2006-21-0060

Advances in HEV Battery Management Systems

Gregory L. Plett
University of Colorado at Colorado Springs, and consultant to Compact Power Inc.

Martin J. Klein
Compact Power Inc.

ABSTRACT

A critical element of a hybrid-electric-vehicle (HEV) propulsion system is the battery management system (BMS), which controls the performance of the HEV battery, the costliest and heaviest component of the propulsion system. This paper examines the relevance and criticality of an HEV BMS as a whole; that is, its general functions and "responsibilities". Of these, its ability to accurately estimate and report the state-of-charge (SOC) is arguably the most important. This paper will explain why SOC estimation is important, and will examine advances in the state of the art of SOC estimation methods, with a focus on Kalman Filter techniques.

HEV BATTERY PACK OVERVIEW

Optimizing the cost, weight, size and reliability of major HEV systems is critical in maximizing the value of the HEV to the end customer. Of the components comprising the propulsion system of an HEV, the costliest is the battery pack, which may represent 30–35% of the total cost of the propulsion system. The battery is among the heaviest components of the propulsion system as well. Therefore, careful design of the battery pack and the BMS can dramatically impact the lifetime affordability of an HEV.

Figure 1 provides a block diagram of a typical battery pack. Comprising the pack are the battery cells, junction block(s), BMS, thermal management system, wiring and connectors, and the pack housing. (While it is possible for the BMS to be physically located outside of, or even remote from, the pack housing itself, for this paper it is assumed to be located within the boundaries of the pack.) In most applications, the cells are wired in series to develop the necessary high voltage.

The primary functions of the battery pack are to store electrical energy produced by the vehicle (during regenerative braking) and to provide electrical energy for use by the vehicle particularly during acceleration or other peak energy demands. The pack needs to do so in a manner that is safe, reliable, and cost efficient. This includes not only minimizing initial purchase costs, protecting the vehicle from voltage surges or drop-outs, and preventing harmful conditions, but also minimizing operational stresses that can shorten the life of the battery cells. These operational stresses include excessive temperature, excessive discharging, and excessive over-charging.

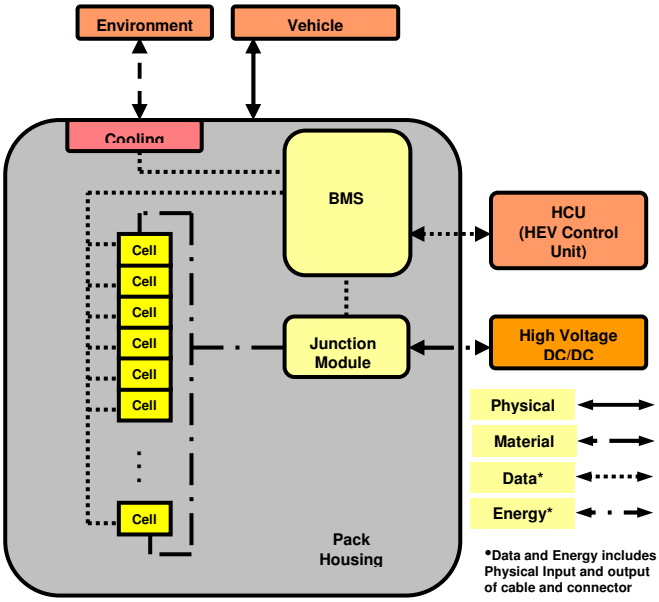

Figure 1
HEV Battery Pack Block Diagram

The pack housing provides the physical housing of the pack's components. The cooling system provides the physical capability for the air (or liquid) thermal media circulation. The junction module provides the high-voltage connection to the vehicle, as well as the necessary relays and safety interlocks for the pack.

Remaining is the BMS, which manages the delivery and acceptance of electrical energy to/from the cells, as well as the operation of the cooling system and junction module. The BMS consists of a printed circuit board (PCB) and connectors (and housing, if necessary). It provides the following functions (among others):

- Cell state monitoring (*e.g.*, voltage, temperature)
- Charge and discharge current measurement and limiting
- Management of the cooling system
- Necessary data conditioning, diagnostics and battery-to-host vehicle communication functions
- High voltage relay energizing and de-energizing.
- SOC and state-of-health (SOH) estimating, including the effects of aging

BATTERY MANAGEMENT AND SOC

Of the functions listed above, accurate SOC and SOH estimation are the most critical functions in optimizing the size and weight of a battery pack, as well as protecting the cells and providing a reliable, "transparent" driving experience.

