State of Charge Estimation of Lithium-ion Batteries with Particle Filter Algorithm

Fei Xia
College of Automation Engineering
Shanghai University of Electric Power
Shanghai, China
xiafeiblue@163.com

Zhicheng Wang
Zhejiang Power Transmission and Transformation Engineering Co., Ltd.
Hangzhou, China
362766001@qq.com

Chuanlin Zhang*
College of Automation Engineering
Shanghai University of Electric Power
Shanghai, China
clzhang@shiep.edu.cn

Abstract—**The number of electric vehicles has increased significantly in recent years. The battery is its power source, so to estimate its state of charge (SOC) is meaningful. The extended Kalman filter (EKF) algorithm and the unscented Kalman particle filter (UKF) are integrated through analytic hierarchy process and Euclidean distance method as the proposed distribution function of the particle filter (PF) algorithm which is developed into algorithm of improved particle filter. The fusion algorithm is applied to estimate the SOC of the third-order Thevenin model of batteries. Experimental results show that the SOC estimation accuracy of the fusion algorithm is better than that of EKPF and UPF.**

Keywords—SOC estimation, third-order Thevenin model, Analytic hierarchy process, Fusion algorithm

I. INTRODUCTION

Governments and enterprises are paying more attention to the development of electric vehicles. As their power source, batteries are core components. Research on lithium-ion batteries is becoming more important, and state of charge (SOC) estimation is critical. Accurate SOC estimation is related to the safe operation of electric vehicles. It is a core parameter of a battery management system that will balance and match the energy of each battery pack according to its value [1]. The accurate estimation of SOC can effectively prevent over-charging or over-discharging of a battery, thereby extending its service life [2].

The main estimation methods for SOC are ampere-hour integration methods [3], the open-circuit voltage method [4], the neural network method [5], the Kalman filter method [6], and particle filtering algorithms [7]. The ampere-hour integration method can produce inaccurate estimates due to inaccurate initial SOC values and large current fluctuations. As an open-loop prediction method, it will gradually accumulate errors [3]. The open-circuit voltage method requires the battery to stand for a long time; hence, it cannot meet real-time requirements, but it is often combined with other methods to calculate the open-circuit voltage at each moment of rest [4]. The neural network method has the advantage that it requires no prior formulas or mathematical methods [5]. However, it requires much training data, the training time is long [6], and the choice of training data has a great impact on the results.

978-1-7281-5414-5/20/$31.00 ©2020 IEEE

The Kalman filter algorithm can make an optimal estimate that meets the minimum variance for dynamic systems [6], and it is applicable to various types of batteries, but the battery model has a greater impact on the estimation accuracy. The Thevenin model is most often used in SOC estimation of lithium batteries [8]. To improve the estimation accuracy of the EKF algorithm, Yanqing Shen [9] combined it with an adaptive chaos genetic algorithm to estimate SOC, and Wei Kexin [10] used a multi-model adaptive EKF algorithm to estimate SOC. The UKF algorithm improves EKF and uses a traceless transform to deal with the nonlinear transfer of the mean and covariance. Based on the Thevenin model, Shen Yanxia and Zhou Yuan [11] used UKF to estimate the SOC of a lithium-ion battery pack under different temperatures and discharge current conditions. UKF was combined with the open-circuit voltage method and ampere-hour integration method to estimate SOC [12].

Although the Kalman filter algorithm can obtain the optimal estimate of the minimum variance through recursion, it depends greatly on the system model, which must have Gaussian noise. For nonlinear systems such as batteries, it is necessary to use a nonlinear statistical filtering algorithm based on Bayesian estimation and Monte Carlo methods such as a particle filtering algorithm [7]. In this regard, Min Ye [13] used an adaptive particle filter algorithm for SOC estimation. The problem of particle degradation in the estimation process of the PF algorithm is the greatest obstacle to its use. The solution is to adopt a better recommended distribution function or to perform resampling. In this regard, the extended Kalman particle filter algorithm (EKPF) is obtained by using EKF as the recommended distribution of PF.

EKPF uses EKF as the recommended distribution function. It uses Taylor expansion to approximate the system linearly, ignoring higher-order terms that introduce linearization errors in the filtering process and reduce the accuracy of the algorithm. Although UPF uses a traceless transformation to deal with nonlinearity, there is no need to linearize the nonlinear system. UPF is only applicable to Gaussian environments. To this end, we propose the weighted fusion of EKF and UKF as the recommended distribution function of the particle filter algorithm to form a new fusion particle filter algorithm. The analytic hierarchy process is a simple, intuitive, convenient, and practical method to determine the weights of evaluation indicators. However, these weights will be subjective. Therefore, based on the analytic hierarchy process, we introduce Euclidean distance to describe the similarity of weights obtained by experts. Its use to delete the opinions of some experts who are far away can somewhat reduce the subjectivity of weight determination and make the distribution of weights more reasonable. The fusion algorithm uses the weight calculation method of AHP and Euclidean distance to calculate the weights of the EKF and UKF algorithms, and performs weighted fusion to obtain the distribution of the importance of the state recommendations, thereby achieving the estimation of the particle state.

