilde{x},k}^- - L_k\mathbb{E}[\tilde{y}_k\tilde{y}_k^{\mathrm{T}}]L_k^{\mathrm{T}}\\ &= \Sigma_{\tilde{x},k}^- - L_k\mathbb{E}[\tilde{y}_k\tilde{y}_k^{\mathrm{T}}](\mathbb{E}[\tilde{y}_k\tilde{y}_k^{\mathrm{T}}])^{-\mathrm{T}}\mathbb{E}[\tilde{x}_k^-\tilde{y}_k^{\mathrm{T}}]^{\mathrm{T}}\\ &= \Sigma_{\tilde{x},k}^- - L_k C_k\Sigma_{\tilde{x},k}^- = (I - L_k C_k)\Sigma_{\tilde{x},k}^-.\end{aligned}$$

The final Kalman filter for linear systems is summarized in Table 2. These are the same equations as presented (without derivation) in reference [4].

4. An approximation for non-linear systems: the extended Kalman filter

The extended Kalman filter is one approach to generalize the KF results to non-linear systems. At each point in time, steps 2 and 4 linearize the non-linear state and output equations around their present operating point using Taylor-series expansions. Steps 1 and 3 approximate the *a priori* state estimate and output estimate using previously computed terms.

EKF step 1: State estimate time update. The state prediction step is approximated as

$$\begin{aligned}\hat{x}_k^- &= \mathbb{E}[f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)|\mathbb{Y}_{k-1}]\\ &\approx f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}, k-1),\end{aligned}$$

where $\bar{w}_{k-1} = \mathbb{E}[w_{k-1}]$. That is, we approximate the expected value of the new state by assuming that it is reasonable to simply propagate $\hat{x}_{k-1}^+$ and $\bar{w}_{k-1}$ through the state equation. Often, $\bar{w}_{k-1} = 0$.

EKF step 2: Error covariance time update. The covariance prediction step is accomplished by first making an approximation for $\tilde{x}_k^-$.

$$\begin{aligned}\tilde{x}_k^- = x_k - \hat{x}_k^- &= f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)\\ &\quad - f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}, k-1).\end{aligned}$$

The second term is expanded as a Taylor series around the prior operating "point" which is the set of values $\{x_{k-1}, u_{k-1}, w_{k-1}, k-1\}$

$$\begin{aligned}\hat{x}_k^- &\approx f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)\\ &\quad + \underbrace{\left.\frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)}{\partial x_{k-1}}\right|_{x_{k-1}=\hat{x}_{k-1}^+}}_{\text{Defined as } \hat{A}_{k-1}}(\hat{x}_{k-1}^+ - x_{k-1})\\ &\quad + \underbrace{\left.\frac{\partial f(x_{k-1}, u_{k-1}, w_{k-1}, k-1)}{\partial w_{k-1}}\right|_{w_{k-1}=\bar{w}_{k-1}}}_{\text{Defined as } \hat{B}_{k-1}}(\bar{w}_{k-1} - w_{k-1}).\end{aligned}$$

This gives $\tilde{x}_k^- \approx \hat{A}_{k-1}\tilde{x}_{k-1}^+ + \hat{B}_{k-1}\tilde{w}_{k-1}$. Substituting this to find the predicted covariance:

$$\Sigma_{\tilde{x},k}^- = \mathbb{E}[(\tilde{x}_k^-)(\tilde{x}_k^-)^{\mathrm{T}}] \approx \hat{A}_{k-1}\Sigma_{\tilde{x},k-1}^+\hat{A}_{k-1}^{\mathrm{T}} + \hat{B}_{k-1}\Sigma_{\tilde{w}}\hat{B}_{k-1}^{\mathrm{T}}.$$

EKF step 3: Output estimate. The system output is estimated to be

$$\hat{y}_k = \mathbb{E}[h(x_k, u_k, v_k, k)|\mathbb{Y}_{k-1}] \approx h(\hat{x}_k^-, u_k, \bar{v}_k, k),$$

where $\bar{v}_k = \mathbb{E}[v_k]$. That is, it is assumed that propagating $\hat{x}_k^-$ and the mean sensor noise is the best approximation to estimating the output.

EKF step 4: Estimator gain matrix. The output prediction error may then be approximated

$$\tilde{y}_k = y_k - \hat{y}_k = h(x_k, u_k, v_k, k) - h(\hat{x}_k^-, u_k, \bar{v}_k, k)$$

Table 3
Summary of the non-linear extended Kalman filter from reference [16]

Non-linear 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 $\Sigma_{\tilde{w}}$ and $\Sigma_{\tilde{v}}$, respectively

Definitions

$\hat{A}_k = \left.\frac{\partial f(x_k,u_k,w_k,k)}{\partial x_k}\right|_{x_k=\hat{x}_k^+}$ $\quad \hat{B}_k = \left.\frac{\partial f(x_k,u_k,w_k,k)}{\partial w_k}\right|_{w_k=\bar{w}_k}$
$\hat{C}_k = \left.\frac{\partial h(x_k,u_k,v_k,k)}{\partial x_k}\right|_{x_k=\hat{x}_k^-}$ $\quad \hat{D}_k = \left.\frac{\partial h(x_k,u_k,v_k,k)}{\partial v_k}\right|_{v_k=\bar{v}_k}$

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: $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1}, \bar{w}_{k-1}, k-1)$
Error covariance time update: $\Sigma_{\tilde{x},k}^- = \hat{A}_{k-1}\Sigma_{\tilde{x},k-1}^+\hat{A}_{k-1}^{\mathrm{T}} + \hat{B}_{k-1}\Sigma_{\tilde{w}}\hat{B}_{k-1}^{\mathrm{T}}$
Output estimate: $\hat{y}_k = h(\hat{x}_k^-, u_k, \bar{v}_k, k)$
Estimator gain matrix: $L_k = \Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}}[\hat{C}_k\Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}} + \hat{D}_k\Sigma_{\tilde{v}}\hat{D}_k^{\mathrm{T}}]^{-1}$
State estimate measurement update: $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$
Error covariance measurement update: $\Sigma_{\tilde{x},k}^+ = (I - L_k\hat{C}_k)\Sigma_{\tilde{x},k}^-$

using again a Taylor-series expansion on the second term.

