Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Power Sources 134 (2004) 252–261
JOURNAL OF POWER SOURCES
www.elsevier.com/locate/jpowsour

Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs
Part 1. Background

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 27 January 2004; accepted 26 February 2004

Available online 9 June 2004

Abstract

Battery management systems (BMS) in hybrid-electric-vehicle (HEV) battery packs must estimate values descriptive of the pack's present operating condition. These include: battery state of charge, power fade, capacity fade, and instantaneous available power. The estimation mechanism must adapt to changing cell characteristics as cells age and therefore provide accurate estimates over the lifetime of the pack.

In a series of three papers, we propose a method, based on extended Kalman filtering (EKF), that is able to accomplish these goals on a lithium-ion polymer battery pack. We expect that it will also work well on other battery chemistries. These papers cover the required mathematical background, cell modeling and system identification requirements, and the final solution, together with results.

This first paper investigates the estimation requirements for HEV BMS in some detail, in parallel to the requirements for other battery-powered applications. The comparison leads us to understand that the HEV environment is very challenging on batteries and the BMS, and that precise estimation of some parameters will improve performance and robustness, and will ultimately lengthen the useful lifetime of the pack. This conclusion motivates the use of more complex algorithms than might be used in other applications. Our premise is that EKF then becomes a very attractive approach. This paper introduces the basic method, gives some intuitive feel to the necessary computational steps, and concludes by presenting an illustrative example as to the type of results that may be obtained using EKF.

Keywords: Battery management system (BMS); Hybrid electric vehicle (HEV); Extended Kalman filter (EKF); State of charge (SOC); State of health (SOH); Lithium-ion polymer battery (LiPB)

1. Introduction

This paper is the first in a series of three that describe 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 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 a gasoline (or diesel) engine and an electric motor. Both may be coupled directly to the power train—resulting in a "parallel hybrid" 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 on the market in the US are parallel hybrid systems and require a battery pack of fairly modest size. Even so, and because of the demanding

* Tel.: +1-719-262-3468; fax: +1-719-262-3589.
E-mail address: glp@eas.uccs.edu (G.L. Plett).
URL: http://mocha-java.uccs.edu.
[1] The author is also consultant to Compact Power Inc., Monument, CO 80132, USA. Tel.: +1-719-488-1600; fax: +1-719-487-9485.

0378-7753/$ – see front matter
doi:10.1016/j.jpowsour.2004.02.031

requirements on a pack of limited capacity, advanced methods must be used to estimate SOC, SOH, and instantaneous power in order to safely, efficiently and aggressively exploit the pack capabilities.

The method we use to estimate these parameters is based on Kalman filter theory. 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 estimation 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 over-charge or over-discharge. Note that there have been other reported methods for SOC estimation that use Kalman filtering [1,2], but the method in this series of papers expands on these results and also differs in some important respects, as will be outlined later [3].

This first paper is an introduction to the problem. It describes the HEV environment and the algorithmic requirement specifications for a BMS. The remainder of the paper is a brief tutorial on the Kalman filter theory necessary to grasp the content of the remaining papers; additionally, a nonlinear extension called the "extended Kalman filter" (EKF) is discussed.

The second paper [3] describes some mathematical cell models that may be used with this method. It also gives an overview of other modeling methods in the literature and shows how an EKF may be used to adaptively identify unknown parameters in a cell model, in real time, given cell input–output data.

The third paper [4] covers the parameter estimation problem; namely, how to dynamically estimate SOC, power fade, capacity fade and so forth. An EKF is used in conjunction with the cell model. The cell model may be fixed, or may itself have adaptable parameters so that the model tracks cell aging effects. Details for a practical implementation are discussed.

We now proceed by discussing requirements for a BMS in the HEV environment, and comparing them to requirements for other battery-powered systems. The additional requirements of HEV justify the use of advanced algorithms. We then review essential Kalman filter theory with the aim being to demystify the steps involved. An example of linear Kalman filtering is given to illustrate the presentation.

2. HEV versus portable electronics BMS environments

In principle, the results of these papers could be applied to manage the performance of any battery-based system, including, for example, hybrid electric vehicles, battery electric vehicles (BEVs) and consumer portable electronics (PE). The HEV environment, however, is particularly harsh—imposing many difficult requirements on the battery cells and BMS—and motivates the use of advanced techniques. In our experience, battery management algorithms developed for portable electronic applications, for example, do not work adequately for the HEV application.

The HEV BMS performs many tasks, including communicate with the vehicle controller, measure cell physical quantities of interest (e.g., cell voltage, current and temperature), and manage cell balancing. Here, we are only interested in the algorithmic considerations as motivated by the requirements imposed by the environment and the vehicle. Fig. 1 shows a simple block diagram for the algorithm function, and a short description is given below:

- *Initialization.* When the vehicle is turned on, the algorithms must be initialized. The predominant dynamics of a cell while at rest is simply "self-discharge". If the level of self-discharge is too high, the state of health should be flagged as a warning or fault condition.
- *SOC update.* Once in every measurement interval, the voltages, temperatures and module current are measured. The cell/pack SOC estimate must be updated based on these measurements.
- *SOH update.* Battery capacity and other parameters change over the lifetime of the pack. These must be continuously estimated in order to maintain safety and to obtain maximum performance from the pack.
- *Maximum available power.* Based on the SOC estimate and its uncertainty, and a dynamic cell model, the BMS must be able to estimate the maximum dis/charge power available at any time that will not cause voltage, SOC, or other design limits to be violated.
- *Equalization.* Series strings of cells with unequal capacities (as all are) will become unbalanced. That is, even if the SOC of all cells start with the same value, they will drift slowly apart as the system operates. The BMS must determine which cells must have their levels of charge altered to keep the pack balanced.