- Accurate SOC estimation allows for optimal and smooth blending of battery power with the internal combustion (IC) engine;
- Maximum charge and discharge power limits based on SOC, temperature and SOH are needed to maximize battery life; and,
- Battery service indicators and diagnostic tools rely on accurate state information.

The BMS must have knowledge of the internal state and parameters of its cells in order to perform many of its functions. However, in most cases, the internal cell states and parameters cannot be directly measured, but must be estimated in some way.

This paper focuses on refinements to an approach for continually and dynamically estimating the state-of-charge of battery cells during HEV operation. Much attention is paid to SOC estimation accuracy because of the potential for optimizing battery usage—and therefore size, weight, cost and reliability—if the reported SOC can be trusted over the operating range and life of the HEV.

The most significant benefit of an accurate SOC estimate is the ability to minimize the number and size of cells needed to provide the range of power and energy required by the propulsion system. HEVs typically operate in an SOC range of 20% to 80%. If the SOC estimate uncertainty is high, several undesirable conditions could arise with the potential for over-discharging and overcharging.

If the SOC estimate is too optimistic (*i.e.*, it reports an available charge greater than reality), the propulsion system may demand power in excess of the ability of the battery to provide it while remaining above the minimum SOC. Several effects could then result:

- The pack will discharge deeper than expected, and therefore take longer to recharge to a median level. Also, if the vehicle is turned off in this state, the battery may not have the power re-start the engine when required;
- The BMS may detect an over-current condition and abruptly reduce available power to the propulsion system, resulting in a perception of poor drivability dynamics; and,
- In the worst case, battery cell damage may occur if no secondary over-discharge protection is available in the BMS.

These conditions will adversely affect customer perception of vehicle performance and reliability. One solution, in the absence of a better SOC estimate, is to add cells to provide the necessary "headroom" to compensate.

If the SOC estimate is too pessimistic (*i.e.*, it reports an available charge lower than reality), several adverse conditions may arise. During acceleration events, the propulsion system may unnecessarily limit the demand for battery power in favor of the internal combustion (IC) engine, resulting in lower fuel efficiency. Or, during deceleration/regen events, the BMS may "allow" recharge energy in excess of the ability of the battery to accept it while remaining below the maximum SOC. Battery cell damage may result in extreme cases. Again, customer perception will suffer, particularly if fuel economy or performance expectations are not met, or battery cells need premature replacement.

The above conditions pose considerable risks in terms of battery cost and reliability. To the extent the SOC estimate operates in a wide range of uncertainty (error), propulsion system designs will require excess battery cost and weight to ensure satisfactory battery and vehicle performance.

The potential for over-discharging and over-charging actually applies to both electric vehicles (EVs) and HEVs. However, HEVs place an additional burden on the SOC estimation algorithm in that, unlike EVs, many HEVs typically do not have a "plug-in" recharging feature. When connected to an external charger, a reasonably well-designed BMS can "reset" the SOC estimate to a high degree of accuracy. During the subsequent driving period, the SOC estimate may drift, but such drift error would be limited to a few hundred miles of driving until the user again plugs in the vehicle. Moreover, an "intelligent" BMS could use each subsequent manual recharge event to further refine the SOC algorithm and further reduce drift error. In a "plug-free" HEV, the SOC estimator must maintain its expected accuracy for tens of thousand of miles, and provide for operation (and memory) not only during normal operation, but also during vehicle shutdown and start-up (key-on, key-off), and fault conditions.

BATTERY PACK DYNAMICS

As mentioned above, the BMS for HEV applications is required to estimate quantities that are descriptive of the present battery pack condition, but that may not be directly measured. Some of these quantities may change rapidly, such as the pack state-of-charge (SOC), which can traverse its entire range within minutes. Others may change very slowly, such as cell capacity, which might change as little as 20% in a decade or more

of regular use. The quantities that tend to change quickly comprise the "state" of the system, and the quantities that tend to change slowly comprise the time varying "parameters" of the system.

Hybrid-electric vehicles are designed so that (ideally) the internal combustion engine provides average vehicle power, and the electric system provides the dynamic variation in the power demanded, resulting in overall high efficiency. One implication of the highly dynamic power demand on the battery system is that the cells are rarely in an electro-chemical equilibrium state. This disqualifies simple methods to estimate the battery internal state.