$$\hat{y}_k \approx h(x_k, u_k, v_k, k) + \underbrace{\left.\frac{\partial h(x_k, u_k, v_k, k)}{\partial x_k}\right|_{x_k=\hat{x}_k^-}}_{\text{Defined as } \hat{C}_k}(\hat{x}_k^- - x_k)$$
$$+ \underbrace{\left.\frac{\partial h(x_k, u_k, v_k, k)}{\partial v_k}\right|_{v_k=\bar{v}_k}}_{\text{Defined as } \hat{D}_k}(\bar{v}_k - v_k).$$

From this, we can compute such necessary quantities as

$$\Sigma_{\tilde{y},k} \approx \hat{C}_k\Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}} + \hat{D}_k\Sigma_{\tilde{v}}\hat{D}_k^{\mathrm{T}},$$
$$\Sigma_{\tilde{x}\tilde{y},k}^- \approx \mathbb{E}[(\tilde{x}_k^-)(\hat{C}_k\tilde{x}_k^- + \hat{D}_k\tilde{v}_k)^{\mathrm{T}}] = \Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}}.$$

These terms may be combined to get the Kalman gain

$$L_k = \Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}}[\hat{C}_k\Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}} + \hat{D}_k\Sigma_{\tilde{v}}\hat{D}_k^{\mathrm{T}}]^{-1}.$$

EKF step 6: Error covariance measurement update.

Finally, the updated covariance is computed as

$$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}} = \Sigma_{\tilde{x},k}^- - L_k\Sigma_{\tilde{y},k}(\Sigma_{\tilde{y},k})^{-\mathrm{T}}(\Sigma_{\tilde{x}\tilde{y},k}^-)^{\mathrm{T}}$$
$$= \Sigma_{\tilde{x},k}^- - L_k\hat{C}_k\Sigma_{\tilde{x},k}^- = (I - L_k\hat{C}_k)\Sigma_{\tilde{x},k}^-.$$

This completes the derivation of the extended Kalman filter. It is summarized in Table 3. These are the same equations as presented (without derivation) in reference [4].

5. Problems with the EKF

The extended Kalman filter is probably the best known and most widely used non-linear Kalman filter. However, it has a number of flaws that can be improved upon fairly easily to improve state estimation. These flaws reside in two assumptions made in order to propagate a Gaussian random state vector x through some non-linear function: one assumption concerns the calculation of the output random variable mean, the other concerns the output random variable covariance.

First, we note that EKF step 1 attempts to determine an output random-variable mean from the state-transition function $f(\cdot)$ assuming that the input state is a Gaussian random variable. EKF step 3 makes a similar calculation for the output function $h(\cdot)$. The EKF makes the simplification

$$\mathbb{E}[\text{fn}(x)] \approx \text{fn}(\mathbb{E}[x]),$$

which is not true in general, and not necessarily even close to true (depending on "how non-linear" the function $\text{fn}(\cdot)$ is). The SPKF to be described will make an improved approximation to the means in steps 1 and 3.

Secondly, in EKF steps 2 and 4, a Taylor-series expansion is performed as part of a calculation designed to find the output-variable covariance. Non-linear terms are dropped from the expansion, resulting in a loss of accuracy. The SPKF uses a different method to compute covariances and will improve these estimates as well.

To give a simple one-dimensional example illustrating these two effects, consider Fig. 2. The non-linear function is drawn, and the input random-variable PDF is shown on the horizontal axis, with mean 1.05. The straight dotted line is the linearized approximation used by the EKF to find the output mean and covariance. The output approximate PDF estimated by EKF is drawn as a dotted line on the vertical axis, where a Gaussian PDF with the same mean and variance of the true data is shown as a solid PDF on the same axis. We notice significant differences between the means and covariances, indicating that EKF is not producing an accurate estimate of either one.

For a two-dimensional example, consider Fig. 3. Frame (a) shows a cloud of Gaussian-distributed random points used as input to this function, and frame (b) shows the transformed set of output points. The actual 95% confidence interval (indicative of a contour of the Gaussian PDF describing the output covariance and mean) is shown as a solid ellipse with the output mean

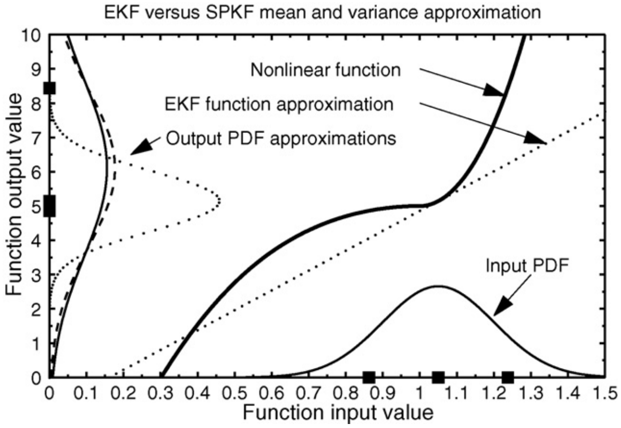

Fig. 2. EKF vs. SPKF mean and variance approximation. The solid line is the non-linear function; the dotted straight line is the linear approximation used by EKF in the neighborhood of input variable equal to 1.05. The solid squares on the axes are the input and output sigma points. The solid-line PDFs are the Gaussian PDFs matching the input and matching the output mean and variance; the dashed-line PDF is the output PDF predicted by the sigma-point method, and the dotted PDF is the output PDF predicted by the EKF method.