In the next several sections we will explore how the HEV BMS requirements differ from other battery application requirements in these respects.

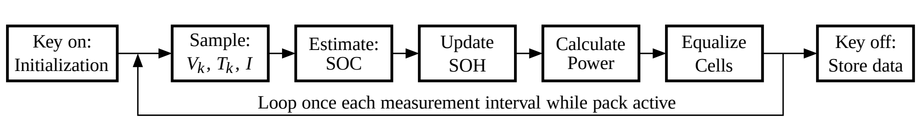

Fig. 1. Algorithms in an HEV BMS.

Table 1
Typical characteristics of three battery operating environments

Characteristic	HEV	BEV	PE
Maximum rate	$\pm 20C$	$\pm 5C$	$\pm 3C$
Rate profile	Very dynamic	Moderate	Piecewise constant
SOC estimation	Very precise	Precise	Crude
Predict available power	Yes	Yes	No
Cell balancing	Continuous	Continuous, or on charge only	On charge only
SOH estimation	Required	Required	Not essential
Lifetime	10–15 years	10–15 years	<5 years

2.1. Rate characteristics

Three different categories of battery-powered systems and typical characteristics of their operating environments are listed in Table 1. We find that the characteristics of the HEV environment are more demanding on battery cells and battery management systems than the other two. For example, HEVs require very high electrical current relative to the capacity of the cells, with present vehicles demanding up to ±20 times the *C*-rate. We expect that future systems will require even higher relative rates. Battery electric vehicles, on the other hand, generally require peak absolute current $< 5C$ and PE are designed for low absolute rates $< 3C$. The rate profile (current as a function of time) for HEV is also very dynamic as HEVs are typically designed so that the battery/motor system handles the instantaneous load transients and the engine handles the average load [5,6]. BEV rate profiles are less dynamic, and many PE devices experience nearly constant power drains. The low power drain of PE results in cells that are always in a near-equilibrium state and allows simple battery management algorithms. For example, SOC may be estimated by measuring the terminal voltage (which, at low *C* rates, is never much different from the open-circuit-voltage) and then performing a table lookup function of open-circuit-voltage versus SOC. In contrast, the high rates and dynamic rate profiles of HEV result in cell electro-chemistry that is rarely in an equilibrium state while the vehicle operates. Therefore, for example, voltage sensing without further processing is a poor SOC estimator.

2.2. SOC estimation

Of primary importance is a method to accurately estimate the SOC of cells in the pack. We will define SOC more carefully later [3], but what is meant is an indication of the fraction of charge remaining in each cell, from 0 to 100%, available to do useful work. To use a vehicular analogy, it is similar to the dashboard gas gauge that reads “Empty” (0%) to “Full” (100%). However, while there exist sensors to accurately *measure* a gasoline level in a tank, there is no sensor available to measure SOC. Instead, SOC must be *estimated* from physical measurements by some algorithm. The technique to be developed in these papers is unique in that it not only provides an accurate estimate of SOC, it also provides dynamic error bounds on the estimate. This is a direct consequence of the way we use Kalman filtering.

For PE, an imprecise SOC estimator is often adequate. However, for peak performance—required in HEV and BEV—an accurate SOC estimate provides the following benefits [7]:

- *Longevity.* If a gasoline tank is over-filled or run empty, no harm is done to the tank. However, over-charging or over-discharging a battery cell may cause permanent damage and result in reduced lifetime. An accurate SOC estimate may be used to avoid harming cells by not permitting current to be passed that would cause damage.
- *Performance.* Without a good SOC estimator, one must be overly conservative when using the battery pack to avoid over/undercharge due to trusting the poor estimate. With a good estimate, especially one with known error bounds, one can aggressively use the entire pack capacity.
- *Reliability.* A poor SOC estimator behaves differently for different driving profiles. A good SOC estimator is constant and dependable, enhancing overall power system reliability.
- *Density.* Accurate SOC and battery state information allows the battery pack to be used aggressively within the design limits, so the pack does not need to be over-engineered. This allows smaller, lighter battery packs.
- *Economy.* Smaller battery systems cost less. Warranty service on a reliable system costs less.

We will spend a considerable portion of these papers describing an accurate method to estimate SOC for these reasons.

2.3. Available power estimation

A second algorithmic BMS requirement is to dynamically estimate the maximum available battery charge and discharge power (or, maximum charge and discharge current). The estimate is computed knowing SOC, temperature and a model of cell dynamics, and must be reliable over the whole SOC and temperature operating range. Upon computing the (charge or) discharge limit, the BMS then guarantees that the vehicle may safely draw this constant power level for a pre-defined number of seconds into the future without exceeding voltage and SOC design limits. We use the EKF

SOC estimate along with the corresponding error bounds to conservatively and accurately predict available power while allowing pack usage to be as aggressive as possible for the available data.

