Journal of Power Sources 198 (2012) 338–350

Contents lists available at SciVerse ScienceDirect

Journal of Power Sources

journal homepage: www.elsevier.com/locate/jpowsour

Battery cell state-of-charge estimation using linear parameter varying system techniques

Y. Hu [a,1], S. Yurkovich [b,*]

[a] The Ohio State University, United States
[b] University of Texas at Dallas, United States

ARTICLE INFO

Article history:
Received 13 June 2011
Received in revised form
20 September 2011
Accepted 21 September 2011
Available online 29 September 2011

Keywords:
Lithium ion batteries
SoC estimation
State estimator
Linear parameter varying systems

ABSTRACT

This paper describes a model based method for real time battery cell state-of-charge (SoC) estimation using linear parameter varying (LPV) system techniques. For this class of methods, the applicable structure is one in which the input to output dynamics of the battery can be described by a discrete parameter varying state variable model that includes the SoC as a state. Within this context, the problem of state-of-charge estimation is viewed as a state estimation problem, so that a state estimator is designed using the model. An LPV system technique, combined with input to state stability criteria, is used to analyze the stability and performance of the estimator. Compared with algorithms available in the current literature, such as those employing an extended Kalman filter and sliding mode observers, this method offers good performance with a guarantee of stability, and possesses user friendly tuning with low computational complexity for easy on-board implementation. Experimental results are given which validate the proposed methodology.

1. Introduction

As hybrid and electric vehicle technology advance further, automotive manufacturers have adopted lithium ion batteries as the electrical energy storage device of choice in current and envisioned vehicles. These high-capacity, high-power batteries provide significant improvement in terms of energy and power density when compared to NiMH and lead-acid batteries used in previous generations of plug-in hybrid electric vehicles (PHEVs), hybrid electric vehicles (HEVs) and electric vehicles (in total, these classes are referred to as P/H/EVs). However, the lithium ion chemistry is one such that even mild over-charging or over-discharging can result in catastrophic failures or premature aging [1]. Therefore, the current strategy to prolong the life of the battery pack is to use the batteries in a conservative fashion. On the other hand, utilizing the batteries to the full extent of their capabilities can provide better fuel economy, drivability and reduced cost (for example, resulting in a smaller battery pack). One solution to simultaneously maximize both battery life and still garner the highest utility from the batteries is to have an effective battery management system that can operate closer to the safety limit to maximize gains.

One of the most important and yet difficult problems in designing an effective battery management system is the estimation of the state-of-charge (SoC) of the battery pack. In previous generations of HEVs, the battery pack is operated in charge-sustaining mode, usually in a small range around 75% (often 60–85%). In such applications, the margin of error is large because the batteries are operated relatively far from their safety limits. With the advent of PHEVs, battery packs often must be used in capacity ranges down to around 25% SoC in charge-depleting mode and then operated in charge-sustaining mode around 25%. Having a more accurate SoC estimate means that even when the batteries are near the limiting SoC, the entire hybrid system can still be controlled so that performance is not compromised by the safety systems, which operate by curbing battery usage when voltage limitations are violated.

Estimating the SoC accurately in real time is a challenging problem. Formally, SoC is defined as the ratio of available ampere-hour (Ah) to the total Ah available when the battery is fully charged (namely, the capacity). When the current through the battery is measured precisely, the SoC (herein represented by the variable z) can be calculated via a Coulomb counting process in the manner

$$z(t) = z(0) - \frac{1}{K}\int_0^t I\,dt, \tag{1}$$

where K is a factor that that is inversely proportional to the capacity of the battery. However, because measurement of electrical current is always corrupted by noise, the integration operation results in a

* Corresponding author. Tel.: +1 972 883 2305; fax: +1 972 883 4653.
E-mail addresses: yiran.hu@gmail.com (Y. Hu), steve.yurkovich@utdallas.edu, yurkovich.1@osu.edu (S. Yurkovich).
[1] He is currently with General Motors Corp., United States.

0378-7753/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jpowsour.2011.09.058

Nomenclature

$\overline{\Gamma}_e$	disturbance and uncertainty in the error dynamics of the observer
$\overline{A}_e$	system matrix for the error dynamics of the observer
χ	slope of a secant line on the OCV to SoC function f
ΔT	maximum change of temperature in one sampling event
$\Delta\{A, B, C, D\}$	uncertainties in system matrices
$\hat{\cdot}$	estimated quantity from the observer – i.e. $\hat{z}$ is the estimated SoC
$\tilde{\cdot}$	error variable between estimated and true
$\{A, B, C, D\}$	state space system matrices for the dynamic voltages in the battery model
C_e	extended output system matrix
C_n	nominal capacity (Ah)
f	open circuit voltage function
I	battery current (A)
i_d	current direction (0/1 = charge/discharge)
K_e	overall observer gain
K_x	observer gain for the dynamic voltage states
K_z	observer gain for SoC state
L_f	lower bound for χ
P	positive definite matrix in the Lyapunov equation
$P_{ij} : i = \{0, 1\}, j = \{c, d\}$	components of the P corresponding to temperature and current direction
Q	positive definite matrix
T	cell temperature
T_s	sampling time (s)
u	input signal to the model (current)
U_f	upper bound for χ
V	Lyapunov function defined for the observer
V_h	hysteresis voltage
V_{oc}	open circuit voltage
w, v	input and output disturbances
X	states representing dynamic voltages in the battery model
z	state representing battery SoC (%)

drift of the estimate away from the true value over time. Additionally, because capacity of the battery degrades over time, current integration, which depends on capacity, will also become inaccurate. Alternatively, one could measure the open circuit voltage (OCV) of the battery because it has a near one-to-one correspondence with the SoC and is independent of the capacity. However, battery dynamics dictate that the OCV can only be measured after the battery has been resting for a significant period of time. Therefore, the OCV is not a useful measurement technique when the battery is under continuous operation. Furthermore, for batteries whose OCV is relatively flat with respect to SoC, this method can be difficult to apply.

Aside from direct (or indirect) measurement, the SoC can also be estimated using other techniques (for a brief summary see [2]) which improves upon the shortcomings of the measurement techniques. The study [3] proposes a fuzzy logic method where the SoC is determined using the frequency response of the battery. This, however, is not practical in an on-board setting where the controller is prohibited from injecting a sinusoidal signal into the battery pack. Similarly, [4] provides a fuzzy logic current integration method where the current integration is adapted based on the operating conditions. Although more practical than the previous method, this approach still suffers from the drawback of the integration process, where the SoC estimate experiences drift over time. Black box approaches such as those employing neural networks have also been used for SoC estimation; for example, [5] shows an estimator constructed using an artificial neural network. These methods can often produce very good results after the network is trained sufficiently. However, the training process is laborious (sometimes non-convergent), especially if the estimator is expected to function under various operating conditions. Furthermore, the black box nature makes these estimators less intuitive.

An alternative class of algorithms used for SoC estimation is one utilizing model based techniques. In these methods, control theoretic techniques are applied to a control-oriented model to estimate the SoC. Generally speaking, the term control-oriented usually refers to a class of models that are of low-order and have a low-degree of nonlinearity, yet have sufficient accuracy for the subsequent control design to produce good results with the nonlinear plant. There are several examples of these algorithms in the literature. First is the extended Kalman filter (EKF) approach [6,7], wherein a slightly nonlinear discrete state space model for the battery is identified, which includes the SoC as a state. Then an extended Kalman filter is applied to the model to estimate the total state. The primary drawback of this approach is the fact that an error covariance matrix (whose size is the order of the model) must be propagated through the system at each sampling instance to calculate the correction gain. This results in a significant number of calculations that may not be suitable for implementation. In addition, because the EKF requires linearization of the plant at each time instance around an unknown operating point (namely the SoC), convergence is not guaranteed.

Another model based approach that has shown success is the sliding mode observer approach; [8,9] show two variations. In this approach, a linear battery model, simplified as compared to the model used in [6], is used to describe the battery dynamics. Then a sliding mode observer is used to estimate the SoC. By allowing the sliding mode gain to dominate the plant uncertainty, this method is able to guarantee that the estimator has desirable convergence properties. The estimation results are good for the data used with this method. However, explicit results are not given for regions where the battery is operated at low temperatures. In our experience, because battery parameters (such as internal resistance) change significantly when temperature is lowered, even when using a sliding mode observer whose gains are tuned to dominate the maximum uncertainty, the estimation errors that result are often quite large. Furthermore, because temperature is a measured quantity, it makes sense that adapting the estimator gains with respect to temperature can improve performance.

In this paper, a robust SoC estimator is designed using a linear parameter varying (LPV) estimator. The model used to describe the battery dynamics is a discrete LPV state variable model which can be obtained via methods discussed in previous work [10–12]. The parameters on which the LPV model operates are SoC and temperature; such parametric variation built into the model improves the accuracy over various operating conditions to reduce the error in the estimation, especially at lower temperatures. Because the SoC is a parameter of the plant and unknown, this structure treats the plant as uncertain; as such, the resulting estimator must be robust to this uncertainty. In addition, the effect of the uncertainty can be analyzed explicitly using input to state stability criterion [13–16]. There are several advantages to this design. First the computational complexity is much less than the EKF approach because no error covariance matrix is propagated forward. Second, stability of the estimator can be guaranteed while the properties of the estimation error can be analyzed explicitly. After the models are developed and discussed, the estimator design is demonstrated using experimental data collected from two different batteries.

2. Battery model

In this section, the model of the battery is described. Because the final algorithm is intended for on-board implementation, the battery model is assumed to be in a discrete state variable form given as

$$\begin{bmatrix} z[k+1] \\ X[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & A(T, i_d) \end{bmatrix} \begin{bmatrix} z[k] \\ X[k] \end{bmatrix} + \begin{bmatrix} \frac{-T_s}{36C_n} \\ B(T, z, i_d) \end{bmatrix} (u[k] + w[k])$$
$$y[k] = f(T, z) + CX[k] + D(T, z, i_d)u[k] + v[k], \tag{2}$$

where T is the temperature, z is the SoC, i_d is the current direction, T_s is the sampling time, C_n is the nominal capacity, $X \in n\Re$ is the state vector representing the battery dynamics, f is the open circuit voltage as a function of T and z, and w and v are zero mean Gaussian noise disturbances that affect the input and output, respectively. In addition, A, B, C, D are the state variable system matrices of appropriate dimensions for the dynamic state X. Note here that the matrix A is assumed to be independent of z (the SoC). Previous modeling results [17] have shown that the dependence of A on z has only a very minor effect on the accuracy of the model. In addition, because z is a quantity to be estimated, it must be considered unknown. As such, including this term does not improve the accuracy of the model.

