A Robust Algorithm for State-of-Charge Estimation With Gain Optimization

Shaheer Muhammad, Muhammad Usman Rafique, Shuai Li, Zili Shao, Qixin Wang, and Nan Guan

***Abstract*—The charging and discharging procedure of a battery is a typical electrochemical process, which can be modeled as a dynamic system. State of charge (SoC) is a commonly used measure to quantify the charge stored in the battery in relation to its full capacity. Recent efforts of optimizing battery performance require more accurate SoC information. The noise in sensor readings makes the estimation even more challenging, especially in battery-operated systems where the supply voltage of the sensor keeps changing. Traditionally used methods of Coulomb counting and extended Kalman filter suffer from the accumulation of noise and common phenomenon of biased noise, respectively. The traditional approach of dealing with ever-increasing demand for accuracy is to develop more complicated and sophisticated solutions, which generally require special models. A key challenge in the adoption of such systems is the inherent requirement of specialized knowledge and hit-and-trial-based tuning. In this paper, we explore a new dimension from the perspective of a self-tuning algorithm, which can provide accurate SoC estimation without error accumulation by creating a negative feedback loop and enhancing its strength to penalize the estimation error. Specifically, we propose a novel method, which uses a battery model and a conservative filter with a strong feedback, which guarantees that worst-case amplification of noise is minimized. We capitalize on the battery model for data fusion of current and voltage signals for SoC estimation. To compute the best parameters, we formulate the linear matrix inequality conditions, which are optimally solved using open-source tools. This approach also features a low computational expense during estimation, which can be used in real-time applications. Thorough mathematical proofs, as well as detailed experimental results, are provided, which highlight the advantages of the proposed method over traditional techniques.**

***Index Terms*—Battery management systems, data fusion, state estimation, state feedback, state of charge (SoC).**

Manuscript received January 9, 2017; revised March 29, 2017; accepted April 24, 2017. Date of publication April 28, 2017; date of current version December 1, 2017. This work was partially supported by the grants from Research Grants Council of the Hong Kong Special Administrative Region, China (GRF 152138/14E and GRF 15222315/15E), National Natural Science Foundation of China through Projects 61272103, 61373049, and 61309004, National 863 Program 2013AA013202, Research Fund for the Doctoral Program of Higher Education of China under Grant 20130191120030, Chongqing cstc2012ggC40005 and cstc2013jcyjA40025, Fundamental Research Funds for the Central Universities (CDJZR14185501) and Chongqing University (2012T0006), and the Hong Kong Polytechnic University (4-ZZD7, G-YK24, G-YM10, and G-YN36). Paper no. TII-17-0044. *(Corresponding author: Shuai Li.)*

The authors are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: postmuhammadshaheer@yahoo.com; m.usman694@gmail.cm; shuaili@polyu.edu.hk; cszlshao@comp.polyu.edu.hk; csqwang@comp.polyu.edu.hk; csguannan@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2699219

I. Introduction

Electric vehicles, smartphones, and other consumer electronics are greatly contributing to the rise of commercial utilization of batteries. Batteries offer an efficient way of storing energy, and with careful considerations, environment-friendly energy ecosystems can be built around them. For a battery, estimating the state of charge (SoC) is an important problem for many applications involving single and multiple batteries. The estimated value of SoC can be utilized to efficiently use the individual cell as well as to schedule usage of cells in the multicell system to maximize the discharge time. Recently, different optimization techniques have been used to maximize the operating time of large-scale multicell batteries. Some techniques exploit the internal properties of recovery and rate discharge effect, such as [1]. Smart algorithms are also being employed to charge battery systems quickly [2]. However, these methods require the information of the current state of cells by measuring SoC. Due to complexity of electrochemical processes in batteries and noise in sensors, many sophisticated algorithms have been proposed for efficient battery monitoring, such as estimation of SoC [3], state of health (SoH) [4]–[6], and remaining useful life [7]. In this paper, we propose a novel SoC estimation method, which performs well under all types of noises and outperforms commonly used algorithms.

An earlier work that uses voltage as the basis of SoC estimation was presented in [8]. Though this work dealt with complexities such as hysteresis, it was concluded that it is difficult to estimate SoC for some battery types such as Nickel-metal hydride. A fuzzy logic approach for SoC estimation was presented in [9], which required training data, which is challenging because of changing properties in different conditions. A complex neural-network-based approach was presented in [10], but it is restricted to lead acid batteries and requires complex network design and computations. A hybrid neural-network- and genetic-algorithm-based approach of SoC estimation of series-connected modules (battery cells) was discussed in [11]. Despite the promising result, the proposed method is relatively complex and has high computational cost. In [12], a complicated mathematical model has been devised, specifically for lead acid batteries, which can predict the SoC and remaining operation time with up to 10% error. A nonlinear estimation method based on the sliding-mode observer is presented in [13]. This paper also discusses the parameters of the battery model through different tests. A clever algorithm was presented in [14]: it combines the weighted sum of (previously discussed) complex voltage-based methods and Coulomb counting techniques. An earlier method based on the Kalman filter was presented in [15]–[18]. In addition to a few similar methods, a famous estimation algorithm based on the extended Kalman filter (EKF) is proposed in [19]–[24]. This paper constructively develops the design of estimation method and detailed simulations and experimental results are shown. However, comparison of the estimated SoC with the actual SoC is not provided.

The simplest technique of calculating SoC relies solely on current measurement and is referred to as Coulomb counting [25]. In this method, the cell voltage is completely disregarded and only current is used. Alternatively, some traditional approaches were based on voltage information. This is challenging because of noise in voltage measurement and a highly nonlinear SoC–voltage relationship of lithium-ion (Li-ion) cells. Also, the SoC–voltage curve is flat for most of the region: there is very small change in voltage corresponding to a larger change in SoC, which makes it infeasible to use voltage as a basis for SoC estimation. However, the method of Coulomb counting also suffers from a major drawback: because of its relative integrative nature, the error keeps accumulating in it and grows significantly over time. Also, as it relies solely on the current measurement (and completely disregards the cell voltage), it can never cope with an incorrect initial estimate of SoC. This is the main reason, which has motivated us to use data fusion to combine the information of current and voltage to get the better estimate of SoC while dealing with noise in measurements.

The main disadvantage of Kalman-filter-based methods is that they can only deal with known and zero-mean Gaussian noise; results are inaccurate if there is bias in noise. Practically, a detailed model of noise is not available, and often, the noise is not zero-mean. In fact, in a wide variety of applications, noise is non-Gaussian [26]. The bias in measurement noise is a frequently occurring problem. As observed in most of the systems, the sensors and embedded controllers are powered by the same battery whose SoC is being monitored. This induces bias in measurement as the battery voltage changes with time. With time, batteries lose their storage capacity (due to calendar and cycle aging) and their SoH reduces, which causes a permanent change in internal resistance of cell, which leads to bias in sensor measurement. The presence of bias is a common factor to consider in estimation, and it leads to the solution of robust filters, such as H_∞, which do not make any assumption about noise [27]. The problem of SoC estimation is further complicated because noise is accumulated over time: persistent estimation error for a long operational period leads to a drastic increase in error.

A key benefit of the proposed solution over the existing approaches is that this algorithm does not require manual tuning. The gain calculation has been formulated as an optimization problem, which not only makes the method easy-to-use but also guarantees that the gain of noise from input to the output is minimal. Conventional recursive formulation of the H_∞ filter also requires a manually iterated procedure for finding gain, as shown in [28].

