Parameters Optimization of Lebesgue Sampling-based Fault Diagnosis and Prognosis with Application to Li-ion Batteries

Wuzhao Yan[1], Bin Zhang[1], and Marcos Orchard[2]

[1] *Department of Electrical Engineering,*
University of South Carolina, Columbia, SC, 29208, USA
wyan@email.sc.edu
zhangbin@cec.sc.edu

[2] *Department of Electrical Engineering,*
Universidad de Chile, Santiago, Chile
morchard@ing.uchile.cl

ABSTRACT

Lebesgue sampling-based fault diagnosis and prognosis (LS-FDP) is developed with the advantage of less computation requirement and smaller uncertainty accumulation. Same as other diagnostic and prognostic approaches, the accuracy and precision of LS-FDP are significantly influenced by the diagnostic and prognostic models. The predicted results will show great discrepancy with the real remaining useful life (RUL) in applications if the model is not accurate. In addition, the fixed model parameters cannot accommodate the varying stress factors that affect the fault dynamics. To address this problem, the parameters in the models are treated as time-varying ones and are adjusted online to accommodate changing dynamics. In this paper, a recursive least square (RLS) based method with a forgetting factor is employed to make the diagnostic and prognostic models online adaptive in LS-FDP. The design and implementation of LS-FDP are based on a particle filtering algorithm and are illustrated with experiments of Li-ion batteries. The experimental results show that the performance of LS-FDP with model adaptation is improved on both battery capacity estimation and RUL prediction.

1. INTRODUCTION

Condition-based maintenance (CBM) becomes an important function in the modern complex industrial systems to maintain reliability, safety, and availability. CBM utilizes sensing data to monitor the occurrences of faults, estimate the fault state, predict the remaining useful life (RUL), and provide a time-based maintenance decision. The failure prognosis is one of the main challenges in the CBM system design because it projects the current fault state into future time instants to predict the RUL without observations.

Prognostic approaches can generally be categorized into two major classes: model-based (or physics-based) and data-driven methods (Lee, 2007; Jardine, Lin, & Banjevic, 2006). Model-based methods apply mathematical models, which describe the physical mechanism of the fault dynamics to forecast the fault growth. Data-driven methods, on the other hand, employ the collected data to derive the fault growth models. Models for the two methods need to be accurate to ensure the accuracy of the prediction of RUL. However, due to various factors including the lack of understanding of the fault dynamics, the uncertainties from future loading, and measurement noise, a model with fixed parameters is insufficent to accurately describe the effects of varying operational and environmental conditions on fault growth and will result in perforance degradation. One of the solutions to overcome this problem is to adopt the model parameter adaptation in the algorithms, which adjusts the parameters in the model adaptively based on new measurements. The parameter adaptation has been studied in traditional Riemann sampling-based fault diagnosis and prognosis (RS-FDP), in which the samples are taken in a periodic manner. Since fault diagnosis and prognosis (FDP) method based on Lebesgue sampling (LS) has been developed (Zhang & Wang, 2014; Yan, Zhang, Wang, Dou, & Wang, 2016) to overcome the high demands on computational resources, it is necessary to study the parameter adaptation in LS-FDP. LS-FDP divides the state axis by a number of predefined states (also called Lebesgue states). The computation of LS-based FDP is triggered only when the value of feature changes from one Lebesgue state to another, or an event happens. This event-based FDP introduces the characteristics of "execute only when necessary", which significantly reduces the computation demands by eliminating unnecessary computation.

This paper thus develops a model based on data-driven method and integrates it into the failure diagnosis and prognosis architecture. Fault growth models are set in a particle filter (PF) framework where the real-time fault state distribution is compared with its baseline (healthy) counterpart to detect a fault. The PF-based prognostic algorithm is then implemented to achieve the prediction of RUL. During the FDP process, a parameter adaptation algorithm is introduced to optimize the parameters to ensure that the diagnostic and prognostic models can describe the nonlinear fault dynamics accurately.

The paper is organized as follows: Section 2 provides the background of the Lebesgue sampling-based particle filter and the recursive least square method. A case study of lithium ion battery is presented to demonstrate the improvement on the prediction result produced by parameter adaptation in Section 3. Conclusions and future research topics are given in Section 4.