This particular battery model structure represents a common approach in approximating the battery dynamics. It is based on the physical intuition that the output voltage of a battery is composed of an OCV, a set of dynamic voltages that are the results of electrochemical effects like charge transferring and diffusion, and a voltage resulting from the internal resistance. The combination of the dynamic voltages and the voltage corresponding to the internal resistance is commonly referred to as the overvoltage of the battery. An additional component often included in battery models of this type is the hysteresis voltage, which describes the fact that the rested battery OCV at a given SoC and temperature can be different depending on the previous excitation. The dynamic equation most commonly used to describe this phenomenon (see [6,18] for example) is a first order dynamic equation in the SoC. For instance, the continuous time equation can be written as

$$\dot{V}_h = \Gamma|I|(H(\dot{z}, T) - V_h), \tag{3}$$

where V_h is the hysteresis voltage, $\Gamma > 0$ provides the time constant, and H is a function that provides the maximum hysteresis voltage for a given SoC and temperature. Examining this equation reveals that if the current input is zero, then the equation reduces to

$$\dot{V}_h = 0. \tag{4}$$

The discrete form of this equation is

$$V_h[k+1] = V_h[k], \tag{5}$$

which is precisely the same form as the dynamic equation for z (and consequently for the OCV) when the current is zero. Because the battery terminal voltage reflects the combined effect of the OCV and the hysteresis, one cannot distinguish between the effects of the OCV and hysteresis based on the output voltage when the current is zero. In other words, if the hysteresis effect is included in the model, then under zero current conditions, the system is not observable from the output. As such, a hysteresis element cannot be included in a state estimator design, and is therefore not included in the remainder of this paper. Note that this does not mean that the hysteresis effect must be ignored altogether. If a hysteresis model is available in real time, one can simply remove the hysteresis voltage computed by this model from the battery voltage measurement prior to applying the state estimator. In so doing, the influence of the hysteresis can be removed from the estimated states.

One aspect of the model that is not addressed in this paper is the model adaptation with respect to aging. The main reason for omitting this is the lack of experimental aging data at the present time. The problem of modeling parameter variation as a result of aging is a topic that will be studied as part of the future work. One possible approach, such as already proposed in previous works involving the extended Kalman filter [6], is to augment the model states with critical parameters such as internal resistance and capacity. The estimator can then be designed to estimate these additional states as well as the SoC. The overall system is capable of adapting to the aging of the battery.

The model described here can be identified in two ways. The first method is to identify a parameterized equivalent model such as done in [10]. In this case, typically an equivalent circuit model consisting of an OCV, an internal resistance, and multiple pairs of parallel RC circuits is used. The number of RC circuit pairs depends on how accurately one wants to model the battery. Typically, a second or third order model is sufficient for capturing the basic charge transfer and diffusion dynamics in the frequency range of 10 mHz–10 Hz. Even though the slower diffusion dynamics can reach as low as 10 μHz, these dynamics are difficult to model for this type of application, where the model structure must be appropriate for control design. For instance, often utilized elements such as the Warburg impedance or transmission line cannot be used here without a meaningful time domain representation. However, the effect of these unmodeled dynamics and uncertainty can be mitigated by applying control theory. Also note that the optimal model order may be slightly different depending on temperature. However, having different model order for different temperatures makes estimator design very difficult. Therefore it is assumed that model order is constant for all temperatures. The model constructed this way can be represented by a continuous-time differential equation, which can be discretized (see Appendix A for more information on this). Another method is to directly identify the model using a black box identification technique (one example can be seen in [17], where a subspace method is used to perform the identification). The model that results is very similar to the discretized equivalent circuit model. Once again, lower order models (such as second or third order) are typically used to capture the basic charge transfer and diffusion dynamics. For more information on the actual identification, interested readers should refer to the aforementioned references.

3. Estimator design

In this section, the design of the SoC estimator is described. The basic structure of this estimator is a state observer that uses the voltage, temperature, and current measurements as feedback signals. Because the SoC is a state of the model, if the states of the observer converge to the true states, SoC estimation is achieved.

The generic form of the state observer design is given as

$$\begin{bmatrix} \hat{z}[k+1] \\ \hat{X}[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & A(T, i_d) \end{bmatrix} \begin{bmatrix} \hat{z}[k] \\ \hat{X}[k] \end{bmatrix} + \begin{bmatrix} \frac{-T_s}{36C_n} \\ B(T, \hat{z}, i_d) \end{bmatrix} u[k] + K_e(y[k] - \hat{y}[k])$$
$$\hat{y}[k] = f(T, \hat{z}) + C\hat{X}[k] + D(T, \hat{z}, i_d)u[k]. \tag{6}$$

Note that this is precisely the form of a model with output feedback for correction. The exact form of the output feedback K_e will be discussed later, except to note that it is a *function* of the difference between the estimator output and the measured voltage output. Because w and v are unknown, they are not included in the estimator. For estimator coefficients that depend on the SoC, the estimated SoC $\hat{z}$ is used in place of the true SoC z.

The output error $\tilde{y}[k] = y[k] - \hat{y}[k]$ can be computed as

$$\tilde{y}[k] = f(T, z) - f(T, \hat{z}) + C\tilde{X}[k] + \Delta D[k]u[k] + v[k],$$

where $\tilde{X}[k] = X[k] - \hat{X}[k]$ and $\Delta D[k] = D(T, z, i_d) - D(T, \hat{z}, i_d)$. The estimation error dynamics can then be computed as

$$\begin{bmatrix} \tilde{z}[k+1] \\ \tilde{X}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & A(T, i_d) \end{bmatrix} \begin{bmatrix} \tilde{z}[k] \\ \tilde{X}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta B[k] \end{bmatrix} u[k] + \begin{bmatrix} \frac{-T_s}{36C_n} \\ B(T, z, i_d) \end{bmatrix} w[k] - K_e(f(T, z) - f(T, \hat{z}) + C\tilde{X}[k] + \Delta D[k]u[k] + v[k]). \quad (7)$$

Next, define

$$\chi[k] = \frac{f(T[k], z[k]) - f(T[k], \hat{z}[k])}{z[k] - \hat{z}[k]}. \quad (8)$$

For a given T, the function f must satisfy a basic property of the OCV function in that it is monotonically increasing with respect to SoC. Because χ represents the slope of a secant line on the graph of f and z, the monotonicity of f implies that χ is always positive. Because the domain of T and z are both compact, $\exists L_f, U_f > 0$ such that

$$0 < L_f \le \chi[k] \le U_f, \quad (9)$$

$\forall k \in \mathbb{N}$. Note here that L_f and U_f can also be functions scheduled on temperature. However, because the *shape* of the OCV functions is likely to be similar even for different temperatures, χ is primarily a function of the SoC rather than temperature. Therefore the variation of L_f and U_f with respect to the temperature will be small, and thus for the sake of simplicity, L_f and U_f are taken to be constants.

Given this,

$$f(T, z) - f(T, \hat{z}) + C\tilde{X}[k] = [\chi[k]\, C] \begin{bmatrix} \tilde{z}[k] \\ \tilde{X}[k] \end{bmatrix} \triangleq C_e[k]\tilde{X}_e[k].$$

In other words, the output error can be written as a linear function of the estimation error, with the exception that part of the linear coefficient, namely χ, is time varying and unknown (since it involves z which is the quantity to be estimated). This suggests that the estimator to be designed can be considered a linear state estimator that is robust to variations in χ. In other words, one can select K_e as a linear function of the output error and design it in such a way that the estimator converges regardless of the true values of χ. Thus, K_e can be written as

$$K_e = \begin{bmatrix} K_z(T, i_d) \\ K_x(T, i_d) \end{bmatrix}, \quad (10)$$

where the components K_z and K_x could depend on T and i_d.

The error dynamics can now be written as

$$\begin{bmatrix} \tilde{z}[k+1] \\ \tilde{X}[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & A(T, i_d) \end{bmatrix} \begin{bmatrix} \tilde{z}[k] \\ \tilde{X}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta B[k] \end{bmatrix} u[k] + \begin{bmatrix} \frac{-T_s}{36C_n} \\ B(T, z, i_d) \end{bmatrix} w[k] - \begin{bmatrix} K_z \\ K_x \end{bmatrix} \left([\chi[k]\, C] \begin{bmatrix} \tilde{z}[k] \\ \tilde{X}[k] \end{bmatrix} + \Delta D[k]u[k] + v[k] \right) = \bar{A}_e[k] \begin{bmatrix} \tilde{z}[k] \\ \tilde{X}[k] \end{bmatrix} + \bar{\Gamma}_e[k], \quad (11)$$

where

$$\bar{A}_e[k] = \begin{bmatrix} 1 - K_z\chi & -K_zC \\ -K_x\chi & A(T, i_d) - K_xC \end{bmatrix}, \quad (12)$$

$$\bar{\Gamma}_e[k] = \begin{bmatrix} 0 \\ \Delta B[k] \end{bmatrix} u[k] + \begin{bmatrix} \frac{-T_s}{36C_n} \\ B(T, z, i_d) \end{bmatrix} w[k] - \begin{bmatrix} K_z \\ K_x \end{bmatrix} (\Delta D[k]u[k] + v[k]). \quad (13)$$

Note that $\bar{A}_e$ determines the dynamics of the error and $\bar{\Gamma}_e$ represents the disturbance or uncertainty that ultimately determines the size of the estimation error. Further note that for brevity in notation, the dependence of K_z and K_x on T and i_d is not expressed in (11), even though K_z and K_x are assumed to be functions of these scheduling variables.