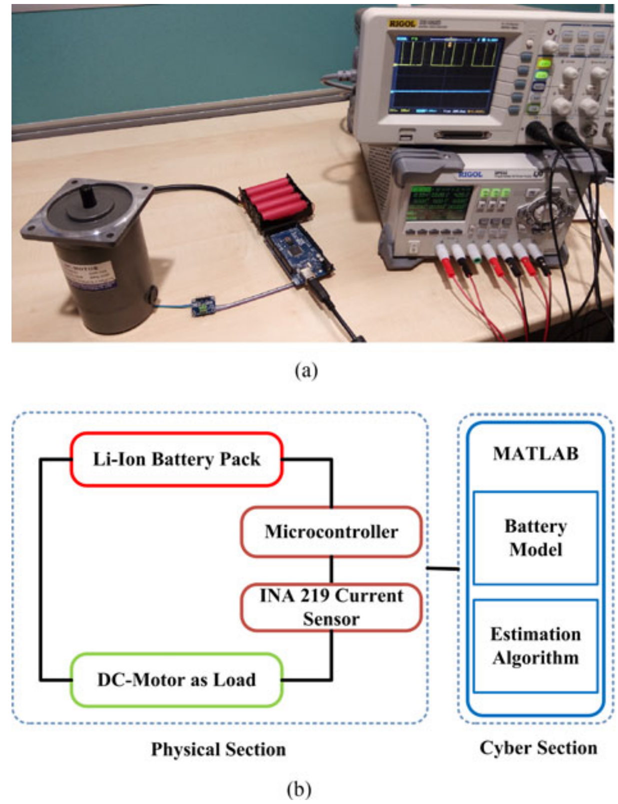


Fig. 1. Experimental setup and overview of the whole system. (a) Evaluation setup including battery, sensor, load (motor), and microcontroller. (b) System overview: hardware and software modules of the complete system.

The proposed method utilizes the battery model and techniques to deduce an exact model of any battery. Availability of an accurate model enables us to use data fusion and estimation techniques for this problem. Our evaluation setup and overview of the system are shown in Fig. 1(a) and (b), respectively. We have developed a novel method based on the battery model and robust estimation algorithm.

The computational expense is a major bottleneck for most of the embedded controllers. While sophisticated solutions might be implemented on computers, embedded systems generally have limited resources. Conventional filters (e.g., EKF and particle filter) require complex calculations to compute gain after every measurement cycle. The conventional way of solving the H_∞ filter is also complex, requiring recursive calculation of algebraic Riccati equations. Additionally, the system gain is set by hit-and-trial of multiple runs, which may be sensitive to the system model and operating conditions. To avoid these problems and to simplify the calculations, we devised the linear matrix inequality (LMI) conditions for the H_∞ filter, which can be optimally solved for best performance using open-source numerical optimization solvers. The optimal filter weight can be calculated by solving the LMI problem offline, prior to real-time estimation, which reduces the computational requirements during operation.

H_∞ is a conservative filter, which is designed to suppress the worst-case performance [29], [30]. Robustness of the H_∞

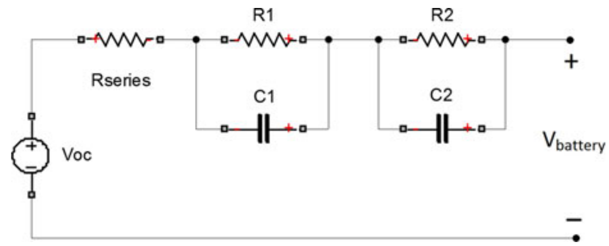


Fig. 2. Second-order model of the battery.

filter under different noise types has been rigorously proved in the literature, such as [31], [32]. Our main contributions in this paper are as follows.

1) A novel method of a conservative filter, based on the battery model, is presented, which guarantees that the worse-case amplification of noise is minimized.
2) Computational effort of the robust SOC estimation method is reduced with offline tuning of the control gain. Self-tuning-based optimal solution of the H_∞ filter has been developed to minimize the system gain.
3) Design of an H_∞ filter, which performs well in the presence of biased noise. In our evaluations, the error of the proposed method is less than 0.32%, whereas the error of the EKF goes as high as 2.43% and error of Coulomb counting is 7.72%. A comparison of different estimation methods under various conditions is validated by experimental results.

In Section II, we develop the battery model and formulate the estimation problem. Section III explains the H_∞ filter design, and Section IV details the LMI conditions for the filter. Implementation details are provided in Section V, and simulation and experimental results are presented in Section VI. Conclusions are given in Section VII.

II. Problem Formulation

In this section, the battery model and its representation are briefly discussed, followed by the formulation of the filtering problem for estimation of SoC.

A. Battery Model

Because of the nonlinear properties of batteries, especially Li-ion, a detailed battery model is required for estimation of SoC. However, the modeling problem has been extensively investigated in the literature. We consider the second-order battery model, widely used by other scholars, presented in [33]. The equivalent circuit of this model is shown in Fig. 2.

Many techniques, such as usage of pulses presented in [34], have been rigorously used to identify the parameters of the battery model. Let us define the state vector as $[v_1 \quad v_2 \quad \mathrm{SOC}]^T$. Current drawn from the battery, i, is considered as input for the model. The circuit in Fig. 2 can help us relate the input to the state vector, as shown in the following equation:

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{\mathrm{SoC}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{R_1C_1} & 0 & 0 \\ 0 & -\frac{1}{R_2C_2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \mathrm{SoC} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_1} \\ \frac{1}{C_2} \\ -\frac{1}{3600 \cdot \mathrm{Cap}} \end{bmatrix} \cdot i + E \cdot v_n. \tag{1}$$

Here, Cap is the total storage capacity of the battery in Ampere-hours. Also, v_1 and v_2 represent the voltage drop across C_1 and C_2, respectively. Vector v_n represents the noise, which is being added in the system, and matrix E relates the noise to the system state. Vector v_n is defined as $[n_{v_1} \quad n_{v_2} \quad n_{\mathrm{SoC}} \quad n_{\mathrm{voltage}}]^T$: it contains noise in states (v_1, v_2, and SoC) and the measured voltage.

Generally, the terminal voltage is considered the system output; however, there is a complication in it. It is widely known that the relationship between open-circuit voltage v_{OC} and SoC is nonlinear. Battery voltage can be represented as a nonlinear function of SoC, as shown in (2). This relationship can be determined experimentally:

$$v_{\mathrm{oc}} = f(\mathrm{SoC}). \tag{2}$$

The output voltage can be written as

$$y = f(\mathrm{SoC}) + \mathrm{v}_1 + \mathrm{v}_2 + \mathrm{R}_{\mathrm{series}} \cdot \mathrm{i}. \tag{3}$$

The linearized output equation can be derived by using Taylor series and ignoring higher order terms, as

$$y = \begin{bmatrix} 1 & 1 & \frac{\partial f(\mathrm{SoC})}{\partial \mathrm{SOC}} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \mathrm{SoC} \end{bmatrix} + R_{\mathrm{series}} \cdot i + F \cdot v_n. \tag{4}$$

Here, F relates the noise vector to the observed output. Matrices A, B, C, and D can be taken from (1) and (4).

For the purpose of estimation, we consider the output to be the estimated state z given by

$$z = Gx + Bu + Jv_n. \tag{5}$$

B. Estimation Problem

Considering independently, there are two methods of estimation of SoC: relying on current (coulomb counting) or by measuring voltage. Coulomb counting is the obvious choice, but it suffers from the accumulation of noise over time, and hence, a drift from the actual value is added. Voltage-based methods are limited because of the nonlinear relation of SoC with voltage. Additionally, Li-ion batteries have a flat voltage–SoC curve, i.e., there is a very small change in voltage for a considerable change in SoC. This is the reason why estimation is used to combine both voltage and current values to get the best result. A typical estimation problem can be formed as (6) and (7). Here, $\hat{x}$ and $\hat{y}$ represent the estimated state and output, respectively. Determining gain, K, is the main objective of filter design. As discussed in subsequent sections, both EKF and H_∞ filter employ different methods to calculate K but they have same structure as shown in (6)–(8):

$$\dot{\hat{x}} = A\hat{x} + K(y - \hat{y}) \tag{6}$$

$$\hat{y} = C\hat{x} \tag{7}$$

$$\hat{z} = G\hat{x} + Bu. \tag{8}$$

III. Robust H$_\infty$ Filter Design

The structure of the H_∞ filter is the same as given in (6)–(8). We define three errors $\tilde{x} = x - \hat{x}$, $\tilde{y} = y - \hat{y}$, and $\tilde{z} = z - \hat{z}$. Accordingly, based on the filter equations, we can express error dynamics as