II. PF ALGORITHM COMBINING EKF AND UKF

A. Update with EKF algorithm

Using the EKF algorithm to update the particles at each state point requires linearization of the equation, first-order Taylor expansion of the nonlinear function around the filter value, and defining the expression as shown below:

$$A_k = \frac{\partial f}{\partial x}\Big|_{x=\hat{x}_k} \quad C_k = \frac{\partial g}{\partial x}\Big|_{x=\hat{x}_k} . \tag{1}$$

Then the system equations can be expressed as

$$x_{k+1} = A_k x_k + B_k u_k + w_k \tag{2}$$

$$y_k = C_k x_k + D_k u_k + v_k . \tag{3}$$

The EKF algorithm is used to update the particle state of the linearized discrete nonlinear system, as shown as follows.

First, initialize the particles. Generate particle swarm $\{x_0^i\}_{i=1}^N$ by prior probability $p(x_0)$; all particles have a weight of $1/N$. The initial covariance is $\hat{P}_0^i = \mathrm{var}(x_0)$.

Use EKF to update particles

$$\hat{x}_{k/k-1}^i = A_{k\text{-}1}\hat{x}_{k-1/k-1}^i + B_{k-1}u_{k-1} \tag{4}$$

$$P_{k/k-1}^i = A_{k-1}P_{k-1/k-1}^i A_{k-1}^T + Q_{k-1} \tag{5}$$

$$K_k = P_{k/k-1}^i C_k^T (C_k P_{k/k-1}^i C_k^T + R_k)^{-1} \tag{6}$$

$$\hat{x}_{k/k}^i = \hat{x}_{k/k-1}^i + K_k[y_k - \mathrm{g}(\hat{x}_{k/k-1}^i, u_k)] \tag{7}$$

$$P_{k/k}^i = (I - K_k C_k)P_{k/k-1}^i . \tag{8}$$

The update generates particles

$$x_k^i \sim q(\hat{x}_k^i \,\big|\, x_{k-1}^i, y_k) = N(\hat{x}_k^i, \hat{P}_k^i) . \tag{9}$$

Calculate particle weights

$$\hat{\omega}_k^i = \omega_{k-1}^i \frac{p(y_k \mid \hat{x}_k^i)\, p(\hat{x}_k^i \mid x_{k-1}^i)}{q(\hat{x}_k^i \mid x_{k-1}^i, y_{1:k})} . \tag{10}$$

Normalize weights

$$\tilde{\omega}_k^i = \hat{\omega}_k^i \Big/ \sum_{i=1}^N \hat{\omega}_k^i . \tag{11}$$

B. Update using UKF algorithm

Generate particle swarm $\{x_0^i\}_{i=1}^N$ by prior probability $p(x_0)$, and make $\overline{x_0^i} = E[x_0^i]$,

$$P_0^i = E[(x_0^i - \overline{x_0^i})(x_0^i - \overline{x_0^i})^T],\ \overline{x}_0^{ia} = E(x_0^{ia}) = \begin{bmatrix} \overline{x}_0^i & 0 & 0 \end{bmatrix},$$

$$P_0^{ia} = E[(x_0^{ia} - \overline{x}_0^{ia})(x_0^{ia} - \overline{x}_0^{ia})^T] = \begin{bmatrix} P_0^i & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & R \end{bmatrix}.$$

Update the particles using the UKF algorithm. This includes time and measurement. Select particles

$$x_{k-1}^{ia} = [\overline{x}_{k-1}^{ia} \quad \overline{x}_{k-1}^{ia} + \sqrt{(n_a+\lambda)P_{k-1}^{ia}} \quad \overline{x}_{k-1}^{ia} - \sqrt{(n_a+\lambda)P_{k-1}^{ia}}].$$

Among them, $n_a = n_x + n_w + n_v$, $x_{k-1}^{ia} = [x_{k-1}^{ix} \quad x_{k-1}^{iw} \quad x_{k-1}^{iv}]^T$.

In other words, x_{k-1}^{ia} is the expanded state vector.