Heuristically, several observations can be made immediately. For linear systems, if the unforced system (namely $\bar{\Gamma}_e = 0$) has asymptotically stable dynamics, then with bounded input, the state response will still be bounded (input to state stability). Therefore, the purpose of K_e is to make the unforced system asymptotically stable. For a parameter varying system, this is not as simple. But because the parameter variation with respect to T is very slow, the idea of gain scheduling suggests that placing the eigenvalues of this matrix inside the unit circle will likely stabilize the system. Along the diagonal of $\bar{A}_e$, there are two portions: $1 - K_z\chi$ and $A(T, z, i_d) - \Delta A[k] - K_xC$. If $|K_x\chi| \ll 1$ and $K_z > 0$, then the eigenvalues of $\bar{A}_e$ will be very close to $1 - K_z\chi$ and the eigenvalues of $A(T, z, i_d) - \Delta A[k] - K_xC$. Because A has stable dynamics, the function of K_z is to provide a greater margin of stability and counter effects of unmodeled dynamics. If this is accomplished, then the system will be stable. To ensure that the estimation error is as small as possible, $\bar{\Gamma}_e$ must be as small as possible. With the exception of the last term of $\bar{\Gamma}_e$, the other terms are the result of model uncertainty and noise, which cannot be influenced once the model is given. The last term in (13) contains uncertainty and disturbance multiplied by K_e (in view of (10)). This suggests that if K_e had relatively small magnitude, greater robustness to uncertainty could be expected. However, because K_e is also used to place the poles, this also means that a faster closed loop response would come at the cost of higher sensitivity to uncertainty and disturbance.

Theses arguments can be formalized via Lyapunov theory. Define a parameter dependent Lyapunov function V by

$$V[k] = \tilde{X}_e^T[k]P(T, i_d)\tilde{X}_e[k], \quad (14)$$

where $P \in n+1 \times n+1\Re$ is a positive definite matrix. Here, P is assumed to depend on T and i_d, even though if a choice of P that is independent of the parameters could be found, then it can and should be used.

For a given V, we have

$$V[k+1] - V[k] = \tilde{X}_e^T[k](\bar{A}_e^T[k]P[k+1]\bar{A}_e[k] - P[k])\tilde{X}_e[k] + 2\bar{\Gamma}_e^T[k]P[k+1]\bar{A}_e[k]\tilde{X}_e[k] + \bar{\Gamma}_e^T[k]P[k+1]\bar{\Gamma}_e[k].$$

Suppose that for some $P(\cdot)$, $Q > 0$, $\bar{A}_e^T[k]P[k+1]\bar{A}_e[k] - P[k] < -Q$, $\forall k$. Note here that Q can be assumed to be independent of the scheduling parameters because if a parameter dependent Q exists, then because the domain of the parameters is compact, a non-parameter dependent Q can always be found to bound the parameter dependent Q. It follows that

$$V[k+1] - V[k] = \tilde{X}_e^T[k](-Q)\tilde{X}_e[k] + 2\bar{\Gamma}_e^T[k]P[k+1]\bar{A}_e[k]\tilde{X}_e[k] + \bar{\Gamma}_e^T[k]P[k+1]\bar{\Gamma}_e[k]. \quad (15)$$

Because the quadratic term on the right-hand-side of (15) dominates for large $\tilde{X}_e$, $\exists m > 0$ such that if $|\tilde{X}_e[k]| > m$, then $V[k+1] - V[k] < 0$. Therefore, $\tilde{X}_e$ will converge to a neighborhood around 0. The size of this neighborhood depends on the magnitude of the remaining terms on the right-hand-side of (15), which

is directly dependent on the magnitude of $\overline{\Gamma}_e$. This analysis confirms analytically the reasoning behind the previous argument on the magnitude of $\overline{\Gamma}_e$.

This analysis also suggests that K_e should be selected so that the LMI $\overline{A}_e^T[k]P[k+1]\overline{A}_e[k] - P[k] < -Q$ has solutions $P(\cdot)$, $Q>0$. This can be done using an LMI solver that simultaneously solves for all three variables. However, such a process is generally very limiting because K_e and P must be in a specific form for the LMI solver to be applicable. Furthermore, because K_e also affects the performance of the estimator in terms of convergence and disturbance rejection, the solution that an LMI solver finds may not be desirable. A practical solution is to use the idea of gain scheduling to find K_e and then confirm the stability of the resulting solution by checking the LMI given previously. The strategy is to hold T constant and solve for K_e and then schedule K_e with respect to T. This strategy makes sense because the temperature process is slower than the electrical process. Therefore the design problem reduces to that of selecting K_e for various fixed temperatures and then interpolating the results with respect to temperature.

When designing K_e for a fixed temperature, the properties of χ can be used to significantly simplify the design problem. As discussed previously, the eigenvalues of $\overline{A}_e$ should be very close to $1-K_z\chi$ and the eigenvalues of $A(T, i_d)-K_xC$. The eigenvalue $1-K_z\chi$, which corresponds to the state z, determines the convergence properties of the SoC component of the observer. Because $\chi>0$, $K_z>0$ is used to bring this eigenvalue inside the unit circle. The further inside the unit circle this eigenvalue is, the better convergence property one has for the SoC. Because χ is generally very small and K_z should also be small to reduce the effect of the disturbance, $1-K_z\chi$ is always positive and is generally very close to the unit circle. Given this interpretation, it is easy to see that for all the values that χ can take, $\chi=L_f$ results in the worst performance for this design problem. Therefore when designing K_e for a fixed temperature, the problem can be simplified by using $\chi=L_f$. If the performance of the resulting estimator is good under this construction, it will be even better for other values of χ.

The overall design process for fixed temperature reduces to placing the closed loop poles of the estimator at a set of values that provide a good compromise between convergence speed and disturbance rejection. Because disturbance rejection implies smaller magnitude for the components of K_e, the poles selected should be very close to the original poles. This is particularly true for the state z since the small magnitude of χ requires a very large K_z to result in a pole far away from the unit circle. Therefore it is usually advisable to place the pole corresponding to z slightly inside the unit circle.

After the pole placement is done satisfactorily, the LMI condition resulting from the Lyapunov analysis must be checked to verify that the closed loop system is stable. The LMI in compact notation can be written as

$$\begin{bmatrix} \overline{A}_e^T[k]P[k+1]\overline{A}_e[k] - P[k] + Q & 0 & 0 \\ 0 & -P[k] & 0 \\ 0 & 0 & -Q \end{bmatrix} \le 0. \tag{16}$$

This form is difficult to use because two time indices occur. Therefore we make the simplification that P takes a linear functional form:

$$P(T[k], i_d[k]) = \begin{cases} P_{0c} + T[k]P_{1c} & \text{if charging} \\ P_{0d} + T[k]P_{1d} & \text{if discharging} \end{cases}. \tag{17}$$

Then it is sufficient to check the LMI

$$\begin{bmatrix} \overline{A}_e^T P(T, i_d)\overline{A}_e[k] - P(T, \{c, d\}) \pm \Delta T P_{1\{c,d\}} + Q & 0 & 0 \\ 0 & -P(T, i_d) & 0 \\ 0 & 0 & -Q \end{bmatrix} \le 0, \tag{18}$$

where ΔT is the maximum amount of temperature change that can occur within one sampling period. The idea of (18) is that given $P(T[k+1], i_d[k+1])$, $P(T[k], i_d[k])$ must be inside the range $P(T[k+1] \pm \Delta T, \{c, d\})$. However, because the dependence on T is linear, it is only necessary to check the corner points of this range. In particular, these are precisely the four points $P(T[k+1]+\Delta T, c)$, $P(T[k+1]+\Delta T, d)$, $P(T[k+1]-\Delta T, c)$, and $P(T[k+1]-\Delta T, d)$. Therefore, instead of checking for all possible $P[k+1]$, $P[k]$, it is sufficient to only check these four possibilities. Following this approach, even though the number of LMIs increases, the problem becomes much more solvable.

3.1. A different look at SoC estimation

The SoC estimator design highlighted the two most important factors that influence the design of any voltage-based SoC estimator: the accuracy of the OCV function and the small slope of the OCV function. A natural question to ask is whether or not we can estimate the OCV instead of SoC to avoid dealing directly with the accuracy of the OCV function. Indeed a similar state estimator construction can be used to estimate the OCV based on the model. To do this, we first perform a change of coordinates to replace the state z with a state that represents the OCV. Recall that the OCV is related to the SoC and temperature via the function f. Let V_{oc} represent the OCV. Then

$$V_{oc}[k+1] \approx V_{oc}[k] - \frac{\delta f}{\delta z}(T[k], z[k])\frac{T_s}{3600C_n}I[k], \tag{19}$$

using a first order Taylor expansion under the assumption that ΔSoC and ΔT are both small over each sampling period. Given (19), the dynamic equation of the system can be written as

$$\begin{aligned} \begin{bmatrix} V_{oc}[k+1] \\ X[k] \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & A(T, z, i_d) \end{bmatrix} \begin{bmatrix} V_{oc}[k] \\ X[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\delta f}{\delta z}(T[k], z[k])\frac{-T_s}{36C_n} \\ B(T, z, i_d) \end{bmatrix} (u[k] + w[k]) \\ y[k] &= [1 \;\; C]X_e[k] + D(T, z, i_d)u[k] + v[k]. \end{aligned} \tag{20}$$

The effect of this nonlinear change of coordinates is that the nonlinear function f is moved from the output equation to the input coefficient. Using this form, we can estimate the OCV using a state estimator given as

$$\begin{aligned} \begin{bmatrix} \hat{V}_{oc}[k+1] \\ \hat{X}[k] \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & A(T, \hat{z}, i_d) \end{bmatrix} \begin{bmatrix} \hat{V}_{oc}[k] \\ \hat{X}[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} \frac{\delta f}{\delta z}(T[k], \hat{z}[k])\frac{-T_s}{36C_n} \\ B(T, \hat{z}, i_d) \end{bmatrix} u[k] + K_e(T, i_d)(y[k] - \hat{y}[k]) \\ \hat{y}[k] &= [1 \quad C]\hat{X}_e[k] + D(T, \hat{z}, i_d)u[k]. \end{aligned} \tag{21}$$

Note that in this equation, $\hat{z}$ is an estimated SoC computed using the estimated OCV and the inverse of the OCV function f. Define

$$\overline{A}_e[k] = \begin{bmatrix} 1 - K_z & -K_zC \\ -K_x & A(T, z, i_d) - \Delta A[k] - K_xC \end{bmatrix}. \tag{22}$$

Note this is very similar to (12) with the exception that $\chi[k]$ has been replaced by 1. Just like $\overline{A}_e$ in (12), this matrix determines the stability of the closed loop estimator. Consequently, the same analysis used previously can be used to conclude that if there exists symmetric P and Q such that

$$\overline{A}_e^T[k]P[k+1]\overline{A}_e[k] - P[k] < -Q, \tag{23}$$

$\forall k$, then the error dynamics will be input-to-state stable. What is different about this linear matrix inequality is that it does not

depend on χ. In fact, if we assume that A_e does not depend on z, then there is no uncertainty in this matrix inequality. Consequently, the pole placement approach discussed previously can be applied very easily. After placing the poles, the previous matrix inequality can be used to ensure that the parameter variation of A_e does not cause the system to go unstable.

