

# Parameters Adaption of Lebesgue Sampling-based Diagnosis and Prognosis for Li-ion Batteries

Wuzhao Yan<sup>1</sup>, Wanchun Dou<sup>2</sup>, Datong Liu<sup>3</sup>, Yu Peng<sup>3</sup>, and Bin Zhang<sup>1</sup>

<sup>1</sup> *Department of Electrical Engineering,  
University of South Carolina, Columbia, SC, 29208, USA*  
wyan@email.sc.edu  
zhangbin@cec.sc.edu

<sup>2</sup> *Department of Computer Science & Technology, State Key Lab. for Novel Software Technology,  
Nanjing University, Nanjing, Jiangsu, 210093, P.R. China*  
douwc@nju.edu.cn

<sup>3</sup> *Department of Automatic Test and Control,  
Harbin Institute of Technology, Harbin, Heilongjiang, 150080, P.R. China*  
Liudatong@hit.edu.cn  
Pengyu@hit.edu.cn

## ABSTRACT

Traditional fault diagnosis and prognosis (FDP) approaches are based on Riemann sampling (RS), in which samples are taken and algorithms are executed in a periodic time interval. With the increase of system complexity, the real-time implementation of this Riemann sampling-based FDP (RS-FDP) becomes a bottleneck, especially for distributed applications. To overcome this problem, a Lebesgue sampling-based FDP (LS-FDP) is proposed. LS-FDP takes samples on the fault dimension axis and provides a need-based diagnostic philosophy in which the algorithm is executed only when necessary. In previous Lebesgue sampling-based FDP, the Lebesgue length is a constant. To accommodate the change of fault dynamics, it is desirable to execute FDP algorithm more frequently when the fault growth is fast while less frequently when fault growth is slow. This requires to change the Lebesgue length adaptively. The goal of this paper is to deliver an improved LS-FDP method with varying Lebesgue length, which enables the FDP to be executed according to needs. The design and implementation of varying Lebesgue length LS-FDP based on a particle filtering algorithm are illustrated with experimental results on Li-ion batteries to verify the performances of the proposed approach. The experimental results show that the new varying LS-FDP is accurate

and time-efficient on long term prognosis and also keeps a closer monitoring on the fast increase of fault size.

## 1. INTRODUCTION

The utilization of embedded systems are increasing in modern vehicle and other complex system design. There are more than 70 distributed microcontrollers (also known as electronic control units-ECUs) in one high-end car (Pattipati, Wang, Zhang, Howell, & Salman, 2011). The ECUs in modern cars perform variety of functions such as stability control, cruise control, oil and coolant monitoring, energy-efficient propulsion, turbo on-and-off, navigation with real-time traffic, and even autonomous driving. To ensure these functions, diagnosis and prognosis are needed to monitor and predict the health condition of safety critical components, such as engine, battery, and transmission.

With the increase of components and subsystems in a complex system, more and more diagnostic and prognostic algorithms are deployed on local processors and embedded systems to alleviate the requirements on the communication bandwidth, power, and computation, thus to improve the reliability of the whole system (Schwabacher & Goebel, 2007; Zhang et al., 2011; Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006). However, these local processors and embedded systems have very limited computational resources. It is difficult or even impossible for traditional fault diagnosis and prognosis (FDP) algorithms to be deployed.

To overcome this bottleneck that prevents the distributed

---

Wuzhao Yan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

FDP in complex systems, Lebesgue sampling-based FDP (LS-FDP) algorithms are developed (Zhang & Wang, 2014), in which a novel FDP philosophy is employed on an “as-needed” basis. It can efficiently reduce the computational cost compared with the traditional RS-FDP. Different from RS-FDP, LS-FDP divides the state axis by a number of pre-defined states (also called Lebesgue states). The FDP will be triggered when the feature value changes from one Lebesgue state to another, or an event happens. After the fault is detected, the prognosis is executed to estimate a distribution of operating time for the fault state reaching each Lebesgue state.

With the characteristic of “execute only when necessary” the computation demands are significantly reduced by eliminating unnecessary computation. When the fault growth is slow, the FDP is executed in a low frequency. While when the fault growth is fast, the FDP is executed more frequently. In previous LS-FDP, the Lebesgue length is constant and fixed, this is not an optimal solution for most fault dynamics that the fault growth is nonlinear. To accommodate the non-linearity of faulty dynamics, it is desirable to adjust the Lebesgue length adaptively and optimally.

Since the Lebesgue states in LS-FDP are selected to ensure the performance for the fastest fault growth scenario, the LS-FDP algorithm can be executed more frequently than necessary when the fault growth is slow. This results in significant over-provisioning of the real-time system hardware. In practice, the system may have multiple fault modes, and different faults have different growth speeds. The computational resources can be dynamically distributed among different FDPs to optimize the performance of microcontroller.

To achieve this goal, the fault growth speed estimated from previous Lebesgue states is used to optimize current Lebesgue state length, from which a new set of Lebesgue states is achieved. Compared with the initial Lebesgue state lengths, the new Lebesgue state lengths are shrunk or stretched, and is re-adjusted every time when prognosis is executed. The prognosis algorithm then predicts the distributions of operating time on these updated Lebesgue states.

The paper is organized as follows: Section 2 provides an overview of the framework of varying Lebesgue length sampling FDP (VLS-FDP). Section 3 develops a particle filtering based VLS-FDP method. A case study based on lithium ion battery is presented to demonstrate the advantages of VLS-FDP in Section 4. Conclusions and future research topics are given in Section 5.