Time update

$$x_{k/k-1}^{ix} = A_{k\text{-}1}x_{k-1}^{ix} + B_{k-1}u_{k-1} \tag{12}$$

$$\overline{x}_{k/k-1}^i = \sum_{j=0}^{2n_a} W_j^{(m)} x_{j,k|k-1}^{ix} \tag{13}$$

$$P_{k/k-1}^i = \sum_{j=0}^{2n_a} W_j^{(c)} [x_{j,k|k-1}^{ix} - \overline{x}_{k/k-1}^i][x_{j,k|k-1}^{ix} - \overline{x}_{k/k-1}^i]^T \tag{14}$$

$$y_{k|k-1}^i = g(x_{k|k-1}^{ix}, x_{k|k-1}^{iv}) \tag{15}$$

$$\overline{y}_{k|k-1}^{i} = \sum_{j=0}^{2n_a} W_j^{(m)} y_{j,k|k-1}^{i} . \quad (16)$$

In the above equation, $\overline{x}_{k/k-1}$ is the weighted sum of one-step predictions for all particle points.

Measurement update

$$P_{y_{k|k-1}y_{k|k-1}} = \sum_{j=0}^{2n_a} W_j^{(c)} [\overline{y}_{j,k|k-1}^{i} - \overline{y}_{k/k-1}^{i}][\overline{y}_{j,k|k-1}^{i} - \overline{y}_{k/k-1}^{i}]^T \quad (17)$$

$$P_{x_{k|k-1}y_{k|k-1}} = \sum_{j=0}^{2n_a} W_j^{(c)} [\overline{x}_{j,k|k-1}^{ix} - \overline{x}_{k/k-1}^{i}][\overline{y}_{j,k|k-1}^{i} - \overline{y}_{k/k-1}^{i}]^T \quad (18)$$

$$K_k = P_{x_{k|k-1}y_{k|k-1}} P_{y_{k|k-1}y_{k|k-1}}^{-1} \quad (19)$$

$$\overline{x}_k^i = \overline{x}_{k/k-1}^i + K_k (y_k - \overline{y}_{k|k-1}^i) \quad (20)$$

$$\hat{P}_k^i = P_{k/k}^i + K_k P_{y_{k|k-1}y_{k|k-1}} K_k^T . \quad (21)$$

Sample particles

$$\hat{x}_k^i \sim q(\overline{x}_k^i \left| \overline{x}_{k-1}^i , y_k \right.) = N(\overline{x}_k^i , \hat{P}_k^i) . \quad (22)$$

Calculate the weights

$$\omega_k^i \propto \frac{p(y_k \mid \hat{x}_k^i) p(\hat{x}_k^i \mid x_{k-1}^i)}{q(\hat{x}_k^i \mid x_{k-1}^i , y_{1:k})} , i = 1, 2, \cdots, N . \quad (23)$$

Normalized weight

$$\tilde{\omega}_k^i = \hat{\omega}_k^i \Big/ \sum_{i=1}^{N} \hat{\omega}_k^i . \quad (24)$$

C. Fusion based on AHP and Euclidean distance method

The particle weights of the EKF and UKF algorithms are calculated based on AHP and Euclidean distance, as follows.

- Suppose that n experts independently score the weights of the two algorithms according to some rules of the comparison method, and we use the analytic hierarchy process to obtain the weight of each expert for each algorithm. Let the weights assigned by the i-th expert to the EKF and UKF algorithms be W_{i1} and W_{i2}, respectively, where i = 1, 2, …, n. Then the weight matrix is

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ \vdots & \vdots \\ w_{n1} & w_{n2} \end{bmatrix} . \quad (25)$$

- Calculate the similarity between the weights obtained by the experts. Let D_{ij} denote the distance between the weights of the i-th and j-th experts. Then

$$D_{i,j} = \sqrt{\sum_{k=1}^{2} (w_{ik} - w_{jk})^2} . \quad (26)$$

- Let $B_i = \sum_{j=1}^{n} D_{i,j}$, i = 1, 2, …, n. If $B_t = max\ (B_1, B_2, \ldots, B_n)$, then removing the t-th position. Generally, the above removal steps can be repeated, and the number of removed is generally 20%–30% of the total.

If the number of experts eliminated is c, then re-sort the remaining (n-c) group of expert weights and perform the following optimization process to obtain the weights of each algorithm:

$$M_j = \frac{1}{n-c} \sum_{i=1}^{n-c} W_{i,j} \quad (j=1,2,\ldots,m) . \quad (27)$$

So, the weight of each algorithm is $M=(M_1, M_2)$, and $M_1+M_2 = 1$.

For the weighted fusion of the particles updated by the EKF and UKF algorithms in this paper, let the existing six experts assign the weights to the two algorithms as shown in Table I.