OVERVIEW OF CELL STATE ESTIMATION METHODS

Since accurate state estimates are required to establish battery power limits, and for driver interface and vehicle control, the associated algorithms must take into account such effects as hysteresis, polarization, time constants, etc. Additionally, these algorithms must be able to track changes in parameters as the cells age in order to maintain accuracy.

There are some estimation methods that, for a variety of application-specific reasons, are better (or worse) than others for use in SOC algorithms.

The Tino method, for example, is a voltage-based algorithm to estimate SOC that uses the approximation

$$\text{voltage} \approx \text{OCV}(\text{SOC}) - \text{I} \times \text{R}$$

$$\text{SOC} \approx \text{OCV}^{-1}(\text{voltage} + \text{I} \times \text{R})$$

where OCV =open circuit voltage

This does not take into account hysteresis, polarization voltages, etc., and so is a very crude estimate of SOC, especially at low temperatures where resistance can be quite high and where hysteresis can also be quite large. It is also poor at extreme values of SOC where the resistance is much larger than at moderate SOC values.

Coulomb-counting methods may be suitable for short periods of battery operation. These methods are initialized using an SOC value derived from the first voltage readings, just like the Tino method. However, coulomb-counting is subject to significant drift (error) and, in the case of HEV use, the counting "registers" rarely have the opportunity to be re-initialized.

This paper presents an approach that uses a mathematical model of cell dynamics to estimate HEV battery state and parameters. This model-based approach is a nonlinear variant of Kalman filtering, first published in 1961 [1,2], which itself is known to be the optimum estimator for a linear system. Kalman filters are ubiquitous in control systems, communications, defense, image processing, space, and GPS navigation applications, they present a single unified approach to BMS estimation tasks without requiring add-on reset mechanisms or correction factors for age, temperature, and so forth, are computationally simple—only requiring linear algebraic operations—and provide the best solution for robust long-term deployment.

Even within the domain of Kalman filtering we have found specific methods to perform better than others—sometimes significantly so. The Extended Kalman Filter (EKF), for example, is perhaps the most common approach to state estimation for nonlinear systems. The intent of this paper is not to present estimation results based on EKF—that has been done elsewhere [3–10]—rather, we point out that EKF has a number of flaws that can be improved upon fairly easily to improve state estimation.

Sigma-point Kalman filtering (SPKF) is an alternate approach to generalizing the Kalman filter to state estimation for nonlinear systems and, based on analysis and actual testing, has been shown to be superior to the methods described so far.

We proceed by first outlining the generic framework of model-based estimation, and then show how the present approach fits into that framework. We then outline the model structure that is used for our cells, give some results and some concluding remarks.

MODEL-BASED ESTIMATION

REQUIRED STRUCTURE OF THE MODEL

To use the given approach to estimate a cell's state, we must have a "state-space" model of its dynamics:

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

$$y_k = g(x_k, u_k, v_k) \quad (2)$$

where x_k is the state vector at discrete-time index k, u_k is the measured system input vector at time k, (perhaps including battery-pack current, temperature, etc.) and w_k is unmeasured "process noise" (modeling inaccuracy of the cell model). The system output is y_k, and v_k models sensor noise. The stochastic inputs w_k and v_k are assumed to be zero-mean white Gaussian random processes with covariance matrices Σ_w and Σ_v, respectively. Equation (1) is called the "state equation", (2) is called the "output equation", and f( ) and g( ) are (possibly nonlinear) functions, specified by the particular cell model used.

To be more specific, the system input vector u_k typically contains the instantaneous cell current i_k. It may also contain the cell temperature T_k, an estimate of the cell's nominal capacity C_k, and/or an estimate of the cell's internal resistance R_k, for example. The system output is typically a scalar but may be vector valued as well. Here we consider the output to be the cell's loaded

terminal voltage—not at-rest open-circuit-voltage (OCV). The system's state vector x_k in some way represents in summary form the total effect of all past input to the system so that the present output may be predicted solely as a function of the state and present input. Values of past inputs are not required. Our method constrains the state vector to include SOC as one component, so that SOC may later be estimated using some form of Kalman filter.