2. BACKGROUND OF LEBESGUE SAMPLING AND RECURSIVE LEAST SQUARE

Traditional PF is developed based on the Riemann sampling framework, and it's proved to be effective to deal with nonlinear systems, especially when the analytical solutions don't exist. PF adopts a set of particles with associated weights to approximate the fault state. The objective is to obtain a new set of particle by propagating the previous particles based on the fault growth model (Gordon, Salmond, & Smith, 1993) and represent the fault state estimation by the new set of particles. The details of PF are described as follows:

Assume a fault state X to be a Markov process with initial distribution $p(x_0)$ and the evolution of the state is defined by $x_k = f_k(x_{k-1}, \omega_k)$ with ω_k being the process noise. The subscript k represents the kth time instant. The measurements Y are assumed to be conditionally independent given X. The measurement model is given as $y_k = h_k(x_k, v_k)$ with v_k as observation noise. Let $x_{0:k} = \{x_0, \cdots, x_k\}$ and $y_{1:k} = \{y_1, \cdots, y_k\}$ represent the state and the available observations up to time k. The *posterior distribution* $p(x_{0:k}|y_{1:k})$ can be achieved by Bayesian rule.

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{1}$$

where $p(y_k|y_{1:k-1})$ is the normalizing constant, $p(y_k|x_k)$ is the likelihood given by the measurement model, $p(x_k|y_{1:k-1})$ is the prior distribution calculated via the Chapman-Kolmogorov equation:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \tag{2}$$

In nonlinear cases, Sequential Monte Carlo (SMC) methods, such as particle filter, are widely used to approximate the *posterior distribution* $\pi_k(x_{0:k}) = p(x_{0:k}|y_{1:k})$. A set of N particles $(w_{k-1}^{(i)}, x_{0:k-1}^{(i)})$ is available to approximate a desired prior distribution $\pi_{k-1}(x_{0:k-1})$, where the superscript $i = 1, 2, \cdots, N$ denotes N particles with location $x_{0:k-1}^{(i)}$ and weight $w_{k-1}^{(i)}$ for the ith particle at the $(k-1)$th time instant. The mission of obtaining the posterior distribution is transferred to efficiently obtain a new set of N particles $(w_k^{(i)}, \bar{x}_{0:k}^{(i)})$, where $\bar{x}_{0:k}^{(i)}$ denotes location of N new particles. In the context of SMC methodology, a Monte Carlo approximation can be obtained as:

$$\pi_k(x_{0:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta\left(x_{0:k} - \bar{x}_{0:k}^{(i)}\right). \tag{3}$$

with $\sum_{i=1}^{N} w_k^{(i)} = 1$, where δ denotes the Dirac-delta function. The weight can be updated in a recursive formula as:

$$\begin{aligned} w\left(\bar{x}_{0:k}^{(i)}\right) &= w_{k-1}^{(i)} h_k\left(y_{1:k}|\bar{x}_{0:k}^{(i)}\right) \quad \text{and} \\ w_k^{(i)} &= \frac{w\left(\bar{x}_{0:k}^{(i)}\right)}{\sum_{i=1}^{N} w\left(\bar{x}_{0:k}^{(i)}\right)}. \end{aligned} \tag{4}$$

2.1. Lebesgue sampling method

With the development of modern technologies, the complexity of system increases rapidly. The utilization of embedded systems in those complex systems design becomes more and more important. Distributed system design is widely accepted in engineering design, especially for complicated systems (Genc & Lafortune, 2007; Qiu, Wen, & Kumar, 2009; Kumar & Takai, 2009; Liu, Qin, & Chai, 2013; Lefebvre, 2014). With this trend, more and more FDP functions are deployed on local processors and embedded systems to alleviate the requirements on communication bandwidth, power, and computation, thus to improve the reliability of the whole system (Schwabacher & Goebel, 2007; Zhang et al., 2011; Chen et al., 2012; Ren, Wang, & Wang, 2015). These embedded systems have very limited computational resources. However, traditional RS-based PF takes samples and executes algorithms in periodic time intervals (Olivares, Cerda Munoz, Orchard, & Silva, 2013; Pola et al., 2015; Xian, Long, Li, & Wang, 2014), which requires significant computational resources. This makes it difficult or even impossible to implement RS-FDP algorithms on the embedded systems.