2.4. Cell balancing

For battery packs to achieve required power levels at reasonable rates, cells must be configured in series strings. (The higher voltage of a series string allows lower current for the same power level.) Over time, these cells may become "out of balance" as small differences in their dynamics—principally, in their Coulombic efficiencies and capacities—cause their states of charge to drift apart from each other. The danger is that one or more cells may then limit the discharge ability of the pack if their SOC is much lower than that of the others, and one or more cells may limit the charging capacity of the pack if their SOC is much higher than that of the others. In an extreme case, the pack can neither be discharged nor charged if one cell is at the low SOC limit and another is at the high SOC limit, even if all other cells are at intermediate values. Packs may be *balanced* or *equalized* by "boosting" (individually adding charge to) cells with SOC too low and "bucking" (individually depleting charge from) cells with SOC too high. In BEV and PE, balancing is often done using voltage-based methods at the end-of-charge point of the charging process. For example, the pack might be charged until the highest cell voltage reaches some defined limit, after which the remaining cells are individually boosted until the entire pack is equalized. This same method may not be used in HEV since cells are never fully charged or discharged; rather, equalization must occur continuously. This significantly complicates the algorithm and hardware requirements. In particular, equalization should be done on a differential SOC basis, not on a voltage basis, requiring a good SOC estimation algorithm. Additionally, we use SOC error bounds to create an equalization "dead band" so that balancing halts if cells are close to equalized to avoid over-stressing cells.

2.5. SOH estimation

For the HEV application, knowledge of battery *state of health* is required. SOH is partially described by a vector of diagnostic flags including simple measurements such as: "Are there any cells with voltage too high or too low?", "Is the pack current too high?", and "Are the temperatures of any cells too high or too low?". Complete SOH estimation also requires more complex estimation: "Are there any cells with SOC above or below design limits?", "Are there any cells with self-discharge rate above some acceptable limit?", "Has the capacity of any cell faded below some minimum acceptable value?", "Does the internal resistance of any cell exceed some limit?", and so forth. This allows a service technician to identify cells in a pack that need to be replaced without the need to replace all the cells. SOH information may also be written to a data log for warranty purposes. In a portable electronic device, elaborate SOH estimation is not required: when the user determines that the battery is no longer giving acceptable performance he or she simply replaces the whole pack.

2.6. Lifetime

Finally, for commercial success, the lifetime of the HEV cells must meet or exceed the lifetime of the vehicle. Replacing a battery pack every few years is not acceptable. Cell electro-chemistry and construction plays a dominant role in longevity, but good BMS algorithms can extend life as well by prohibiting pack use that over-stresses cells, thus preventing damage.

The pack's long life has other ramifications; for example, we know that the BMS must be able to make accurate predictions over the entire lifetime of the cells. This implies that the BMS must estimate or track all relevant cell parameters as time goes by. By adapting to changing cell characteristics, the BMS can accurately estimate available power over the entire life of the battery system, and will not allow current so high that cells are damaged. We use EKF methods to do this as well.

3. Linear Kalman filtering

Many of the algorithm requirements just described prescribe estimating parameters of a battery pack that may not be directly measured. We find that Kalman filtering provides an elegant and powerful solution. Kalman filtering is an established technology for dynamic system state estimation that is in common use in many fields including: target tracking, global positioning, dynamic systems control, navigation, and communication, but is not widely known in the battery field. The Kalman filter comprises a set of recursive equations that are repeatedly evaluated as the system operates. We will not directly derive these equations here; rather, our hope is that the following discussion will aid intuition into the method's workings. The reader is referred to Kalman's original paper and several textbooks [8–12] for further derivation details.

Very generally, any causal dynamic system—including a battery cell—generates its outputs as some function of the past and present inputs. It is often also convenient to think of the system having a "state" vector (which may not be directly measurable) where the state summarizes the effect of all past inputs on the system. Present system output may be computed with present input and present state only—past input values need not be stored. We will apply Kalman filter theory by viewing each cell in the battery pack to be a dynamic system whose inputs include the current and temperature experienced by the cell and whose output is the (loaded) terminal voltage. The state vector may include SOC, relaxation dynamics and hysteresis effects, for example.

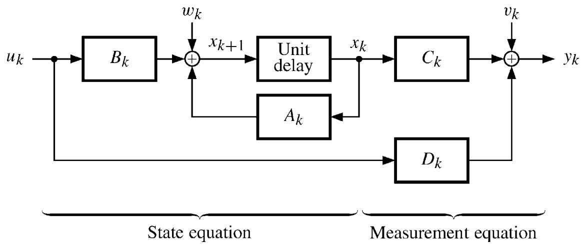

Fig. 2. Diagram of linear discrete-time system in state-space form.

To be efficiently computable by the BMS, we will consider a discrete-time version of the cell dynamics. Each measurement interval, indexed by integer valued time index k (e.g., perhaps once per second) the model updates its state and output values based on its input. A very general framework that we may use is a "state-space" model of discrete-time lumped linear dynamic systems:

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

$$y_k = C_k x_k + D_k u_k + v_k. \quad (2)$$

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 linear combination of states and input plus $v_k \in \mathbb{R}^m$, which models "sensor noise" that affects the measurement of the system output in a memoryless way, but does not affect the system state. 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. This equation is also illustrated in the block diagram of Fig. 2.