$$\begin{aligned}\dot{\tilde{x}} &= A\tilde{x} + Ev_n - K\tilde{y}\\ &= (A - KC)\tilde{x} + (E - KF)v_n\end{aligned} \tag{9}$$

$$\tilde{y} = C\tilde{x} + Fv_n \tag{10}$$

$$\tilde{z} = G\tilde{x} + Jv_n. \tag{11}$$

As mentioned, we use $\tilde{z}$ as output to evaluate the robustness of the system against noise v_n; we consider the L_2 gain from v to z. For this evaluation, only (9) and (11) affect the result, and they can be rewritten as

$$\begin{aligned}\dot{\tilde{x}} &= A\tilde{x} + Ev_n - K\tilde{y}\\ &= (A - KC)\tilde{x} + (E - KF)v_n\end{aligned} \tag{12a}$$

$$\tilde{z} = G\tilde{x} + Jv_n. \tag{12b}$$

A. H_∞ Canonical Model

Consider a system with x, y, and w as its state variable, system output, and additive noise, respectively, and follows mathematical equations described below:

$$\dot{x} = Ax + Bw \tag{13a}$$

$$y = Cx + Dw. \tag{13b}$$

Remark 1: For the H_∞ problem given in (12), it falls into the form of (13) with $(A - KC)$, $(E - KF)$, G, and J in (12) corresponding to A, B, C, and D in (13), respectively.

B. Filter Formulation

Problem 1 (H_∞ Design): For the system described in (13), design parameters such that its L_2 gain is less than γ, i.e.,

$$\|y\|_{L_2} < \gamma\|w\|_{L_2} + \text{const.} \tag{14}$$

Remark 2: The L_2 gain of a linear system is equal to the H_∞ norm of its transfer function. Both of them describe the robustness of a system against external disturbances.

IV. LMI Conditions for an H_∞ Filter

A. Motivation of LMI

It is important to understand the significance of LMI conditions for H_∞ estimation or control. As we will soon find below, LMIs offer a flexible theoretical way of dealing with many aspects of control and estimation problems. Development of numerical methods has long been employed to solve LMIs. However, new numerical methods such as interior point are much faster, efficient, and they can be used to optimally solve LMIs [35]. An advantage of formulating problems in LMI is that their optimal solution can be found out using the free open-source software. One of the most famous free tools to solve LMI is CVX, which can be used in MATLAB [36]. In summary, on the one hand, LMIs offer the advantage of ease of manipulation and design, and on the other hand, we just have to formulate the problem in LMIs, and then, we can use efficient numerical solvers to optimally solve those LMIs.

The traditional H_∞ filter design problem involves manually finding out the best possible γ and solving the algebraic Riccati equation. This can be a tedious calculation, especially considering the nonlinearity, which is eradicated by linearization at different operating points. Computation of gain K, as shown in (6), becomes complex at every step, as it involves large computation to solve the Riccati equation. Alternatively, we propose an LMI-based H_∞ filter. We can leverage LMI to minimize γ using numerical optimization. Additionally, as discussed below, we may find the optimal solution of LMI only once; this removes the online computational cost.

B. Derivation of LMI Conditions

Let us develop the LMI conditions for this filtering problem.

Theorem 1: The following matrix inequalities suffice the L_2 gain requirement (14):

$$\begin{bmatrix} A^TP + PA & PB & C^T \\ B^TP & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0,\ P > 0. \tag{15}$$

Proof: Step 1: Note: The L_2 gain defined in (14) is in integral form, as the L_2 gain is defined over time horizon. For proof, we use its equivalent differential form based on the definition of $V = x^TPx$ and prove $\dot{V} \leq (\gamma w^Tw - \gamma^{-1}y^Ty)$. For the condition

$$\dot{V} \leq \gamma w^Tw - \gamma^{-1}y^Ty, \tag{16}$$

integrating from time 0 to t on both sides of (16) yields

$$\begin{aligned}V(t) - V(0) \leq{}& \gamma\int_0^t w^T(\tau)w(\tau)d\tau\\ &- \gamma^{-1}\int_0^t (y^T(\tau)y(\tau))d\tau.\end{aligned} \tag{17}$$

Equation (17) can be rewritten in terms of the L_2 norms as

$$V(t) - V(0) \leq \gamma\|w\|^2_{L_2} - \gamma^{-1}\|y\|^2_{L_2}. \tag{18}$$

Recall that $V = x^T P x, P > 0$. Thus, $V(t) \geq 0$ for any x at any time t. So,

$$-V(0) \leq V(t) - V(0) \leq \gamma \|w\|_{L_2}^2 - \gamma^{-1} \|y\|_{L_2}^2$$

is equivalent to

$$-\gamma V(0) \leq \gamma^2 \|w\|_{L_2}^2 - \|y\|_{L_2}^2. \tag{19}$$

Note that $V(0) = x^T(0)Px(0)$ is a constant. Clearly, (19) is already in the form of (14) by choosing the constant in (14) as $\gamma V(0)$.

Step 2: Let $V = x^T P x$ for $P > 0$ and symmetric. Then

$$\dot{V} = x^T P \dot{x} + \dot{x}^T P x. \tag{20}$$

Substituting (13) into (20) yields

$$\dot{V} = x^T P A x + x^T A^T P x + x^T P B w + w^T B^T P x. \tag{21}$$

In addition, with (13), we have

$$\gamma w^T w - \gamma^{-1} y^T y = \gamma w^T w - \gamma^{-1}(Cx + Dw)^T(cx + Dw). \tag{22}$$

Therefore

$$\begin{aligned} &- \dot{V} + \gamma w^T w - \gamma^{-1} y^T y \\ &= -x^T P A x - x^T A^T P x - x^T P B w - w^T B^T P x \\ &+ \gamma w^T w - \gamma^{-1}(Cx + Dw)^T(Cx + Dw) \end{aligned}$$

which can be expressed as

$$\begin{bmatrix} x \\ w \end{bmatrix}^T W \begin{bmatrix} x \\ w \end{bmatrix}$$

where

$$W = \begin{bmatrix} -(PA + A^T P + \gamma^{-1} C^T C) & PB - \gamma^{-1} C^T D \\ B^T P - \gamma^{-1} D^T C & \gamma I - \gamma^{-1} D^T D \end{bmatrix}.$$

To guarantee $-\dot{V} + \gamma w^T w - \gamma^{-1} y^T y > 0$ all the time, we need the following matrix inequality to hold:

$$\begin{bmatrix} -(PA + A^T P + \gamma^{-1} C^T C) & PB - \gamma^{-1} C^T D \\ B^T P - \gamma^{-1} D^T C & \gamma I - \gamma^{-1} D^T D \end{bmatrix} > 0. \tag{23}$$

Step 3: Equation (23) is already in a proper matrix form, but we can use the Schur complement to make it more compact. Using the Schur complement, we can prove that the following one is equivalent to (23):

$$\left[\begin{array}{cc|c} -(A^T P + PA) & -PB & -C^T \\ -B^T P & \gamma I & -D^T \\ \hline -C & -D & \gamma I \end{array}\right] > 0. \tag{24}$$

To prove so, partition the matrix in (24) as indicated by the lines into four block matrices. According to the Schur complement, (24) is equivalent to

$$\begin{bmatrix} -(A^T P + PA) & -PB \\ -B^T P & \gamma I \end{bmatrix} - \begin{bmatrix} -C^T \\ -D^T \end{bmatrix} (\gamma I)^{-1} \cdot \begin{bmatrix} -C & -D \end{bmatrix} > 0, \gamma I > 0$$

which is clearly identical to (23). ■

The H_∞ design problem includes K in both A and B of the canonical model. To see this, we replace A, B, C, and D in (15) of the canonical form (13) with $(A - KC)$, $(E - KF)$, G, and J in (12) as

$$\begin{bmatrix} (A - KC)^T P + P(A - KC) & P(E - KF) & G^T \\ (E - KF)^T P & -\gamma I & J^T \\ G & J & -\gamma I \end{bmatrix} < 0,$$

$$P > 0$$

where K and P are both unknown. By defining a new variable $Q = K^T P$, it becomes

$$\begin{bmatrix} A^T P - C^T Q + PA - Q^T C & PE - Q^T F & G^T \\ E^T P - F^T Q & -\gamma I & J^T \\ G & J & -\gamma I \end{bmatrix} < 0,$$

$$P > 0$$

which are LMIs relative to Q, P, and γ.