## 2. VLS-FDP

### 2.1. Fault Growth Modeling

Assume that the fault growth model can be described by a continuous-time differential equation:

$$\dot{a} = F(a, u) \quad (1)$$

where  $a$  is the fault dimension,  $u$  is the system control inputs which can influence the fault growth, and  $F(\cdot)$  is a nonlinear function that describes the fault growth under current operating scenario. The feature, denoted by  $y$ , is extracted from testing data and serves as the measurement for the FDP algorithm. To simplify the problem, the mapping between  $y$  and  $a$  is described as  $y = a$ , and  $y$  is employed as the indicator of the fault state, which is the measured real-time battery capacity in the case of battery life prediction.

Lebesgue sampling-based model for the fault growth dynamics in discrete-time can be described as:

$$\hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D, \hat{a}(t_k)) \quad (2)$$

where  $\hat{a}(t_k)$  is the Lebesgue state of fault dimension,  $t_k$  is the  $k$ th sampling instant, and  $D$  is the Lebesgue state length.

### 2.2. Lebesgue Sampling based Diagnosis

The LS-based diagnosis is described as follows:

1. Divide the state range into a series of Lebesgue states  $\{F_1, F_2, \dots, F_f\}$ ;
2. If the feature value  $y$  causes a transition from one Lebesgue state to another (Astrom & Bernhardsson, 2002; McCann & Le, 2008), i.e., an event happens, the diagnostic algorithm is executed. Otherwise update the time stamp and wait;
3. Calculate the current fault state distribution and determine whether a fault is detected.

### 2.3. Lebesgue Sampling based Prognosis

For LS-FDP, the prognosis is conducted along the state axis to calculate the time to failure (TTF) or remaining useful life (RUL) distributions of operation time to reach the defined Lebesgue states. The model for LS-based prognosis is given as:

$$t_{k+1} = t_k + g_t(D, \hat{a}(t_k)) \quad (3)$$

This function describes the time distribution of the fault state to arrive at each Lebesgue states. The prediction horizon is the number of Lebesgue states from the current Lebesgue state to the state defined as failure threshold. Compared to RS-based prognosis, this prediction horizon is usually small and will significantly reduce the computation cost.

Figure 1 shows the flow chart of Lebesgue sampling-based prognosis. Suppose at state  $F_d$ , the diagnosis algorithm de-

tests a fault and prognosis needs to be implemented to calculate the distributions of operating time when the fault size reaches Lebesgue states  $\{F_{d+1}, \dots, F_f\}$ .

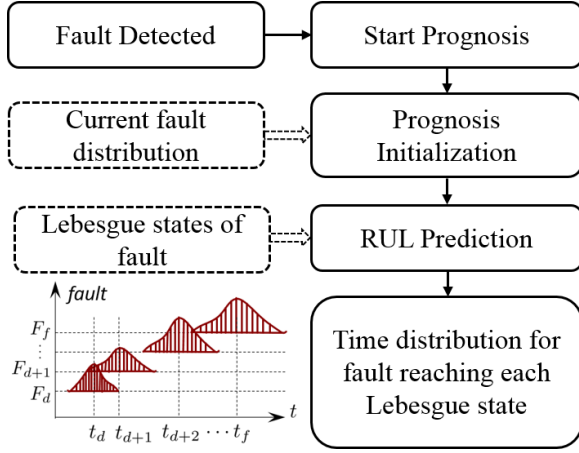


Figure 1. Flow chart of Lebesgue sampling-based prognosis

The initial condition for prognosis is the time distribution to reach the current Lebesgue state  $F_d$ . In a particle filtering-based algorithm, this time distribution can be obtained by conducting prediction based on the fault growth model in Equation (2) for all particles with  $F_d$  being set as the threshold.

#### 2.4. The concept of VLS-FDP

This section will develop the complete VLS-FDP framework with an overview of the proposed solutions to overcome the limitations of the fixed Lebesgue length LS-FDP. The innovation of this method is that Lebesgue length of the diagnosis and prognosis algorithm is no longer fixed. Instead, Lebesgue length is online adjusted adaptively to accommodate the nonlinearity of the fault dynamics. The concept is illustrated in Figure 2, the battery degradation data is from (He, Williard, Osterman, & Pecht, 2011). In LS-FDP method, as shown in Figure 2(a), the Lebesgue states are equally distributed on the state axis, which means the state changes are the same, no matter the fault grows fast or slowly. In VLS-FDP, the Lebesgue state length are changed based on the fault growth speed. The fault growth speed is characterized as the ratio between the Lebesgue state length of two successive Lebesgue states and the operation time between them. It is clear in Figure 2 (b) that the Lebesgue states are nonuniform.

### 3. ALGORITHM DESIGN

In this section, a particle filtering-based algorithm is designed in the framework of varying length Lebesgue sampling.

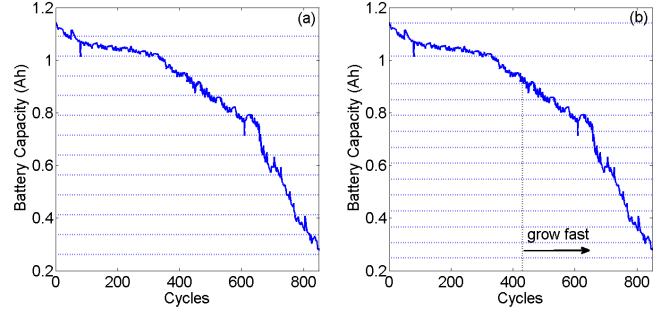


Figure 2. Illustration of VLS-FDP. (a) LS-FDP with fixed Lebesgue state length; (b) VLS-FDP with varying Lebesgue state length