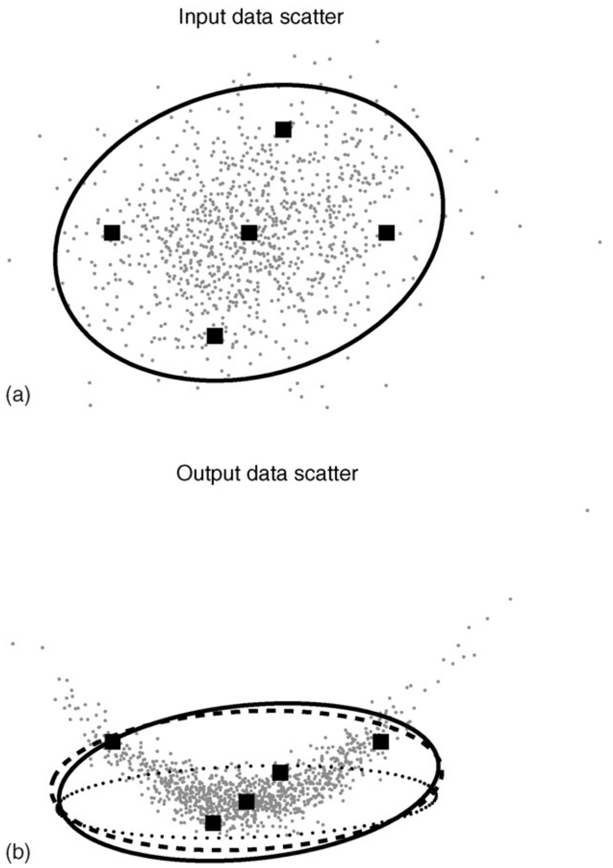

Fig. 3. Two-dimensional Gaussian random data (gray points in (a)) processed by a non-linear function to become the gray points in (b). The solid ellipses are the true 95% confidence bounds, the black squares are the input and output sigma points; the dashed ellipse is the 95% confidence bound produced by the sigma-point method, and the dotted ellipse is the 95% confidence bound produced by the EKF method.

being at the center of the ellipse. The dotted ellipse shows the covariance predicted by EKF, with the EKF mean being at the center of that ellipse. Again, EKF is very far from the truth.

In both examples, SPKF greatly outperforms EKF. We discuss why in the next section.

6. Sigma-point Kalman filters

We have seen that the EKF approach to generalizing the KF to non-linear systems is to linearize the equations at each sample point using a Taylor-series expansion. This amounts to a first-order approximation of the required terms, with the questionable assumption being that the second- and higher-order terms are insignificant. Additionally, the EKF does not accurately account for the uncertainty of the underlying random variable in that the EKF equations are expanded around the *a priori* mean, with covariance expected to scale according to the slope of the function at this point only. The true *a posteriori* spread may be significantly different if the function being linearized is in fact quite non-linear in the neighborhood of the *a priori* mean. These approximations may result in large losses in estimation accuracy and have been observed to result in unstable filters [17,18,12].

Sigma-point Kalman filtering (SPKF) is an alternate approach to generalizing the Kalman filter to state estimation for non-linear systems. Rather than using Taylor-series expansions to approximate the required covariance matrices; instead, a number of function evaluations are performed whose results are used to compute an estimated covariance matrix. This has several advantages: (1) derivatives do not need to be computed (which is one of the most error-prone steps when implementing EKF), also implying (2) the original functions do not need to be differentiable, and (3) better covariance approximations are usually achieved, relative to EKF, allowing for better state estimation, (4) all with comparable computational complexity to EKF.

SPKF estimates the mean and covariance of the output of a non-linear function using a small fixed number of function evaluations. A set of points (*sigma points*) is chosen to be input to the function so that the (possibly weighted) mean and covariance of the points exactly matches the mean and covariance of the *a priori* random variable being modeled. These points are then passed through the non-linear function, resulting in a transformed set of points. The *a posteriori* mean and covariance that are sought are then approximated by the mean and covariance of these points. 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.

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

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

with columns 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* [19,20] 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 $\{\gamma, \alpha^{(\mathrm{m})}, \alpha^{(\mathrm{C})}\}$ if we define the weighted mean as $\bar{x} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}_i$, the weighted covariance as $\Sigma_{\tilde{x}} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^{\mathrm{T}}$, $\mathcal{X}_i$ as the ith column of $\mathcal{X}$, and both $\alpha_i^{(m)}$ and $\alpha_i^{(\mathrm{c})}$ as real scalars with the necessary (but not sufficient) conditions that $\sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} = 1$ and $\sum_{i=0}^{p} \alpha_i^{(\mathrm{C})} = 1$. 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) [21–24,18,12] and the *Central Difference Kalman Filter* (CDKF) [25,26,8]—are summarized in Table 4. The UKF is derived from the point of view of estimating covariances with data rather than Taylor series. The CDKF is derived quite differently—it uses Stirling's formula to approximate derivatives rather than using Taylor series—but the final method is essentially identical. The CDKF has only one "tuning parameter" h, which makes implementation simpler. It also has marginally higher theoretic accuracy than UKF [26], so we focus on this method in the application sections later. Specifically, we used $h = 1.2$ (which was hand-tuned to give somewhat better results than the default value of $h = \sqrt{3}$), and $L = \dim\{x\} = 6$.

Before introducing the SPKF algorithm, we reexamine the examples of Figs. 2 and 3 using sigma-point methods. In the

Table 4
Weighting constants for two sigma-point methods

	γ	$\alpha_0^{(\mathrm{m})}$	$\alpha_k^{(\mathrm{m})}$	$\alpha_0^{(\mathrm{c})}$	$\alpha_k^{(\mathrm{c})}$
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 $(10^{-2}\le\alpha\le 1)$. Note that this α is different from $\alpha^{(\mathrm{m})}$ and $\alpha^{(\mathrm{c})}$. κ is either 0 or $3-L$. β incorporates prior information. For Gaussian RVs, $\beta=2$. h may take any positive value. For Gaussian RVs, $h=\sqrt{3}$.

one-dimensional example in Fig. 2, three input sigma points are needed and map to the output three sigma points shown. The mean and variance of the sigma-point method is shown as a dashed-line PDF and closely matches the true mean and variance. For the two-dimensional example in Fig. 3, five sigma points represent the input random-variable PDF, as shown in frame (a). These five points are transformed to the five output points in frame (b). We see that the mean and covariance of the output sigma points (dashed ellipse) closely match the true mean and covariance and are much better than EKF.

Will the sigma-point method always be so much better? The answer depends on the degree of non-linearity of the state and output equations—the more non-linear the better SPKF should be with respect to EKF. Given that our battery model is fairly linear—the non-linearities arise only in the hysteresis state and the OCV function—we expect modest improvements. Here, we proceed to develop the SPKF algorithm so that we can demonstrate results shortly.