GENERAL FORM OF THE ESTIMATOR

Model-based estimation, based on these equations, is a recursive process to update an estimate $\hat{x}_k$ of the true state x_k. Assuming for the moment that the model perfectly represents the cell, we can use the model to estimate what is happening in the cell in real time. The following sequence of steps are repeatedly executed:

- The actual input current to the cell is measured. This value is used as input to the model.
- The model predicts what the cell voltage must then be, based on the input current and on the model's internal estimate of the cell state and parameters.
- The actual cell voltage is measured. If the model's state and parameters are exact, then there is no difference between the actual cell voltage and the model's estimate. Any difference is because of an error in the estimate of cell state or parameters.
- The cell-model state and parameter estimates are adapted to lower the cell voltage estimation error.
- The updated state/parameter estimates are output to be used for whatever purposes are desired.
- This process repeats every sampling interval (*e.g.*, every one second).

This same approach works even when the cell model is not in fact perfect, or when there is noise on the sensors, but adaptation of the model's state and parameter estimates must be made more slowly because of the additional uncertainty.

In order to optimally perform the adaptation step, the algorithm must internally weigh uncertainties in the model, uncertainties in the present state estimate, and uncertainties in the sensor inputs. These are represented mathematically by covariance matrices of the appropriate variables. The variables themselves are understood to hold the expected value of the quantities that they represent. Since the algorithm must know these uncertainties, it has the added capability of being able to determine error bounds (for example, 3-sigma or 6-sigma error bounds) on all estimated quantities.

To more precisely discuss the mathematical operations performed by the estimator, we first define some notation. A superscript "–" denotes a predicted quantity, a superscript "+" denotes an updated estimate of that quantity, a circumflex "^" denotes an estimated quantity, a tilde "~" denotes an estimation error, and Σ_x denotes the covariance of its subscripted variable. Further, we define $E[\ ]$ to be the statistically expected value of its argument, and $\mathbf{Y}_k = \{y_0,\ldots,y_k\}$ to be the history of measurements until time k.

This process is a kind of Kalman filtering, which is the optimal state estimator for linear systems. The batteries are nonlinear, so we use different varieties of the Kalman filter known as extended Kalman filtering and sigma-point Kalman filtering. This paper presents the methods in some detail, and gives results showing what kind of estimation may be achieved using LiPB high-power cells in an HEV environment.

Step 1: State estimate time update

The first step computes the state estimate time update, which predicts the present value of the state given past measurements:

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

Step 2: Error covariance time update

The second step determines the predicted state-estimate error covariance matrix $\Sigma_{\tilde{x},k}^-$ based on *a priori* information and the system model.

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

Step 3: Estimate system output

The third step is to estimate the system's output using present *a priori* information.

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

Step 4: Estimator gain matrix

The fourth step is to compute the optimal gain factor used when updating the state estimate, which is

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

Step 5: State estimate measurement update

The fifth step is to update the state estimate using the measured cell voltage, the predicted cell voltage, and the estimator gain matrix.

$$\hat{x}_k^+ = E[x_k \mid \mathbf{Y}_k] = \hat{x}_k^- + L_k(y_k - \hat{y}_k).$$

Step 5 is computed in the same way for all variants of the Kalman filter, so is not elaborated on in the sequel. Notice that the output estimation error is scaled by the gain matrix and used to adapt $\hat{x}_k$. The gain matrix tends to have large entries for states whose value is uncertain (high covariance) and small entries for measurements with a high degree of sensor noise. The gain matrix optimally combines new and old information in the filter.

Step 6: Error covariance measurement update

The final step of the update mechanism is to update the state-error covariance matrix.

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

THE EXTENDED KALMAN FILTER

As mentioned earlier in this paper, EKF has two flaws related to assumptions made in order to propagate a Gaussian random state vector x_k through some nonlinear 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( ) assuming that the input state is a Gaussian random variable. EKF step 3 makes a similar calculation for the output function h( ). EKF makes the simplification $E[\mathrm{fn}(x)] \approx \mathrm{fn}(E[x])$, which is not generally true, and possibly not even close to true. 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, the output-variable covariance is found by linearizing the nonlinear equations, resulting in a loss of accuracy. SPKF uses a different method to compute covariances, improving these estimates as well.