#### 3.1. Particle filter for VLS-based diagnosis

An unobserved fault process  $X$  is assumed to be a Markov process characterized by initial distribution  $p(x_0)$  and the transition probability  $p(x_k | x_{k-1})$  defined by  $x_k = f_k(x_{k-1}, w_k)$  with  $w_k$  being the process noise. The subscript  $k$  represents the  $k$ th events. The observation is assumed to be conditionally independent on  $X$ . The distribution of  $Y_k$  given  $X_k$ ,  $(Y_k | X_k)$ , is defined by  $y_k = h_k(x_k, v_k)$  and  $v_k$  is the measurement noise. Let  $x_{0:k} = \{x_0, \dots, x_k\}$  and  $y_{1:k} = \{y_1, \dots, y_k\}$  denote the states and measurements from beginning to the  $k$ th event. The objective is to estimate the posterior distribution  $p(x_{0:k} | y_{1:k})$ , which in the framework of Bayesian theory is achieved by prediction and filtering.

In nonlinear cases, most of the analytical solutions do not exist. Sequential Monte Carlo (SMC) methods, such as particle filter, are widely used to provide approximate solution. The particle filter approach approximates the posterior distribution at the  $(k-1)$ th event by a set of  $N$  particles  $(w_{k-1}^{(i)}, x_{0:k-1}^{(i)})$ , where superscript  $i$  denotes the  $i$ th particle,  $w_{k-1}^{(i)}$  and  $x_{0:k-1}^{(i)}$  are the weight and location of the particles at the  $(k-1)$ th event, respectively. The purpose is to achieve a new set of  $N$  particles  $(w_k^{(i)}, \bar{x}_{0:k}^{(i)})$  to approximate the distribution  $\pi_k(x_{0:k})$ , where  $\bar{x}_{0:k}^{(i)}$  is the location of new particles. In SMC framework, a Monte Carlo approximation can be obtained as:

$$\pi_k(x_{0:k}) = \sum_{i=1}^N w_k^{(i)} \delta(x_{0:k} - \bar{x}_{0:k}^{(i)}) \quad (4)$$

where  $\delta$  denotes the Dirac-Delta function,  $\sum_{i=1}^N w_k^{(i)} = 1$ . The weight of the particle is updated by a recursive method:

$$w(\bar{x}_{0:k}^{(i)}) = w_{k-1}^{(i)} h_k(y_{1:k} | \bar{x}_{0:k}^{(i)}) \quad (5)$$

$$w_k^{(i)} = \frac{w(\bar{x}_{0:k}^{(i)})}{\sum_{i=1}^N w(\bar{x}_{0:k}^{(i)})}$$

An LS-based diagnostic model is used to implement the par-

particle filtering based fault diagnosis. The model is augmented from (2) and is given as:

$$\begin{cases} x_d(t_k + 1) = f_b(x_d(t_k) + n(t_k)) \\ \hat{a}(t_{k+1}) = \hat{a}(t_k) + f_t(D, \hat{a}(t_k)) \cdot x_d(t_k) + w_a(t_k) \\ y(t_k) = \hat{a}(t_k) + v(t_k) \end{cases} \quad (6)$$

where the nonlinear mapping  $f_b(x)$  is given by:

$$f_b(x) = \begin{cases} 1, & \text{if } \|x - 1\| \leq \|x - 0\| \\ 0, & \text{otherwise} \end{cases}$$

where  $x_d$  is a boolean value that indicates the *normal* (0) or *faulty* (1), respectively,  $\hat{a}$  is the Lebesgue state that indicates the fault size,  $w_a$  and  $v$  are process and measurement noises,  $n$  is an independent and identically distributed uniform white noise, the initial condition is given as  $x_d = 0$ , indicating that there is no fault initially.

In this model,  $t_k$  is the event stamp indicating there is a state transition, the measurement  $y(t_k)$  is directly mapped from the fault size  $\hat{a}(t_k)$ , for battery case, it is the capacity measured by Coulomb counting method.

During the diagnostic process, the algorithm will be executed only when two successive measurements trigger a transition of Lebesgue states.

### 3.2. Particle filter for VLS-based prognosis

In LS-FDP framework, the prediction horizon is reduced to the Lebesgue state number from the current Lebesgue state  $F_d$  to the state that indicates a failure threshold  $F_j$ . Due to the switching of prediction horizon from the time axis to the state axis, the prediction horizon is greatly reduced, which results in reduction of a large amount of computation and the accumulation of uncertainties.

The prognostic model for LS-FDP is given as:

$$t_{k+1} = t_k + g_t(D, \hat{a}(t_k)) + w_t(t_k) \quad (7)$$

where  $D$  is the Lebesgue state length and  $w_t(t_k)$  is the model noise.

With this model, the particle algorithm estimates the time distribution on the time axis. Note that the output of diagnosis is a fault state distribution defined on the state axis, which cannot be used in LS-based prognosis. The reason is that LS-based prognosis needs an initial condition of time distributions on the current Lebesgue state. To address this problem, a new set of  $M$  particles ( $w_L^{(i)}, t_L^{(i)}$ ) is adopted to initialize the prognostic process, where the subscript  $L$  denotes the Lebesgue state,  $w_L^{(i)}$  and  $t_L^{(i)}$  are particle weights and particles on the time axis, respectively. The initial weights of the particles are uniform ( $w_L^{(i)} = 1/M$ ). The RUL pdf is calculated under the condition of Lebesgue state  $L = F_f$  by (7).