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 as described below.

SPKF step 1: State estimate time update. 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: $\hat{x}_{k-1}^{a,+}=[(\hat{x}_{k-1}^{+})^{\mathrm{T}},\bar{w},\bar{v}]^{\mathrm{T}}$, and the augmented *a posteriori* covariance estimate: $\Sigma_{\tilde{x},k-1}^{a,+}=\mathrm{diag}(\Sigma_{\tilde{x},k-1}^{+},\Sigma_w,\Sigma_v)$. These factors are used to generate the $p+1$ sigma points

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

From the augmented sigma points, the $p+1$ vectors comprising the state portion $\mathcal{X}_{k-1}^{x,+}$ and the $p+1$ vectors comprising the process-noise portion $\mathcal{X}_{k-1}^{w,+}$ are extracted. The process equation is evaluated using all pairs of $\mathcal{X}_{k-1,i}^{x,+}$ and $\mathcal{X}_{k-1,i}^{w,+}$ (where the subscript i denotes that the ith vector is being extracted from the original set), yielding the *a priori* sigma points $\mathcal{X}_{k,i}^{x,-}$ for time step k. That is,

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

Finally, the *a priori* state estimate is computed as

$$\hat{x}_k^-=\mathbb{E}[f(x_{k-1},u_{k-1},w_{k-1},k-1)|\mathbb{Y}_{k-1}]$$
$$\approx\sum_{i=0}^{p}\alpha_i^{(\mathrm{m})}f(\mathcal{X}_{k-1,i}^{x,+},u_{k-1},\mathcal{X}_{k-1,i}^{w,+},k-1)=\sum_{i=0}^{p}\alpha_i^{(\mathrm{m})}\mathcal{X}_{k,i}^{x,-}.$$

SPKF step 2: Error covariance time update. Using the *a priori* sigma points from step 1, the *a priori* covariance estimate is computed as

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

SPKF step 3: Estimate system output y_k. The system output is estimated by evaluating the model output equation using 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

$$\hat{y}_k=\mathbb{E}[h(x_k,u_k,v_k,k)|\mathbb{Y}_{k-1}]$$
$$\approx\sum_{i=0}^{p}\alpha_i^{(\mathrm{m})}h(\mathcal{X}_{k,i}^{x,-},u_k,\mathcal{X}_{k-1,i}^{v,+},k)=\sum_{i=0}^{p}\alpha_i^{(\mathrm{m})}\mathcal{Y}_{k,i}.$$

SPKF step 4: Estimator gain matrix L_k. To compute the estimator gain matrix, we must first compute the required covariance matrices.

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

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

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

SPKF step 6: Error covariance measurement update. The final step is calculated directly from the optimal formulation: $\Sigma_{\tilde{x},k}^{+}=\Sigma_{\tilde{x},k}^{-}-L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}}$. The SPKF solution is summarized in Table 5.

7. Application to battery management systems

7.1. Cell and cell test description

In order to compare the various Kalman filtering methods' abilities to estimate SOC and SOH, we gathered data from a prototype LiPB cell. The cell comprises a $\mathrm{LiMn_2O_4}$ cathode, an artificial graphite anode, is designed for high-power applications, has a nominal capacity of 7.5 Ah and a nominal voltage of 3.8 V. SOC estimation results using EKF have been previously reported for this cell, which we call a GEN3 cell [1–6], and may be used as a benchmark for comparison. Note that the results in the companion paper [7] are for a newer generation cell, which we call a G4 cell.

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 16 "urban dynamometer driving schedule" (UDDS) cycles, separated by 40 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. 4(a), and rate as a function of time for one of the UDDS cycles

Table 5
Summary of the non-linear sigma-point Kalman filter

Non-linear 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 Σ_w and Σ_v, respectively	
Definitions: let $x_k^a = [x_k^{\mathrm{T}}, w_k^{\mathrm{T}}, v_k^{\mathrm{T}}]^{\mathrm{T}}$, $\mathcal{X}_k^a = [(\mathcal{X}_k^x)^{\mathrm{T}}, (\mathcal{X}_k^w)^{\mathrm{T}}, (\mathcal{X}_k^v)^{\mathrm{T}}]^{\mathrm{T}}$, $p = 2 \times \dim(x_k^a)$	
Initialization: for $k = 0$, set $\hat{x}_0^+ = \mathbb{E}[x_0]$ $\Sigma_{\tilde{x},0}^+ = \mathbb{E}[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^{\mathrm{T}}]$	$\hat{x}_0^{a,+} = \mathbb{E}[x_0^a] = [(\hat{x}_0^+)^{\mathrm{T}}, \bar{w}, \bar{v}]^{\mathrm{T}}$ $\Sigma_{\tilde{x},0}^{a,+} = \mathbb{E}[(x_0^a - \hat{x}_0^{a,+})(x_0^a - \hat{x}_0^{a,+})^{\mathrm{T}}]$ $= \mathrm{diag}(\Sigma_{\tilde{x},0}^+, \Sigma_w, \Sigma_v)$
Computation: for $k = 1, 2, \ldots$ compute	
State estimate time update	$\mathcal{X}_{k-1}^{a,+} = \{\hat{x}_{k-1}^{a,+}, \hat{x}_{k-1}^{a,+} + \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}, \hat{x}_{k-1}^{a,+} - \gamma\sqrt{\Sigma_{\tilde{x},k-1}^{a,+}}\}$ $\mathcal{X}_{k,i}^{x,-} = f(\mathcal{X}_{k-1,i}^{x,+}, u_{k-1}, \mathcal{X}_{k-1,i}^{w,+}, k-1)$ $\hat{x}_k^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{X}_{k,i}^{x,-}$
Error 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	$\mathcal{Y}_{k,i} = h(\mathcal{X}_{k,i}^{x,-}, u_k, \mathcal{X}_{k-1,i}^{v,+}, k)$ $\hat{y}_k = \sum_{i=0}^{p} \alpha_i^{(\mathrm{m})} \mathcal{Y}_{k,i}$
Estimator gain matrix	$\Sigma_{\tilde{y},k} = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{Y}_{k,i} - \hat{y}_k)(\mathcal{Y}_{k,i} - \hat{y}_k)^{\mathrm{T}}$ $\Sigma_{\tilde{x}\tilde{y},k}^- = \sum_{i=0}^{p} \alpha_i^{(\mathrm{c})} (\mathcal{X}_{k,i}^{x,-} - \hat{x}_k^-)(\mathcal{Y}_{k,i} - \hat{y}_k)^T$ $L_k = \Sigma_{\tilde{x}\tilde{y},k}^- \Sigma_{\tilde{y},k}^{-1}$
State estimate measurement update	$\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$
Error covariance measurement update	$\Sigma_{\tilde{x},k}^+ = \Sigma_{\tilde{x},k}^- - L_k \Sigma_{\tilde{y},k} L_k^{\mathrm{T}}$

