

# A Universal State-of-Charge Algorithm for Batteries

Bingjun Xiao                      Yiyu Shi                      Lei He  
 Electrical Engineering Dept., University of California, Los Angeles  
 Los Angeles, California, 90095, U.S.

## ABSTRACT

State-of-charge (SOC) measures energy left in a battery, and it is critical for modeling and managing batteries. Developing efficient yet accurate SOC algorithms remains a challenging task. Most existing work uses regression based on a time-variant circuit model, which may be hard to converge and often does not apply to different types of batteries. Knowing open-circuit voltage (OCV) leads to SOC due to the well known mapping between OCV and SOC. In this paper, we propose an efficient yet accurate OCV algorithm that applies to all types of batteries. Using linear system analysis but without a circuit model, we calculate OCV based on the sampled terminal voltage and discharge current of the battery. Experiments show that our algorithm is numerically stable, robust to history dependent error, and obtains SOC with less than 4% error compared to a detailed battery simulation for a variety of batteries. Our OCV algorithm is also efficient, and can be used as a real-time electro-analytical tool revealing what is going on inside the battery.

**Categories and Subject Descriptors:** B.8.2 [Hardware]: -Performance and Reliability - Performance Analysis and Design Aids

**General Terms:** Algorithms, Modeling

**Keywords:** Battery, State of Charge, Circuit Analysis

## 1. INTRODUCTION

Renewable energy has become a national goal for the United States. It has been anticipated that by 2015 10% of the total energy consumption in the nation will come from renewable sources, and the number will increase to 25% by 2025. One bottleneck is energy storage, as the peak time of energy harvesting is not necessarily the same as that of energy consuming. The battery is probably the most widely used energy storage device [1,2]. Despite its ever-increasing importance, many challenges remain unsolved to characterize and manage the battery. Among them, one fundamental issue is the estimation of state-of-charge (SOC).

SOC, represented in percentage, indicates the amount of energy available in a battery. It is needed for controlling battery-powered systems. For example, the battery of a

hybrid car should be charged with  $20\% \leq \text{SOC} \leq 95\%$  [3]. It is also used in dynamic power management [4], battery scheduling [5,6], battery-efficient communication traffic shaping and routing [7] for circuits and IT systems.

There have been many attempts in literature to estimate SOC by synthesizing circuit models based on measured voltage and current at battery terminals. For example, [8] uses an extended Kalman filter, [9] employs a circuit model containing resistances, capacitors and diodes, and [10] combines voltage translation via a first-order RC network and Coulomb counting by integrating current over time. The circuit models have to be time variant as battery SOC has a strong history and temperature dependency.

However, these circuit models that mimic the behavior of the battery (see Fig. 2(a)-2(b)) face the following problems: First, the simplified models cannot exactly match the performance of the battery. The estimation error gets increasingly larger until the algorithm breaks down. Second, to perform real-time estimation, initial model parameters must be set properly to ensure the convergence of the algorithm. These parameters are quite sensitive to battery types and discharge current profiles, and can affect the algorithm robustness significantly. Researchers have been well aware of these problems for decades. To avoid using circuit models, a recent work [11] applied a radial basis function (RBF) neural network. However, the parameters in the network still need to be properly chosen to guarantee the convergence, and there is no guarantee of accuracy, stability or robustness.

In contrast, we develop in this paper a universal yet efficient SOC algorithm by system analysis in the frequency domain without using any circuit models for batteries. We directly extract the battery open-circuit voltage from the voltage and current at the battery terminals. Then SOC can be inferred from OCV, a well-established practice in the field. The only assumption we make is that the SOC is constant within a time window of certain width and the battery is a linear or weakly non-linear system. The advantages of our OCV calculation are obvious:

1. It does not require any off-line training or initial setup, and can be applied to any battery type with any discharge current profile.
2. Even if a large estimation error has occurred at certain time due to unexpected environmental change, our algorithm can quickly converge to the correct SOC.
3. The complexity of the algorithm is low and it can be used as a real-time electro-analysis tool for battery diagnosis.

Experimental results show that the SOC can be extracted

\* Address comments to lhe@ee.ucla.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC'10, June 13-18, 2010, Anaheim, California, USA

Copyright 2010 ACM 978-1-4503-0002-5 /10/06...\$10.00

on-line with less than 4% error for different battery types and discharge current.

The remainder of the paper is organized as follows. We introduce the background about OCV and SOC in Section 2, and propose our algorithm in Section 3. Section 4 presents the implementation details and analyzes the proposed algorithm. We present experiments in Section 5 and conclude in Section 6.