Given a model as Eqs. (1) and (2), we may wish to estimate the unmeasured state x_k of the corresponding physical system, in real time, in a dynamic environment, given knowledge of the system's measured input/output signals. The Kalman filter is the optimum method to do so under certain assumptions. By modeling a cell's dynamics with the desired unknown quantities (e.g., SOC) as members of the model state vector, the Kalman filter will automatically compute the best estimate of their present values.

Some assumptions are made when deriving the filter equations. First, 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:

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

where $\mathbb{E}[\cdot]$ is the statistical expectation operator and a superscript T is the matrix/vector transpose. The assumptions on the noise processes w_k and v_k are rarely (never) met in practice, but the consensus of the literature is that the method still works very well. Our own results corroborate the robustness of the Kalman filter.

Table 2
Summary of the linear Kalman filter from [11]

Linear state-space model[a]

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$
$$y_k = C_k x_k + D_k u_k + v_k$$

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: $\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u_{k-1}$

Error covariance time update: $\Sigma_{\tilde{x},k}^- = A_{k-1}\Sigma_{\tilde{x},k-1}^+ A_{k-1}^{\mathrm{T}} + \Sigma_w$

Kalman 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:
$\hat{x}_k^+ = \hat{x}_k^- + L_k[y_k - C_k\hat{x}_k^- - D_k u_k]$

Error covariance measurement update: $\Sigma_{\tilde{x},k}^+ = (I - L_k C_k)\Sigma_{\tilde{x},k}^-$

[a] w_k and v_k are independent, zero-mean, Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively.

The Kalman filter problem is then: Use the entire observed data $\{u_0, u_1, \ldots, u_k\}$ and $\{y_0, y_1, \ldots, y_k\}$ to find the minimum mean squared error estimate $\hat{x}_k$ of the true state x_k. That is, with the assumptions on w_k and v_k and a system modeled as Eqs. (1) and (2), solve[2]

$$\hat{x}_k = \arg\min_{\hat{x}\in\mathbb{R}^n} \mathbb{E}[(x_k - \hat{x})^{\mathrm{T}}(x_k - \hat{x})|u_0, u_1, \ldots, u_k, y_0, y_1, \ldots, y_k].$$

The solution to this problem is widely known and is presented in Table 2. (For more details on deriving the equations, see for example [8–12].) The heart of the solution is a set of computationally efficient recursive relationships that involve both an estimate of the state itself, and also the covariance matrix $\Sigma_{\tilde{x},k} = \mathbb{E}[\tilde{x}_k\tilde{x}_k^{\mathrm{T}}]$ of the state estimate error $\tilde{x}_k = x_k - \hat{x}_k$. The covariance matrix indicates the uncertainty of the state estimate, and may be used to generate error bounds. A "large" $\Sigma_{\tilde{x},k}$ (one with large singular values) indicates a high level of uncertainty in the state estimate; a "small" $\Sigma_{\tilde{x},k}$ (one with small singular values) indicates confidence in the estimate.

The discrete-time Kalman filter computes two different estimates of the state and covariance matrix each sampling interval. The first estimate, $\hat{x}_k^-$, is based on the prior state

[2] The system must also be "observable", a condition implying that it is in fact possible to estimate the state from the output. The systems we will explore meet this requirement, so we will not discuss it in detail.

estimate as computed in the previous iteration, $\hat{x}_{k-1}^{+}$, propagated forward in time one sample interval using a model of system dynamics. It is computed *before* any system measurements are made, and is denoted by superscript "−". The second estimate, $\hat{x}_k^{+}$, "tunes up" the first estimate *after* measuring the system output y_k. It is given the superscript "+". The state and covariance estimates $\hat{x}_k^{+}$ and $\Sigma_{\tilde{x},k}^{+}$ are then more accurate than $\hat{x}_k^{-}$ and $\Sigma_{\tilde{x},k}^{-}$ as they incorporate knowledge gleaned from the measurement y_k, and should be used by the BMS to report SOC estimates (etc.) to the vehicle.

The Kalman filter is initialized with the best available information on the state and error covariance:

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

Often, these quantities are not precisely known, and initialization must be performed in an ad hoc manner. This is not a problem as the Kalman filter is known to be very robust to poor initialization, and will quickly converge to the true values as it runs.

Following initialization, the Kalman filter repeatedly performs two steps each measurement interval. First, it *predicts* the value of the present state, system output, and error covariance: $\hat{x}_k^{-}$, $\hat{y}_k$, and $\Sigma_{\tilde{x},k}^{-}$. Secondly, using a measurement of the physical system output, it *corrects* the state estimate and error covariance to $\hat{x}_k^{+}$ and $\Sigma_{\tilde{x},k}^{+}$.

