An intelligent online prognostic approach is proposed for predicting the remaining useful life (RUL) of lithium-ion (Li-ion) batteries based on artificial fish swarm algorithm (AFSA) and particle filter (PF), which is an integrated approach combining model-based method with data-driven method. The parameters, used in the empirical model which is based on the capacity fade trends of Li-ion batteries, are identified dependent on the tracking ability of PF. AFSA-PF aims to improve the performance of the basic PF. By driving the prior particles to the domain with high likelihood, AFSA-PF allows global optimization, prevents particle degeneracy, thereby improving particle distribution and increasing prediction accuracy and algorithm convergence. Data provided by NASA are used to verify this approach and compare it with basic PF and regularized PF. AFSA-PF is shown to be more accurate and precise.

1. Introduction

In recent years, green energy has gained increasing support among the people and electric energy has become much more popular. Lithium-ion (Li-ion) batteries offer advantages such as a high energy ratio, high voltage, desirable cryogenic properties, low self-discharge ratio, and lack of the memory effect [1]. Li-ion batteries are therefore widely used in electric vehicles, satellites, aerospace craft, and other important domains. Being the core cell of several types of electronic equipment and complex systems [2], Li-ion batteries are a critical component of the electronic system as a whole, and the failure of Li-ion batteries can lead to system malfunction, downed systems, and even human casualties and financial losses [3].

The capacity of Li-ion batteries inevitably degrades over time. Hence, monitoring the state of health of Li-ion batteries in operation is of great necessity. The monitored data can be used to predict the remaining useful life (RUL) of Li-ion batteries, which can provide the basis for maintenance decisions and make the use of Li-ion batteries much more convenient and safer. Currently, the optimal prediction of the RUL of Li-ion batteries has aroused intense discussion in the Prognostic and Health Management (PHM) domain locally and abroad [1, 4].

Several studies have focused on predicting the RUL of Li-ion batteries [5–7]. Zhang and Lee reviewed various aspects of recent research and developments in Li-ion battery PHM and summarized the techniques, algorithms, and models used in RUL prediction [8]. Regarding research on methods of forecasting the RUL of Li-ion batteries, Saha et al. from the Prognostics Center of Excellence (PCoE) of NASA were first to develop a number of performance degradation tests for Li-ion batteries under various test conditions [9]. The parameters measured were comprehensive. The researchers obtained a large amount of test data and took the lead in using the Bayesian estimation method to forecast the RUL of Li-ion batteries. The core idea of this method is to build a probability density distribution (PDF) of states based on all available information [10] to establish a way to express and manage the uncertainty of prediction. The researchers forecast the RUL of Li-ion batteries by establishing a particle filter- (PF-) based method. Later, Saha et al. proposed a forecasting method combining support vector machine with
PF on the basis of [10]. This method was able to calculate the PDF of the prediction value, rather than mean time to failure, accurately and precisely.

Given the complexity of the system organization of Li-ion batteries, the appropriate operating conditions are changeable and the system often suffers from noise jamming, test data, model error, load variation, operating conditions, and other uncertain factors need to be taken into consideration [10]. PF is an effective solution for these problems in PHM for nonlinear non-Gaussian systems. PF is able to dynamically adjust model parameters of nonstationary conditions and predict the unknown parameters of the battery empirical model by tracking historical data. Accurate prediction as well as a confidence evaluation of the prediction value can be obtained by means of PF. The estimation of confidence interval is often characterized by PDF. PF describes the posteriori estimate of a state using a set of particles with weight; this method results in a probability expression for the prediction value. This description, which is based on Monte Carlo, is equivalent to a real posterior probability density function, and effectively expresses the uncertainty of the prediction outcome, thereby making PF an ideal method for state tracking and prediction [11].

However, basic PF has a serious problem of particle degeneracy [12]. Addressing this problem improperly results in "particle collapse," a severe case of sample impoverishment [13] where the weights of particles accumulate on a small interval is often characterized by PDF. PF describes the posterior density of the prediction value. This description, which is based on Monte Carlo, is equivalent to a real posterior probability density function, and effectively expresses the uncertainty of the prediction outcome, thereby making PF an ideal method for state tracking and prediction [11].