is plotted in Fig. 4(b). We see that SOC increases by about 5% during each UDDS cycle, but is brought down about 10% during each discharge between cycles. The entire operating range for these cells (10% SOC to 90% SOC marked as dashed lines in Fig. 4(a)) is excited during the cell test.

The data was used to identify parameters of the cell models to be described in the next sections. 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% SOC to 90% SOC.

7.2. Enhanced self correcting model description

In order to examine and compare performance of different Kalman filters, 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" (ESC) cell model from references [3,5]. This model includes effects due to open-circuit-voltage, internal resistance, voltage time constants, and hysteresis.

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

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

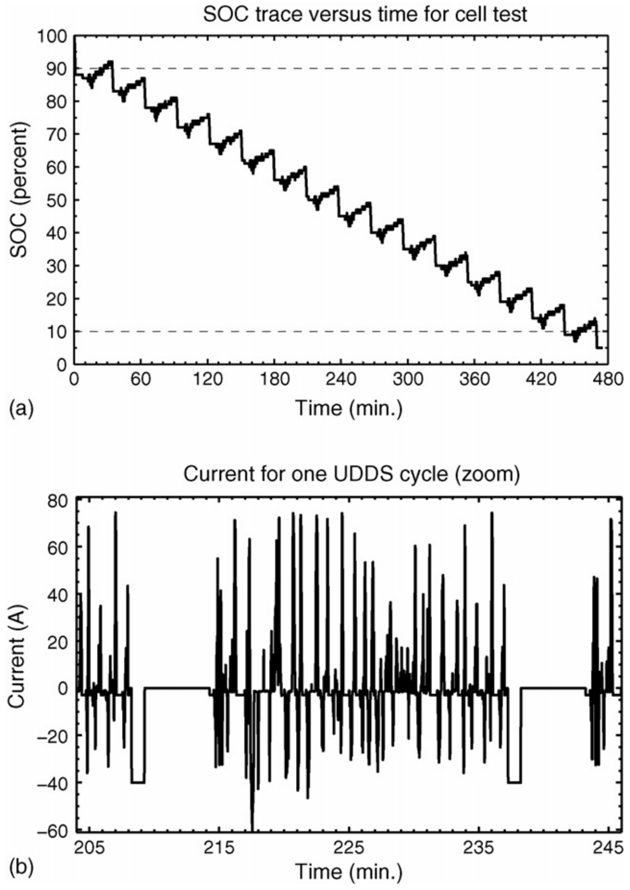

Fig. 4. Plots showing SOC vs. time and rate vs. time for UDDS cell tests. SOC is shown in (a); rate for one UDDS cycle is shown in (b).

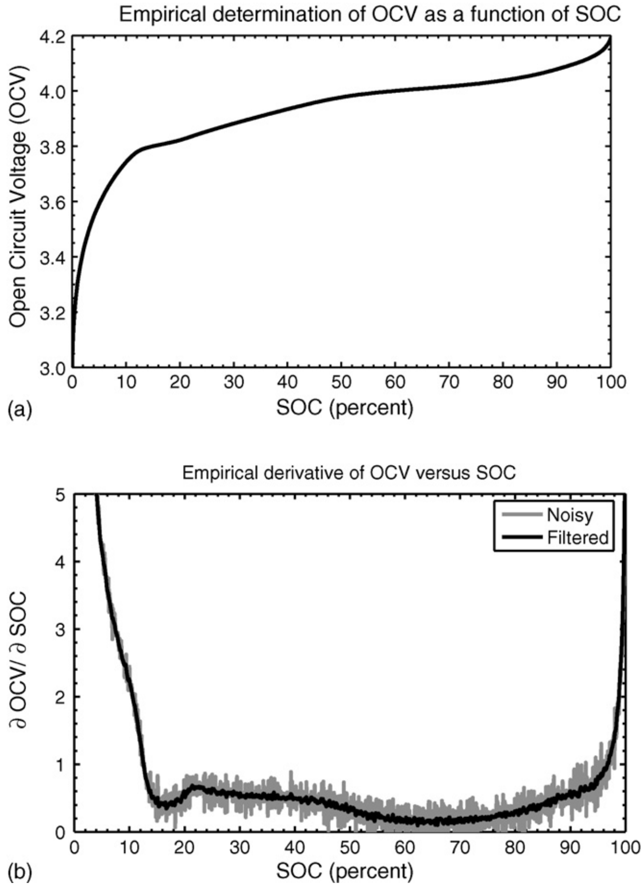

Fig. 5. Plots of (a) open-circuit-voltage as a function of state-of-charge, and (b) derivative of OCV as a function of SOC. In (b), raw, noisy version shown as gray, filtered derivative shown as black.

where η_i is the cell Coulombic efficiency at current i_{k-1}, ΔT represents the inter-sample period (in seconds), and C represents the cell capacity (in As).

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 γ 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.

The open-circuit-voltage as a function of state-of-charge for these cells is plotted in Fig. 5(a). This is an empirical relationship found by cell testing. For the purpose of computations involving OCV, the final curve was digitized at 200 points and stored in a table. Linear interpolation is used to look up values in the table.