To overcome this bottleneck, a novel Lebesgue sampling-based FDP framework was developed (Wang & Zhang, 2014; Zhang & Wang, 2014; Yan et al., 2016), in which FDP algorithms are implemented "as-needed". In LS-FDP, the state axis is divided into a number Lebesgue states, as shown in Figure 1. The blue lines represent the Lebesgue state F_i, the LS-FDP is executed only when the value of feature changes from one Lebesgue state to another (an event happens). If the new feature and the previous one stay in the same Lebesgue state, there is no event and the FDP algorithm does not execute. If the new observation changes from one Lebesgue state to another one, there is an event and the algorithm needs to be executed. This philosophy significantly reduces the computation demands by eliminating unnecessary computation.

The model of the fault growth can be described as follows:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D(t_k), \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_a(t_k) \quad (5)$$

where $\hat{a}(t_k)$ is the fault dimension at t_k time instant of the k-th event, $D(t_k)$ is a set of Lebesgue state length adjusted at t_k, which is defined as the distance between the neighboring Lebesgue states: $D_i(t_k) = F_{i+1}(t_k) - F_i(t_k)$, $\dot{\hat{a}}(t_k)$ is the degradation speed, $\omega_a(t_k)$ is the noise, and $f_t(\cdot)$ is a nonlinear function that represents the nonlinear fault growth.

Different from RS-based prognosis, LS-based prognosis is conducted along the state axis to calculate the distributions of operation time to reach the defined Lebesgue states directly. The model for LS-based prognosis is given as:

$$t_{k+1} = t_k + g_t(D(t_k), \hat{a}(t_k), \dot{\hat{a}}(t_k)) + \omega_t(t_k) \quad (6)$$

where $\omega_t(t_k)$ represents the uncertainties, and $g_t(\cdot)$ is a nonlinear function that describes the time distribution of the fault state arriving at each Lebesgue states. The prediction horizon is the number of Lebesgue states from the current Lebesgue state F_d to the state defined as failure threshold F_f. Compared to RS-based prognosis, this prediction horizon is usually small and will significantly reduce the computation cost.

The output of diagnosis is the estimation of fault state. Different from RS-FDP, it cannot be used as the initial distribution for the prognosis in the LS framework. The state distribution needs to be transferred into a time distribution by a short-term prediction method, which is shown in Figure 2. At time t_k when an event happens on the current Lebesgue state F_d, fault diagnosis is executed and each particle is processed to calculate the state probability density function (pdf) at t_k. Note that at time instant t_k, some particles does not reach the current Lebesgue state. To get the time distribution on F_d, a short-term prediction is used for those particles not yet reach F_d. This short-term prediction produces future fault state distribution, which is compared against the current Lebesgue state F_d through the law of total probability to achieve the time distribution for fault state reaching the threshold defined at F_d.

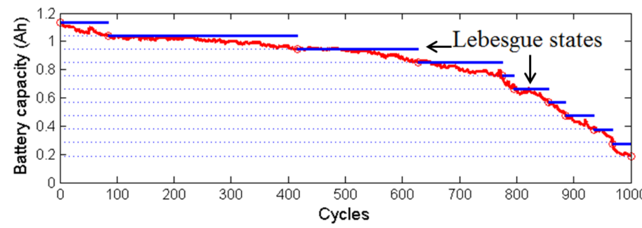

Figure 1. Illustration of LS with fixed Lebesgue state length

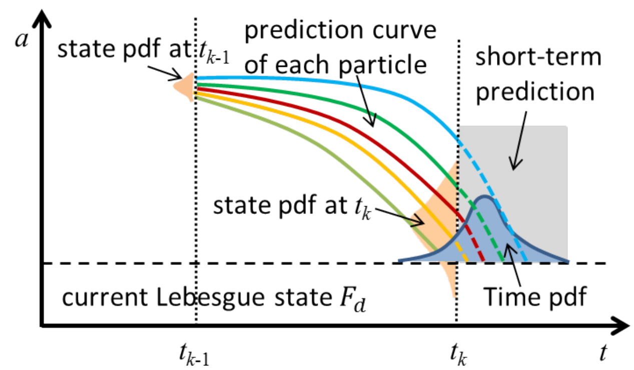

Figure 2. Conversion from state distribution to time distribution

3. APPLICATION WITH LITHIUM-ION BATTERY

In this section, the parameter adaptation based on a particle filtering algorithm is demonstrated with an application to the capacity degradation Lithium-ion batteries. The results are compared against those from PF without parameter adaptation to illustrate the advantages of parameter adaptation. Lithium-ion battery is a safety critical component, and is widely used due to the advantages in high energy density, high cycle life, good resistance to memory, and less weights. Since the life and state of the batteries are not directly observable, diagnosis and prognosis are critical for estimating the battery state (Sidhu, Izadian, & Anwar, 2015; Orchard, Hevia-Koch, Zhang, & Tang, 2013; Pola et al., 2015; Olivares et al., 2013), such as state-of-health (SOH), state-of-charge (SOC), and remaining useful life (RUL).