C. Solving LMI

Many available solvers can be used to find the optimal solution of the LMI. Formally, we have developed the following optimization problem:

$$\begin{aligned} &\min \quad \gamma \\ &\text{subject} \quad to \end{aligned}$$

$$\begin{bmatrix} A^T P - C^T Q + PA - Q^T C & PE - Q^T F & G^T \\ E^T P - F^T Q & -\gamma I & J^T \\ G & J & -\gamma I \end{bmatrix} < 0,$$

$$P > 0. \tag{25}$$

Any LMI solver can be used to solve this LMI to get P and Q. Then, we can substitute these to get the gain K as $K = P^{-1} Q^T$. In this work, we used the CVX [36] solver for the optimal solution of the LMI.

Remark 3: We can see that the proposed robust H_∞ filter has been formulated as an optimization problem. The solution not only guarantees the best result, but also removes the overhead of tuning by hit-and-trial.

V. Implementation Details

A. Battery Modeling

The cell used in this study is an 18650 Li-ion cell having a nominal voltage of 3.7 V and storage capacity of 3.6 Ah. First of all, based on experiments, we find the model parameters of the battery based on the model presented in Section II-A. Current and voltage of a fully charged cell were noted to find the parameters during discharge. Polynomials of the different order were tried in order to find the best match for our battery. We found out that the following ninth-order polynomial is sufficient

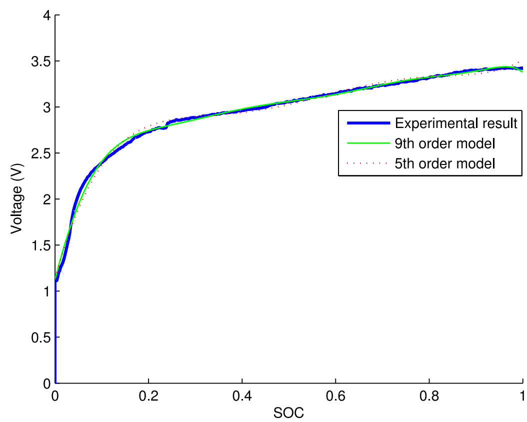


Fig. 3. Comparison of simulation with experimental results.

to predict the voltage based on information of SoC as in (2):

$$v_{oc} = f(\mathrm{SoC}) = a_1 S^9 + a_2 S^8 + a_3 S^7 + a_4 S^6 + a_5 S^5 + a_6 S^4 + a_7 S^3 + a_8 S^2 + a_9 S^1 + a_{10}$$

where S represents the SoC for compactness. We get the following values of these coefficients: $a_1 = -1955.12$, $a_2 = 8703.78$, $a_3 = -16064.97$, $a_4 = 15786.72$, $a_5 = -8739.73$, $a_6 = 2557.63$, $a_7 = -243.74$, $a_8 = -61.91$, $a_9 = 19.60$, and $a_{10} = 1.12$. The suitable order of polynomial is dependent on the battery and its SoC–voltage curve. Any polynomial that can successfully represent this relationship without overfitting can be used. To highlight this, we show the model predicted by a fifth-order polynomial as well in Fig. 3; it can be seen that ninth-order model is more accurate, especially in the nonlinear region when SoC is close to 1.

The battery parameters were found out to be: $R_{\text{series}} = 0.0051\,\Omega$, $R_1 = 0.0067\,\Omega$, $C_1 = 9327.2$ F, $R_2 = 0.0243\,\Omega$, and $C_2 = 675.3$ F. For a detailed process of finding all parameters of the battery model, the reader is referred to [34].

Let us validate the correctness of our battery model. We performed a simple experiment of running a small motor with a fully charged battery and noted the current and voltage of the cell. We provided same inputs to the model and compared the results of modeling with the experiment, as shown in Fig. 3. We can see that simulation matches the actual experiment. INA219 from Texas Instruments was used for sensing current and voltage in experiments.

B. Computational Expense for Gain Recalculation

For traditional methods, such as EKF, the filter gain is recalculated at every iteration, which is quite complex, often requiring matrix inversions. The standard approach of implementing the H_∞ filter requires the solution of the algebraic Riccati equation, which demands higher computation time. Our gain calculation process, as explained in Section IV-C, ensures the optimal solution according to the system. However, the computational overhead must be considered for real-time computing. About recalculation of gain, we have the following remark.

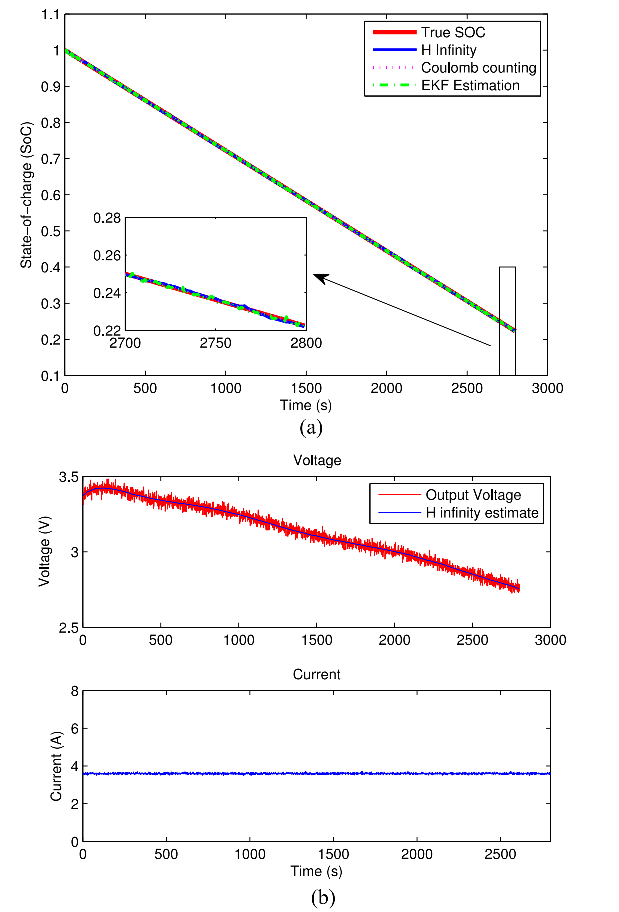


Fig. 4. Simulation result of SoC estimation with minimal noise. (a) SoC over time. (b) Current and voltage signals over time (the actual current is 3.6 A).

Remark 4: Gain recalculation is not required at every iteration. The recalculation process, which requires more time, should be dictated by the battery model and errors in linearization. Variation in the linear model is dependent on SoC, and we can leverage it to reduce computational costs.

In the ideal case, where the system model is linear, our approach demands the calculation of gain only once (because H_∞ is a conservative filter) as can be seen in Section IV-C. However, due to the nonlinear system model, we have to linearize at different operating points to get an accurate matrix C, as shown in (4). To minimize error in linearization, we need to recalculate gain as soon as C has changed significantly, which is dependent on SoC.

Proposition 1: Instead of calculating gain by solving LMI (see Section IV-C) at every step, we only solve it when SoC has changed by at least ΔS_c.