The partial derivative of OCV with respect to SOC, required by EKF but not SPKF, is plotted in Fig. 5(b). This relationship was computed by first taking finite differences between points in the OCV plot in Fig. 5(a) and dividing by the distance between points (i.e., Euler's approximation to a derivative). The resulting data is too noisy to be of practical use, as shown in the gray line of Fig. 5(b). It was filtered using a zero-phase low-pass filter, resulting in the black line of Fig. 5(b), which is used in EKF. This relationship is also digitized at 200 points, and linear

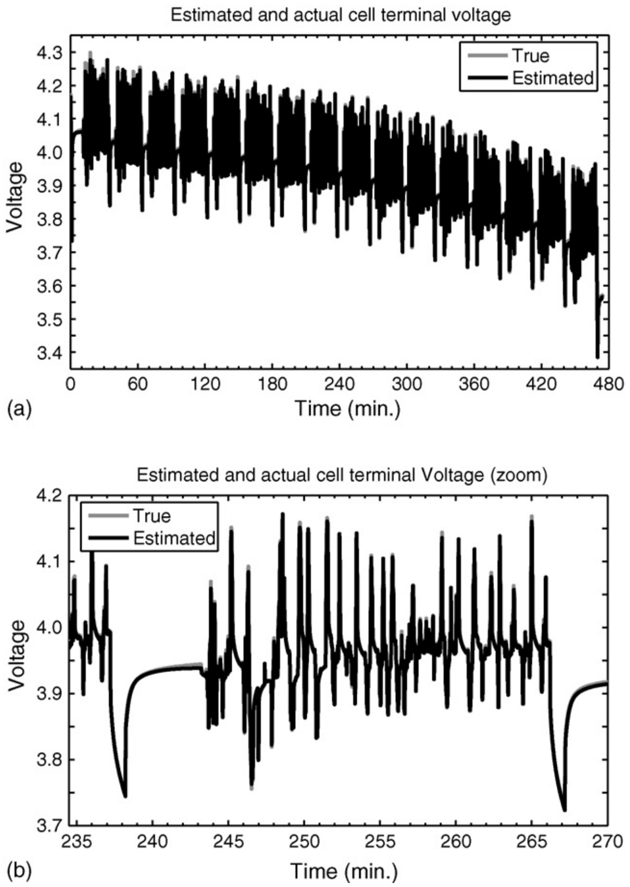

Fig. 6. Voltage prediction using cell model. Gray is true voltage, black is estimated voltage. In (b) a zoom of voltage prediction for one UDDS cycle at around 50% SOC is shown.

interpolation into the table of values is used when computations requiring this function are performed.

Other parameters are fit to the cell model using a method previously described ([5], Section 4). In particular, the model employs four low-pass filter states ($n_f = 4$), a nominal capacity of 7.5 Ah, and an inter-sample interval of $\Delta T = 1$ s. Further, $A_f = \text{diag}\{0.9624, 0.8509, 0.9981, 0.99999\}$, $B_f = 10^{-4} \times [1, 1, 1, 1]^T$ (chosen to scale all states—including the hysteresis and SOC state—to roughly the same dynamic range), $G = [-0.5256, -1.3258, -0.1855, 0.0012]$, $\gamma = 2.2523$, $\eta = 1$, $M = 74.7$ mV, the charging resistance was 2.6 mΩ, and the discharging resistance was 2.7 mΩ. These are the same values used to generate the results in [5]. There is very close agreement between the cell model voltage prediction and the cell true voltage. This is illustrated in Fig. 6(a). To better illustrate the model's fidelity, refer to the zoom on one UDDS cycle in the 50% SOC region, shown in Fig. 6(b).

7.3. Examples of EKF versus SPKF

Using the data fit to the ESC model, we ran an EKF and an SPKF to predict SOC. Note that the EKF results in this section are identical to those in [6] and are replicated here for clarity of comparison with the SPKF results. All "tuning" parameters were the same between EKF and SPKF; the model and all cell-test data files used were identical.

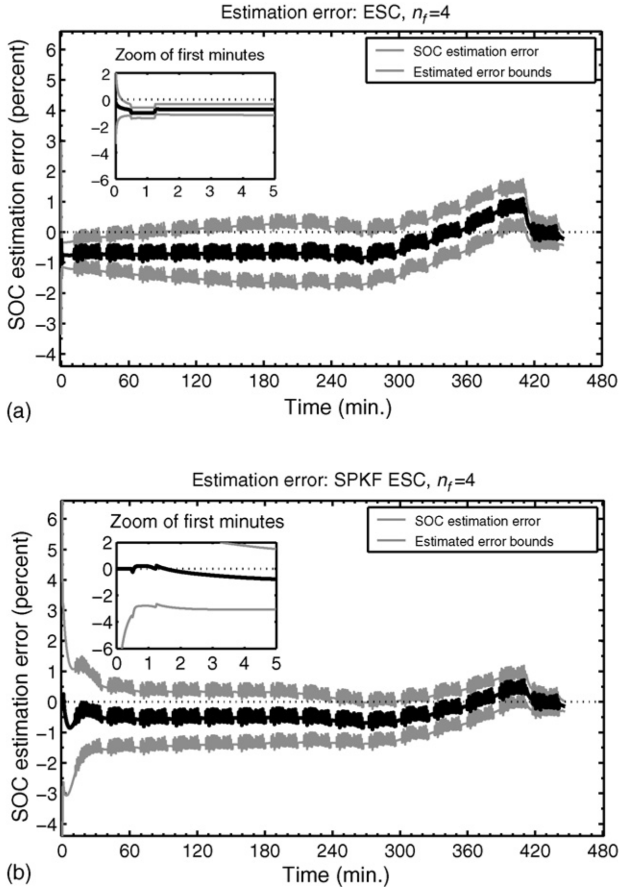

Fig. 7. SOC estimation error for: (a) EKF vs. (b) SPKF, when correctly initialized.