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

THE SIGMA POINT KALMAN FILTER

We have seen that the EKF approach to state estimation for nonlinear systems is to linearize the equations at each sample point using a Taylor-series expansion. Sigma-point Kalman filtering is an alternate approach to generalizing the Kalman filter to state estimation for nonlinear systems. Instead of using Taylor-series expansions to approximate the required covariance matrices, a small fixed number of function evaluations are performed instead. This has several advantages: (1) derivatives do not need to be computed (which is one of the most error-prone steps of EKF), also implying (2) the original functions do not need to be differentiable, (3) better covariance approximations are usually achieved than using EKF, allowing for better state estimation, and (4) all with comparable computational complexity to EKF. A set of points (sigma points) is chosen so that the (possibly weighted) mean and covariance of the points exactly matches the mean and covariance of the *a priori* random variable. These points are then passed through the nonlinear function, resulting in a transformed cloud of points. The *a posteriori* mean and covariance that are sought are then approximated by the mean and covariance of this cloud. Note that the sigma points comprise a fixed small number of vectors that are calculated deterministically—unlike Monte Carlo or particle filter methods.

Specifically, if an input random vector x has dimension n, mean $\bar{x}$, and covariance $\Sigma_{\tilde{x}}$, then $p+1=2n+1$ 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 $S=\sqrt{\Sigma}$ computes a result such that $\Sigma = SS^T$. Gamma is an algorithm tuning parameter that controls the a priori covariance of the sigma points. Common values are listed in Table I. Usually, the efficient Cholesky decomposition [11,12] is used, resulting in lower-triangular S (although, any other method may be used). The weighted mean and covariance of $\mathcal{X}$ agree with the original mean and covariance if we define the weighted mean as

$$\bar{x} = \sum_{i=0}^{p} \alpha_i^{(m)} \mathcal{X}_i,$$

the weighted covariance as

$$\Sigma_{\tilde{x}} = \sum_{i=0}^{p} \alpha_i^{(c)} (\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^T,$$

$\mathcal{X}_i$ as the *i*th column of $\mathcal{X}$, and both $\alpha_i^{(m)}$ and $\alpha_i^{(c)}$ as real scalars with the necessary (but not sufficient) conditions that

$$\sum_{i=0}^{p} \alpha_i^{(m)} = 1 \text{ and } \sum_{i=0}^{p} \alpha_i^{(c)} = 1.$$

The various sigma-point methods differ only in the choices taken for these weighting constants. Values for the two most common methods—the Unscented Kalman Filter (UKF) [13–18] and the Central Difference Kalman Filter (CDKF) [19–21]—are summarized in Table I. The CDKF has only one "tuning parameter" h, which makes implementation simpler. It also has marginally higher theoretic accuracy than UKF [20].

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.

TABLE I: TWO SETS OF WEIGHTING CONSTANTS.

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

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

SPKF step 1: State estimate time update

At each measurement interval, the time update is computed by first forming the augmented *a posteriori* state estimate vector for the previous time interval:

$$\hat{x}^{a,+}_{k-1} = \left[(\hat{x}^{+}_{k-1})^T, 0, 0\right]^T,$$

and the augmented *a posteriori* covariance estimate:

$$\Sigma^{a,+}_{\tilde{x},k-1} = \mathrm{diag}(\Sigma^{+}_{\tilde{x},k-1}, \Sigma_w, \Sigma_v).$$

These are used to generate the $p+1$ sigma points:

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

From the augmented set, $p+1$ vectors comprising the state portion $\mathcal{X}^{x,+}_{k-1}$ and $p+1$ vectors comprising the process-noise portion $\mathcal{X}^{w,+}_{k-1}$ are extracted. The state equation is evaluated using all pairs of $\mathcal{X}^{x,+}_{k-1,i}$ and $\mathcal{X}^{w,+}_{k-1,i}$, yielding the *a priori* sigma points

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

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

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

SPKF step 2: Error covariance time update

Using the *a priori* sigma points from step 1, the *a priori* covariance estimate is computed as

$$\Sigma^{-}_{\tilde{x},k} = \sum_{i=0}^{p} \alpha^{(c)}_i \left(\mathcal{X}^{x,-}_{k,i} - \hat{x}^{-}_{k}\right)\left(\mathcal{X}^{x,-}_{k,i} - \hat{x}^{-}_{k}\right)^T.$$

SPKF step 3: Estimate system output

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

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

The output estimate is then

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

SPKF step 4: Estimator gain matrix

To compute the estimator gain matrix, we must first compute the required covariance matrices.