## 2. BACKGROUND

[10] is a good example of the general framework for SOC estimation by combining two types of estimation methods, i.e.,

$$\text{SOC} = \alpha \text{SOC}_c + (1 - \alpha) \text{SOC}_v, \quad (1)$$

where  $\text{SOC}_c$  is the Coulomb-counting based SOC and  $\text{SOC}_v$  is the voltage-based SOC.  $\alpha \in [0, 1]$  is the weight factor.  $\text{SOC}_c$  is estimated based on the amount of charge that has been extracted from the battery, and can be simply calculated as

$$\text{SOC}_c(t) = \text{SOC}_c(0) - \frac{1}{Q} \int_0^t I(t) dt, \quad (2)$$

where  $Q$  is some constant to relate the current with charges.

$\text{SOC}_v$  is an estimation based on the open-circuit voltage (OCV) of the battery, which is defined as the voltage between the anode and cathode of the battery when there is no external load connected and no external current between the terminals, even though there may be current internally (e.g., self-discharge current). It has been shown in many studies that there is a time-independent bijection between OCV and SOC. For example, in [8, 12], the relationship between OCV and SOC is given as

$$\text{OCV}(t) = a \text{SOC}(t) + b, \quad (3)$$

where  $a$  and  $b$  are some constants determined by the measurement. A more accurate description, as shown in Fig. 1, is discussed in [13].

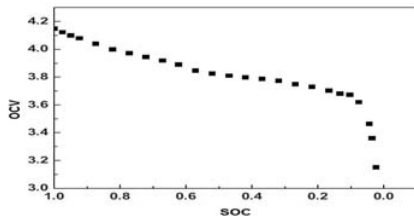


Figure 1: Dependency of OCV on SOC for a Lithium Ion Battery. [13]

Clearly,  $\text{SOC}_c$  has a strong dependency on history, and  $\text{SOC}_v$  is less dependent and is therefore needed to avoid error accumulation over time and to correct major upsets in  $\text{SOC}_c$ . However, it may not be feasible to disconnect the battery from the load and measure its OCV from time to time. Therefore, it is necessary to find out some alternative approaches that can obtain the OCV directly from measured voltage and current data. Towards this end, many different approaches have been proposed, all of which are based on certain types of simplified battery models.

For example, [10] uses a simplified RC pair as shown in Fig. 2(a) in connection with a resistor and a voltage source to model the battery.  $R$  is the battery resistance and  $R_{ct}$

in series with  $C_o$  approximate mass-transfer impedance. All the elements are time-variant according to SOC, and are obtained by online regression from the measured voltage and current. Note that the voltage source value is exactly the OCV. Unfortunately, the model ignores many effects such as the Warburg potential which models the diffusion [14], and it only works well for certain cases. Similarly, [13] and [9] use the models in Fig. 2(b) and Fig. 2(c) respectively. In Fig. 2(b),  $C$  models the chemical diffusion of the electrolyte within the battery (i.e., it is not a purely electrical capacitance); The resistances  $R_b$ ,  $R_c$  and  $R_d$  model all forms of energy loss in the battery (i.e., electrical and nonelectrical losses); The diodes allow for  $R_c$  to be the value of resistance for charging and  $R_d$  the value of resistance for discharging and have no physical significance in the battery except for modeling purposes only. In Fig. 2(c),  $R_s$  represents the ohmic resistance of the battery, including the electrolyte, electronic contacts, particle-to-particle contact resistance, etc.;  $R_n$  and  $C_n$  describe the slow migration of  $\text{Li}^+$  through the surface films;  $R_{ct}$  and  $C_{dl}$  represent the Faradaic charge transfer resistance and double-layer capacitance of the electrodes, respectively; The diffusion impedance  $Z_W$  of the anode and cathode models Warburg diffusive behavior. Both models are too complicated and yet cannot guarantee the convergence. In general, their models are developed for specific types of batteries, and parameters need to be tuned for individual battery due to process variation.