This will imply that in a complete battery cycle (SoC from 0 to 1, or 1 to 0), the number of calculations of gain will be $\frac{1}{\Delta S_c}$. It is of vital importance to note that, in general, SoC changes significantly slowly as compared to the processing speed of embedded systems. Because of the conservative nature of H_∞, we can leverage this property to reduce computations. For our ninth-order model as explained in Section V-A, we concluded that $\Delta S_c = 0.005$ is sufficient (which implies that, at maximum,

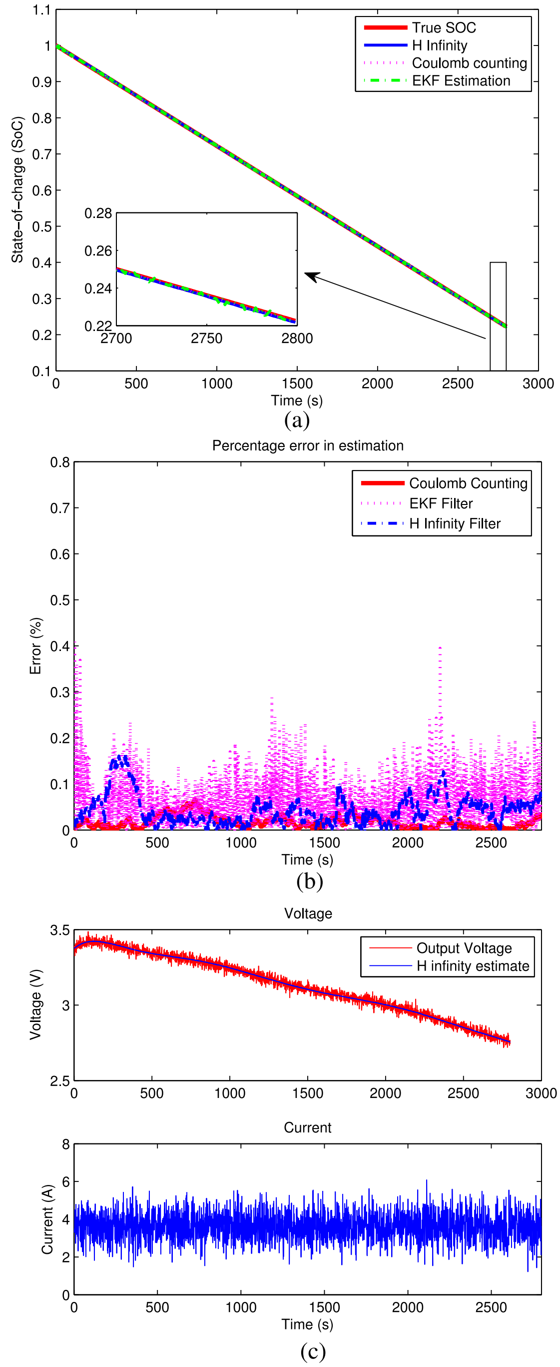


Fig. 5. Estimation result with unbiased noise. (a) Simulation result of SoC with unbiased noise. (b) Error in SoC estimation. (c) Current and voltage signals with unbiased noise (the actual current is 3.6 A).

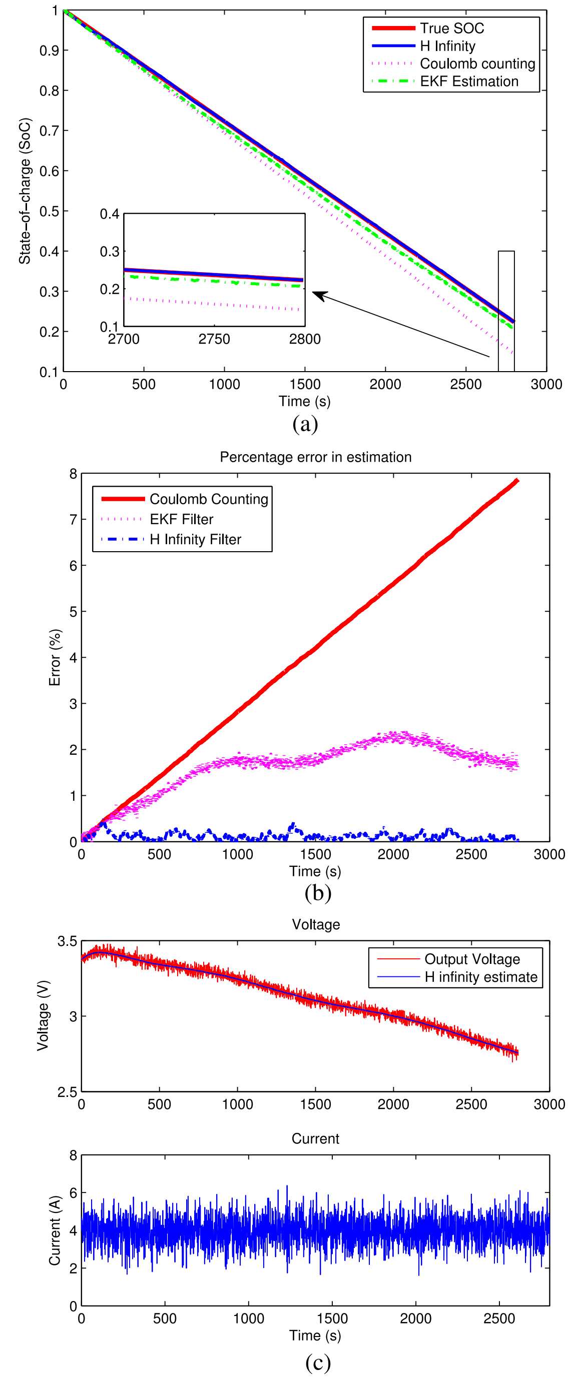


Fig. 6. SoC estimation in the presence of biased noise. It can be seen that the proposed method outperforms other methods. (a) Simulation result of SoC with biased noise. (b) Error in SoC in the presence of biased noise. (c) Original current of 3.6 A has biased noise of 0.36 A and variance of noise is 0.36A^2.

it will be recalculated 200 times), which is much smaller than continuous calculation lasting up to thousands of seconds. It is important to note that a suitable value of ΔS_c depends on thesystem model; for lower order polynomials, we found out values as high as $\Delta S_c = 0.1$ to be satisfactory.

It is proposed that for all steps of gain calculation (which have an upper bound of $\frac{1}{\Delta S_c}$), the gain is calculated offline and stored in the form of a lookup table in the embedded controller. For our case, we only had to calculate gain at 200 operating points, which are marked by a difference of 0.005 in SoC (on a scale of 0 to 1) using Section IV-C. This calculation, and storage in memory, implies that the embedded controller only has to perform simple multiplications, expressed in (6)–(8) because of prior calculation for the optimal solution of the developed LMI conditions. This enables even the simple and inexpensive controllers (such as Arduino UNO in our case) to

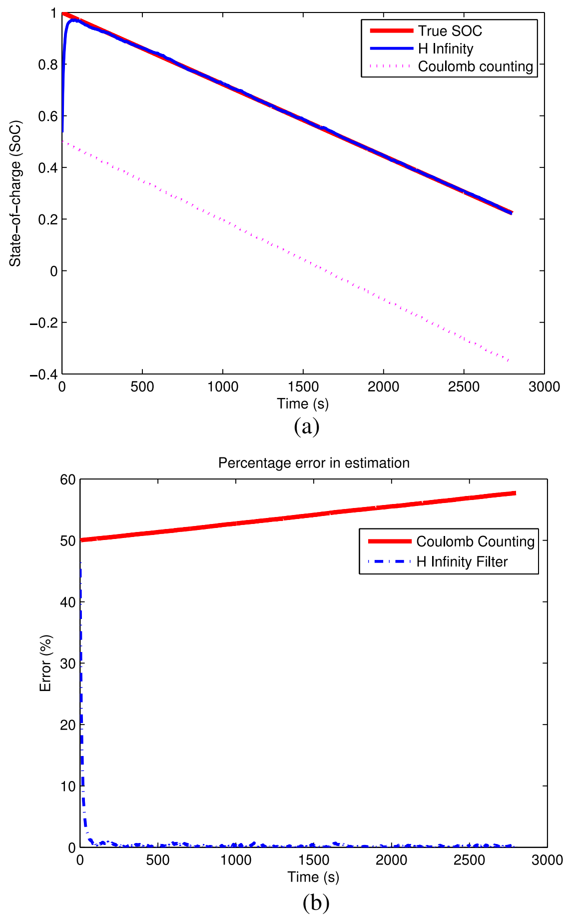


Fig. 7. Result when the initial estimate is inaccurate, showing the robustness of the proposed method. The initial estimate of SoC is 0.5, whereas the actual value is 1. (a) Filter performance in the case of incorrect initial estimate. (b) Error in the case of inaccurate initial estimate.