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

$$\Sigma^{-}_{\tilde{x}\tilde{y},k} = \sum_{i=0}^{p} \alpha^{(c)}_i \left(\mathcal{X}^{x,-}_{k,i} - \hat{x}^{-}_{k}\right)\left(\mathcal{Y}_{k,i} - \hat{y}_k\right)^T$$

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

SPKF step 6: Error covariance measurement update

The final step is calculated directly from the optimal formulation:

$$\Sigma^{+}_{\tilde{x},k} = \Sigma^{-}_{\tilde{x},k} - L_k\Sigma_{\tilde{y},k}L^T_k.$$

ENHANCED-SELF-CORRECTING CELL MODEL

The model that we use in this paper is one that we have called the "enhanced self-correcting" (ESC) cell model [5–10]. In order to use the Kalman methods we propose to estimate SOC, the cell model must be represented in the discrete-time state-space form of (1) and (2) constraining SOC to be a member of the state vector. The difference between the models depends only on the definitions of x_k, u_k, y_k, f( ) and g( ).

The basis for the SOC state-equation is developed as follows: If $z(t) = \mathrm{SOC}$, we know that

$$z(t) = z(0) - \int_0^t \frac{\eta(i(\tau))\, i(\tau)}{C}\, d\tau, \tag{3}$$

where C is the nominal capacity of the cell, i(t) is the cell current at time t, and $\eta(i(t))$ is the Coulombic efficiency of the cell. A discrete-time approximate recurrence may then be written as

$$z_{k+1} = z_k - \frac{\eta(i_k)\, i_k\, \Delta t}{C}, \tag{4}$$

where Δt is the sampling period (in hours). Equation (4) is used to include SOC in the state vector of the cell model as it is in state equation format already, with SOC as the state and i_k as the input.

The dynamics of the change of polarization voltage are also captured by a state equation. We add "filter states" with linear dynamics:

$$[f_{k+1}] = [\mathrm{diag}(\alpha)][f_k] + i_k.$$

The vector α has N filter "poles", with $|\alpha| < 1$ for stability, corresponding to time constants of the polarization voltage dynamics. We use $N = 2$.

A further phenomenon captured by a state equation is that of hysteresis. A cell that has recently undergone a charge event will have a higher rest voltage than one that has undergone a discharge event, even at the same SOC. That is, voltage does not decay to OCV, but to OCV plus/minus a factor based on the hysteresis of the cell. We note that hysteresis is not a phenomenon generally associated with lithium-ion systems, since most applications have been in the light portable electronics area where SOC accuracy is not as critical as in the HEV application and where temperatures are not as extreme. It is, however, very pronounced at low temperatures and can lead to SOC errors as large as ±40% if the estimate is based simply on OCV (even with full cell relaxation.)

A hysteresis state implementing a linear-time-varying difference equation may be modeled as:

$$h_{k+1} = \exp\left(-\left|\frac{\eta(i_k) i_k \gamma \Delta t}{C}\right|\right) h_k + \left(1 - \exp\left(-\left|\frac{\eta(i_k) i_k \gamma \Delta t}{C}\right|\right)\right) M. \tag{5}$$

M represents the maximum hysteresis voltage at the present temperature, and γ is a hysteresis rate constant.

The three components of the system state are combined:

$$x_k = \begin{bmatrix} f_k^T & h_k & z_k \end{bmatrix}^T. \quad (6)$$

The corresponding equations for f_k, h_k, and z_k also combine to form the vector function f( ).

The cell terminal voltage is modeled by the output equation g( ). With the states of the system as defined, the ESC model computes:

$$y_k = \mathrm{OCV}(z_k) + C[f_k] - Ri_k + h_k. \quad (7)$$

The voltage is computed as the sum of the open-circuit-voltage at the present SOC, plus a weighted sum of the polarization voltage states, minus ohmic losses, plus hysteresis. A further constraint on (7) is that during a constant-current dis/charge, the polarization filter voltages must converge to zero so that $y_k \rightarrow \mathrm{OCV(SOC)} - I \times R$ (plus hysteresis) [5].

The ESC-model form is now defined. In order to implement the model for a specific cell electrochemistry, however, we require knowledge of the parameters of the model. Specifically, we must determine the OCV versus SOC relationship, the filter time constants α, the number of filter states N, hysteresis rate factors, and so forth. Details on how this has been done on the cells in question in this paper may be found in [6].