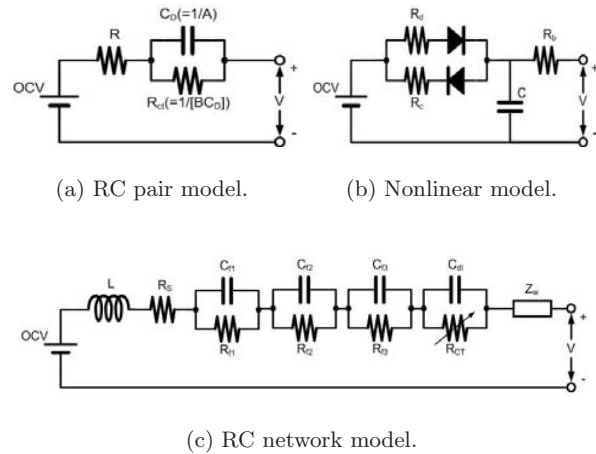


Figure 2: Examples of battery models used in literature.

In the remainder of the paper, we will propose a universal approach to extract OCV and also demonstrate its accuracy, efficiency, robustness and universal applicability.

## 3. ALGORITHM

Batteries are in general considered as a slowly time-varying nonlinear system with voltage source [9, 10, 13]. Within a short time window, we can assume that it can be treated as a time-invariant linear system, and the SOC and accordingly the OCV can be treated as a constant. As such, we propose to use a window based OCV extraction technique. For the simplicity of presentation, although all functions to be discussed in this section are time-discrete, we will loosely use time-continuous notations (e.g.  $v(t)$ ) and explicitly ex-

press them as discrete samples (e.g.  $v(t_1), v(t_2), \dots$ ) only when necessary. To avoid confusion, we use  $*$  to denote convolution and  $\times$  to denote multiplication.

From the superposition theorem, in each time window the terminal voltage  $v(t)$  of a battery can be decomposed as

$$v(t) = v_{zi}(t) + v_{zs}(t) \quad (4)$$

$$= v_{zi}(t) + h(t) * i(t) \quad (5)$$

where  $v_{zi}(t)$  is the zero-input response corresponding to the terminal voltage with no discharge current, and  $v_{zs}(t)$  is the zero-state response corresponding to the terminal voltage with discharge current  $i(t)$  as input and the voltage source shorted.  $h(t)$  is the impulse response of the linear system modeling the battery. Note that the validity of the convolution in (5) is based on the assumption of linearity. All existing works try to explicitly express  $h(t)$  as some analytical functions obtained from the detailed battery model. In this paper, we will directly extract OCV without using any model.

At this moment, we assume that we are extracting the SOC in the time window  $0 \leq t \leq t_w$ , and at  $t < 0$  the discharge current is always zero. This assumes that before  $t = 0$  the battery is disconnected from the load. We will remove this assumption later as the window is shifted. With this assumption and ignoring the self-discharge effect, the zero-input response is actually the open-circuit voltage, i.e.,

$$v_{zi}(t) = \text{OCV} \times u(t), \quad 0 \leq t \leq t_w, \quad (6)$$

where  $u(t)$  is a unit step function

$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (7)$$

The multiplication of  $u(t)$  indicates that we are only interested in  $t \geq 0$  region. It will facilitate the algorithm later.

To see how this can be done, we first search for an  $f(t)$  which satisfies

$$f(t) * i(t) = \delta(t) \quad 0 \leq t \leq t_w, \quad (8)$$

where  $\delta(t)$  is the Dirac delta function, i.e.,

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases} \quad (9)$$

Note that we require  $f(t)$  to satisfy (8) only in the window, and the equality does not need to hold outside it.

The time-discrete algorithm to solve for  $f(t)$  is illustrated in Alg. 1, where  $n$  is the total number of sampling points in the window and  $t_1, t_2, \dots, t_n$  are the sampling time points. The key idea is to “inverse-convolute” the samples. The process is similar to that of solving the inverse of a matrix using elementary transformation. Again, we require the convoluted result to be equal to the delta function only in the window.