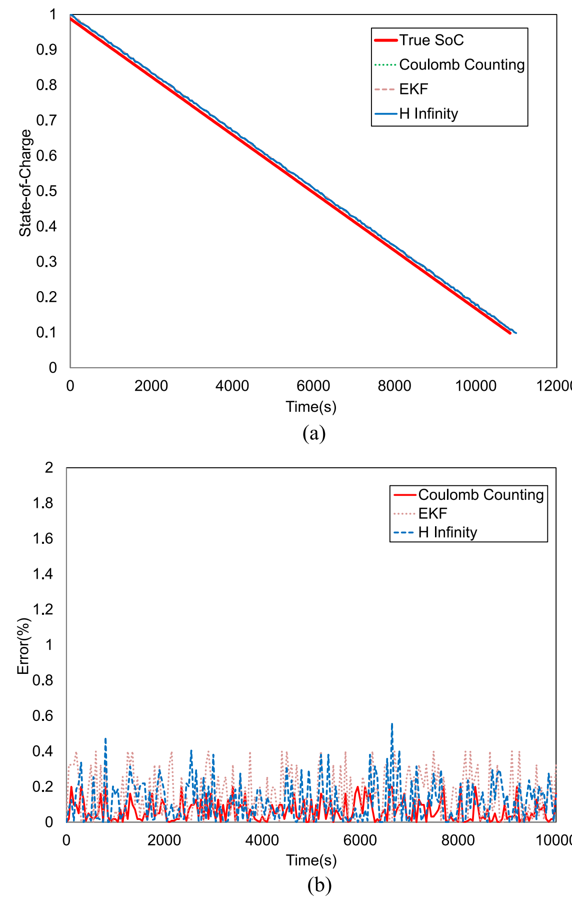


Fig. 8. Experimental result with unbiased noise. (a) Experimental result of SoC estimation (the noise has zero mean). (b) Error in SoC estimation.

easily find the optimal solution online without any additional burden.

VI. Results

The proposed method has been rigorously evaluated for robustness in both simulations and experimental tests. It is important to note that in all evaluations, we have used the nonlinear system model; the linearized version is used only for the purpose of filtering. This makes all our evaluations realistic by inducing the errors due to linearization as well modeling errors. Such errors are generally ignored by works that are limited to simulations only. In simulations and experiments, we first investigate the case of unbiased noise, where all estimation methods work well. Then, we evaluate these techniques in the presence of biased noise: this shows the advantage of the proposed method over others.

A. Baseline for Comparison

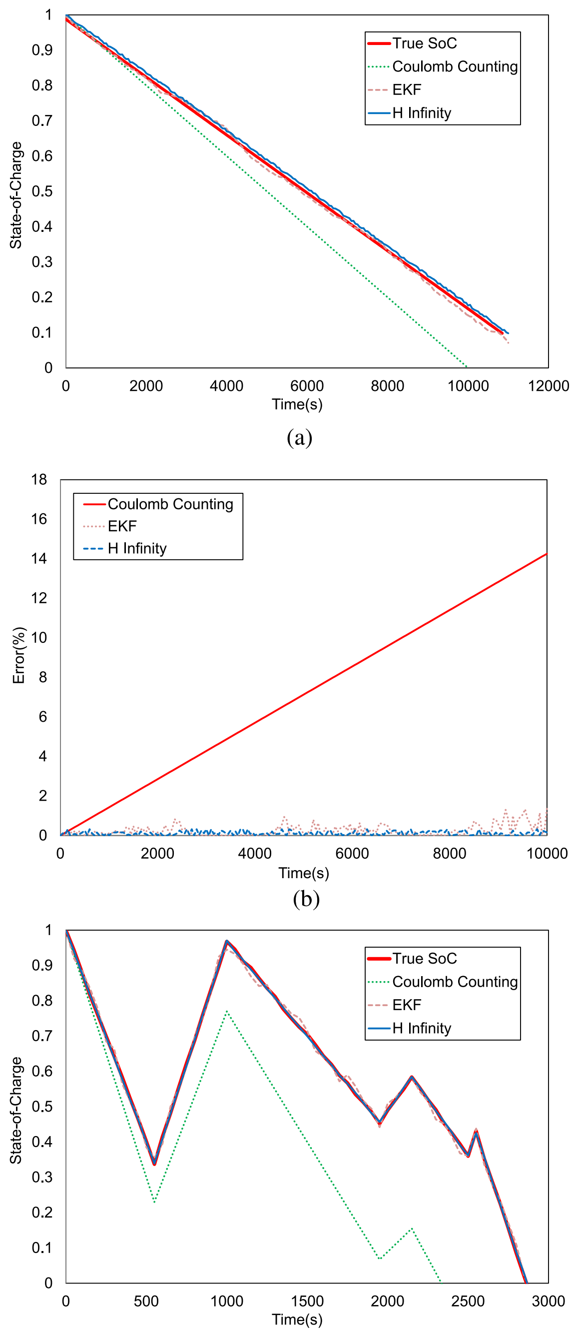
Before we evaluate the performance of the proposed H_∞ filter, let us first have a look at the baseline of the methods we will be comparing with. We compare the performance of our filter with two commonly used methods for SoC estimation, which are Coulomb counting and EKF filter.

The reason for selecting Coulomb counting is straightforward: it is simplest and the most commonly used method. The EKF method has been included in the comparison because it has been rigorously studied, and experimental results, though with limited noise types, are available in the literature, such as in [19].

1) Coulomb Counting: Coulomb counting is a simple method of estimating SoC, which relies solely on current measurement. It can be represented as

$$\mathrm{SoC} = \mathrm{SoC}_i - \frac{1}{3600 \cdot \mathrm{Cap}} \cdot \int_0^t i(t)dt \qquad (26)$$

where SoC_i is the initial SoC, Cap is the battery storage capacity (Ah), and i is the current (A). Because of noise in current measurements, there is some error in SoC estimation. Since Coulomb counting, presented in (26), is an integrative process, the error keeps accumulating over time and grows large for longer operating time. The other alternative is to use voltage measurement for SoC estimation. As discussed before, the SoC–voltage relationship is highly nonlinear and even worse is the fact that the SoC–voltage curve of Li-ion batteries is flat, i.e.,



(a)

(b)

(c)

Fig. 9. Experiment result in the presence of biased noise. (a) Experimental result of SoC estimation with biased noise. (b) Error in SoC estimation. (c) Experimental result of SoC estimation with biased noise under extreme conditions.

for a large change in SoC, there is only a small difference in voltage.

2) EKF Estimation: Let us have a brief look at the design of the EKF filter for the system defined in (1) and (4). Prior state estimation is given by

$$\dot{\hat{x}}^{-} = Ax + Bu.$$

Output estimation is

$$\hat{y} = C\hat{x}^{-} + Du.$$

The prior covariance matrix P^{-} can be calculated using

$$P^{-} = APA^{T} + Q$$

where Q is the process noise and P is the covariance matrix calculated previously. The Kalman gain L can be computed by

$$L = P^{-}C^{T}[CP^{-}C^{T} + R]^{-1}$$

where R is measurement noise. After measurement update y, the final state estimate is

$$\hat{x} = \hat{x}^{-} + L[y - \hat{y}]$$

and covariance matrix P is updated by

$$P = (I - LC)P^{-}.$$

B. Filter Gains

For the H_{∞} filter, the matrices E and F, as explained in (1) and (4), are used as tuning parameters. The values used for these matrices are

$$E = 10^{-3}\begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.07 & 0 \end{bmatrix},$$

$$F = \begin{bmatrix} 0 & 0 & 0 & 100 \end{bmatrix}.$$

For the EKF, we need a process noise matrix Q and a measurement noise matrix R. The matrices used for the EKF are

$$Q = 10^{-3}\begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.07 \end{bmatrix}, \quad R = 100.$$

As shown in simulation and experimental results, the EKF filter outperforms other methods in the presence of Gaussian noise; this shows the correct tuning of filter parameters.