The prediction step, also known as the *time update*, computes the expected state value at the next measurement point. It is accomplished by propagating the system input through the system model dynamics, assuming the expected process noise of zero:

$$\hat{x}_k^{-} = A_{k-1}\hat{x}_{k-1}^{+} + B_{k-1}u_{k-1}.$$

The state uncertainty is also updated:

$$\Sigma_{\tilde{x},k}^{-} = A_{k-1}\Sigma_{\tilde{x},k-1}^{+}A_{k-1}^{\mathrm{T}} + \Sigma_w.$$

If the system is stable, $A_{k-1}\Sigma_{\tilde{x},k-1}^{+}A_{k-1}^{\mathrm{T}}$ is contractive, reducing uncertainty. An undriven stable system state always decays toward zero, so certainty of the state estimate is improved over time. The process noise Σ_w term always increases uncertainty, as we cannot measure w_k to determine more accurately how it is affecting the state.

Following the output measurement, the state correction step, also known as the *measurement update* is

$$\hat{x}_k^{+} = \hat{x}_k^{-} + L_k[y_k - (C_k\hat{x}_k^{-} + D_k u_k)].$$

That is, the updated state estimate equals the predicted state estimate plus a weighted correction factor. The term in the square brackets is equal to the measured cell voltage minus the predicted cell voltage from the cell model: $\hat{y}_k = C_k\hat{x}_k^{-} + D_k u_k$. The difference $y_k - \hat{y}_k$ may be nonzero due to measurement noise, an incorrect state estimate $\hat{x}_k^{-}$, or an inaccurate cell model. It represents the "new information" in the measurement, and so the sequence of differences is often called the "innovations process" for this reason. If any innovation is large, the corresponding state update tends to be large. If an innovation is small, the state update tends to be small.

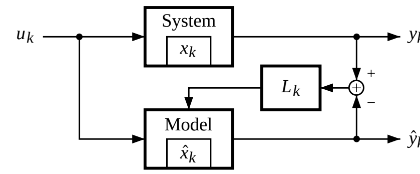

Fig. 3. Diagram of state update.

Each innovation is weighted by *Kalman gain* vector L_k in the state update

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

If the present state estimate is very uncertain, $\Sigma_{\tilde{x},k}^{-}$ is "large" and the values in L_k tend to be large, forcing a large update. If the present state estimate is certain, the values in L_k tend to be small, and the state-estimate update will be small. Also, if sensor noise is large, Σ_v is large, causing L_k to be small and the update to be small. The Kalman gain may be thought of as a signal-to-noise (SNR) ratio balancing factor that has high gain if the innovations signal has relatively high SNR, and has low gain if the SNR is low. Kalman filter convergence is faster for high SNR.

The covariance correction step is

$$\Sigma_{\tilde{x},k}^{+} = (I - L_k C_k)\Sigma_{\tilde{x},k}^{-}.$$

The state uncertainty always decreases due to the new information provided by the measurement.

With this basic understanding, we can view the Kalman filter macroscopically as depicted in Fig. 3. The true system has a measured input u_k and a measured output y_k. It also has an unmeasured internal state x_k. A model of the system runs in parallel with the true system, simulating its performance. This model has the same input u_k and has output $\hat{y}_k$. It also has internal state $\hat{x}_k$, which has known value as it is part of the model simulation. The true system output (a scalar, in our case) is compared with the model output, and the difference is an output error, or innovation. This innovation is converted to a vector value by multiplying with the Kalman gain L_k, and used to adapt the model state $\hat{x}_k$ to more closely approximate the true system's state. The state estimate and uncertainty estimates are updated through computationally efficient recursive relationships.

In conclusion, the Kalman filter provides a theoretically elegant and time-proven method to filter measurements of system input and output to produce an intelligent estimate of a dynamic system's state. The equations involve basic matrix operations that are easy to implement on digital-signal-processing (DSP) chips. A side effect of the Kalman filter is that the state uncertainty matrix is automatically produced, giving an indication of the error bound on the estimate. For the Kalman filter assumptions, 95.4% of the time the true unknown state x_k is bounded by

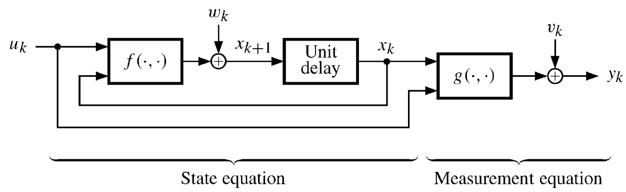

Fig. 4. Diagram of nonlinear discrete-time system in state-space form.

$\hat{x}_k \pm 2\sqrt{\text{diag}(\Sigma_{\tilde{x},k})}$ or 99.7% of the time x_k is bounded by $\hat{x}_k \pm 3\sqrt{\text{diag}(\Sigma_{\tilde{x},k})}$.

4. Extended Kalman filtering

The Kalman filter is the optimum state estimator for a linear system with the assumptions as described. If the system is nonlinear, then we may use a linearization process at every time step to approximate the nonlinear system with a linear time varying (LTV) system. This LTV system is then used in the Kalman filter, resulting in an extended Kalman filter (EKF) on the true nonlinear system. Note that although EKF is not necessarily optimal, it often works very well.

We model the nonlinear system as:

$$x_{k+1} = f(x_k, u_k) + w_k, \tag{3}$$

$$y_k = g(x_k, u_k) + v_k. \tag{4}$$

As before, w_k and v_k are zero-mean white Gaussian stochastic processes with covariance matrices Σ_w and Σ_v, respectively. Now, $f(x_k, u_k)$ is a nonlinear state transition function and $g(x_k, u_k)$ is a nonlinear measurement function. The nonlinear system is also illustrated in the block diagram of Fig. 4.