The primary difference between this estimator and the previous estimator used for SoC estimation is that the feedback is not affected by the quantity χ. Consequently, the pole placement is not as restrictive as is the case for the SoC estimation. Thus a natural question to ask is why not just estimate the OCV and then use the OCV to calculate the SoC based on the inverse of f? The answer is that while this is certainly valid, the problem of the small slope of f cannot be avoided altogether. When we calculate the SoC using the OCV, the small errors in the estimated OCV become magnified by the inverse function of f. In addition, because the derivative of f, which depends on the estimated SoC, is used as the input coefficient in the OCV equation, the uncertainty in the true SoC has the same effect as an input disturbance. Therefore, in either case, the estimation result will depend on the accuracy of the OCV function as well as the nonlinearity of df/dz.

This alternative estimation scheme does offer an advantage when compared to the direct SoC estimation scheme. When estimating the SoC directly, we must be concerned about the estimate exceeding the physical limits (over 100% or below 0%). Therefore in practical implementations, a saturation function must be used to limit the resulting SoC. Because this saturation is performed directly on a dynamic state, the stability properties can be altered. However, if the OCV is estimated, then we are not concerned with setting an artificial limit on the resulting estimate because the measurement has the same units. A negative voltage, for example, will only increase the feedback error which would prompt the estimator to correct the estimation accordingly.

3.2. Comments

In the above, the design of the SoC estimator (as well as an OCV estimator) is described in mathematical detail. Some important points that might otherwise be lost among the equations should be re-emphasized. First, the analysis shows that for any voltage-based SoC estimator, the quantity that ultimately determines how well the estimator will work is L_u, the smallest slope of the OCV function. Larger L_u will make the SoC more sensitive to voltage variation, which allows us to obtain more from the voltage measurement signal. Second, there is always a conflict between convergence speed and tracking performance. Increased convergence speed will inevitably decrease tracking robustness, while increased tracking robustness will result in a slower estimator. Hence the designer must decide which attribute is more important. Furthermore, this conflict also couples with the problems that arise when L_u is small. In such cases, a faster convergence requirement can significantly degrade the tracking robustness. Design examples are provided in the next section to illustrate these issues more clearly.

4. Design examples

In this section, the design algorithms described previously are illustrated using data collected from two separate batteries. Due to chemistry differences, the estimator behavior is very different in these two cases. The differences highlight the design tradeoff described in the previous section.

4.1. EIG battery

In this section, a constant temperature design example is provided using data collected from an EIG battery that has a capacity of 20 Ah and nominal voltage of 3.6 V. This battery uses a Li[NiCoMn]O_2 based cathode and a graphite-based anode. The model of the battery is identified using the data collected from the battery operating under an asymmetrical step profile. This model includes both an OCV map as a function of the SoC and the dynamic system matrices required by (2). The OCV is measured by repeatedly discharging the battery by 10% and then resting for 1 h to record the OCV for that SoC. The SoC of the battery for the current profile is computed by integrating a post-processed current (noise and sensor offsets are removed).

The A and C matrices for the model equation (2) are given as

$$A = \begin{bmatrix} 0.9877 & 0 \\ 0 & 0.7954 \end{bmatrix} \quad C = [-1 \ -1]. \tag{24}$$

The SoC equation for the system (accounting for the capacity and the sampling time) is given by

$$z[k+1] = z[k] - \frac{0.5}{20 \times 36} u[k]. \tag{25}$$

Figs. 2–4 show the input coefficients b_1, b_2 and the internal resistance D as functions of SoC and current direction, respectively. Fig. 5 plots the open circuit voltage as a function of the SoC.

Given these coefficients, a SoC estimator can be designed using the method described in the previous section. First evaluate the derivative of the OCV function f, shown in Fig. 6. As we can see from this plot, this derivative is bounded between 0.004 V/% and 0.011 V/%. Therefore K_e can be designed for $\chi = 0.004$. In other words, $C_e = [0.004, C]$.

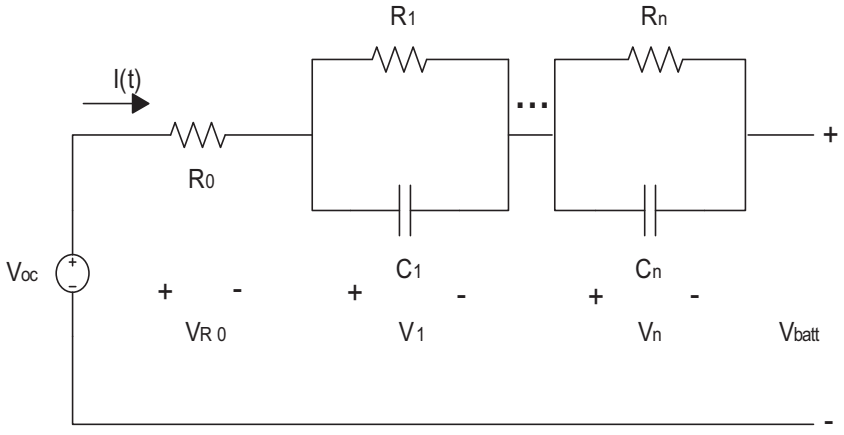

Fig. 1. Equivalent circuit used for battery model.

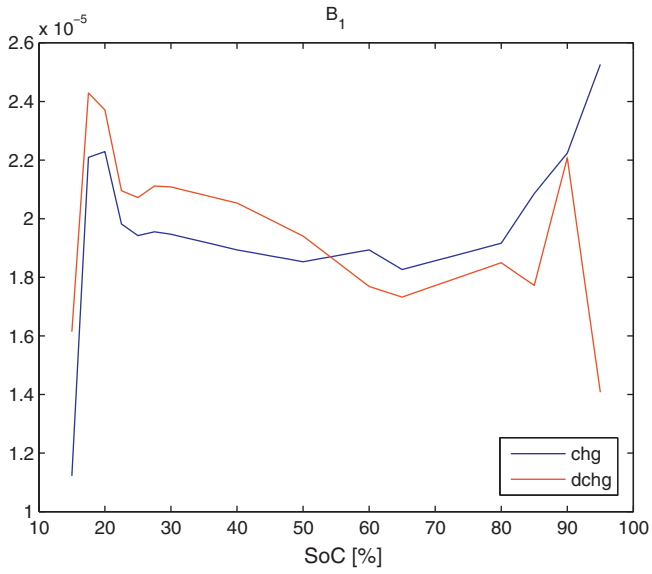

Fig. 2. B_1 coefficient.

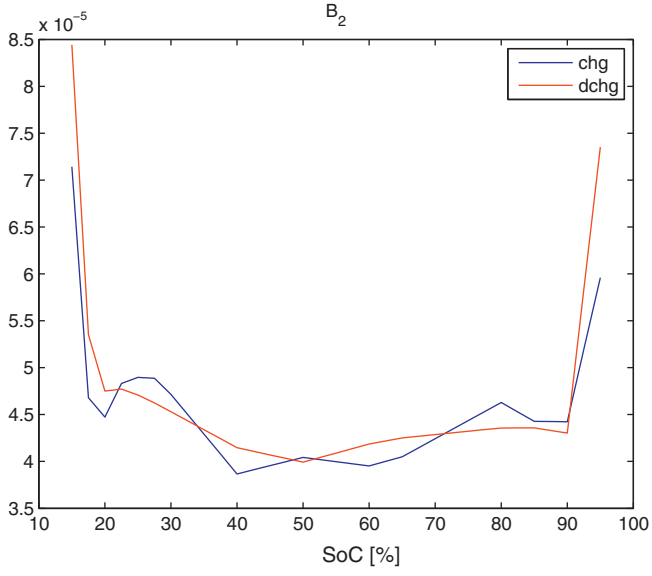

Fig. 3. B_2 coefficient.

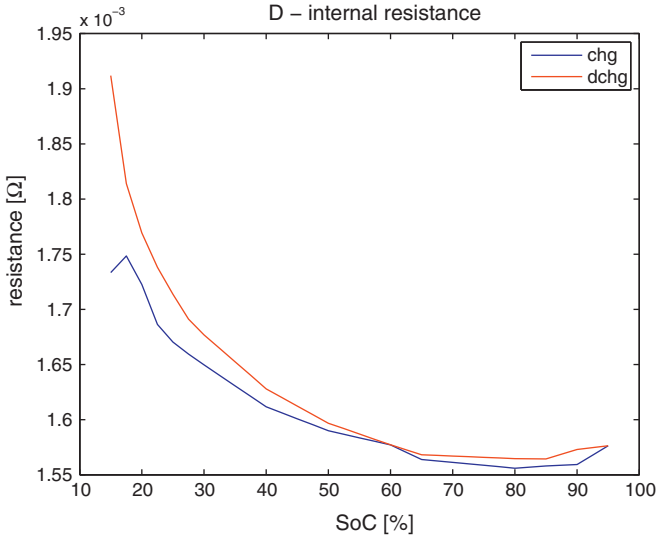

Fig. 4. D coefficient.