The difference between LS-FDP and VLS-FDP is that the Lebesgue state length  $D$  in VLS-FDP is not a constant, but changed based on the diagnostic result. The whole process is illustrated in Figure 3 and is described as follows:

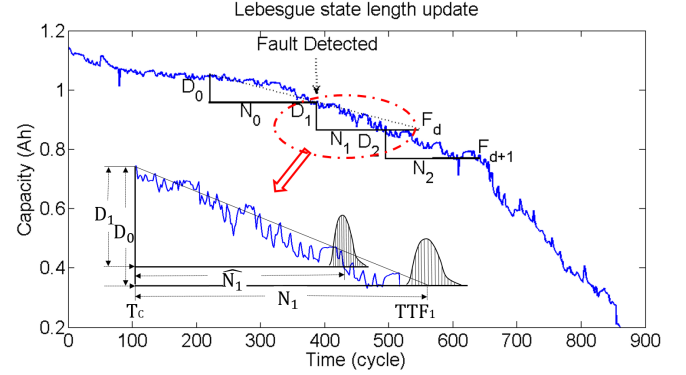


Figure 3. The process of Lebesgue state length update during the prognosis process.

The procedure is described as follows:

1. With a new measurement, check if an event happens. If an event happens, run diagnosis algorithm to detect the fault with the initial set of Lebesgue states and Lebesgue state length;
2. When the fault is detected at the Lebesgue state  $F_d$ , the prognostic algorithm is executed with an initial Lebesgue state length  $D_0$  and the Lebesgue states are ( $F_1, \dots, F_{d-1}$ ). The time interval between Lebesgue state  $F_{d-1}$  and  $F_d$  is  $N_0$  (in the diagnostic process), the slope is calculated  $S_0 = D_0/N_0$ ;
3. Lebesgue state length for prognosis is  $D_0$  when the prognosis is initialized on Lebesgue state  $F_d$ . The time interval to reach  $F_{d+1}$  calculated by (7) is  $N_1 = TTF_1 - T_c$  for the prognosis on Lebesgue state  $F_d$ , where  $TTF_1$  is the mean of the predicted time to failure (TTF),  $T_c$  is the current time instant. The Lebesgue state length between Lebesgue state  $F_d$  and  $F_{d+1}$  is updated by  $D_1 = S_0 \times N_1$ , and the TTF and time interval are also updated to  $\hat{TTF}_1$  and  $\hat{N}_1$ , and the new slope is  $S_1 = D_1/\hat{N}_1$ ;
4. The prognosis at Lebesgue state  $F_{d+1}$  starts with Lebesgue state length  $D_1$  and slope  $S_1$ , time interval  $N_2$  is also calculated by (7). Lebesgue state length  $D_2$  is given as  $D_2 = S_1 \times N_2$ , then update the time interval  $\hat{N}_2$  to calculate the slope  $S_2 = D_2/\hat{N}_2$ ;
5. Repeat the step 3) and 4) until the Lebesgue state reaches the prognosis threshold  $F_f$ . This step yields the TTF and RUL distribution;
6. At the next time instant, when a new measurement becomes available, repeat step 1) to 5). Note that the Lebesgue state lengths for diagnosis are also updated based on the calculation result in the previous prognostic

process. When a new event happens, the time interval is updated by the ground truth of cycle life to  $\bar{N}_1$ , the slope  $\bar{S}_1 = D_1/\bar{N}_1$  and  $D_2$  are used as the initial condition for the new recursive prognostic loop.

Note that, after the prognostic process, the length of each Lebesgue state is changed. The new Lebesgue states are used for the following prognostic process as the initial set of Lebesgue states. So the whole FDP process will be executed with a different frequency, which is determined by the fault growth speed.

#### 4. EXPERIMENTAL RESULTS

In this section, the proposed VLS-FDP scheme will be demonstrated with a particle filtering algorithm with an application to the prediction of the capacity degradation of a Li-ion battery. Battery is a safety critical component that provides power for most autonomous systems, such as computers, robots, electrical vehicles, and unmanned aircraft (Saha, Goebel, Poll, & Christophersen, 2009; Zhang, Tang, DeCastro, Roemer, & Goebel, 2014). Since the life and state of the batteries are not directly observable, diagnosis and prognosis are critical for estimating the battery state (K. Goebel, Saha, & Saxena, 2008; K. F. Goebel et al., 2006; Sidhu, Izadian, & Anwar, 2015), such as state-of-health (SOH), state-of-charge (SOC), and remaining useful life (RUL).

In this experiment, the SOH of a Li-ion battery with 1.1 Ah rated capacity is used to verify the proposed VLS-FDP algorithm, which is compared with RS-FDP and LS-FDP algorithms. The degradation of the capacity is obtained from charge-discharge cycle tests carried by an Arbin BT2000 battery test system under room temperature at a discharge current of 1.1 A. The charge-discharge cycle is cut off at pre-determined cut-off voltages. The failure threshold for the SOH is set to be 0.25Ah and the battery capacity reaches this threshold at 854th cycle.

##### 4.1. RS-FDP

To implement diagnosis and prognosis, a fault growth model for RS-FDP is developed by model fitting:

$$C(t+1) = C(t) - \alpha \cdot p_1 \cdot |p_2 + p_3 \cdot t + p_4 \cdot t^2|^{p_5} + w(t) \quad (8)$$

where  $C$  is battery capacity,  $t$  is the time index which is cycle number in this experiment,  $p = [5e^{-5}, -225, 5.6, -0.0135, 0.5]$  are parameters,  $\alpha$  is a hyper model parameter with mean of  $3.8e^{-3}$  and variance of  $5e^{-5}$ , and  $w$  is a model noise.