TABLE I. WEIGHTS OF EKF AND UKF ALGORITHMS GIVEN BY VARIOUS EXPERTS

Evaluation expert	**Weight**	
	Weight one w1	***Weight two w2***
Expert one	0.452	0.548
Expert two	0.464	0.536
Expert three	0.466	0.534
Expert four	0.451	0.549
Expert five	0.468	0.532
Expert six	0.455	0.545

Then we can calculate the Euclidean distance between experts according to formula (33). The results are shown in Table II, from which it is seen that B5 = max (B1, B2,..., Bn) = 0.073539, so the recommendation of the fifth expert for weight allocation should be excluded. Therefore, the optimized weights for the EKF and UKF algorithms are M = (0.4576, 0.5424).

Based on the weights of the EKF and UKF algorithms obtained by the weight calculation methods of the above AHP and Euclidean distance, the particles updated by the two algorithms are weighted and fused to obtain a new particle set, and we normalize the weights of this new set.

Resampling the new particle set, we first calculate the effective number of particles and compare it to the threshold to determine whether resampling is required. If the number of effective particles is less than the threshold, then we resample to get a new set of particles. Finally, the state estimation is

$$\hat{x}_k \approx \sum_{i=1}^{N} \tilde{\omega}_k^i x_k^i . \tag{28}$$

TABLE II. EUCLIDEAN DISTANCE BETWEEN EXPERTS

Euclidean distance	***Expert one***	***Expert one***	***Expert two***	***Expert three***	***Expert four***	***Expert five***	***Expert six***	***Total***
Expert one	0	0.0169705	0.0197989	0.0014142	0.0226274	0.0042426	0.065054	0
Expert two	0.0169705	0	0.002828	0.018385	0.005657	0.012728	0.056569	0.0169705
Expert three	0.0197989	0.002828	0	0.021213	0.002828	0.015556	0.062224	0.0197989
Expert four	0.0014142	0.018385	0.021213	0	0.024042	0.005657	0.070711	0.0014142
Expert five	0.0226274	0.005657	0.002828	0.024042	0	0.018385	0.073539	0.0226274
Expert six	0.0042426	0.012728	0.015556	0.005657	0.018385	0	0.056569	0.0042426

III. Experimental Verification And Analysis

When using the PF algorithm combining EKF and UKF to estimate the SOC of a battery in the third-order Thevenin model, SOC estimation is first carried out based on the data collected in the experiment at a constant current discharge of 0.5C. The number of particles N used in the algorithm is 50, and the effective particle threshold is set to 2N/3.

Using this fusion algorithm, the estimated value of the lower-end voltage of the lithium battery model at constant current discharge is shown in Fig. 1. To prove the accuracy of the algorithm, the estimated results of the fusion algorithm, the improve UPF calculation value and the improve EKPF calculation value are compared with reference UI values in Fig. 1.

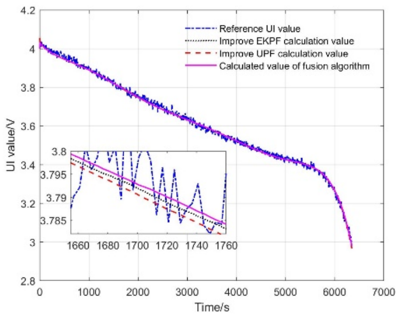

Fig. 1. Voltage estimation value of fusion algorithm under constant current conditions

It can be seen from Fig. 1. that the estimated value of the fusion algorithm is closest to the mean value with the noise reference value, followed by the improved UPF algorithm, and then the improved EKPF algorithm, which shows that the model observations estimated by the fusion algorithm have higher accuracy.

For lithium battery third-order with Thevenin model under constant current conditions, the SOC estimation results of the particle filter algorithm combining the EKF and UKF algorithms are compared with the ampere-hour integration method, EKPF method and UPF method. The comparisons of estimated SOC values is shown in Fig. 2. It can be seen from Fig. 3. that the estimated error values of SOC from the EKPF algorithm, UPF algorithm, and fusion algorithm are compared. To more clearly and intuitivel see the accuracy of SOC estimation by each algorithm, the SOC reference value estimated by the ampere-hour integration method is compared to the SOC value estimated by each algorithm, and the absolute value of the difference is obtained.