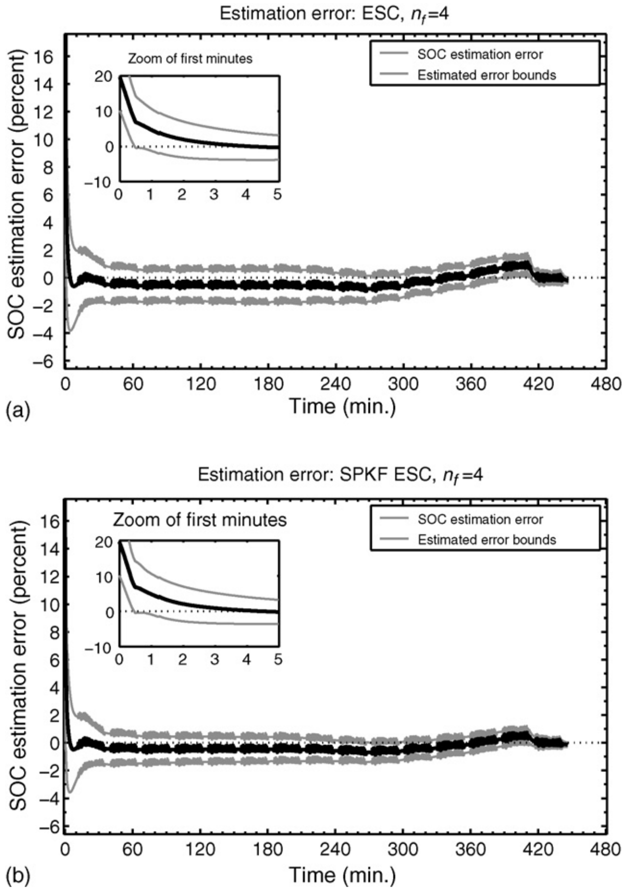

Fig. 8. SOC estimation error for: (a) EKF vs. (b) SPKF, when not correctly initialized.

SOC estimation error is plotted in Fig. 7 for both technologies when the filters are correctly initialized (to initial SOC =100%). The EKF RMS SOC estimation error is 0.64% while the SPKF RMS SOC estimation error is 0.49%—an improvement of over 23%. The maximum absolute SOC estimation error was 1.10% for EKF and 0.90% for SPKF—an improvement of 18%. The bounds correctly encompassed the zero point 95.11% of the time for SPKF, but only 74.73% of the time for EKF.

In Fig. 8, we compare the results for EKF versus SPKF when the filters were incorrectly initialized to SOC=80% rather than the correct 100%. Both filters had equivalent uncertainty matrices on the states, so recovery from the initial error took a similar amount of time. Error bounds for the SPKF method were slightly tighter, however.

The EKF had an RMS SOC estimation error of 0.75%, while the SPKF had an RMS SOC estimation error of 0.69%. This is an 8% improvement, which at first appears small compared to the first trial case. Notice that the RMS error is here dominated by the initial convergence of the estimators, which is nearly equivalent for both technologies. The lower error of SPKF may be attributed mostly to its convergence closer to the true SOC after the initial transient. Also, the SPKF bounds were correct 97.86% of the time, while the EKF bounds were correct only 93.86% of the time. These numeric results are replicated in Table 6 for easy comparison. In steady state we observe SPKF to improve RMS

Table 6
Comparison of EKF vs. SPKF in UDDS test results predicting SOC

	Correctly initialized			Incorrectly initialized	
	RMS error (%)	Maximum error (%)	Bounds error (%)	RMS error (%)	Bounds error (%)
EKF	0.64	1.10	25.27	0.75	6.14
SPKF	0.49	0.90	4.89	0.69	2.14
Improvement	23	18	81	8	65

error by about 20% and to greatly improve prediction of error bounds because of a better state error covariance estimate.

8. Conclusions

This paper has considered several methods for state estimation of a battery cell with application to battery-management systems of hybrid-electric vehicles. These algorithms are based on optimal estimation theory (also known as sequential probabilistic inference) and encompass several members of the Kalman filter family. In particular, we have shown how the Kalman filter, the extended Kalman filter and the sigma-point Kalman filters can be derived and applied to state estimation.

In prior work we have shown how the extended Kalman filter can be used in battery-management systems. Here, we have shown that the sigma-point Kalman filter may also be used for LiPB-cell state estimation, at an equivalent computational complexity. Data from testing using a third-generation prototype LiPB cell shows that SPKF gives superior results. The RMS SOC estimation error is lower, the maximum SOC estimation error is lower, and the error bounds produced on the estimate are more accurate.

The state estimate produced by SPKF can also be used to predict available power [27], or to effect equalization via SOC [6]. Furthermore, the estimate can be made more accurate if the cell parameters are also estimated in real time to account for any manufacturing differences between cells, and to track the effects of aging. Details on how to do this using SPKF are discussed in the companion to this paper [7].

Acknowledgments

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, and 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.

Appendix A. Proof of Gaussian recursion

In this appendix, we prove the claims of Eqs. (3) and (4) using the definitions of Eqs. (5)–(11). That is, we find $\mathbb{E}[x_k|\mathbb{Y}_k]$ under the basic assumption that all densities remain Gaussian. We first derive a fundamental equation governing Gaussian estimation, and then apply it to the problem at hand.

Theorem. *If x and y are jointly Gaussian vectors with means $\bar{x}$ and $\bar{y}$ and joint covariance Σ, then $p(x|y)$ is Gaussian with mean $\bar{x} + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\bar{y})$ and covariance $\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}$, and thus $\mathbb{E}[x|y] = \bar{x} + \Sigma_{xy}\Sigma_{yy}^{-1}(y-\bar{y})$.*

Proof. First, we write,

$$p(x \mid y) = \frac{p(x,y)}{p(y)} \propto \exp \times \left(-\frac{1}{2} \left\| \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \right\|^2_{\Sigma^{-1}} + \frac{1}{2}\|y-\bar{y}\|^2_{\Sigma_{yy}^{-1}} \right), \tag{12}$$

where $\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}$ is the joint covariance matrix. Then, substitute the transformation

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix} = \begin{bmatrix} I & \Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix} \times \begin{bmatrix} \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Sigma_{yy}^{-1}\Sigma_{yx} & I \end{bmatrix},$$

maintaining the inner matrix without expansion. The terms in the exponent of (12) become

$$\exp\left(-\frac{1}{2}(x - \bar{x} - \Sigma_{xy}\Sigma_{yy}^{-1}(y-\bar{y}))^{\mathrm{T}}(\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx})^{-1}(x - \bar{x} - \Sigma_{xy}\Sigma_{yy}^{-1}(y-\bar{y})) \right),$$

verifying the claim.