In this experiment, the SOH of a Lithium-ion battery with 1.1Ah rated capacity under 1.1A charge/discharge capacity is used to verify the proposed parameter adaptation method based on PF. The charge-discharge cycle of the battery is tested by Arbin BT2000 system under room temperature. The charging and discharging of the battery are cut off at the given cutoff voltage. The capacity degradation curve vs charging-discharging cycle is obtain by Coulomb counting. The failure threshold is defined as 0.35Ah and the battery capacity degrades to this value at the 810th cycle.

3.1. Recursive least square method

In the PF-based FDP method, a model is needed to project the current particle set to a future particle set. To make the model adaptive to changing operating conditions and environmental factors that affect the battery degradation, model parameter adaptation is adopted in the FDP process. Several optimization methods have been developed in FDP (Lin, Lee, Chen, & Tseng, 2008; Laosiritaworn & Chotchaithanakorn, 2009; Bergstra, Bardenet, Bengio, & Kégl, 2011; Fu, Fei, Guangming, & Li, 2009). Due to its advantages of simplicity, recursive least square (RLS) is used in this paper. The RLS algorithm is implemented as follows (Zhang et al., 2011):

1. Define a cost function as:

$$J(\theta) = \frac{1}{2} \cdot \sum_{i=1}^{N} \lambda^{N-i} \left[C(i) - C\left(\hat{\theta}(i-1)\right) \right]^2 \tag{7}$$

   where N is the length of the measurement, λ is a forgetting factor, which is usually given in the range of $0 < \lambda \leq 1$, and $\hat{\theta}(i)$ is the parameters to be optimized.

2. Calculate the derivatives $\phi(i)$ with respect to parameters θ:

$$\phi(i) = \left. \frac{dC(\theta)}{d\theta} \right|_{\theta=\hat{\theta}(i-1)} \tag{8}$$

3. Parameters are updated as:

$$\hat{\theta}(i) = \hat{\theta}(i-1) + P(i)\phi(i) \left[C(i) - C\left(\hat{\theta}(i-1)\right) \right] \tag{9}$$

   and $P(i)$ is given as:

$$P(i) = \frac{P(i-1)}{\lambda} \left[1 - \frac{\phi(i)\phi^T(i)P(i-1)}{\lambda + \phi^T(i)P(i-1)\phi(i)} \right] \tag{10}$$

3.2. Models for diagnosis and prognosis

To implement the diagnosis in PF algorithm for the battery capacity degradation, 40 uniformly distributed Lebesgue states are initially defined in the battery's full capacity of 1.1Ah based on the battery capacity and computation resource. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another. A fault growth model is developed by data-driven method, and is given as follows:

$$\begin{aligned} C(t_{k+1}) =& C(t_k) - p_d \cdot C(t_k) \cdot D(t_k) \cdot \\ & sgn(C(t_k) - C(t_{k-1})) + \omega_C(t_k) \end{aligned} \tag{11}$$

where C is the battery capacity, p_d is the model parameter, t_k is the time instant when an event happens, $sgn(\cdot)$ is a sign function that indicates the changing trend, and ω_C is the model noise.

To conduct prognosis in LS framework, the output of diagnosis needs to be transformed into the operation time distribution. The conversion from state distribution to the operation time distribution is achieved as discussed in Figure 2.

LS-based prognosis is conducted along capacity axis to predict the time distribution when the capacity degradation arrives each Lebesgue state directly. The model for prognosis is given as (Yan et al., 2016):

$$t_{k+1} = t_k + p_p \cdot C(t_k) \cdot D(t_k) \cdot exp\left(-\dot{C}(t_k)\right) + \omega_k(t_k) \tag{12}$$

where p_p is the model parameter and ω_k is the model noise.

Note that the Lebesgue length $D(t_k)$ is no longer a constant after the execution of the FDP algorithm since it is adaptively adjusted according to the fault growth speed to monitor the system closer.

3.3. Experimental results comparison

For the diagnosis model, p_d is the target parameter that needs to be optimized. The flow chart of RLS based optimized method is shown in Figure 3. The measurement and the mean of the state estimation from diagnosis is used to produce the error, which is used as the input of the parameter adaptation algorithm and to adjust p_d.