For RS-based diagnosis, a set of 500 particles are used in the algorithm based on our computational resource. Figure 4 shows the diagnostic results at the 472nd cycle. The mean of capacity estimation is 0.87414 and the 95% confidence interval is [0.8497, 0.8945]. The upper sub-figure of this fig-

ure is the measurement compared with the filtered estimation. The bottom sub-figure shows the comparison of initial baseline pdf compared with the real-time estimated pdf at the 472nd cycle. Note that the diagnostic algorithm is executed 472 times, i.e., every time when a new measurement becomes available.

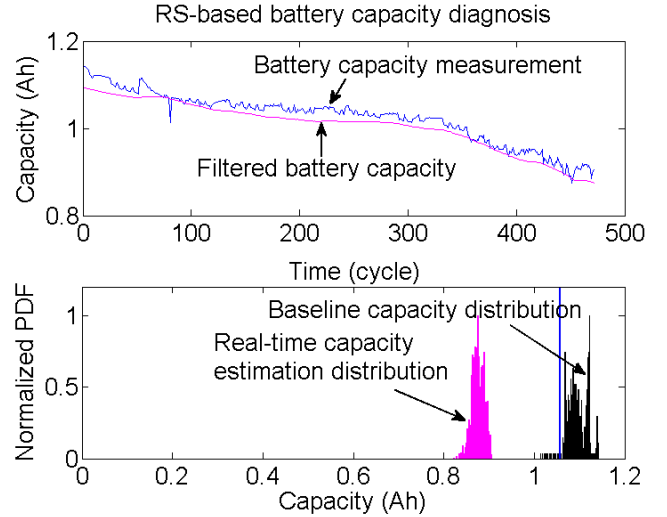


Figure 4. Experimental result of RS-based diagnosis.

With an estimation of the current battery capacity as the initial condition, the prognosis is executed to conduct the long-term prediction and estimation of RUL. Figure 5 shows the expected value, upper and lower bounds of 95% confidence interval of the battery capacity pdf at each future cycle. Note that only 20 particles are used for prognosis because of a large prediction horizon of 525 cycles. The battery capacity pdf at each cycle is compared against the failure threshold to obtain the RUL pdf, as shown in the histogram on the horizontal axis. The law of total probabilities is used in this process.

In this figure, the predicted result of the mean of the failure time is at the 748th cycle and the RUL life is 276 cycles. The distance between the prediction and ground truth is 106 cycles. The 95% confidence interval of the RUL pdf is [628, 987], which means that the uncertainty accumulated along the prediction horizon is very large.

##### 4.2. LS-FDP

In LS-FDP, the input of the algorithm is the feature, which is divided into a series of Lebesgue states. If the new measurement causes a transition of Lebesgue state, i.e., an event happens, the diagnostic algorithm is executed. The time instant when an event happens is called the “event stamp”. The sequence of the event stamps formulates a time series that is used as the input of real time diagnostic algorithms. The output of fault diagnosis is the fault state distribution, which is used to calculate the probability to declare a fault. The test is

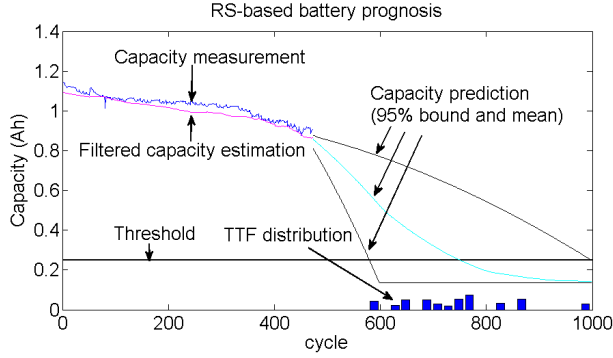


Figure 5. Experimental result of RS-based prognosis.

performed between the estimate pdf of fault and the baseline pdf from healthy condition.

To implement LS-FDP for the battery capacity degradation, 40 Lebesgue states are defined based on the battery's full capacity of 1.1Ah and the computation resource. With this setting, the diagnostic algorithm is executed only when the capacity degrades from one Lebesgue state to another, i.e., an event happens. The model for diagnosis is given by:

$$C(t_{k+1}) = C(t_k) - p_d \cdot D \cdot \text{sgn}(C(t_k) - C(t_{k-1})) + w_C(t_k) \quad (9)$$

where  $p_d$  is the model parameter,  $t_k$  is the event stamp indexes,  $\text{sgn}(\cdot)$  gives the sign, and  $w_C$  is the model noise.

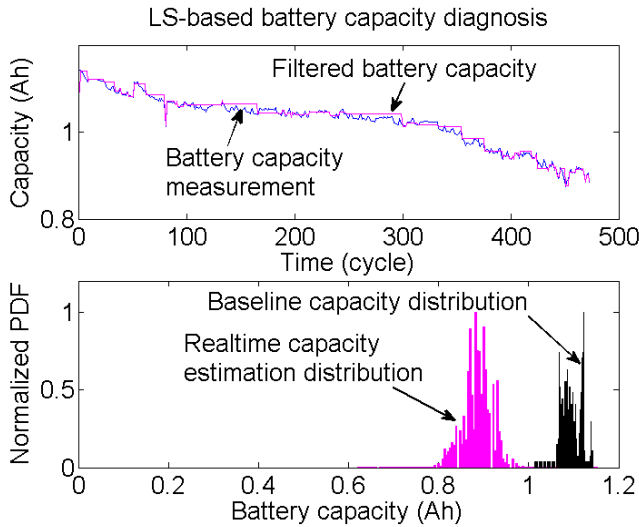


Figure 6. LS-based diagnosis for battery at the 472nd cycle.