With  $f(t)$ , we can compute  $v_f(t) = f(t) * v(t)$  from (4)-(7) as

$$v_f(t) = f(t) * v(t) \quad (10)$$

$$= f(t) * [\text{OCV} \times u(t) + v_{zs}(t)] \quad (11)$$

$$= \text{OCV} \times u_f(t) + f(t) * v_{zs}(t) \quad (12)$$

$$= \text{OCV} \times u_f(t) + f(t) * i(t) * h(t) \quad (13)$$

$$= \text{OCV} \times u_f(t) + \delta(t) * h(t) \quad (14)$$

$$= \text{OCV} \times u_f(t) + h(t) \quad 0 \leq t \leq t_w, \quad (15)$$

where  $u_f(t)$  is marked as

$$u_f(t) = f(t) * u(t). \quad (16)$$

---

**Algorithm 1** The algorithm to calculate  $f(t)$ .

---

```

1: INPUT: Sampled  $i(t_i), 0 \leq t_1 < \dots < t_n \leq t_w$ 
2: OUTPUT:  $f(t_i), 0 \leq t_1 < \dots < t_n \leq t_w$ 
3: for  $j = 1$  to  $n$  do
4:    $f_{norm}(t_j) = f(t_j) = \delta(t_j)/i(t_1)$ 
5:    $i_{norm}(t_j) = i_f(t_j) = i(t_j)/i(t_1)$ 
6: end for
7: for  $i = 2$  to  $n$  do
8:   for  $j = n$  to  $i$  do
9:      $f(t_j) = f(t_j) - f_{norm}(t_{j-i+1})i_f(t_i)$ 
10:     $i_f(t_j) = i_f(t_j) - i_{norm}(t_{j-i+1})i_f(t_i)$ 
11:   end for
12: end for

```

---

The frequency-domain response of the battery can be considered as finite as reported by measured response spectrum [13]. Then according to the final value theorem [15], we have

$$\lim_{t \rightarrow \infty} h(t) = \lim_{s \rightarrow 0} sH(s) = 0. \quad (17)$$

Accordingly,

$$\lim_{t \rightarrow \infty} \frac{v_f(t)}{u_f(t)} = \text{OCV}. \quad (18)$$

This means that when we use a large  $t$ ,  $h(t)$  approaches zero and  $\frac{v_f(t)}{u_f(t)}$  gives a good approximation of OCV in the current time window. In practice, we choose a time point where  $u_f(t)$  reaches its maximum (so that the error introduced by non-zero  $h(t)$  is minimized) in the window to evaluate OCV.

After the extraction of OCV, we can acquire the impulse response of the system in the current time window

$$h(t) = v_f(t) - \text{OCV} \times u_f(t). \quad (19)$$

Note that  $h(t)$  obtained from the above equation can only be accurate when  $t$  is small ( $|h(t)|$  is large). Since we cannot really calculate OCV at  $t = \infty$ , the error in obtained OCV can become significant at large  $t$  when  $h(t)$  gets close to zero. In practice, we simply set  $h(t)$  to zero when  $|h(t)| \leq \epsilon$  where  $\epsilon$  is a small positive value.

After finishing the OCV extraction in the current window, we can repeat the same process to extract the OCV in the next window. However, it is necessary to remove the assumption that the discharge current is always zero before the window. The history discharge current will impact the zero-input response in the current window, as the battery cannot return to steady state immediately. Accordingly, we need to remove from the samples the voltage response caused by the history current in the previous window. Since we already have the impulse response in the previous window (19) as well as the discharge current in the current window, the removal can be conducted as follows:

$$v'(t) = v(t) - \int_{t-t_e}^{t_w} i(\tau)h(t-\tau)d\tau \quad (20)$$

where  $t_e$  is the end time of the previous window. After this process, the system can be modeled again as a system with zero-input before the current time window. We can repeat the OCV algorithm described above by using  $i(t)$  and  $v'(t)$  instead of  $v(t)$  to extract the OCV as well as to update the impulse response.

Obviously, the only situation where the above algorithm will fail in is that  $u_f(t)$  also converges to zero as  $t$  approaches infinity. If  $u_f(t)$  is always zero for  $t \geq t_0$  with some  $t_0$ , or to be more specific, if  $u_f(t_i)$ , the time-discrete samples of  $u_f(t)$ , are zeros for  $i \geq 2$ , then the proposed algorithm

breaks down. However in this case the corresponding discharge current is actually constant and has component only at  $s = 0$  in the frequency domain. It becomes impossible to extract OCV for any voltage-based extraction methods. Fortunately, the battery model becomes almost a resistance network when the discharge current is approximately constant. And OCV can be simply calculated as

$$\text{OCV} = V(t) - I(t)R_{eff}, \quad (21)$$

where  $R_{eff}$  is the effective terminal resistance. There is a unique mapping between the  $R_{eff}$  and the SOC. Accordingly, from the previous extracted SOC and assuming that  $SOC$  and  $R_{eff}$  vary slowly with time, we can obtain the  $R_{eff}$  in the current window from table-look-up. Then from (21) the new OCV can be obtained.