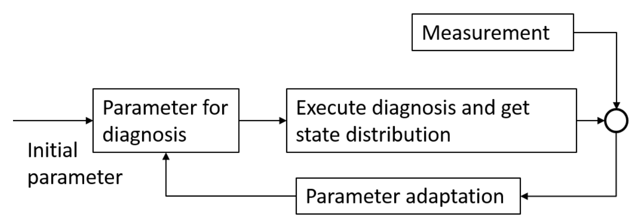

Figure 3. The flowchart of parameter adaptation in the diagnosis process.

Figure 4 shows the diagnostic results with/without parameter adaptation at the 400th cycle. The particle filtering algorithm uses 500 particles to approximate the state distribution. The subfigure (a) shows the comparison of capacity from Coulomb counting (blue) against the estimated mean value from diagnosis (magenta) with/without parameter adaptation. Note that the flat segments mean no event and diagnosis is not executed.

The diagnosis is initially executed with $p_d = 1.2$. The expected value of battery capacity with and without parameter adaptation are 0.9476 and 0.9534, respecitlvey, and the corresponding 95% confidence intervals are $[0.8872,\ 1.0054]$ and $[0.8930,\ 1.0112]$, respecitlvey. As shown in Figure 4 (a), the mean of the diagnosis result from the algorithm with parameter adaptation shows a more accurate state estimation diagnosis. The real-time state distribution against the baseline distributions at the 400th cycle for the algorithm with/without parameter adaptation are shown in Figure 4 (b). It shows that the two distributions do not have significant difference. Note

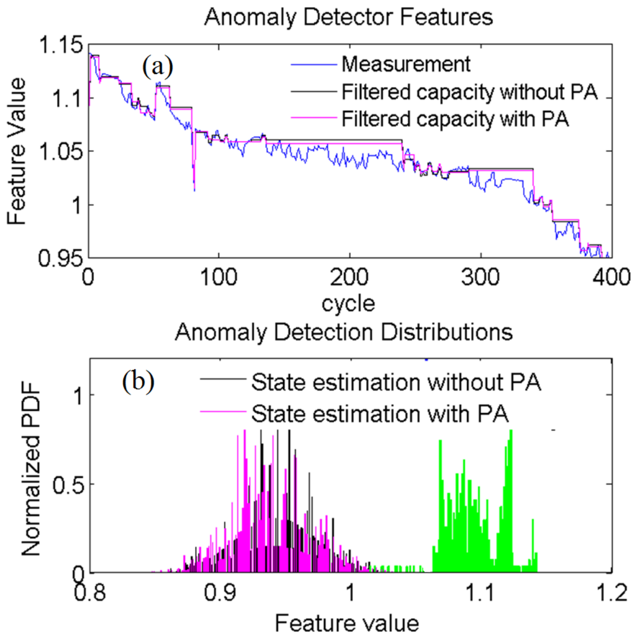

Figure 4. The comparison of diagnosis results with/without parameter adaptation. a): The comparison of capacity from Coulomb counting against the estimated mean values. b): The capacity distributions from the algorithm with the parameter adaptation against the baseline distribution (green).

that, in the past 400 cycles, although 400 measurements are received, there are only 53 events. Therefore, the LS-based diagnosis only runs 53 times. Compared with traditional RS-based diagnosis that needs to run 400 times, the reduction of computation is (400-53)/400=86.8% and computation is 7.55 times faster in running numbers.

Parameter adaptation based on RLS method is also used to adjust the prognosis parameters. The procedure of parameter adaptation in prognosis is shown in Figure 5. The predicted time distributions to reach a selected future Lebesgue states and the real time distribution to reach the same selected Lebesgue state are needed as the reference. The details are listed as follows:

1. At the current Lebesgue state F_C, the prognosis program is executed with the initial parameters. The time distribution for battery capacity degrades to reach the next Lebesgue state F_{C+1} is produced and stored as $\hat{P}r_{C+1}$.
2. When the measurement reaches F_{C+1}, the diagnosis is executed and a state distribution is achieved.
3. The achieved state distribution is converted to a time distribution Pr_{C+1} by the method in Figure 2.
4. The difference of the mean values of $\hat{P}r_{C+1}$ and Pr_{C+1} is used as the error in the RLS optimization, which will generate a new set of parameters to be used in the next prognosis process.
5. Repeat the steps above to optimize the parameters during the FDP process.