Figure 6 shows the diagnostic results with a particle filtering algorithm at the 472nd cycle. In the upper sub-figure, the blue curve is the trajectory of capacity from Coulomb counting and the magenta curve is from the particle filtering. Note that the flat magenta segments mean no event and diagnosis is not executed. The lower sub-figure shows the capacity distri-

bution at the current cycle, where the black distribution is the baseline probability density function (pdf) while the magenta histogram is the real-time capacity pdf from diagnosis. In the past 472 cycles, although 472 measurements are received, there are only 76 events. Therefore, the LS-based diagnosis only runs 76 times. Compared with traditional RS-based diagnosis that needs to run 472 times, the reduction of computation is  $(472-76)/472=83.9\%$  and computation is 6.21 times faster.

Different from the diagnosis that yields fault state distribution at each time instant when an event occurs, prognosis estimate the time distribution on fault state reaching each Lebesgue state. The output of diagnosis is a capacity distribution at current time instant. It cannot be used for prognosis directly and has to be transformed into the operation time distribution. To implement prognosis in LS-FDP framework, the operation time distribution is achieved by predicting all the particles to the current Lebesgue state.

LS-based prognosis is conducted on fault dimension axis to predict the time-to-Lebesgue-state directly. The diagnostic model (9) cannot be used in prognosis. A new model for prognosis is proposed as:

$$t_{k+1} = t_k + p_p \cdot D \cdot \exp(-\dot{C}(t_k)) + w_k(t_k) \quad (10)$$

where  $p_p$  is the model parameter and  $w_k$  is the model noise.

Figure 7 shows the prognostic results with 500 particles at the 472nd cycle. Compared to RS-based prognosis with large horizon (525 cycles) and small number of particles (20), the LS-based prognosis only has a prognathic horizon of 24 Lebesgue states and can afford 500 particles. The reduction of computation time is  $(2.822281-0.011541)/2.822281=99.59\%$  and the computation is about 244 times faster. Note that in RS-based prognosis, only 20 particles (at the cost of performance) are used to make real-time implementation possible. Note also that LS-FDP offers better performance than RS-FDP in terms of TTF prediction due to short prognostic horizon.

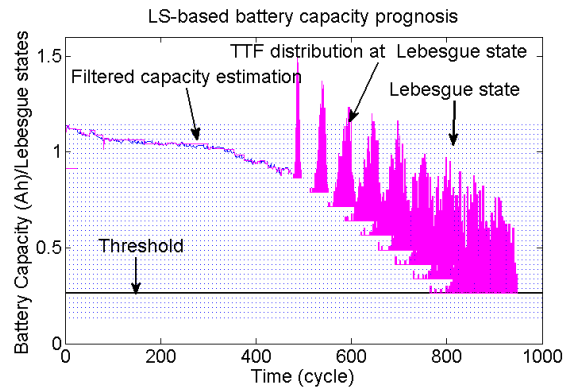


Figure 7. LS-based prognosis at the 472nd cycle.



### 4.3. VLS-FDP

VLS-FDP is developed based on LS-FDP, the battery's full capacity range is divided into 40 Lebesgue state as in LS-FDP. When fault is firstly detected, this Lebesgue state length is the initial Lebesgue state length in VLS-FDP. The model for diagnosis is the same as LS-FDP in model (9).

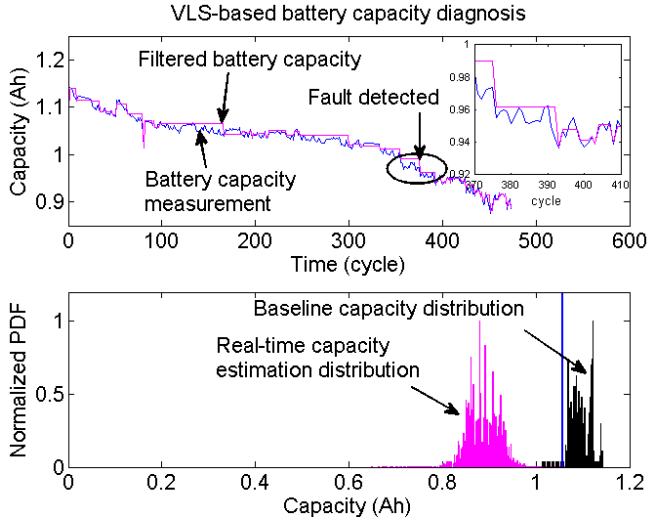


Figure 8. VLS-based diagnosis for battery at the 472nd cycle.

Figure 8 shows the diagnostic results at the 472nd cycle. The upper sub-figure is the battery capacity measurement by Coulomb counting method (blue curve) compared with the filtered capacity estimation (magenta). Note that the length for each Lebesgue state has been changes after the fault being detected compared with the case before fault being detected, which is an indicator of a closer monitoring on fault growth.

The lower sub-figure shows the fault distribution at the 472nd cycle, in which the black histogram is the baseline distribution, and the magenta one is the real-time battery capacity distribution from diagnosis.

The VLS-FDP prognosis is conducted on the state axis to predict the time-to-Lebesgue-state, the diagnostic model is the same as (9).

Figure 9 shows the prognostic result at the 472nd cycle. In this figure, the prediction horizon is 50, which is a little bigger than the LS-FDP with uniform Lebesgue length. This is the trade-off for closer monitoring of SOH when the capacity degradation becomes faster.

To make the figure clear, only the time distribution at selected Lebesgue states are plotted. As shown in Figure 9, the predicted TTF for this battery is 846.9 and the RUL is 374.9 cycles. The 95% confidence interval of the TTF is [819.6 861.9]. Compared with the ground truth TTF of 854, the difference is 7.1 cycles. Note that Lebesgue states for VLS-based prognosis are distributed unequally along the state axis,

which is different from the case of LS-based prognosis. When the growth speed is fast, the prognosis is executed with a higher frequency and the length of the neighboring execution is reduced.