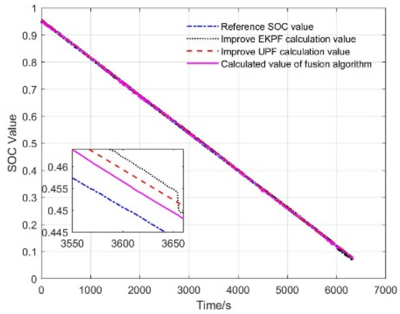

Fig. 2. Estimated SOC value of fusion algorithm under constant current

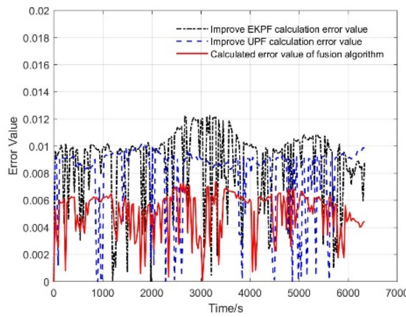

Fig. 3. Estimation error of SOC under constant current condition of fusion algorithm

It can be seen from Fig. 2. and Fig. 3. that the particle filter algorithm using the EKF and UKF algorithms for SOC estimation error is within 0.78%, which is significantly less than the estimation error of EKPF 1.23% , and is also less than the UPF algorithm in constant current SOC estimation error 1.04% under the same conditions. It shows that the particle filter algorithm combining the EKF and UKF algorithms has lower estimation error than that of other algrithm.

IV. CONCLUSION

To more accurately estimate battery SOC, we adopt the analytic hierarchy process and Euclidean distance calculation to combine the EKF and UKF algorithms as the recommended distribution function of the PF algorithm to form a fusion algorithm to estimate the SOC of a lithium battery. To verify the accuracy of the model and the proposed method, the SOC estimation result of the fusion algorithm at 0.5C constant current discharge is compared to the results of the EKPF and UPF algorithms using the EKF and UKF algorithms as the distribution funciton respectively for the PF algorithm. It is found that the error value of the fusion algorithm is the smallest, within 0.78%, which proves its effectiveness.

REFERENCES

[1] Joaquín Klee Barillas, Jiahao Li, Clemens Günther, Michael A. DanzerA comparative study and validation of state estimation algorithms for Li-ion batteries in battery management systems [J]. Applied Energy 155 (2015) 455-462.

[2] Xiaopeng Tang, Yujie Wang, Zonghai Chen. A method for state-of-charge estimation of LiFePO4 batteries based on a dual-circuit state observer [J]. Journal of Power Sources 296 (2015) 23-29.

[3] Hu X, Li S, Peng H, Sun F. Robustness analysis of State-of-Charge estimation methods for two types of Li-ion batteries. J Power Sources 2012, 217:209–19.

[4] Roscher M A, Sauer D U. Dynamic electric behavior and open-circuit-voltage modeling of LiFePO4-based lithium ion secondary batteries [J]. J Power Sources, 2011, 196(1):331-6.

[5] Chao Dong, Guanlan Wang. Estimation of Power Battery SOC Based on Improved BP Neural Network[C]. Proceedings of 2014 IEEE International Conference on Mechatronics and Automation August 3 - 6, Tianjin, China.

[6] Chao Hu, Byeng D. Youn, Jaesik Chung. A multiscale framework with extended Kalman filter for lithium-ion battery SOC and capacity estimation[J]. Applied Energy 92 (2012) 694–704.

[7] HU Shi-qiang, JING Zhong-liang. Overview of particle filter algorithm[J]. Control and Decesion,2005, 20(04):361-365+371.

[8] Hongwen He, Rui Xiong, Xiaowei Zhang, Fengchun Sun. State-of-Charge Estimation of the Lithium-Ion Battery Using an Adaptive Extended Kalman Filter Based on an Improved Thevenin Model[J]. IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 60, NO. 4, MAY 2011.

[9] Yanqing Shen. A chaos genetic algorithm based extended Kalman filter for the available capacity evaluation of lithium-ion batteries[J]. Electrochimica Acta 264 (2018) 400-409.

[10] WEI Kexin, CHEN Qiaoyan. Electric Vehicle Battery SOC Estimation Based on Multiple-model Adaptive Kalman Filter [J]. Proceedings of the CSEE, 2012, 32(31):19-26+214.

[11] SHEN Yan-xia, ZHOU Yuan. State of charge estimation of lithium-ion battery based on unscented Kalman filter [J]. Chinese Journal of Power Sources, 2014, 38(05):828-831.

[12] WANG Qiu-ting, XIAO Duo, JIN Hu. Estimation of state of charge for Li-ion battery based on robust UKF [J]. Chinese Journal of Power Sources, 2013, 37(08):1388-1392.

[13] Min Ye, Hui Guo, Binggang Cao. A model-based adaptive state of charge estimator for a lithium-ion battery using an improved adaptive particle filter[J]. Applied Energy 190(2017): 740-748.