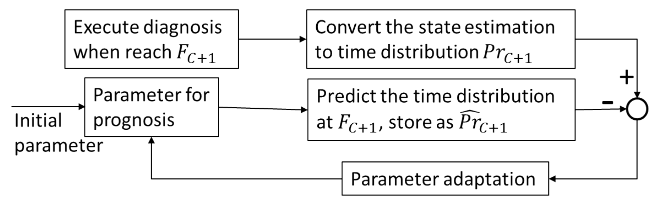

Figure 5. The flowchart of parameter adaptation in prognosis process.

Figure 6 shows the prognostic results with 500 particles at the 400th cycle. The predicted pdfs (magenta) for the fault to reach each Lebesgue state are illustrated. Initially, the parameter in the prognosis model is set to be $p_p = 2$. Figure 6 (a) shows the prognosis results without parameter adaptation in the prognosis algorithm. The predicted time to failure (TTF) for this battery is 1108.8 and the RUL is 708.8 cycles. The 95% confidence interval of the TTF is [1083.0 1134.6]. Compared with the ground truth TTF of 810, the difference is 298.8 cycles. This means that if the model parameters are not accurate due to lack of knowledge or changing operating conditions, the algorithm without parameter adaptation cannot provide an accurate RUL for decision-making.

The prognosis results with parameter adaptation are shown in Figure 6 (b). With the proposed method, the predicted TTF for this battery is 802.8 cycles and the RUL is 402.8 cycles. The 95% confidence interval of the TTF is [788.8 816.7]. Compared with the ground truth TTF of 810, the difference is 7.2 cycles. The prediction accuracy is highly improved from 298.8 cycles to 7.2 cycles by the RLS based parameter adaptation.

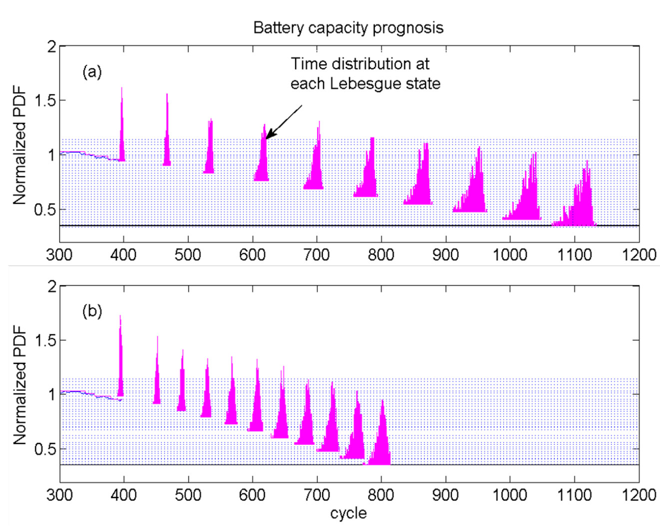

Figure 6. LS-based prognosis at the 400th cycle.

Figure 6 shows that, without parameter adaptation, the prognosis result shows significant prediction error (298.8 cycles). When parameter adaptation is introduced, the prediction accuracy is greatly improved and the prediction error reduces to 7.2 cycle. The reduction of prediction error is (298.8-7.2)/298.8=97.59%.

The comparison of diagnostic and prognostic results of FDP algorithms with/without parameter adaptation are summaried and compared in Table 1. Compared with FDP algorithm without parameter adaptation, the diagnosis result with parameter adaptation has an accurate state estimation, the prognosis with parameter adaptation has a much smaller prediction error (298.8 cycles vs. 7.2 cycles).

Table 1. Comparison of the FDP algorithm with/without parameter adaptation for Battery

Diagnosis results	Without para. adapt.	With para. adapt.
Capacity expectation	0.9476	0.9534
Capacity 95% CI	[0.8930 1.0112]	[0.8872 1.0054]
Execution numbers	53 (100%)	53(100%)
Prognosis results	Without para. adapt.	With para. adapt.
True TTF	810	810
Estimate TTF	1108.8	802.8
95% CI of TTF	[1083.0 1134.6]	[788.8 816.7]

The above comparison of prediction accuracy is only illustrated in Figure 6 at the 400th cycle. To illustrate the improvement of prediction accuracy in the entire battery life, the comparison of prediction accuracy is conducted in terms of $\alpha - \lambda$ metrics (Saxena, Celaya, Saha, Saha, & Goebel, 2010) is adopted.