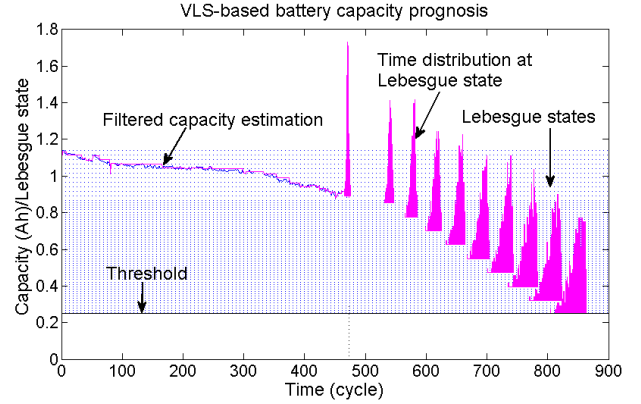


Figure 9. VLS-based prognosis at the 472nd cycle.

### 4.4. Comparison of RS-FDP, LS-FDP, and VLS-FDP

Diagnostic and prognostic results of RS-based, LS-based, and VLS-based algorithms are compared in Table 1 with the same benchmark. Compared with RS-based prognosis with a horizon of 525 cycles and small number (20) of particles at the 472th cycle, the LS-based and VLS-based prognosis have a horizon of 24 and 50 Lebesgue states, and can afford 500 particles. The computation time for LS-based and VLS-based prognosis are only 0.41% and 1.36% of the RS-based prognosis, respectively. Compared to LS-based prognosis, VLS-based prognosis is a little computational expensive. The reason is that VLS-based prognosis keeps a closer monitoring on the SOH after the 472nd cycles by shrinking the Lebesgue state length dynamically to accommodate an accelerated degradation speed.

Table 1. Comparison of Traditional RS-FDP, LS-FDP and VLS-FDP for Battery

	RS-FDP	LS-FDP	VLS-FDP
<b>Diagnosis particles</b>	500	500	500
Capacity expectation	0.8741	0.8854	0.8854
Capacity 95% CI	[0.8469 0.8986]	[0.8285 0.9422]	[0.8290 0.9432]
Execution numbers	472 (100%)	76 (16.1%)	78 (16.5%)
<b>Prognosis particles</b>	20	500	500
True TTF	854	854	854
Estimate TTF	748	831.8	846.9
95% CI of TTF	[628 987]	[795.3 850.3]	[819.6 861.9]
Prognostic horizon	525	24	50
Computation time	2.822281 (100%)	0.011541 (0.41%)	0.038488 (1.36%)

Accuracy is one of the most important properties in FDP. In order to compare the accuracy of three FDP methods,  $\alpha - \lambda$  matrix is introduced in (Saxena, Celaya, Saha, Saha, & Goebel, 2010), as shown in Figure 10 with  $\alpha = 0.3$ . The

matrix is defined as:

$$[1 - \alpha] \cdot r_t(t_k) \leq r^l(t_k) \leq [1 + \alpha] \cdot r_t(t_k) \quad (11)$$

where  $r^l$  is the predicted RUL at the  $l$ th time instant,  $r_t$  is the ground truth TTF,  $\alpha$  is the accuracy modifier (Saxena et al., 2010).

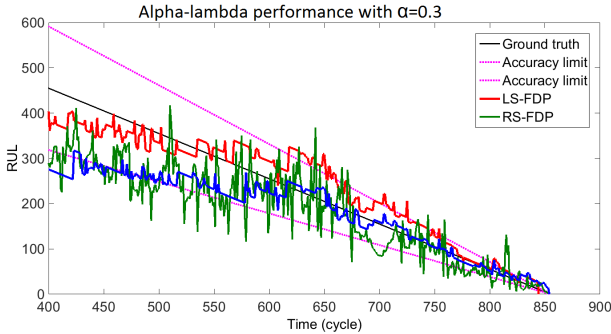


Figure 10. Prediction accuracy comparison among RS-FDP, LS-FDP, and VLS-FDP.

Because of the short prediction horizon and small uncertainty accumulation, the prediction accuracy for LS-FDP and VLS-FDP are higher than that of RS-FDP, as shown in Figure 10. The TTF of LS-FDP and VLS-FDP reach the accuracy zone quickly, and the prediction results are stable. The result of RS-FDP exceeds the accuracy limits sometimes, which means the estimation of TTF is not uniformly accurate. The high prediction accuracy of LS-FDP and VLS-FDP is achieved with much lower computation cost compared with RS-FDP.

The advantages of varying Lebesgue length in diagnosis and prognosis are obvious by the comparison in Table 1. For diagnosis, these three methods have comparable performances. In terms of prognosis, the VLS-FDP shows better performances in different aspects. First, VLS-FDP has better accuracy and precision than RS-FDP by comparing the confidence interval (CI) of the TTF. It reduces the computation resource greatly with similar diagnostic performance. RS-FDP has a large prediction horizon, which requires more computation time and resources. Second, VLS-FDP also avoids the large amount of uncertainty accumulation during the long-horizon prediction. Third, VLS-FDP dynamically distributes the limited computation resources between different fault growth stages, which is an improvement of LS-FDP.

The application of Lebesgue sampling method in FDP provides a natural solution for real-time FDP implementation, especially for those systems with limited computation resources. The improved Lebesgue sampling varying length method distributes the computation resources dynamically within a system. Similar to LS-FDP, the prediction horizon of VLS-FDP is very small compared with that of RS-FDP,

which is beneficial for managing the uncertainty in prognosis.