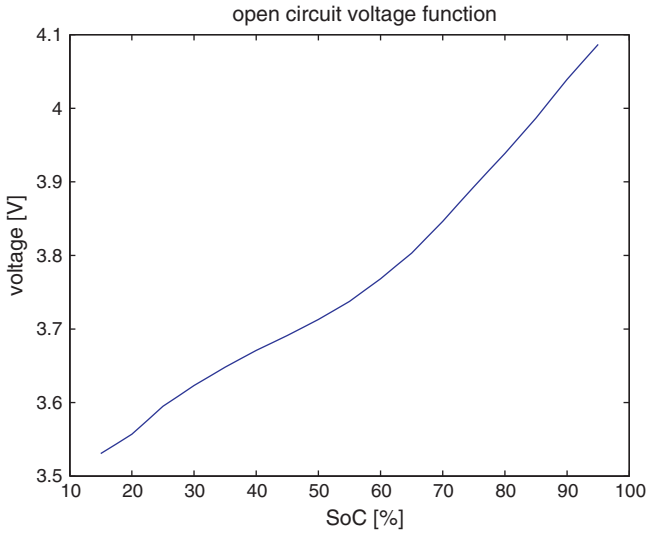

Fig. 5. Open circuit voltage.

The next step is to select the pole locations. Aside from the integrator pole in the state z (which is on the unit circle), the other two poles are quite fast already. For example, the pole at 0.9877 will cause an initial condition on x_1 to reduce to less than 3% of its value in 300 samples (150 s), which is quite fast for vehicle applications. Therefore there is no need to select poles for these two states to be much faster than their current values. Moreover, the penalty of having faster poles is higher gains, which causes the system to be more reactive to measurement errors and noise. Consequently, we select [0.96, 0.72] as the pole locations.

The main question for the design is where to put the pole corresponding to the SoC. Clearly, the pole must be moved inside the unit circle. The effect of the pole location can be seen using a simulation. First, artificial Gaussian zero mean random noise is added to the current and voltage measurement. Then additional pulses of various magnitude and time duration are added to the current measurement. This simulates non-zero-mean measurement errors that can mislead a simple current integrator. Fig. 7 shows an example of the input disturbance. Note that the size of the disturbance is quite large so that its effect is very significant on the system.

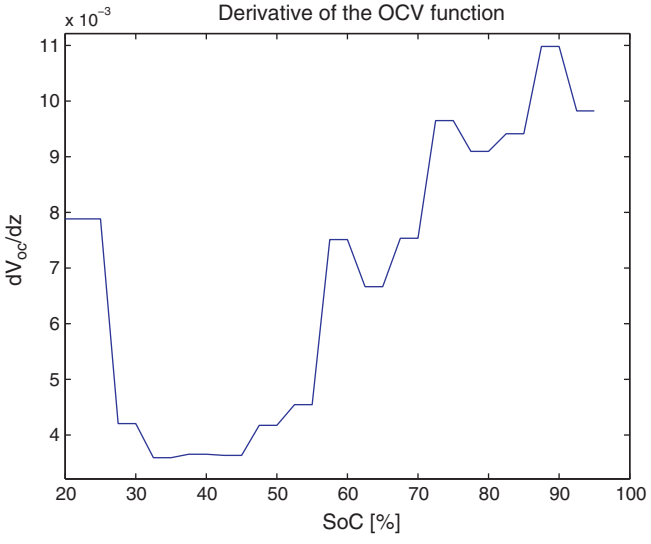

Fig. 6. Derivative of the OCV function.

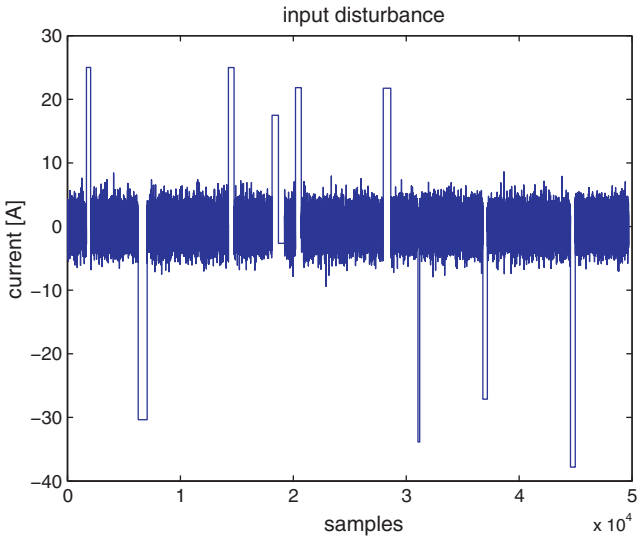

Fig. 7. Disturbance added to the input current.

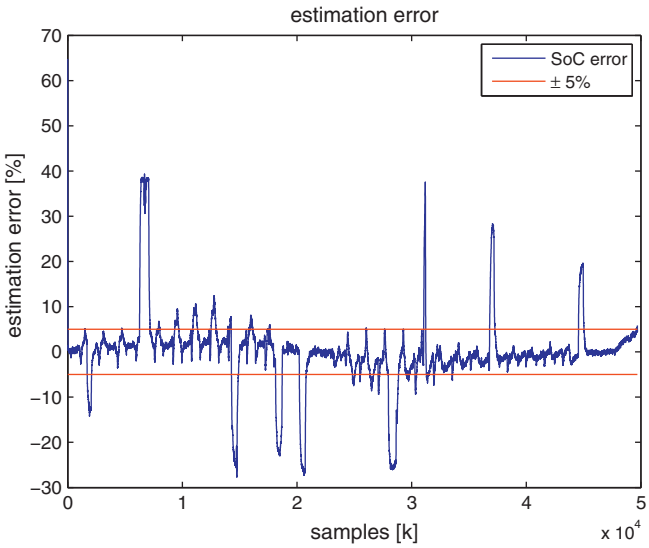

Fig. 9. Pole at 0.99 causes high sensitivity to input/output disturbance.

Placing a pole at 0.99 results in $k_z = 8.8740$. The result of this feedback gain is that the estimator converges very quickly from an initial error, evident in Fig. 8 where the estimator is used to correct the SoC trajectory that was initialized incorrectly. After only 20 samples (10 s), the estimator is within 5% of the true SoC. However, the cost of this is that the input and measurement noise can often cause the estimation error to be quite large, which is seen in Fig. 9.

Next a much slower pole of 0.9991 is used, which results in $k_z = 0.8945$. As expected, the convergence of the estimator is much slower. As shown in Fig. 10, the estimator converges from a very large initial error to within 5% of the true SoC in 1500 samples, or approximately 750 s. However, as seen in Fig. 11, the disturbance now has a much smaller effect on the estimation error.

Note that stable experimental performance does not necessarily indicate stability under all operating conditions. Therefore, it must be confirmed that the designs are stable for all other values of χ. To do this, P and Q must be found such that (18) holds for all $\chi \in [L_f, U_f]$.

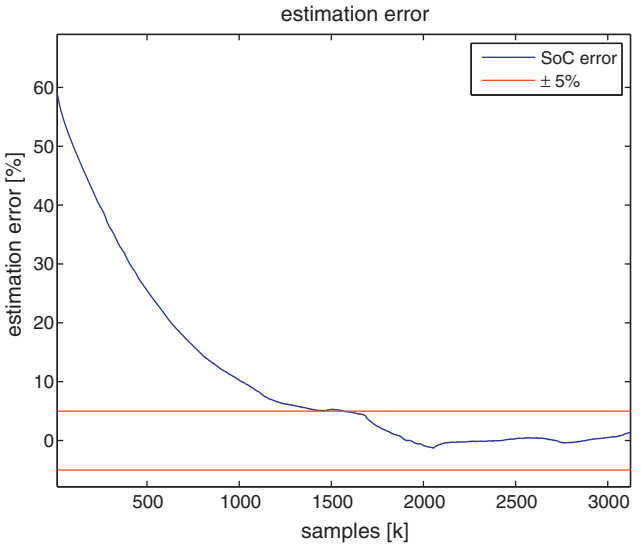

Fig. 10. Pole at 0.9991 results in a much slower convergence.

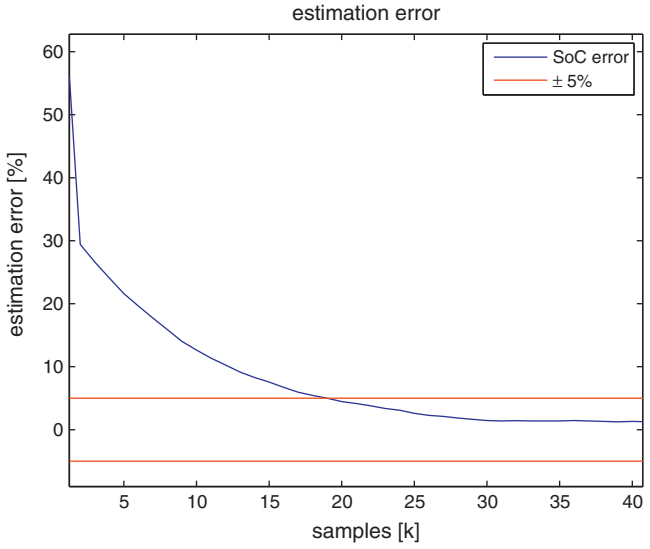

Fig. 8. Pole at 0.99 results in quick convergence.