At each time step, $f(x_k, u_k)$ and $g(x_k, u_k)$ are linearized by a first-order Taylor-series expansion. We assume that $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are differentiable at all operating points (x_k, u_k). Then [13, Theorem 1],

$$f(x_k, u_k) \approx f(\hat{x}_k, u_k) + \underbrace{\left.\frac{\partial f(x_k, u_k)}{\partial x_k}\right|_{x_k=\hat{x}_k}}_{\text{Defined as } \hat{A}_k} (x_k - \hat{x}_k), \tag{5}$$

$$g(x_k, u_k) \approx g(\hat{x}_k, u_k) + \underbrace{\left.\frac{\partial g(x_k, u_k)}{\partial x_k}\right|_{x_k=\hat{x}_k}}_{\text{Defined as } \hat{C}_k} (x_k - \hat{x}_k). \tag{6}$$

Combining (3) and (4) with (5) and (6), we have the linearized equations describing the true system state as a function of itself, known inputs comprising u_k and $\hat{x}_k$, and unmeasurable noise inputs w_k and v_k:

$$x_{k+1} \approx \hat{A}_k x_k + \underbrace{f(\hat{x}_k, u_k) - \hat{A}_k \hat{x}_k}_{\text{Not a function of } x_k} + w_k,$$

$$y_k \approx \hat{C}_k x_k + \underbrace{g(\hat{x}_k, u_k) - \hat{C}_k \hat{x}_k}_{\text{Not a function of } x_k} + v_k.$$

Table 3
Summary of the nonlinear extended Kalman filter from [14]

Nonlinear state-space model[a]

$$x_{k+1} = f(x_k, u_k) + w_k$$
$$y_k = g(x_k, u_k) + v_k$$

Definitions

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

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: $\hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-1})$

Error covariance time update: $\Sigma_{\tilde{x},k}^- = \hat{A}_{k-1}\Sigma_{\tilde{x},k-1}^+\hat{A}_{k-1}^{\mathrm{T}} + \Sigma_w$

Kalman gain matrix: $L_k = \Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}}[\hat{C}_k\Sigma_{\tilde{x},k}^-\hat{C}_k^{\mathrm{T}} + \Sigma_v]^{-1}$

State estimate measurement update: $\hat{x}_k^+ = \hat{x}_k^- + L_k[y_k - g(\hat{x}_k^-, u_k)]$

Error covariance measurement update: $\Sigma_{\tilde{x},k}^+ = (I - L_k\hat{C}_k)\Sigma_{\tilde{x},k}^-$

[a] w_k and v_k are independent, zero-mean, Gaussian noise processes of covariance matrices Σ_w and Σ_v, respectively.

By using these approximations, the EKF algorithm may be developed. The terms labeled “Not a function of x_k” replace the $B_k u_k$ and $D_k u_k$ known input terms in the standard Kalman filter. The final algorithm is summarized in Table 3.

In spirit, the EKF is very similar to standard KF. The initialization step itself is identical. Each iteration, a prediction and a correction step are done. In EKF, the propagation step to predict the present state uses the nonlinear model in the same way that KF uses the linear model. The error covariance propagation and Kalman gain equations are identical to those of the KF, except that now the linearized $\hat{A}_k$ matrix replaces A_k and $\hat{C}_k$ replaces C_k. The state estimate update is identical, except now $\hat{y}_k = g(\hat{x}_k^-, u_k)$ and the error covariance update only differs in using $\hat{C}_k$ rather than C_k.

Before proceeding, we should note that EKF is not the only possible nonlinear extension of KF. In particular, unscented and NPR Kalman filters [15–17] are alternative methods that may provide even better estimates than EKF without the need to differentiate the model. To date, we have not explored these methods in detail.

5. Example of Kalman filtering

In order to illustrate some of the concepts outlined in this paper, we present a simple example of linear Kalman filtering. We consider the system defined by the linear circuit in Fig. 5. We find the continuous-time state-space model of

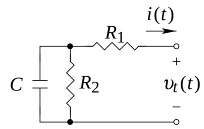

Fig. 5. Simple linear circuit.

this circuit to be:

$$\dot{v}_c(t) = -\frac{1}{R_2C}v_c(t) - \frac{1}{C}i(t) + \frac{1}{C}w(t),$$
$$v_t(t) = v_c(t) - R_1 i(t) + v(t),$$

where $v_c(t)$ is the capacitor voltage as a function of time, $i(t)$ the current exciting the circuit, and $v_t(t)$ the terminal voltage, as indicated in the diagram. This circuit is a crude linear model of a battery cell if both C and R_2 are large and R_1 is small. R_2 is the resistor governing self-discharge, and R_1 is the internal resistance of the cell. The signal $w(t)$ is an uncertain input that affects the system state (capacitor voltage) and might model error in the current sensor, such as a quantization error. According to the Kalman filter assumptions, $w(t)$ must be white and Gaussian, although in practice the method still works very well in most cases where these assumptions are only approximately met. The signal $v(t)$ might model voltage sensor noise that does not affect the system state (e.g., quantization noise on the sensor). It is also assumed to be white and Gaussian, but again, the Kalman filter often works very well even in cases where these assumptions are only approximately met.