## 4. IMPLEMENTATION AND ANALYSIS

### 4.1 Further Speedup

In the algorithm proposed above, the bottleneck of runtime is mainly in the step to solve  $f(t) * i(t) = \delta(t)$  for  $f(t)$  and the following step to calculate  $v_f(t) = f(t) * v(t)$  and  $u_f(t) = f(t) * u(t)$ . Actually these two steps can be combined into one and there is no need to explicitly calculate  $f(t)$ . The overall algorithm is shown in Alg. 2, where  $n$  is the total number of sampling points in one window. For the simplicity of presentation, we still use the first time window  $0 \leq t \leq t_w$  as an example.

---

**Algorithm 2** The algorithm to combine the steps of deconvolution and convolution.

---

```

1: INPUT: Sampled  $i(t_i), v(t_i), 0 \leq t_1 < \dots < t_n \leq t_w$ 
2: OUTPUT:  $v_f(t_i), u_f(t_i), 0 \leq t_1 < \dots < t_n \leq t_w$ 
3: for  $j = 1$  to  $n$  do
4:    $v_{norm}(t_j) = v_f(t_j) - v(t_j)/i(t_1)$ 
5:    $u_{norm}(t_i) = u_f(t_i) - u(t_i)/i(t_1)$ 
6:    $i_{norm}(t_j) = i_f(t_j) - i(t_j)/i(t_1)$ 
7: end for
8: for  $i = 2$  to  $n$  do
9:   for  $j = n$  to  $i$  do
10:     $v_f(t_j) = v_f(t_j) - v_{norm}(t_{j-i+1})i_f(t_i)$ 
11:     $u_f(t_j) = u_f(t_j) - u_{norm}(t_{j-i+1})i_f(t_i)$ 
12:     $i_f(t_j) = i_f(t_j) - i_{norm}(t_{j-i+1})i_f(t_i)$ 
13:   end for
14: end for

```

---

Furthermore, the time complexity of the algorithm is  $O(n^2)$ , where  $n$  is the number of samples. Note that both of the two for-loops with the variable  $j$  can be computed in parallel and thus it is possible to reduce the complexity further to  $O(n)$  with parallel computing platform.

### 4.2 Numerical Stability

The numerical stability of the algorithm depends mainly on whether  $i(t_1)$  is too small to keep the division in Algo. 2 accurate. However we can always select the starting point of the window where  $i(t_1)$  is not so small. The worst case is that the current keeps close to zero so that we cannot find a proper starting point. In this case the state of the system is close to open circuit and we can approximate the OCV with the terminal voltage and therefore the problem is not a concern.

### 4.3 Robustness

Suppose there occurs an error  $\epsilon$  at the  $k$ th extraction of OCV. We denote  $\text{OCV}_{err} = \text{OCV}_{org} + \epsilon$ .  $\epsilon$  spreads to the  $(k+1)$ th extraction via the  $k$ th impulse response  $h_{err}(t) =$

$h_{org}(t) - \epsilon u_f(t)$ . If all of the  $n$  samples in  $h_{err}(t)$  are used to remove the impact of the discharge current in the previous window, then combining (16) and (8), it is easy to see that the error in (20) accumulated via  $v'(t)$  is also bounded by  $\epsilon$ . Accordingly, the error will never accumulate as the window shifts.

In addition, as claimed previously, we directly set  $h(t)$  to zero at large  $t$ , so the error spread on large  $t$  is removed and only that on small  $t$  is spread to the  $(k+1)$ th extraction. This mechanism further speeds up the convergence of the algorithm. Supposing we keep the samples of  $h(t)$  at a ratio of  $\eta$  and also assuming that the error is uniformly distributed on each sample which is an ideal case, the convergence time would be  $t_{cov} = t_w \log_{\eta^{-1}} \frac{\epsilon_{init}}{\epsilon_{end}}$ , where  $\epsilon_{init}$  is the initial error and  $\epsilon_{end}$  is the tolerance bound.

### 4.4 Applicability to Different Battery Types

Clearly, no specific circuit models are assumed by our algorithm. When the algorithm is applied to a battery with new materials, there is no need to adjust any part of our algorithm. This will be further verified by experiments later. Note that a measurement of a complete discharge process is needed to characterize the mapping between SOC and OCV for the new battery.