We can now proceed to verify Eqs. (3) and (4) by finding $\mathbb{E}[x_k|\mathbb{Y}_k]$. First, we define $\tilde{x}_k^- = x_k - \hat{x}_k^-$ where $\hat{x}_k^- = \mathbb{E}[x_k|\mathbb{Y}_{k-1}]$ and $\tilde{y}_k = y_k - \hat{y}_k$ where $\hat{y}_k = \mathbb{E}[y_k|\mathbb{Y}_{k-1}]$. By assumption, $\tilde{x}_k^-$ and $\tilde{y}_k$ are jointly Gaussian, and by construction have zero mean. Then, by Theorem 1 we have that $\mathbb{E}[\tilde{x}_k^-|\tilde{y}_k] = \Sigma_{\tilde{x}\tilde{y},k}\Sigma_{\tilde{y},k}^{-1}\tilde{y}_k$, which we define to be $L_k\tilde{y}_k$.

Secondly, we must show that $\mathbb{E}[\tilde{x}_k^-|\mathbb{Y}_{k-1}] = \mathbb{E}[\tilde{x}_k^-]$ and therefore $\tilde{x}_k^-$ is uncorrelated with $\mathbb{Y}_{k-1}$ (independent because Gaussian). This is straightforward by substitution:

$$\mathbb{E}[\tilde{x}_k^-|\mathbb{Y}_{k-1}] = \mathbb{E}[x_k - \mathbb{E}\{x_k|\mathbb{Y}_{k-1}\}|\mathbb{Y}_{k-1}] = 0 = \mathbb{E}[\tilde{x}_k^-].$$

Therefore, we can write

$$\begin{aligned}\mathbb{E}[\tilde{x}_k^- | \mathbb{Y}_k] &= \mathbb{E}[\tilde{x}_k^- \mid \mathbb{Y}_{k-1}, \tilde{y}_k] \\ &= \underbrace{\mathbb{E}[\tilde{x}_k^- | \tilde{y}_k]}_{L_k(y_k - \hat{y}_k)} = \underbrace{\mathbb{E}[x_k | \tilde{y}_k]}_{\hat{x}_k^+} - \underbrace{\mathbb{E}[\hat{x}_k^- | \tilde{y}_k]}_{\hat{x}_k^-}.\end{aligned}$$

From this last line, we solve for the *a posteriori* state estimate $\hat{x}_k^+ = \hat{x}_k^- + L_k(y_k - \hat{y}_k)$, which verifies Eq. (3) □.

The covariance of $\tilde{x}_k^+$ may be computed using Eq. (3).

$$\begin{aligned}\Sigma_{\tilde{x},k}^+ &= \mathbb{E}[\{(x_k - \hat{x}_k^-) - L_k\tilde{y}_k\}\{(x_k - \hat{x}_k^-) - L_k\tilde{y}_k\}^{\mathrm{T}}] \\ &= \Sigma_{\tilde{x},k}^- - L_k \underbrace{\mathbb{E}[\tilde{y}_k(\tilde{x}_k^-)^{\mathrm{T}}]}_{\Sigma_{\tilde{y},k}} L_k^{\mathrm{T}} - \underbrace{\mathbb{E}[\tilde{x}_k^- \tilde{y}_k^{\mathrm{T}}]}_{L_k\Sigma_{\tilde{y},k}} L_k^{\mathrm{T}} + L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}} \\ &= \Sigma_{\tilde{x},k}^- - L_k\Sigma_{\tilde{y},k}L_k^{\mathrm{T}},\end{aligned}$$

which verifies Eq. (4).

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 2. Simultaneous state and parameter estimation, Int. J. Power Sources, 10.1016/j.jpowsour.2006.06.004.

[8] 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.

[9] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in: IEE Proceedings, Part F, vol. 140, 1993, pp. 107–113.

[10] A. Doucet, On Sequential Simulation-Based Methods for Bayesian Filtering, Tech. Rep. CUED/F-INFENG/TR 310, Cambridge University Engineering Department, 1998.

[11] A. Doucet, J. de Freitas, N. Gordon, Introduction to sequential Monte Carlo methods, in: A. Doucet, J. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, Berlin, 2000.

[12] 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).

[13] R. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME—J. Basic Eng. 82 (Series D) (1960) 35–45.

[14] The Seminal Kalman Filter Paper (1960) http://www.cs.unc.edu/welch/kalman/kalmanPaper.html, accessed 20 May 2004.

[15] S. Haykin, Kalman filters, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley Inter-Science, New York, 2001, pp. 1–22.

[16] 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. 123–74.

[17] E. Wan, R. van der Merwe, A. Nelson, Dual estimation and the unscented transformation, in: Neural Information Processing Systems 12, MIT Press, 2000, pp. 666–672.

[18] E. Wan, R. van der Merwe, The unscented Kalman filter for nonlinear estimation, Proceedings of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, 2000.

[19] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C: The Art of Scientific Computing, second ed., Cambridge University Press, 1992.

[20] G. Stewart, Matrix Algorithms, vol. I: Basic Decompositions, SIAM, 1998.

[21] 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.

[22] S. Julier, J. Uhlmann, A General Method for Approximating Nonlinear Transformations of Probability Distributions, Tech. rep., RRG, Department of Engineering Science, Oxford University, November 1996.

[23] 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.

[24] S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (3) (2004) 401–22.

[25] M. Nørgaard, N. Poulsen, O. Ravn, Advances in Derivative-Free State Estimation for Nonlinear Systems, Technical re IMM-REP-1998–15, Dept. of Mathematical Modeling, Tech. Univ. of Denmark, 28 Lyngby, Denmark, April 2000.

[26] M. Nørgaard, N. Poulsen, O. Ravn, New developments in state estimation for nonlinear systems, Automatica 36 (11) (2000) 1627–1638.

[27] G. Plett, High-performance battery-pack power estimation using a dynamic cell model, IEEE Trans. Vehicular Technol. 53 (5) (2004) 1586–1593.