The metrics is defined as:

$$[1-\alpha]\cdot r_t(t_k) \le r^l(t_k) \le [1+\alpha]\cdot r_t(t_k) \qquad (13)$$

where r^l is the predicted RUL at the lth time instant, r_t is the ground truth RUL, α is the accuracy modifier (Saxena et al., 2010).

Figure 7 shows the α-λ metrics with α=0.3 for the algorithm with and without parameter adaptation. For the result from the algorithm without parameter adaptation, the parameter p_p is set to be 2. Since there is no correction action to adjust the initial p_p, the algorithm is executed with $p_p = 2$ for the whole FDP process. The predicted RUL is longer than the ground truth, which means that the results cannot provide useful information for the CBM and decision-making. On the contrary, the algorithm with parameter adaptation adjusts the initial parameter automatically based on the error between the means of the predicted time distribution and the posterior distribution. At the 77th cycle, the parameter adaptation is triggered for the first time, and p_p is adjusted, which brings the predicted RUL inside the accuracy zone and stay in the zone in the future prediction.

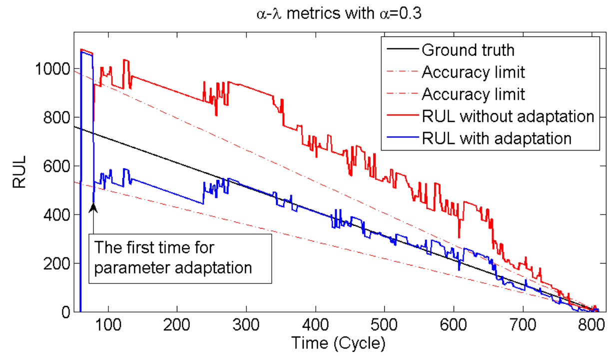

Figure 7. Prognostic accuracy comparison for the algorithm with/without parameter adaptation.

4. CONCLUSIONS

Many parameter adaptation methods have been developed in traditional Riemann sampling-based framework with great success in the past decades. A new Lebesgue sampling-based FDP is introduced with a philosophy of "execution when needed" to reduce the computation. Since the development of LS-FDP there is no research on the parameter adaptation in this area. This paper proposed a parameter adaptation method based on recursive least square to optimize the parameters in diagnostic and prognostic models to accommodate the uncertainties from the unknown future change of fault mechanism. A particle filtering-based algorithm is developed with an application to the SOH of Lithium-ion batteries. Experimental results for LS-FDP with/without parameter adaptation are presented and compared to demonstrate the effectiveness of the proposed parameter adaptation scheme in enhancing the state estimation accuracy and prediction accuracy.

REFERENCES

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems* (pp. 2546–2554).

Chen, C., Brown, D., Sconyers, C., Zhang, B., Vachtsevanos, G., & Orchard, M. E. (2012). An integrated architecture for fault diagnosis and failure prognosis of complex engineering systems. *Expert Systems with Applications*, *39*(10), 9031 - 9040.

Fu, L., Fei, Q., Guangming, S., & Li, Z. (2009, June). Optimization-based particle filter for state and parameter estimation. *Journal of Systems Engineering and Electronics*, *20*(3), 479-484.

Genc, S., & Lafortune, S. (2007, April). Distributed diagnosis of place-bordered petri nets. *Automation Science and Engineering, IEEE Transactions on*, *4*(2), 206-219. doi: 10.1109/TASE.2006.879916

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993, April). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, *140*(2), 107-113. doi: 10.1049/ip-f-2.1993.0015

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Kumar, R., & Takai, S. (2009, July). Inference-based ambiguity management in decentralized decision-making: Decentralized diagnosis of discrete-event systems. *Automation Science and Engineering, IEEE Transactions on*, *6*(3), 479-491. doi: 10.1109/TASE.2009.2021330

Laosiritaworn, W., & Chotchaithanakorn, N. (2009). Artificial neural networks parameters optimization design of experiments: An application in materials modeling.

Lee, J. (2007). A systematic approach for developing and deploying advanced prognostics technologies and tools: methodology and applications. In *Proceedings of the second world congress on engineering asset management, harrogate, uk* (pp. 1195–1206).

Lefebvre, D. (2014, Oct). Fault diagnosis and prognosis with partially observed petri nets. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, *44*(10), 1413-1424. doi: 10.1109/TSMC.2014.2311760

Lin, S.-W., Lee, Z.-J., Chen, S.-C., & Tseng, T.-Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied soft computing*, *8*(4), 1505–1512.