RESULTS OF TESTING

In this section we present some results from testing the SPKF on a battery pack comprising forty high-power LiPB cells described elsewhere [6]. At the time of this testing, the cells in the pack had undergone considerable abuse from previous testing, had lost about 7% of their original capacity, and had increased resistances. Nevertheless, the SPKF methods, using a model that represents new-cell dynamics, produce very good results. Note that more complete results of testing SPKF versus EKF are available in [22–23]. Our purpose here is to present some more recent testing data and to point out a few features of the algorithm.

Raw measured data for the test, captured using the "byte pipe" mechanism of the Aerovironment ABC-150 cycler, are presented in Fig. 2. All 40 cell voltages are plotted, although they are too similar to easily distinguish. Pack current and average cell temperatures for this test are also plotted. Note that the test comprised a 1C constant-current charging portion, a dynamic drive cycle portion, a rest, the drive cycle repeated, and another rest. This test is designed to validate specific aspects of the SPKF algorithm: The SOC during constant-current events should converge to a straight line; the SOC during rest events should converge to the value predicted by OCV (neglecting hysteresis, which is small at the temperature of this test); the SOC during a drive cycle should have the same pattern computed by coulomb counting.

In Fig. 3, five lines are plotted: One is the SOC as estimated in the BMS by the Kalman filters (thick black line), another is the SOC as estimated in post-processing by "C" code on a PC (red dashed line), another is SOC as estimated in post-processing using the Tino method (thin gray line), and two are Coulomb-counting methods estimated in post-processing using either the ABC-150 current-sensor log or the BMS current-sensor log (remaining thin dashed lines: black and blue, respectively). We note a few things:

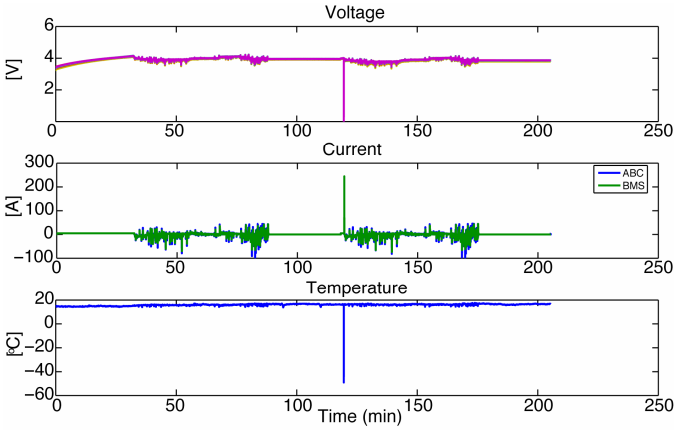

Figure 2. Measurement data from tests.

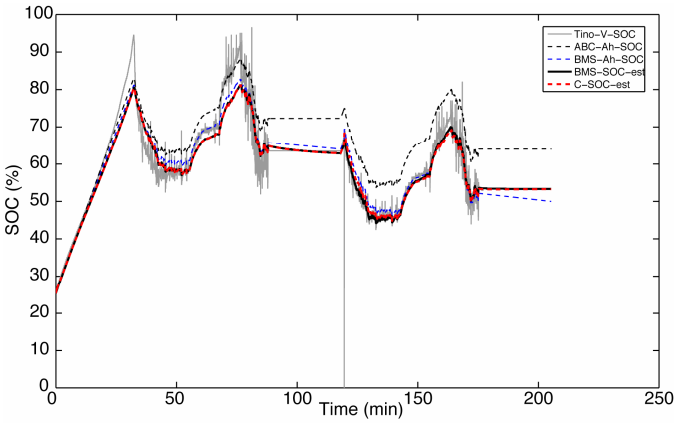

Figure 3. SOC estimation results.

- Both Coulomb-counting methods are post-processed using the recorded ABC-150 current and BMS current. A cell capacity of 4.67Ah is assumed (it is a reasonable value for this pack in its present state—it might be optimized more, but this begs the question of how we would adaptively optimize capacity without a Kalman filter). The Coulomb-counting methods are initialized using an SOC value derived from the first voltage readings, just like the Tino method.
- The post-processed "C" Kalman filter is initialized slightly differently than the BMS Kalman filter, so we expect that their initial values might be somewhat different (especially if the pack has not rested before the test began).

The BMS Kalman filter is initialized some time before the test began. In some cases it is possible that the BMS

Kalman filter was not reset/initialized right before the test, but there is no way to determine this from the recorded data. Despite different initializations, we expect the C and BMS SPKFs to converge to each other.