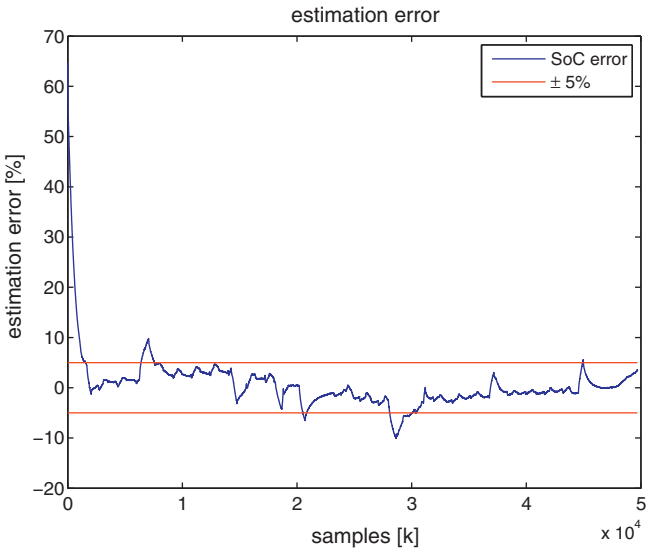

Fig. 11. Pole at 0.9991 provides the ability to mitigate input/output disturbance.

To this end, the Matlab LMI toolbox is used to find feasible solutions for P and Q. For the pole at 0.99, the following P and Q result:

$$P = \begin{bmatrix} 0.1 & -7.5 & 2.1 \\ -7.5 & 5512.8 & -69.3 \\ 2.1 & -69.3 & 2080.1 \end{bmatrix} \quad Q = \begin{bmatrix} 0.0041 & -0.6407 & -0.1400 \\ -0.6407 & 125.8615 & 23.2645 \\ -0.1400 & 23.2645 & 480.5619 \end{bmatrix}.$$

Similarly, for the pole at 0.9991, we have

$$P = \begin{bmatrix} 7.5 & 136.2 & 9.2 \\ 136.2 & 7020.4 & -91.1 \\ 9.2 & -91.1 & 2555.4 \end{bmatrix} \quad Q = \begin{bmatrix} 0.0163 & -0.9209 & -0.1554 \\ -0.9209 & 140.2092 & 21.0865 \\ -0.1554 & 21.0865 & 604.1492 \end{bmatrix}.$$

The fact that P and Q exist (in both cases) concludes that the estimator is stable using either set of gains.

Given that both designs are stable, it is up to the user to select the design that best satisfies the design objectives. For vehicle applications, we note that very large SoC error is rare. Therefore, there is very little need for fast convergence, and robustness to sensor noise or offset is much more important. Consequently, the robust performance offered by the second design, albeit with slower convergence, is a better solution.

4.2. A123 26650 cells

In this section, a multi-temperature design example is provided using data gathered from an A123 26650 lithium ion iron-phosphate battery, which has a $LiFePO_4$-based cathode and graphite-based anode. This particular battery has a capacity of 2.15 Ah and a nominal voltage of 3.3 V. For this battery, a model is identified for operating temperatures between −5 °C and 45 °C and for SoC between 10% and 90%. Once again the model contains a measured OCV–SoC function. One simplification that is made is that the OCV is selected to be independent of temperature. This is due to lack of data points in lower SoC regions for lower temperature data. This simplification necessarily introduces estimation errors for lower temperature data. Nevertheless, when a more precise OCV function becomes available, it can simply be used in place of this simplified OCV function to yield better results.

The datasets used with this battery are asymmetrical step profiles, containing current steps of 2 C, 4 C, and 6 C that are designed to allow the battery SoC to traverse between 10 and 90%. The same profile is used for all temperatures in the range described previously. Note that the range limits on temperature and SoC are dictated by the availability of experimental data. For example, on a battery cycler, it is difficult to design a profile that reaches 0% or 100% effectively. Often because the current control is not exact, a profile that tries to reach 0% or 100% can cause the undervoltage or overvoltage safety systems to stop the experiment before completion. Given that the battery is never operated outside of the 10 to 90% range in vehicle operation, we limit the SoC to be within this range. Nevertheless, if the dataset contains operation in other ranges, the methodology discussed here applies equally.

Once again, the first task is to evaluate the derivative of the open circuit voltage function with respect to SoC; the plot of the derivative is given in Fig. 12. First note that the lowest point on the plot occurs in two sections: between 35% and 55% SoC and between 70% and 85% SoC, where the lowest value reached is 0.0004 V/%. Compared with the same plot for the EIG battery seen in Fig. 6, this number is almost exactly one order of magnitude smaller. Consequently, this suggests that SoC estimation for this battery is inherently a much more difficult problem than with the EIG battery. Furthermore, a large difference in magnitude exists between the section where the lowest points occur and the remaining sections. This suggests that the SoC estimation will be more accurate when the battery is operating outside of these two zones. This is somewhat welcome news because if the battery starts to approach the low and high SoC regions where the df/dz is large, the estimator should be able to estimate the SoC more accurately, thus allowing the energy management system information to avoid taking the battery further into the low and high SoC regions. However, this also has an undesirable aspect because the regions between 35% and 55% SoC and between 70% and 85% are precisely where vehicles operate in charge-sustaining modes. The region between 70% and 85% is the region in which HEVs typically operate while the region around 35% is frequented by PHEVs after they enter the charge-sustaining mode. Consequently, this is not the ideal battery for applying the SoC estimator design. Some of the results later will reflect this analysis.

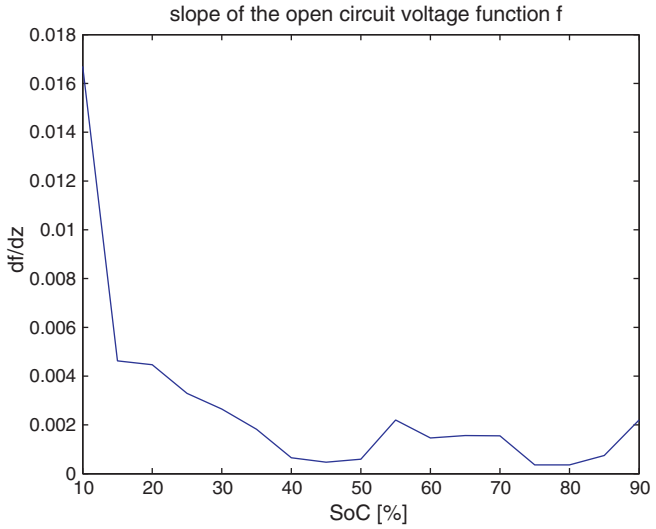

Fig. 12. A123 lithium ion iron-phosphate battery: slope of the OCV function f.

For the purpose of the estimator design, L_f from (9) can be selected as 0.0004. The two eigenvalues of the A matrix as functions of the temperature are given in Table 1. This table is obtained by evaluating the eigenvalues of the A matrices in the identified model at each temperature in the table. We select the poles of the closed loop matrix as [0.99, 0.91], which are smaller than the minimum values of the two poles as functions of the temperature. The reason for this choice is that we require the closed loop system to respond faster than the open loop system, but only slightly because the state z will be relatively slow. The location for the pole corresponding to z is selected as 0.996. While a faster pole would provide faster convergence, because the L_f is so small, a very large feedback gain would be required to achieve faster performance. Therefore we opted for a slower pole and thus a smaller feedback gain.

To see the performance of the estimator with the choice of the poles noted above, we utilize the step profile data that was used for modeling. For the 25 °C step profile data, Fig. 13 shows the estimated SoC compared with the SoC calculated from current integration (noted as "measured"). Fig. 14 shows the estimation error and the input disturbance. From these two figures, we can see that because this battery has a much flatter OCV function than the EIG battery example we have seen previously, the estimator is much

Table 1
Eigenvalues of the A matrix as functions of T.

Temperature (°C)	Pole 1	Pole 2
45	0.9920	0.9657
35	0.9916	0.9477
25	0.9906	0.9399
15	0.9916	0.9367
5	0.9921	0.9255
0	0.9972	0.9229
−5	0.9971	0.9126

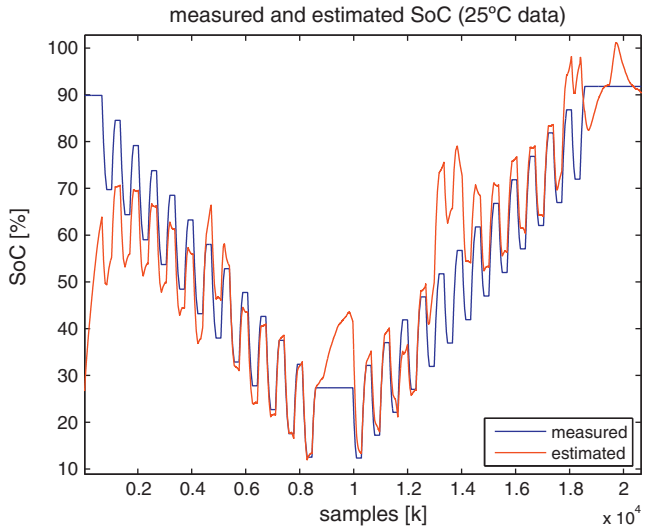

Fig. 13. SoC estimated using the 25 °C data compared with the measured data.

more susceptible to disturbance and modeling errors. However, the overall performance at room temperature is still very good as the estimator provides accuracy with an error of less than 5%, except under offset disturbance.

In Figs. 15 and 16, the same estimator is used with the 45 °C step profile data. Because the modeling error is even smaller at 45 °C, the estimator performance at this temperature is even better than with the 25 °C data.

Figs. 17 and 18 show the estimator performance over the 5 °C step profile data. In this example, we see that the performance is worse than the room temperature case. There are two reasons for this. First, the modeling error at lower temperatures is much higher than that at higher temperatures. Consequently, the magnitude of the uncertainties that influence the size of the dynamic error is much higher. Second, the OCV function for lower temperatures is different from the OCV function at higher temperatures. Because df/dz is very small for this battery, what might be considered small differences in OCV can influence the SoC estimation significantly. To investigate this further, we further moved the closed loop pole of z out toward the unit circle to decrease the effect of modeling error

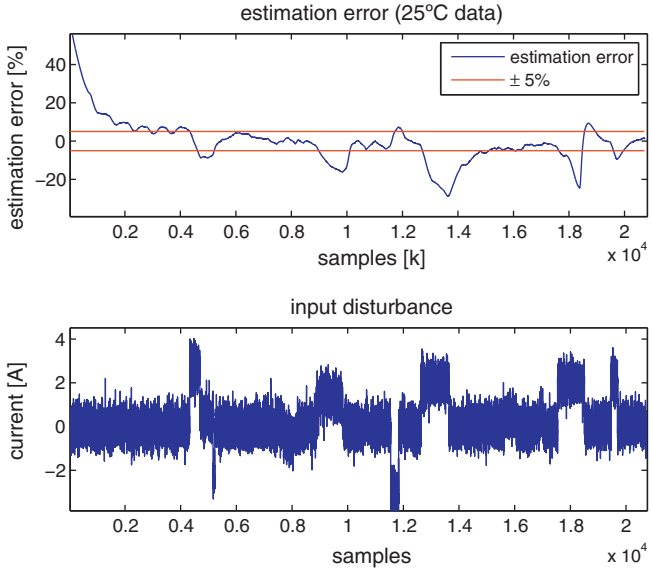

Fig. 14. SoC estimation error and input disturbance with 25 °C data.