### 4.5 Inhibition on Non-Ideal Factors

Though we assumed previously that the system should be time-invariant within a window, the algorithm actually has a natural inhibition on the error introduced by this assumption. To illustrate it clearly, we rewrite the equation used to calculate  $v_f(t)$  in a matrix form as

$$\begin{pmatrix} i(t_1) & i(t_2) & \dots & i(t_n) \\ 0 & i(t_1) & \dots & i(t_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & i(t_1) \end{pmatrix} \begin{pmatrix} v_f(t_n) \\ v_f(t_{n-1}) \\ \vdots \\ v_f(t_1) \end{pmatrix} = \begin{pmatrix} v(t_n) \\ v(t_{n-1}) \\ \vdots \\ v(t_1) \end{pmatrix} \quad (22)$$

Since the inverse of an upper-triangular matrix is still upper-triangular, it is easy to infer that the deviation of the voltage response at  $t_i$  due to non-ideal factors,  $\delta v(t_i)$ , will only impact  $v_f(t_i), v_f(t_{i+1}), \dots$ , and all the samples before  $t_i$  are still accurate. Accordingly, the samples at small  $t$  play more critical role than samples at large  $t$  for OCV estimation. And the deviation of  $v(t)$  at small  $t$  due to non-ideal factors is also small. This concludes that our algorithm can still be accurate even though the system is slowly varying in the time.

## 5. EXPERIMENTAL RESULT

Extensive experiments are conducted to demonstrate the promising performance of the proposed algorithm. We verify our algorithm via dualfoil5 [16], a popular battery simulator whose simulation result is very close to measurement data. The input of the simulator can be either detailed current waveform, load or power at the terminal of a battery. The material of the battery used in simulation can be chosen from a library. The output of the simulator includes SOC and OCV of the battery as well as the current and voltage waveform at the terminal of a battery. The working platform of the experiment is MATLAB 7.01 running at a 1.73GHz clock frequency.

### 5.1 Accuracy Verification

The accuracy of our algorithm is verified with four common discharge profiles of a battery, i.e., periodical discharge

current similar to the one used in [13], piecewise constant discharge current similar to the one reported in [10], constant load of  $0.2\Omega/m^2$ , and constant power of  $100W/m^2$ . In the experiment we choose MCMB 2528 graphite as the negative active material,  $LiPF_6$  in EC:DMC (liquid) as the electrolyte and  $CoO_2$  (Cobalt dioxide) as the positive active material. Before extraction of this type of battery, the lookup tables to map OCV and the effective terminal resistance  $R_{eff}$  to SOC are built based on simulation data offered by dualfoil5 through a discharge process from fully-charged to 15% SOC with a discharge current of  $20A/m^2$ . We set the length of the window as 6s and the sampling interval as 0.06s (the smallest time step the battery simulator dualfoil5 can reach). The battery is first fully charged before conducting each experiment. The results of SOC extraction via our algorithm are shown in Fig. 3. The figure shows that the

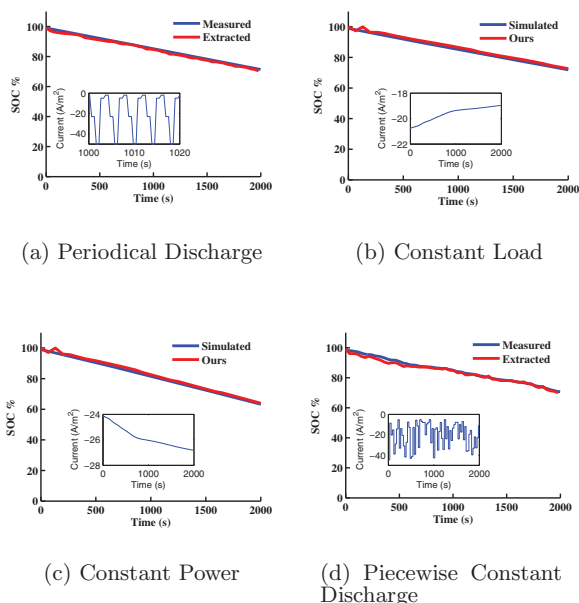


Figure 3: Comparison between our algorithm and simulation for different discharge profiles.

extracted SOC fits well in all the four situations. Our algorithm can, therefore, be used in most operation situations without any manual adjustment which was a big concern in most existing SOC estimation methods as pointed out by [9].

## 5.2 Robustness

We further study the robustness of our algorithm against a sudden burst of error. In Fig. 4, we set the the initial OCV for the algorithm to 3.584V (corresponds to 15% SOC) while the actual SOC at that time point is 100%. All the other experimental settings remain the same as Fig. 3(d). The result is depicted in Fig. 4. From the figure we can see that the algorithm converges quickly to the correct SOC. It shows that voltage-based algorithm is more robust, which is also mentioned in [10].