Note the transient at around time 120 minutes where all readings are incorrect. This is most likely caused by a glitch in the communication between the BMS and the ROS computer; that is, it is likely that the measurements were correct but were not logged correctly. Two seconds of data were lost.

It is important to note that the SPKF SOC estimator has a built-in mechanism for determining whether a measurement is statistically valid (the pre-fit residual method: if a measurement is more than six sigma from its expected value, where sigma is continuously updated depending on cell state, the measurement is considered erroneous and discarded). This mechanism kept the bad measurement from disturbing the operation of the SPKF.

All methods appear to have initialized well. The Tino SOC is bad at the transient, but the SPKF SOC estimate is very good. Notice that the bias has been correctly identified—using a method beyond the scope of this paper—even though the BMS-Ah SOC is dropping quickly at the end of the test, the BMS SPKF SOC has converged to the correct value. If one believes steady-state Tino estimates, the SOC error is less than 2% at all times.

CONCLUSION

This paper presents results of using model-based estimation, particularly the sigma-point Kalman filter, to estimate SOC for LiPB cells in an HEV battery pack. The method works very well, typically producing RMS SOC estimation error on the order of 2% for room-temperature operation. Some other features of the algorithm include: the ability to provide error bounds on the estimated quantities, and the ability to detect erroneous sensor data using the internal state and uncertainties.

REFERENCES

1. R. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME—Journal of Basic Engineering 82, Series D (1960) 35–45.
2. The Seminal Kalman Filter Paper (1960), http://www.cs.unc.edu/~welch/kalman/kalmanPaper.html, accessed 20 May 2004.
3. 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).
4. 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).
5. 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).
6. G. Plett, Results of Temperature-Dependent LiPB Cell Modeling for HEV SOC Estimation, in: CD-ROM Proceedings of the 21st Electric Vehicle Symposium (EVS21), (Monaco: April 2005).
7. G. Plett, Dual and Joint EKF for Simultaneous SOC and SOH Estimation, in: CD-ROM Proceedings of the 21st Electric Vehicle Symposium (EVS21), (Monaco: April 2005).
8. G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 1: Background, Journal of Power Sources 134 (2) (2004) 252–61.
9. G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 2: Modeling and identification, Journal of Power Sources 134 (2) (2004) 262–76.
10. G. Plett, Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs—Part 3: Parameter estimation, Journal of Power Sources 134 (2) (2004) 277–92.
11. W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C: The Art of Scientific Computing, 2nd Edition, Cambridge University Press, 1992.
12. G. Stewart, Matrix Algorithms, Vol. I: Basic Decompositions, SIAM, 1998.
13. S. Julier, J. Uhlmann, H. Durrant-Whyte, A new approach for filtering nonlinear systems, in: Proceedings of the American Control Conference, 1995, pp. 1628–32.
14. 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).
15. 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.
16. S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, Proceedings of the IEEE 92 (3) (2004) 401–22.
17. E. Wan, R. van der Merwe, The unscented Kalman filter for nonlinear estimation, in: Proc. Of IEEE Symposium 2000 (AS-SPCC), Lake Louise, Alberta, Canada, 2000.
18. 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, Ch. 7, pp. 221–82.
19. M. Nørgaard, N. Poulsen, O. Ravn, Advances in derivative-free state estimation for nonlinear systems, Technical rep. IMM-REP-1998-15, Dept. of Mathematical Modeling, Tech. Univ. of Denmark, 28 Lyngby, Denmark (April 2000).

20. M. Nørgaard, N. Poulsen, O. Ravn, New developments in state estimation for nonlinear systems, Automatica 36 (11) (2000) 1627–38.
21. 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.
22. G. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. Introduction and state estimation, Submitted to International Journal of Power Sources.
23. G. Plett, Sigma-point Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. Simultaneous state and parameter estimation, Submitted to International Journal of Power Sources.

CONTACT

Gregory Plett is Associate Professor of Electrical Engineering at the University of Colorado at Colorado Springs and consultant to Compact Power Inc. He may be reached at:

Dept. 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
Tel: +1–719–262–3468, Fax: +1–719–262–3589,
E-mail: glp@eas.uccs.edu,
URL: http://mocha-java.uccs.edu

Martin Klein is Director of Engineering at Compact Power Inc. He may be reached at:

Compact Power Inc.,
1857 Technology Drive, Troy, MI 48230 USA
Tel: +1–248–291–2379
E-mail: mklein@compactpower.com
URL: http://www.compactpower.com