Liu, Q., Qin, S., & Chai, T. (2013, July). Decentralized fault diagnosis of continuous annealing processes based on multilevel pca. *Automation Science and Engineering, IEEE Transactions on*, *10*(3), 687-698. doi: 10.1109/TASE.2012.2230628

Olivares, B., Cerda Munoz, M., Orchard, M., & Silva, J. (2013, Feb). Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena. *Instrumentation and Measurement, IEEE Transactions on*, *62*(2), 364-376. doi: 10.1109/TIM.2012.2215142

Orchard, M., Hevia-Koch, P., Zhang, B., & Tang, L. (2013, Nov). Risk measures for particle-filtering-based state-of-charge prognosis in lithium-ion batteries. *Industrial Electronics, IEEE Transactions on*, *60*(11), 5260-5269. doi: 10.1109/TIE.2012.2224079

Pola, D., Navarrete, H., Orchard, M., Rabie, R., Cerda, M., Olivares, B., ... Perez, A. (2015, June). Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles. *Reliability, IEEE Transactions on*, *64*(2), 710-720. doi: 10.1109/TR.2014.2385069

Qiu, W., Wen, Q., & Kumar, R. (2009, April). Decentralized diagnosis of event-driven systems for safely reacting to failures. *Automation Science and Engineering, IEEE Transactions on*, *6*(2), 362-366. doi: 10.1109/TASE.2008.2009093

Ren, Y., Wang, A., & Wang, H. (2015, March). Fault diagnosis and tolerant control for discrete stochastic distribution collaborative control systems. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, *45*(3), 462-471. doi: 10.1109/TSMC.2014.2358635

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, *1*(1), 20.

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium: Artificial intelligence for prognostics.* Retrieved from `http://www.aaai.org/Library/Symposia/Fall/2007/fs07-02-016.php`

Sidhu, A., Izadian, A., & Anwar, S. (2015, Feb). Adaptive nonlinear model-based fault diagnosis of Li-ion batteries. *IEEE Transactions on Industrial Electronics*, *62*(2), 1002-1011.

Wang, X., & Zhang, B. (2014, Dec). Real-time lebesgue-sampled model for continuous-time nonlinear systems. In *Decision and control (cdc), 2014 ieee 53rd annual conference on* (p. 4367-4372). doi: 10.1109/CDC.2014.7040070

Xian, W., Long, B., Li, M., & Wang, H. (2014, Jan). Prognostics of lithium-ion batteries based on the verhulst model, particle swarm optimization and particle filter. *Instrumentation and Measurement, IEEE Transactions on*, *63*(1), 2-17. doi: 10.1109/TIM.2013.2276473

Yan, W., Zhang, B., Wang, X., Dou, W., & Wang, J. (2016, March). Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, *63*(3), 1804-1812. doi: 10.1109/TIE.2015.2494529

Zhang, B., Sconyers, C., Byington, C., Patrick, R., Orchard, M., & Vachtsevanos, G. (2011, May). A probabilistic fault detection approach: Application to bearing fault detection. *Industrial Electronics, IEEE Transactions on*, *58*(5), 2011-2018. doi: 10.1109/TIE.2010.2058072

Zhang, B., & Wang, X. (2014). Fault diagnosis and prognosis based on lebesgue sampling. In *Annual conference of the prognostics and health management society 2014* (Vol. 5).

BIOGRAPHIES

Wuzhao Yan received his B.A and Ph.D in Department of Physics in University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. After working in Amperex Technology Limited, Dongguan, China, for 3 years, he went to Department of Electrical Engineering in University of South Carolina. Now he focuses on the algorithms of diagnosis and prognosis for lithium ion battery.

Bin Zhang received the B.E. and M.E. degrees from the Nan-

jing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2007. He is currently with the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA. His research interests are prognostics and health management and intelligent systems.

Marcos Orchard received his B.S. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile. He received his M.S. and Ph.D.degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He is currently work as an Associate Professor with the Department of Electrical Engineering at Universidad de Chile, Associate Researcher at the Advanced Mining Technology Center, Associate Researcher at the Advanced Center for Electrical and Electronic Engineering, and Project Leader at the Lithium Innovation Center. Dr. Orchard is an internationally recognized expert in the field of fault diagnosis and prognosis, Bayesian filtering in dynamic non-linear systems, stochastic processes and sequential Monte Carlo methods.