## 5.3 Runtime

The estimation time of the extraction mainly depends the number of points in the observation window. Given the

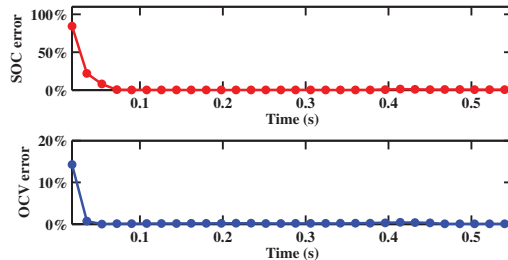


Figure 4: Robustness test. Error of extracted SOC and OCV quickly decreases to nearly zero after several extraction steps.

width of the window, the runtime is then decided by the sampling rate. The sampling rate is often set as high as possible to acquire more data points for more accurate extraction. However the runtime also increases with the increasing sampling rate. The upper limit of the sampling rate can be decided by the condition when the estimation time exceeds the sampling interval such that the extraction process cannot keep in pace with the sampling process. In this experiment, the width of the observation window is set to be 6s and the sampling interval is swept from 0.06s to 0.36s with a step of 0.06s. All the other experimental settings remain the same as Fig. 3(d). The result is shown in Fig. 5. The upper limit of sampling rate is approximately 69

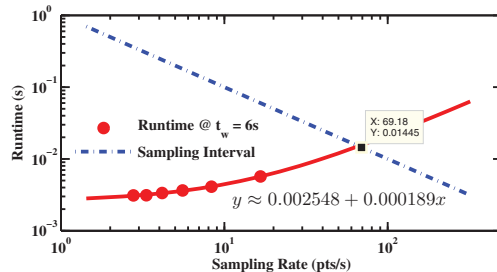
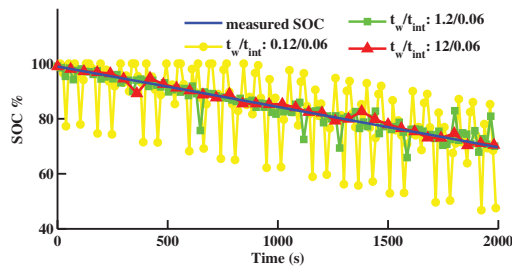


Figure 5: Runtime in logarithm scale. The relationship between runtime and sampling rate is fitted and the upper limit of sampling rate is extrapolated.

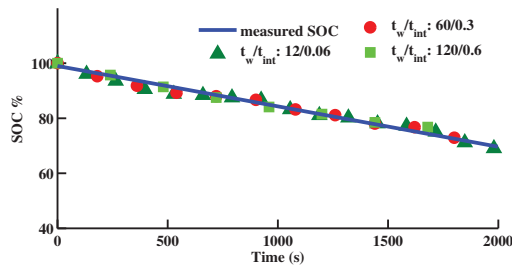
pts/s, which means that SOC can be updated every 14ms. It provides a promising prospect of the realization of online extraction of SOC, e.g. every 2ms, when further speedup can be achieved after the transplantation of the algorithm from MATLAB to C or hardware implementation.

## 5.4 Window Width and Sampling Rate

The selection of a proper window width is critical. It mainly determines the number of samples in a window and the corresponding accuracy within the window. There is a tradeoff between the window width and the accuracy. In this experiment, the sampling rate is set at 0.06s and the window width is tested at 0.12s, 1.2s and 12s, i.e. the number of samples in one window is 2, 20 and 200 respectively. All the other experimental settings remain the same as Fig. 3(d). The result is shown in Fig. 6(a). It shows that the extraction result of SOC becomes inaccurate when the window width is too small. Therefore a proper window width can be decided



(a) Increase window length while keeping sampling rate constant.



(b) Increase window length while keeping number of samples in the window constant.

**Figure 6: The relationship between window width and accuracy.**  $t_w$  is the window width and  $t_{int}$  is the sampling interval.

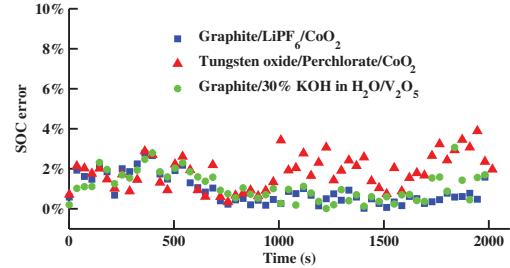
such that there are at least tens of samples in the window to guarantee the accuracy. For example, in Fig. 6(b), the window width with 200 samples always obtains good accuracy for different sampling rates and can be processed in time concerning the result in section 5.3.