We can convert this model to discrete time using standard techniques [18]:

$$v_{c,k+1} = e^{-T/(R_2C)}v_{c,k} - R_2(1 - e^{-T/(R_2C)})i_k + w_k,$$
$$v_{t,k} = v_{c,k} - R_1 i_k + v_k.$$

Here, $v_{c,k}$ is the capacitor voltage at time index k, i_k the current measurement (assumed held constant over a measurement interval), and $v_{t,k}$ the terminal voltage measurement.

We run a linear discrete-time Kalman filter on this system, with parameters for the simulation given in Table 4. The input to the system is a white Gaussian process with variance on each sample equal to Σ_i. Note that the initial state estimate was (purposefully) poor, and the initial state error covariance estimate was initialized too small. Even so, the filter recovers quickly and gives a good state estimate.

Some of the important signals involved in the filter are plotted in Fig. 6. In Fig. 6(a), we see actual capacitor voltage and estimated capacitor voltage plotted as a function of time.

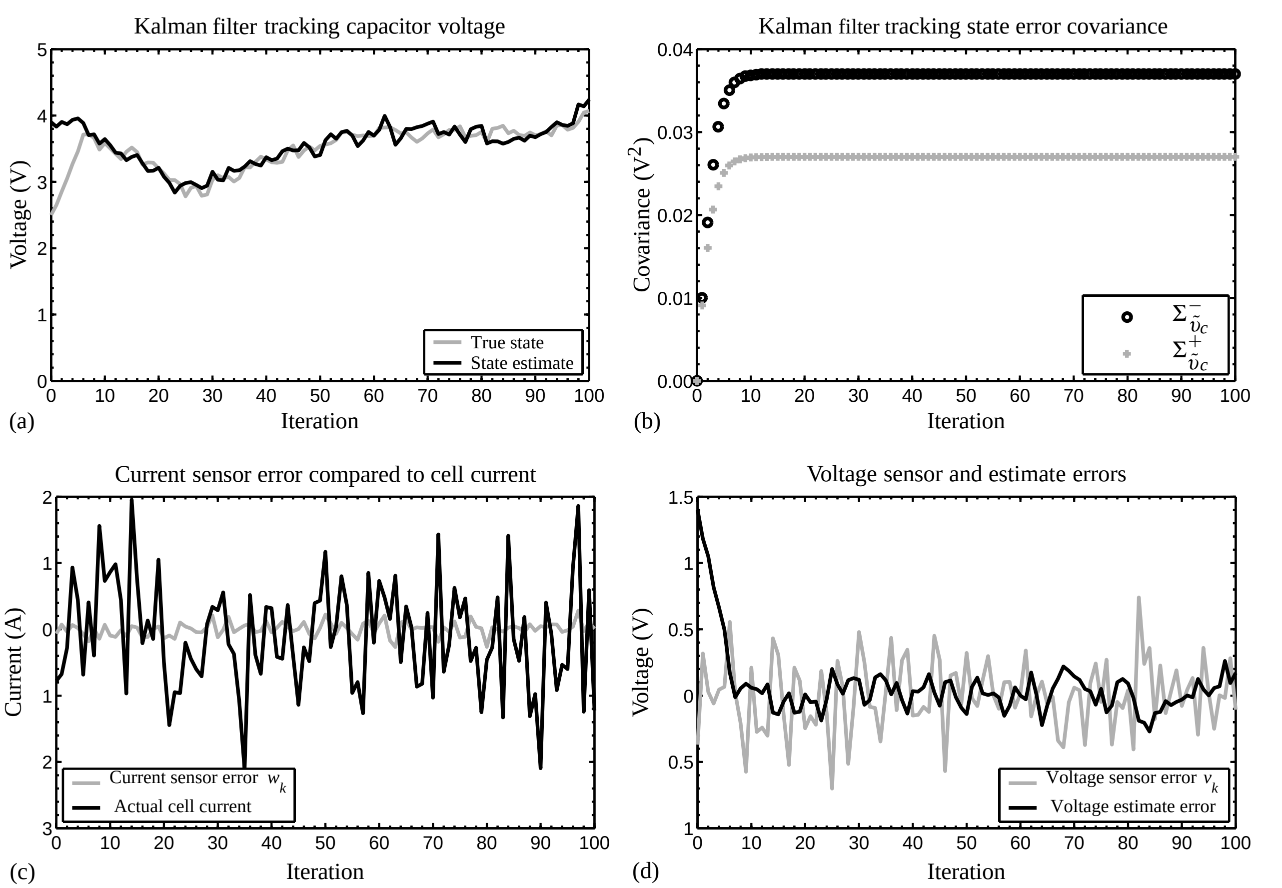

Fig. 6. Kalman filter example: (a) true state, and state estimate are plotted; (b) covariance values are plotted; (c) cell current and current-sensor error are plotted; (d) voltage sensor error and voltage estimate error are plotted.