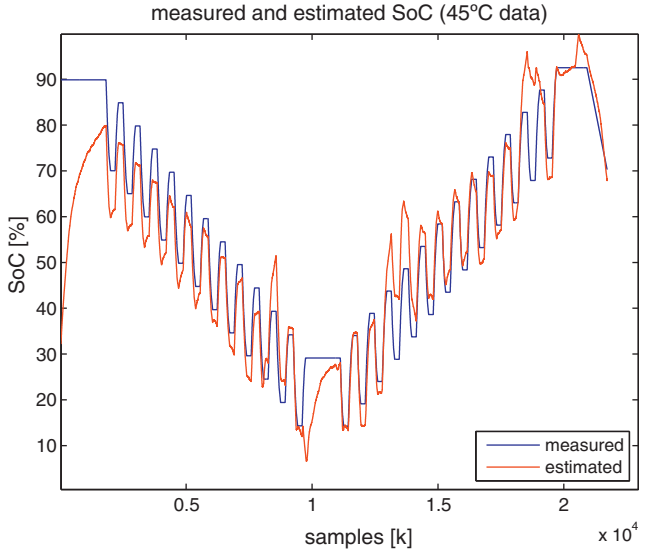

Fig. 15. SoC estimated using the 45 °C data compared with the measured data.

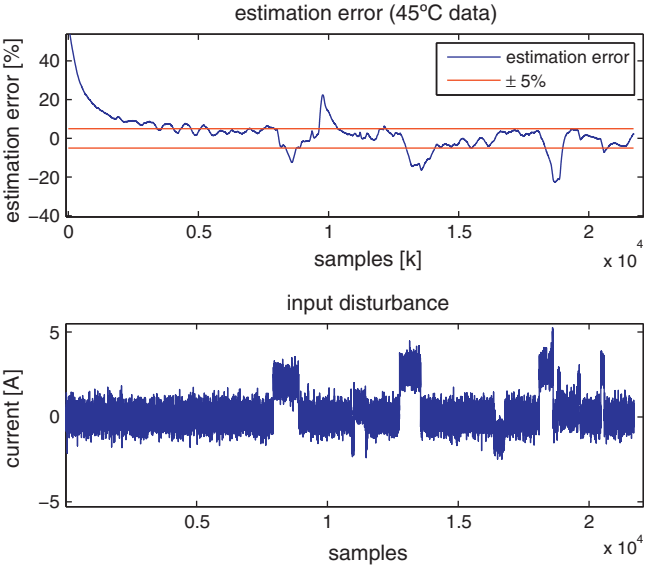

Fig. 16. SoC estimation error and input disturbance with 45 °C data.

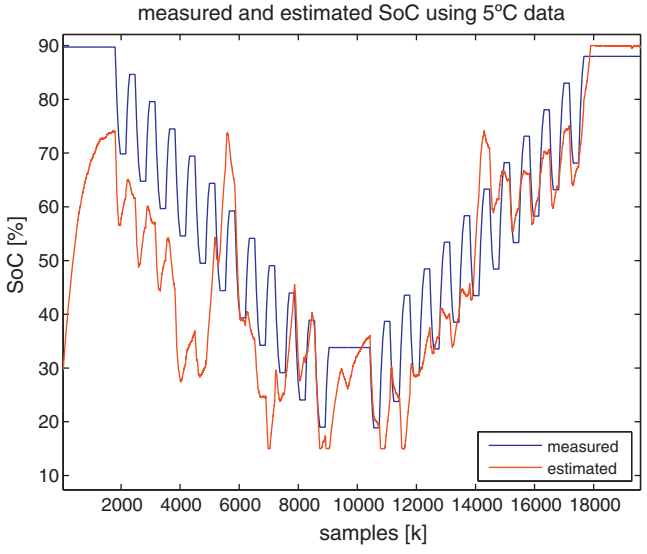

Fig. 17. SoC estimated using the 5 °C data compared with the measured data.

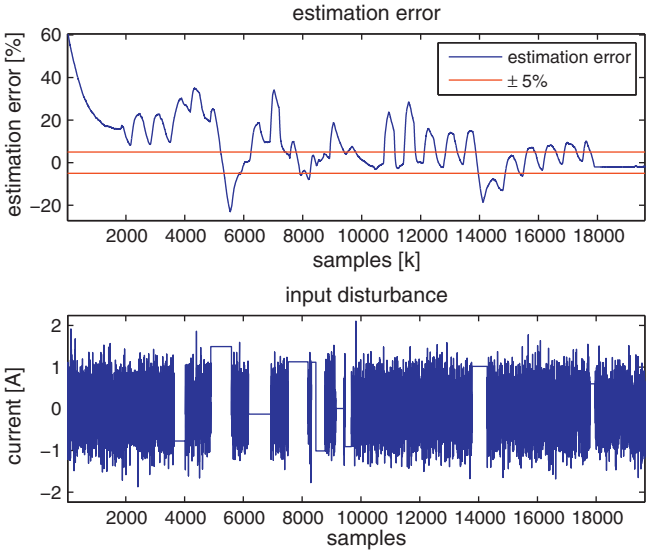

Fig. 18. SoC estimation error and input disturbance with 5 °C data.

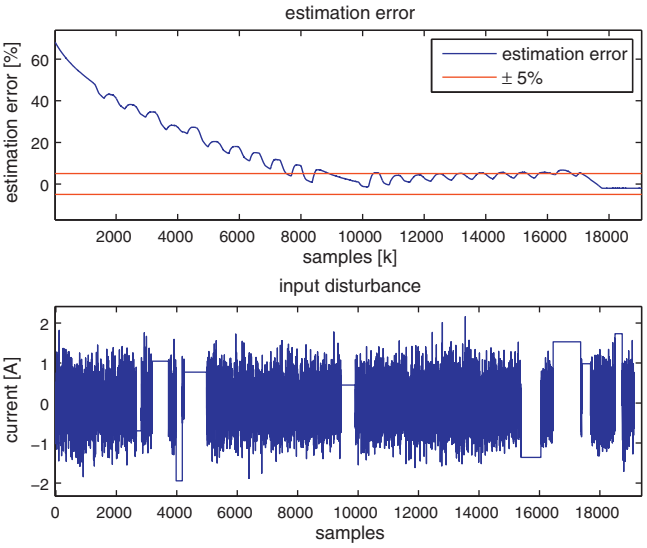

Fig. 20. SoC estimation error and input disturbance with a slower pole.

on the estimator performance. As Figs. 19 and 20 show, the slower pole dramatically reduced the convergence speed. However, the tracking performance after convergence improved dramatically.

An example will serve to illustrate the operation and performance of the OCV estimator. The 25 °C data is used for this purpose. As we can see from Fig. 23, the estimated OCV converges to the OCV calculated based on the measured SoC, and tracks well after a period of convergence. Therefore the OCV estimator discussed in the previous section is a valid alternative to the SoC estimator.

The results here suggest that for this particular battery, the OCV as a function of temperature should also be described. Otherwise, the accuracy of the lower temperature estimation is significantly affected. Due to absence of a better OCV function, for further illustration a multiple temperature example is provided that uses model based simulation data. In this example, a current profile that features simultaneous temperature and SoC variation is used to excite the model; subsequently, the estimator is used to estimate the SoC using the model output as inputs. As always, random signals are added to both the input and output to simulate model uncertainty and noise. As Figs. 21 and 22 show, despite the temperature

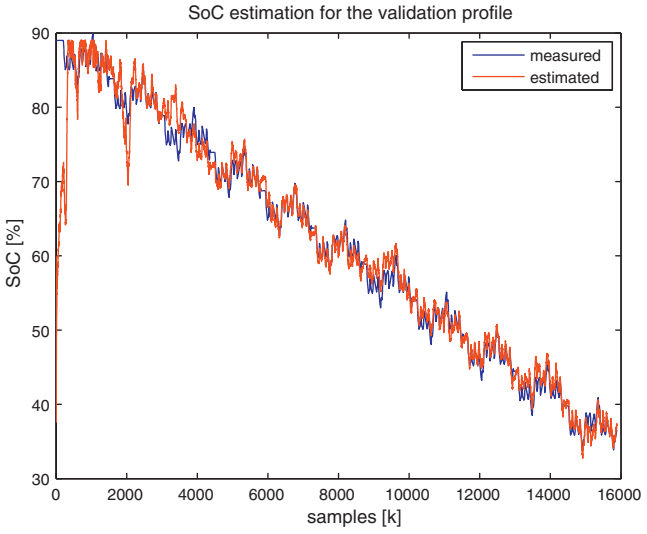

Fig. 21. SoC using the multi-temperature design.

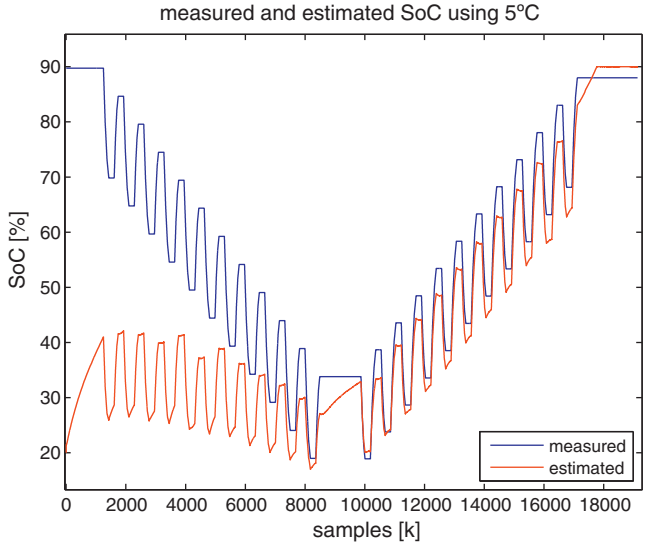

Fig. 19. SoC estimated using the 5 °C data and a slower pole.