C. Simulations

We analyzed the results of an ideal case, where there is little noise in current and voltage. The SoC estimation result is shown in Fig. 4(a), and the current and voltage signals used for this simulation are given in Fig. 4(b).

Next, let us analyze the effect of unbiased (zero mean) noise on the performance of all estimation methods. The current and voltage signals for this evaluation are shown in Fig. 5(c); variance of noise in current is 0.5 A^2. The resulting SoC estimation is shown in Fig. 5(a), and the error in estimation is shown in Fig. 5(b). The interesting inference from this test is the robustness of all methods, including the Coulomb counting, which performs well when the noise has zero mean.

Now, let us consider the case where noise has a bias. The noise current signal has a mean of 0.36 A and variance of 0.36 A^2. The SoC is shown in Fig. 6(a), the error in estimation is given in Fig. 6(b), and resulting current and voltage signals are shown in Fig. 6(c). We can see that in the presence of biased noise, the EKF does not perform so well, while the H_∞ filter has given good estimate.

Let us examine the effect of incorrect initial estimate, i.e., when we do not know the starting SoC of the battery. The initial estimate of SoC is 0.5, while the actual SoC is 1. The resulting performance of the filter is shown in Fig. 7(a), and the error in estimates can be seen in Fig. 7(b). This evaluation has been done in the case of noise having a variance of 0.5 A^2 and a mean of 0.36 A. As Coulomb counting relies exclusively on current, it cannot deal with inaccuracy in initial estimate, while we see the robustness of the H_∞ filter in such a scenario as well.

D. Experimental Results

Let us initially consider the case where noise is unbiased. The result of this experiment is shown in Fig. 8(a), and the error in estimation is shown in Fig. 8(b). The variance of current noise was almost 0.2 A^2. Because of small load (motor), the experiment took longer to be completed. We see that in this near-perfect scenario, both H_∞ and Coulomb counting perform well.

However, if we have noise with bias in it, the situation is completely different. The next experiment has a 0.2-A bias in current reading. The resulting estimation is shown in Fig. 9(a) and the error is plotted in Fig. 9(b). We see that now H_∞ is still robust, but Coulomb counting has accumulated large noise. Moreover, the algorithm has been tested under extreme conditions. In this experiment, we negate all the safety guidelines for the Li-ion battery and discharge current as well as charge at very high and erratic rate. This is done in order to achieve a relatively complex curve for SoC and test the performance of the algorithm. It can be seen from Fig. 9(c) that coulomb counting accumulates much more error in this case, but H_∞ is still robust and more accurate than EKF.

VII. Conclusion

We have shown that the proposed method can robustly estimate SoC by using simple but accurate battery model and employing a conservative filtering technique. The H_∞ filter is solved optimally by formulating it as an LMI problem. The separation of computation (calculation of gain once and iterative implementation of estimator) enables the proposed method to be implemented on embedded controllers without compromising real-time operation requirements. Note that superiority of the proposed algorithm has been demonstrated on the standard battery model; specialized or complex modeling is not required. We extend the robustness of current methods by using data fusion techniques. Our detailed results confirm the improved performance over both Coulomb counting and Kalman filtering. Since the H_∞ filter makes no such assumption of noise type, it can deal with biased noise as well. It is important to note that perhaps the simplest method of Coulomb counting also performs well in the presence of unbiased noise, and it is immune to noise in voltage sensing. A performance comparison of filters under different conditions for the Li-ion battery is presented in Table I. It is important to see that Coulomb counting outperforms other techniques in the presence of unbiased noise because both H_∞ and EKF have modeling and linearization errors. Performance of filters under different and extreme conditions have been observed in different batteries. The comparison of error in estimation methods in the lead-acid battery is presented in Table II.

TABLE I
Comparison of Error in Estimation Methods in the Li-Ion Battery

Scenario		Coul. Count	EKF	H_∞
Nominal System	Mean	**0.006**	0.060 ± 0.002	0.066 ± 0.002
	Max	**0.011**	0.354	0.289
Unbiased Noise	Mean	**0.03**	0.035	0.06 ± 0.002
	Max	**0.089**	0.146	0.24
Biased Noise	Mean	3.93 ± 4.95	1.69 ± 0.35	**0.07 ± 0.004**
	Max	7.725	2.43	**0.32**

TABLE II
Comparison of Error in Estimation Methods in the Lead-Acid Battery

Scenario		Coul. Count	EKF	H_∞
Nominal System	Mean	**0.007**	0.051 ± 0.003	0.071 ± 0.004
	Max	**0.0126**	0.362	0.311
Unbiased Noise	Mean	**0.043**	0.051	0.07 ± 0.004
	Max	**0.096**	0.192	0.4
Biased Noise	Mean	5.28 ± 6.1	1.8 ± 0.38	**0.08 ± 0.003**
	Max	8.871	2.76	**0.35**

In summary, if initial SoC is known and there is no bias in current noise, the Coulomb counting method can be used. For robustness, however, an advanced approach is required. It has been shown that the H_∞ filter performs better when there is bias in current noise, and it requires minimal computations during estimation.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

[1] H. Kim and K. Shin, "Scheduling of battery charge, discharge, and rest," in *Proc. 30th IEEE Symp. Real-Time Syst. Symp.*, 2009, pp. 13–22.
[2] L. He *et al.*, "RAC: Reconfiguration-assisted charging in large-scale lithium-ion battery systems," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1420–1429, May 2016.
[3] A. B. da Cunha, B. R. de Almeida, and D. C. da Silva, Jr, "Remaining capacity measurement and analysis of alkaline batteries for wireless sensor nodes," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 6, pp. 1816–1822, Jun. 2009.
[4] J. Yu, "State-of-health monitoring and prediction of lithium-ion battery using probabilistic indication and state-space model," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2937–2949, Nov. 2015.
[5] J. Yu, "Health degradation detection and monitoring of lithium-ion battery based on adaptive learning method," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1709–1721, Jul. 2014.
[6] M. Landi and G. Gross, "Measurement techniques for online battery state of health estimation in vehicle-to-grid applications," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1224–1234, May 2014.

[7] D. Liu, W. Xie, H. Liao, and Y. Peng, "An integrated probabilistic approach to lithium-ion battery remaining useful life estimation," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 3, pp. 660–670, Mar. 2015.

[8] M. Verbrugge and E. Tate, "Adaptive state of charge algorithm for nickel metal hydride batteries including hysteresis phenomena," *J. Power Sources*, vol. 126, no. 1, pp. 236–249, 2004.

[9] P. Singh, C. Fennie, D. Reisner, and A. Salkind, "A fuzzy logic approach to state-of-charge determination in high performance batteries with applications to electric vehicles," in *Proc. 15th Electr. Veh. Symp.*, vol. 15, 1998, pp. 30–35.

[10] C. Chan, E. Lo, and S. Weixiang, "The available capacity computation model based on artificial neural network for lead–acid batteries in electric vehicles," *J. Power Sources*, vol. 87, no. 1, pp. 201–204, 2000.

[11] Y. Lee, W. Wang, and T. Kuo, "Soft computing for battery state-of-charge (BSOC) estimation in battery string systems," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 229–239, Jan. 2008.

[12] P. Pascoe and A. Anbuky, "VRLA battery discharge reserve time estimation," *IEEE Trans. Power Electron.*, vol. 19, no. 6, pp. 1515–1522, Nov. 2004.