Table 4
Parameters used in simulation

Variable	Value
T (s)	1
R_1 (mΩ)	10
R_2 (MΩ)	10
C (kF)	1
Σ_i (A^2)	1
Σ_w (A^2)	0.01
Σ_v (V^2)	0.1
$v_{c,0}$ (V)	3.9
$\hat{v}^+_{c,0}$ (V)	2.5
$\Sigma^+_{\tilde{v}_c,0}$ (V^2)	0.0

Although we intentionally use a poor estimate of $\tilde{v}^+_{c,0}$ for the purpose of example, the Kalman filter quickly converges to a close neighborhood of the true voltage. It never actually converges to the exact voltage due to the unmeasured noise w_k that is continuously driving the system. The Kalman filter makes an optimum trade-off between believing the sensor reading and believing the model to achieve the best possible state estimate.

In Fig. 6(b) we see the uncertainty matrix of the Kalman filter evolving over time. Two values are plotted: $\Sigma^-_{\tilde{v}_c,k}$ and $\Sigma^+_{\tilde{v}_c,k}$. We see that the former quantity (state uncertainty before output measurement) is always greater than the latter (state uncertainty after output measurement). We also see that, in this case, these two variables settle quickly to steady-state values. This is common in linear time-fixed systems, but is not expected in nonlinear or time-varying systems. Therefore, we cannot use steady-state values for $\Sigma^-_{\tilde{x},k}$ and $\Sigma^+_{\tilde{x},k}$ in our systems.

In Fig. 6(c) and (d) we illustrate that this example does not solve a trivial problem. In frame (c), we see the actual cell current compared to the current-sensor error w_k as a function of iteration. We see that the current-sensor error is a significant fraction of the actual cell current. In frame (d), we see that the voltage sensor error v_k is also *larger* than the voltage estimate error $\tilde{v}_c = v_c - \hat{v}_c$. Even though both sensors are very noisy, the Kalman filter is able to compute an optimally clean estimate of the true capacitor voltage.

6. Conclusions

In this paper we have described the algorithmic requirements of the BMS and its operating environment in the HEV scenario. The particular demands of HEV justify the use of advanced algorithms. We have also reviewed the Kalman filter and extended Kalman filter methods, explaining the motive of each computational step. We presented an example to clarify the discussion. In the following two papers, we will combine this information in such a way that we are able to meet the algorithmic requirements of the BMS.

In order to estimate SOC, we will use a state-space model of the cell dynamics, with SOC as a member of the model state vector. An EKF is then able to estimate SOC. It remains to find a good cell model function and to identify the parameters of this model. When the model is found, the parameters of the model must be determined using a system identification procedure. All of this is described in [3].

Once we have a good cell model, we may estimate SOC. We will also need to estimate SOH, including capacity fade, power fade, self-discharge, and be able to adjust cell model parameters to account for cell aging. We must also be able to compute discharge/charge power. This is discussed in [4].

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 and Dan Rivers and others are gratefully acknowledged.

References

[1] P. Lürkens, W. Steffens, Ladezustandsschätzung von Bleibatterien mit Hilfe des Kalman-Filters, etzArchiv, vol. 8, No. 7, July 1986, pp. 231–236 (in German, English title: State of charge estimation of lead-acid batteries using a Kalman filtering technique).
[2] C. Barbier, H. Meyer, B. Nogarede, S. Bensaoud, A battery state of charge indicator for electric vehicle, in: Proceedings of the International Conference of Institution of Mechanical Engineers, Automotive Electronics, London, UK, May 17–19, 1994, pp. 29–34.
[3] 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.
[4] G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs. Part 3. State and parameter estimation, J. Power Sources 134 (2) (2004) 277–292.
[5] K.L. Butler, M. Ehsani, P. Kamath, A Matlab-based modeling and simulation package for electric and hybrid electric vehicles design, IEEE Trans. Veh. Technol. 48 (6) (1999) 1770–1778.
[6] M. Ehsani, Y. Gao, K. Butler, Application of electrically peaking hybrid (ELPH) propulsion system to a full size passenger car with simulated design verification, IEEE Trans. Veh. Technol. 48 (6) (1999) 1779–1787.
[7] S. Dhameja, Electric Vehicle Battery Systems, Newnes Press (an imprint of Butterworth-Heinemann), Boston, 2002.
[8] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME—J. Basic Eng., Ser. D 82 (1960) 35–45.
[9] The Seminal Kalman Filter Paper, 1960, Accessed 20 May 2003. http://www.cs.unc.edu/~welch/kalman/kalmanPaper.html.
[10] S. Haykin, Adaptive Filter Theory, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 1996.
[11] S. Haykin, Kalman filters, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley/Interscience, New York, 2001, pp. 1–22.
[12] J. Burl, Linear Optimal Control: H_2 and H_∞ Methods, Addison Wessley, Menlo Park, CA, 1999.
[13] J. Marsden, A. Tromba, Vector Calculus, 3rd ed., Freeman, 1998, pp. 243–247.
[14] E. Wan, A. Nelson, Dual extended Kalman filter methods, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley/Interscience, New York, 2001, pp. 123–174.

[15] M. Nørgaard, N. Poulsen, O. Ravn, Advances in derivative-free state estimation for nonlinear systems, Technical Report IMM-REP-1998-15, Technical University of Denmark, 2000.
[16] 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.
[17] E. Wan, R. van der Merwe, The unscented Kalman filter, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, Wiley/Interscience, New York, pp. 221–282.
[18] C.-T. Chen, Linear System Theory and Design, Oxford University Press, New York, 1998.