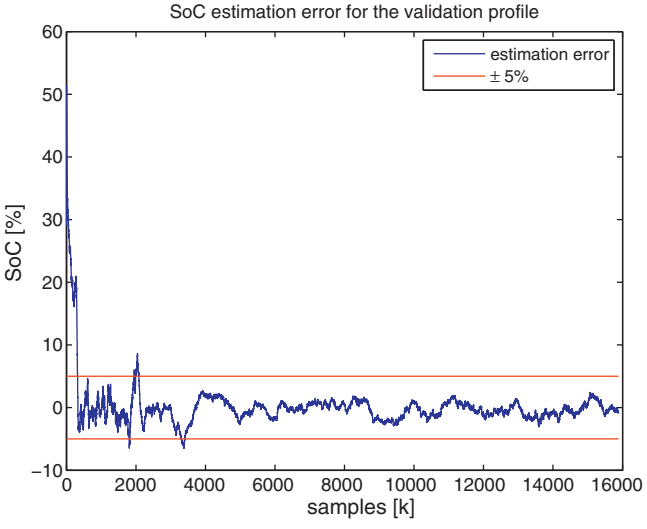

Fig. 22. SoC estimation error using the multi-temperature design.

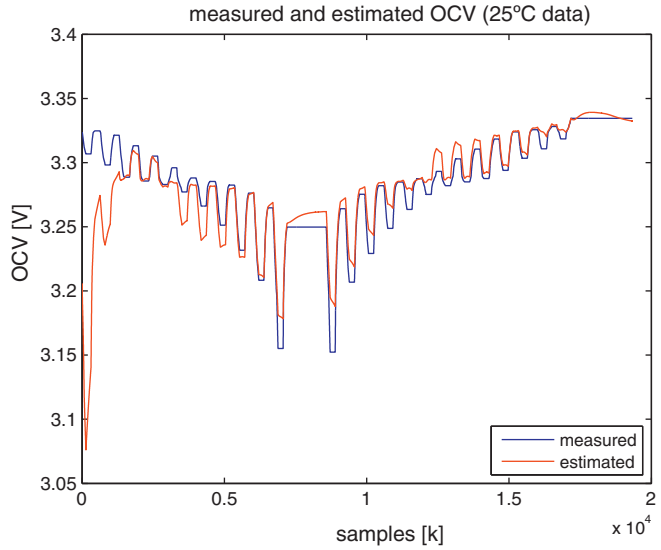

Fig. 23. OCV estimated using the 25 °C step profile data.

variation and the uncertainty, approximately the same performance in terms of estimation error results. After the estimator converges, the overall estimation error is kept well within 5% SoC.

5. Conclusion

In this paper, a SoC estimator is designed using linear parameter varying system techniques. The advantages of this design are threefold. First, this estimator does not require iterative calculation of the feedback gain such as in designs using the extended Kalman filter. Because of this, the computational burden is relatively small so that implementation is more feasible. Secondly, the stability of the estimator can be confirmed analytically based on linear matrix inequalities formulated using input-state stability criterion. Lastly, the performance of the estimator, in terms of convergence and tracking, can be tuned very easily depending on user requirements.

A fundamental fact emerging form this work is that the most important component of a voltage based SoC estimator designed using state estimation is the open circuit voltage of the battery. This is the only component aside from direct current integration that provides the user with a reliable measure of the SoC. For batteries such as the EIG battery whose OCV has relatively steep slope, a small error can be tolerated since the voltage changes significantly enough when the battery SoC changes. But for batteries such as the A123 (whose characteristics are common to all lithium-ion iron-phosphate chemistry), the inherent flatness of the OCV makes it critical to have as good of an OCV model as possible. Consequently, an important continuation of this work is to improve the OCV model, especially as a function of temperature. Such a model can be obtained with a well designed measurement experiment. Even though such an experiment may be tedious or time consuming, its impact on the accuracy of the estimator cannot be underestimated. Furthermore, the data from this experiment would lead to formulation of a hysteresis model, which combined with an accurate OCV model would provide even better estimates.

Appendix A. Equivalent circuit model discretization

The equivalent circuit model used in [10] is provided in Fig. 1. This model comprised of an open circuit voltage, an internal resistance, and multiple R/C circuits that represent the battery dynamics. The dynamics of each parallel RC circuit are represented by a first order differential equation of the form

$$\frac{dV_i}{dt} = -A_i V_i + A_i B_i I, \tag{A.1}$$

where A_i and B_i represent the input coefficient and the time constant, respectively. Note that A_i and B_i can be functions of the parameters as well, but for the purpose of discretization, it is assumed that the parameters do not change within each sampling instance.

Because the OCV is a static function of the SoC, discretizing the OCV dynamics can be done by discretizing the dynamics of the SoC. The continuous-time dynamics of SoC denoted by z are given by

$$\dot{z} = -\frac{1}{3600 C_n} I, \tag{A.2}$$

where C_n is the nominal capacity of the battery.

The overall battery terminal voltage is given by the sum of all the components

$$V_{batt} = V_{oc} - R_0 I - \sum_{i=1}^{n} V_i. \tag{A.3}$$

Given a selected sampling period T_s and a continuous linear system of the form

$$\dot{x} = ax + bu, \tag{A.4}$$

where $x, u \in \Re$ and the coefficients a, b and the input u are assumed to be constant over each sampling period, the discrete equivalent of the system is given by

$$x[k+1] = e^{-aT_s} x[k] + \frac{b}{a}(1 - e^{-aT_s}) u[k]. \tag{A.5}$$

Using this fact, Eqs. (A.1) and (3) can be discretized as

$$V_i[k+1] = a_i V[k] + b_i I[k], \tag{A.6}$$

$$V_h[k+1] = \gamma(I[k]) V_h[k] + \kappa(I[k]) H(\dot{z}, T), \tag{A.7}$$

where

$$a_i = e^{-A_i T_s},$$
$$b_i = B_i(1 - e^{-A_i T_s}),$$
$$\gamma(I[k]) = e^{-\Gamma |I[k]|},$$
$$\kappa(I[k]) = 1 - e^{-\Gamma |I[k]|}.$$

Eq. (A.2) simply represents an integrator, and can therefore be discretized as

$$z[k+1] = z[k] - \frac{T_s}{3600 C_n} I[k]. \tag{A.8}$$

The complete model can then be written in state variable form as

$$\begin{bmatrix} z[k+1] \\ V_1[k+1] \\ \vdots \\ V_n[k+1] \\ V_h[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & a_1 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & a_n & 0 \\ 0 & 0 & \dots & 0 & \gamma(I[k]) \end{bmatrix} \begin{bmatrix} z[k] \\ V_1[k] \\ \vdots \\ V_n[k] \\ V_h[k] \end{bmatrix} + \begin{bmatrix} -\frac{T_s}{3600 C_n} & 0 \\ b_1 & 0 \\ \vdots & \vdots \\ b_n & 0 \\ 0 & \kappa(I[k]) \end{bmatrix} \begin{bmatrix} I[k] \\ H(\dot{z}, T) \end{bmatrix}, \tag{A.9}$$

where the output equation (battery terminal voltage) is given by

$$V_b[k] = V_{oc}(z[k]) - R_0[k] I[k] - \sum_{i=1}^{n} V_i[k] - V_h[k]. \tag{A.10}$$

After discretization, this system becomes essentially indistinguishable from the discrete model form (2).

References

[1] E. Meisner, G. Richter, Journal of Power Sources 144 (2005) 438–460.
[2] S. Piller, M. Perrin, A. Jossen, Journal of Power Sources 96 (2001) 113–120.
[3] A. Salkind, S.C. Fennie, T.P. Atwater, D. Reisner, Journal of Power Sources 80 (1999) 293–300.
[4] S. Malkhandi, Engineering Applications of Artificial Intelligence 19 (2006) 479–485.
[5] C. Cai, D. Du, Z. Liu, J. Ge, 7th World Congress on Intelligent Control and Automation, 2008.
[6] G. Plett, Journal of Power Sources 134 (2004) 262–276.
[7] B. Bhangu, P. Bentley, D. Stone, C. Bingham, IEEE Transactions on Vehicular Technology 54 (3) (2005).
[8] I. Kim, Journal of Power Sources 163 (2006) 584–590.
[9] F. Zhang, G. Liu, L. Fang, Proceedings of the 7th World Congress on Intelligent Control and Automation, 2008.
[10] Y. Hu, S. Yurkovich, Y. Guezennec, B.J. Yurkovich, Journal of Power Sources 1 (2011) 449–457.
[11] Y. Hu, S. Yurkovich, Y. Guezennec, B.J. Yurkovich, Control Engineering Practice 17 (2009) 1190–1201.
[12] Y. Hu, B.J. Yurkovich, S. Yurkovich, Y. Guezennec, ASME Dynamic Systems and Control Conference, 2009.
[13] E. Sontag, Y. Wang, SIAM Journal of Control Optimization 39 (1) (2000) 226–249.
[14] E. Sontag, Y. Wang, Systems and Control Letters 24 (5) (1995) 351–359.
[15] J. Daafouz, J. Bernussou, Systems and Control Letters 43 (2001) 355–359.
[16] G. Millerioux, L. Rosier, G. Bloch, J. Daafouz, IEEE Transactions on Automatic Control 49 (2004) 1385–1389.
[17] Y. Hu, S. Yurkovich, Journal of Power Sources 196 (2011) 2913–2923.
[18] M. Verbrugge, E. Tate, Journal of Power Sources 126 (2004) 236–249.