[13] I. Kim, "Nonlinear state of charge estimator for hybrid electric vehicle battery," *IEEE Trans. Power Electron.*, vol. 23, no. 4, pp. 2027–2034, Jul. 2008.

[14] Y. He, W. Liu, and B. Koch, "Battery algorithm verification and development using hardware-in-the-loop testing," *J. Power Sources*, vol. 195, no. 9, pp. 2969–2974, 2010.

[15] G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 1. background," *J. Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.

[16] G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 2. modeling and identification," *J. Power Sources*, vol. 134, no. 2, pp. 262–276, 2004.

[17] G. Plett, "Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. state and parameter estimation," *J. Power Sources*, vol. 134, no. 2, pp. 277–292, 2004.

[18] G. Dong, J. Wei, and Z. Chen, "Kalman filter for onboard state of charge estimation and peak power capability analysis of lithium-ion batteries," *J. Power Sources*, vol. 328, pp. 615–626, 2016.

[19] Z. Chen, Y. Fu, and C. Mi, "State of charge estimation of lithium-ion batteries in electric drive vehicles using extended Kalman filtering," *IEEE Trans. Veh. Technol.*, vol. 62, no. 3, pp. 1020–1030, Mar. 2013.

[20] G. Dong, J. Wei, C. Zhang, and Z. Chen, "Online state of charge estimation and open circuit voltage hysteresis modeling of LiFePO 4 battery using invariant imbedding method," *Appl. Energy*, vol. 162, pp. 163–171, 2016.

[21] X. Tang, Y. Wang, and Z. Chen, "A method for state-of-charge estimation of $LiFePO_4$ batteries based on a dual-circuit state observer," *J. Power Sources*, vol. 296, pp. 23–29, 2015.

[22] Y. Wang, C. Zhang, and Z. Chen, "A method for state-of-charge estimation of $LiFePO_4$ batteries at dynamic currents and temperatures using particle filter," *J. Power Sources*, vol. 279, pp. 306–311, 2015.

[23] X. Liu, J. Wu, C. Zhang, and Z. Chen, "A method for state of energy estimation of lithium-ion batteries at dynamic currents and temperatures," *J. Power Sources*, vol. 270, pp. 151–157, 2014.

[24] L. Zhong, C. Zhang, Y. He, and Z. Chen, "A method for the estimation of the battery pack state of charge based on in-pack cells uniformity analysis," *Appl. Energy*, vol. 113, pp. 558–564, 2014.

[25] S. Piller, M. Perrin, and A. Jossen, "Methods for state-of-charge determination and their applications," *J. Power Sources*, vol. 96, no. 1, pp. 113–120, 2001.

[26] S. Kassam, *Signal Detection in Non-Gaussian Noise*. Berlin, Germany: Springer, 2012.

[27] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[28] Y. Chen, D. Huang, D. Feng, and K. Wei, "An h filter based approach for battery SOC estimation with performance analysis," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, 2015, pp. 1613–1618.

[29] K. Zhang, B. Jiang, P. Shi, and J. Xu, "Analysis and design of robust fault estimation observer with finite-frequency specifications for discrete-time fuzzy systems," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1225–1235, Jul. 2015.

[30] L. Qiu, Y. Shi, F. Yao, G. Xu, and B. Xu, "Network-based robust control for linear systems with two-channel random packet dropouts and time delays," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1450–1462, Aug. 2015.

[31] L. Zhang, Y. Zhu, P. Shi, and Y. Zhao, "Resilient asynchronous filtering for Markov jump neural networks with unideal measurements and multiplicative noises," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2840–2852, Dec. 2015.

[32] Z. Zhang, C. Lin, and B. Chen, "New decentralized filter design for nonlinear interconnected systems based on Takagi–Sugeno fuzzy models," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2914–2924, Dec. 2015.

[33] M. Chen, G. A. Rinc ón-Mora, "Accurate electrical battery model capable of predicting runtime and IV performance," *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504–511, Jun. 2006.

[34] Y. Hu, S. Yurkovich, Y. Guezennec, and B. Yurkovich, "A technique for dynamic battery model identification in automotive applications using linear parameter varying structures," *Control Eng. Pract.*, vol. 17, no. 10, pp. 1190–1201, 2009.

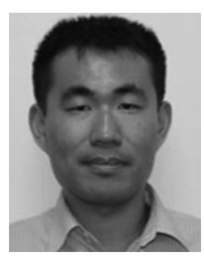
[35] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, vol. 15. Philadelphia, PA, USA: SIAM, 1994.

[36] M. Grant, S. Boyd, and Y. Ye, *CVX: Matlab Software for Disciplined Convex Programming*. Austin, TX USA: CVX Research, Inc., 2008.

Shaheer Muhammad received the B.Sc. degree in electrical engineering from the University of Engineering, Taxila, Pakistan, in 2011, and the M.S. degree in electrical engineer from Air University, Islamabad, Pakistan, in 2014. He has been working toward the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, since 2016.

He is a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University. His research interests include embedded systems, robotics, and control systems.

Muhammad Usman Rafique received the bachelor's and master's degrees in mechatronics engineering from the National University of Sciences and Technology, Karachi, Pakistan, in 2007 and 2010, respectively. He is working toward the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.



Shuai Li received the B.E. degree in precision mechanical engineering from Hefei University of Technology, Hefei, China, in 2005, the M.E. degree in automatic control engineering from the University of Science and Technology of China, Hefei, in 2008, and the Ph.D. degree in electrical and computer engineering from Stevens Institute of Technology, Hoboken, NJ, USA, in 2014.

After graduation, he joined the Hong Kong Polytechnic University as a Research Assistant Professor. His research interests include distributed estimation, optimization and control of networked systems, kinematic and dynamic analysis of robotic systems, and neural networks.

Dr. Li is on the Editorial Boards of *Neural Computing and Applications* and the *International Journal of Distributed Sensor Networks*.

Zili Shao received the B.E. degree in electronic mechanics from the University of Electronic Science and Technology of China, Sichuan, China, in 1995, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Dallas, Dallas, TX, USA, in 2003 and 2005, respectively.

He has been an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, since 2010. His research interests include embedded software and systems, storage systems, and related industrial applications.

Qixin Wang received the B.E. and M.E. degrees in computer science and technology from Tsinghua University, Beijing, China, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2008.

He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. He has published around 15 first/lead author refereed papers on top journals and conferences, and a total of around 40 papers/articles on various academic venues. His main research interests include cyber-physical systems, real-time and embedded systems, and wireless network reliability.

Dr. Wang received an IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS Best Paper Award in 2008, and has one paper chosen as the featured article by the IEEE TRANSACTIONS ON MOBILE COMPUTING 2008 May issue. He has served many top conference TPCs and is serving the Editorial Board of Taylor & Francis's journal *Cyber-Physical Systems*. He is a member of the ACM.

Nan Guan received the B.E. and M.S. degrees in computer science from Northeastern University, Shenyang, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from Uppsala University, Uppsala, Sweden, in 2013.

He worked as a Faculty Member with the Institute of Embedded Systems, Northeastern University. He is currently an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include real-time embedded systems and cyber-physical systems.

Dr. Guan received the EDAA Outstanding Dissertation Award in 2014, the Best Paper Award of IEEE Real-Time Systems Symposium in 2009, the Best Paper Award of International conference on Design, automation and test in Europe in 2013, and the Best Poster Award at the Ph.D. forum of IEEE International Parallel and Distributed Processing Symposium in 2012. He was the Program co-Chair of EMSOFT 2015, and has served for the TPC of RTSS, Design Automation Conference, real-time and embedded technology and applications symposium, International Conference on Embedded Software, Euromicro Conference on Real-Time Systems, Conference on Languages, Compilers, and Tools for Embedded Systems, Conference on Embedded and Real-Time Computing Systems and real-time computing systems and applications, and Asia and South Pacific Design Automation Conference in recent years.