### 5.5 Applicability to Different Battery Types

Finally, we study the applicability of our algorithm to different battery types. Three kinds of combinations of battery materials are selected to act as the active positive material, the electrolyte and the negative positive material of a battery. The three combinations are Graphite /  $LiPF_6$  /  $CoO_2$ , which is used in the previous experiment, Tungsten oxide / Perchlorate /  $CoO_2$  and Graphite / 30% KOH in  $H_2O$  /  $V_2O_5$ . For each type of battery, a measurement of a discharge process from fully charged to 15% SOC is conducted and a new look-up table containing SOC, OCV and  $R_{eff}$  is built. No adjustment to cater manually the OCV algorithm to new battery types is used. All the other experimental settings remain the same as Fig. 3(d). The result is shown in Fig. 7. It can be seen that the error is always within an acceptable range of 4% for three different battery types, which proves that our algorithm is universally applicable to different battery types.

## 6. CONCLUSIONS

In this paper, we have proposed a simple yet accurate al-



**Figure 7: Applicability test of our algorithm to different battery types.**

gorithm to calculate open-circuit voltage (OCV) based on terminal voltage and current of the battery. We use frequency domain linear system analysis but not any detailed circuit model. Accordingly, the method can be applied to any kind of discharge current profile and any battery types without modification. Experiments show that SOC can be obtained efficiently based on the well-established mapping between SOC and OCV, with less than 4% SOC error compared to the detailed battery simulation.

## 7. REFERENCES

- [1] L. J. Fu, H. Liu, Y. P. Wu, E. Rahm, R. Holze, and H. Q. Wu, "Surface modifications on electrode materials for lithium ion batteries," *Solid State Sci.*, 2006.
- [2] Y. F. Chen and J. W. Evans, "Three-dimensional thermal modeling of lithium-polymer batteries under galvanostatic discharge and dynamic power profile," *Journal of The Electrochemical Society*, 1994.
- [3] J. R. Pillai and B. Bak-Jensen, "Electric vehicle based battery storages for future power system regulation services," in *Nordic Wind Power Conference*, 2009.
- [4] L. Benini, G. Castelli, A. Macii, and R. Scarsi, "Battery-driven dynamic power management," in *IEEE Design & Test of Computers*, pp. 53–61, 2001.
- [5] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi, "Extending lifetime of portable systems by battery scheduling," in *Proc. European Design and Test Conf. (DATE)*, pp. 197–203, 2001.
- [6] C. F. Chiasserini and R. R. Rao, "Energy efficient battery management," *IEEE J. on Selected Areas in Communications*, pp. 1235–1245, 2001.
- [7] C. F. Chiasserini and R. R. Rao, "Routing protocols to maximize battery efficiency," in *Proc. of Milcom*, pp. 496–500, 2000.
- [8] S. Pang, J. Farrell, J. Du, and M. Barth, "Battery state-of-charge estimation," in *Proc. Amer. Control Conf.*, pp. 25–27, 2001.
- [9] J. Chiasson and B. Vairamohan, "Estimating the state of charge of a battery," *IEEE Transactions on Control Systems Technology*, pp. 465–470, 2004.
- [10] M. Verbrugge and E. Tate, "Adaptive state of charge algorithm for nickel metal hydride batteries including hysteresis phenomena," *Journal of Power Sources*, pp. 236–249, 2004.
- [11] H. Guo and J. J. Z. Wang, "Estimating the state of charge for ni-mh battery in hev by rbf neural network," in *International Workshop on Intelligent Systems and Applications*, pp. 1–4, 2009.
- [12] *Genesis Application Manual, 5th ed.* H. E. P. Inc., 2000.
- [13] P. Moss, G. Au, E. Plichta, and J. P. Zheng, "An electrical circuit for modeling the dynamic response of li-ion polymer batteries," *Journal of The Electrochemical Society*, pp. A986–A994, 2008.
- [14] B. E. Conway, *Electrochemical Supercapacitor: Scientific Fundamental and Technology Applications*. Kluwer Academic/Plenum Publisher, 1999.
- [15] R. H. Cannon, *Dynamics of Physical Systems*. Courier Dover Publications, 2003.
- [16] "FORTRAN Programs for the Simulation of Electrochemical Systems," in <http://www.cchem.berkeley.edu/jsngrp/fortran.html>.