## 5. CONCLUSIONS AND FUTURE WORKS

Many diagnostic and prognostic methods were developed based on traditional Riemann sampling framework with great success in the past decades. The application of RS-FDP on distributed FDP is limited because of its high computation cost. A new FDP methodology is introduced with a philosophy of "execution when needed" to reduce the computation cost, which makes the long-term online prognosis possible to be computed on an embedded system, such as the micro-controllers in a car and a mobile phone. However, the close monitoring on the fault size is sacrificed to some extent, especially when the fault growth is accelerated. This paper proposes varying Lebesgue state length in the LS-FDP framework. A particle filtering-based algorithm is developed with an application to the diagnosis and prognosis of Li-ion battery SOH.

In the VLS-FDP, models for diagnosis and prognosis are designed separately because diagnosis and prognosis are carried on state and time axis, respectively. The Lebesgue length for the diagnostic and prognostic models changes according to the past fault growth speed. Experimental results for RS-FDP, LS-FDP, and VLS-FDP on a Li-ion battery SOH are presented to demonstrate the effectiveness of the proposed algorithms.

In the future work, data collected from partial charge/discharge cycle and non-constant current charge/discharge with different current will be used to test our methods, new parameter adaption methods will be adopted to accommodate the real vehicle scenarios.

## ACKNOWLEDGMENT

The project is sponsored by the ASPIRE grant program at the University of South Carolina and the Woodrow W. Everett, Jr. SCEE Development Fund in cooperation with the Southeastern Association of Electrical Engineering Department Heads.

## REFERENCES

- Astrom, K., & Bernhardsson, B. (2002, Dec). Comparison of riemann and lebesgue sampling for first order stochastic systems. In *Decision and control, 2002, proceedings of the 41st ieee conference on* (Vol. 2, p. 2011-2016 vol.2). doi: 10.1109/CDC.2002.1184824
- Goebel, K., Saha, B., & Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. In *62nd meeting of the society for machinery failure prevention technology (mfpt)* (pp. 119-131).
- Goebel, K. F., Yan, W., Eklund, N. H. W., Hu, X.,



- Avasarala, V., & Celaya, J. (2006). *Defect classification of highly noisy nde data using classifier ensembles* (Vol. 6167). Retrieved from <http://dx.doi.org/10.1117/12.659704>  
doi: 10.1117/12.659704
- He, W., Williard, N., Osterman, M., & Pecht, M. (2011). Prognostics of lithium-ion batteries based on dempster-shafer theory and the bayesian monte carlo method. *Journal of Power Sources*, 196(23), 10314–10321.
- McCann, R., & Le, A. (2008, Oct). Lebesgue sampling with a kalman filter in wireless sensors for smart appliance networks. In *Industry applications society annual meeting, 2008. ias '08. ieee* (p. 1-5). doi: 10.1109/08IAS.2008.9
- Pattipati, K., Wang, B., Zhang, Y., Howell, M., & Salman, M. (2011). Fault diagnosis and prognosis in a network of embedded systems in automotive vehicles. In *Nsf-nist-uscar workshop on cyber-physical systems*.
- Saha, B., Goebel, K., Poll, S., & Christophersen, J. (2009, Feb). Prognostics methods for battery health monitoring using a bayesian framework. *Instrumentation and Measurement, IEEE Transactions on*, 58(2), 291-296. doi: 10.1109/TIM.2008.2005965
- 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. *Industrial Electronics, IEEE Transactions on*, 62(2), 1002-1011. doi: 10.1109/TIE.2014.2336599
- Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley and Sons.
- 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., Tang, L., DeCastro, J., Roemer, M., & Goebel, K. (2014). Autonomous vehicle battery state-of-charge prognostics enhanced mission planning. *Int. J. Prognost. Health Manage*, 5(2), 1–12.
- 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.

**Wanchun Dou** received the PhD degree in mechanical and electronic engineering from Nanjing University of Science and Technology, China, in 2001. From April 2001 to December 2002, he did his postdoctoral research in the Department of Computer Science and Technology, Nanjing University, China. He is now a Full Professor at the State Key Laboratory for Novel Software Technology, Nanjing University, China. His research interests include workflow, cloud computing and service computing.

**Datong Liu** received the B.Sc. and M.Sc. degrees in Department of Automatic Test and Control from Harbin Institute of Technology (HIT), Harbin, China in 2003 and 2005, respectively. During 2001 to 2003, he also minored in the Computer Science and Technology in HIT. He received the Ph.D. degree in major of measurement and instrumentation from HIT in 2010. He is now an associate professor in the Department of Automatic Test and Control, HIT. His research interests include automatic test, machine learning and data mining for test data processing of complex system, data-driven prognostics and health management, lithium-ion battery prognostics, system health management for aerospace, etc.

**Yu Peng** received the B.S. degree in Measurement Technology and Instrumentation, Harbin Institute of Technology (HIT), Harbin, China, in 1996, and M.Sc. and Ph.D. degrees in Instrumentation Science and Technology, HIT, in 1998 and 2004, respectively. He is a full Professor with the Department of Automatic Test and Control, School of Electrical Engineering and Automation, HIT. His research interests include automatic test technologies, virtual instruments, system health management, reconfigurable computing, etc.

**Bin Zhang** received the B.E. and M.E. degrees from the Nanjing 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. Before that, he was with R&D, General Motors, Detroit, MI, USA, with Impact Technologies, Rochester, NY, USA, and with the Georgia Institute of Technology, Atlanta, GA, USA. His research interests are prognostics and health management and intelligent systems.