2 Mathematical Problems in Engineering

2. Methodologies

2.1. Artificial Fish Swarm Algorithm. Artificial fish swarm algorithm (AFSA) was proposed by Li et al. [20] in 2002 as a new bionic random searching optimization algorithm based on simulating the ecological behaviors of fish swarm in the wild. This algorithm mainly imitates four behaviors of fish swarm, that is, preying behavior to find the optimal solution, swarming behavior to gather the fish swarm in the optimal region, following behavior to free the fish swarm from local optimal solutions, and random behavior to find the optimal solution in a larger scope. With the merits of rapid and global optimization, insensitivity to initial values and selections of parameters, robustness, easy operation, and so on, AFSA has been applied widely in pattern recognition, neural networks, and parameter estimation [21–23]. To illustrate the algorithm clearly and expeditiously, we create the following rules: the individual state of the artificial fish (AF) is denoted by a vector \( X = (x_1, x_2, \ldots, x_n) \), where \( x_i (i = 1, 2, \ldots, n) \) is the variable whose optimal value is to be searched for; the food consistency of the position where the AF is currently located is denoted by \( Y = f(X) \), where \( Y \) denotes the value of the objective function. The distance between two AFSA individuals is expressed as \( d_{ij} = \|X_i - X_j\| \), \( v \) represents the vision distance, \( s \) represents the largest step-length of the AF, and \( r \) represents a random number ranging from zero to one.

The behaviors are described as follows.

**Preying Behavior.** Let \( X_i \) denote the current state of the AF and randomly select a state \( X_j \) (1) within the scope of perception \( (d_{ij} < v) \). In this study, we only discuss the maximum problem. If \( Y_i < Y_j \), \( X_i \) moves a step toward \( X_j \); otherwise, we select a state \( X_j \) randomly again and justify whether it satisfies the above requirement or not. After several attempts, if the requirements still cannot be satisfied, \( X_i \) moves a step randomly. The following briefly shows the process of the preying behavior:

\[
X_{\text{next}} = X_i + r \cdot s \cdot \frac{X_j - X_i}{\| X_j - X_i \|} \tag{1}
\]

**Swarming Behavior.** One stipulation for AF is set in the AFSA; that is, AF always tries to swim toward the center of its adjacent fellow swarm-members and avoids being overcrowded, which directs the swarming behavior. Let \( X_i \) denote the current state of the AF, \( n_j \) denote the number of the fellow swarm-members within the scope of perception \( (d_{ij} < v) \), \( X_c \) denote the center of the fellows around \( X_i \), and \( Y_c \) denote the corresponding food consistency of \( X_c \). The AF swims naturally as it searches around \( n_j \) and \( X_c \), if \( Y_c / n_j > \delta \cdot Y_i \), which means that the center of the swarm-members has a considerable amount of food and is not crowded. Thus, the AF moves toward the center; otherwise, it executes the preying behavior. The following briefly shows the process of the swarming behavior:

\[
X_{\text{next}} = X_i + r \cdot s \cdot \frac{X_c - X_i}{\| X_c - X_i \|} \tag{2}
\]

**Following Behavior.** Let \( X_i \) denote the current state of the AF, and find \( X_j \) of which \( Y_j \) is the largest value within the scope...
of perception ($d_{i,j} < v$). If $Y_j/n_j > \delta \cdot Y_i$, this means that the fellow $X_j$ has high food consistency and the fish surrounding $X_j$ is not very crowded. Thus, $X_j$ moves a step toward $X_j$; otherwise, it executes the preying behavior.

**Random Behavior.** In nature, random behavior is the default activity of the preying behavior; that is, the AF moves a step randomly in its field of vision. The behavior seems to be a random behavior; however, in this process, the AF searches for food or fellow swarm-members in a larger scope. A few iterations later, the AF would gather around in some local extrema; then a large number of AF would gather around the area of the extremum which is the comparatively optimal area. In this way, we could easily obtain the global extremum to smoothly achieve the optimal value [19, 20].

### 2.2. Particle Filter

Particle filter (PF) is a statistical filtering technique for implementing the recursive Bayesian filtering using the Monte Carlo simulations [24]. In the PF approach, the state probability density function (PDF) is approximated by a set of particles (points) representing sampled values from an unknown state space and a set of associated weights denoting discrete probability masses. The particles are generated and recursively updated from a nonlinear process model that describes the evolution of the system under analysis in time with a measurement model, a set of available measurements, and an a priori estimate of the state PDF.

PF methods assume that the state equations can be modeled as a first order Markov process with the outputs being conditionally independent. This model can be written as

$$x_k = f(x_{k-1}) + \omega_k,$$

$$y_k = h(x_k) + \nu_k,$$  

where $x$ denotes the state, $y$ is the output or measurements, and $\omega_k$ and $\nu_k$ are samples from a noise distribution, respectively. Sequential importance resampling (SIR) is a widely used particle filtering algorithm, which approximates the filtering distribution denoted as $p(x_k | y_0, \ldots, y_k)$ by a set of $P$ weighted particles $\{x^i_k, w^i_k : i = 1, \ldots, P\}$. The importance weights $w^i_k$ are approximations to the relative posterior probabilities of the particles such that

$$\int f(x_k) \cdot p(x_k | y_0, \ldots, y_k) \, dx_k \approx \sum_{i=1}^{P} w^i_k f(x^i_k),$$

$$\sum_{i=1}^{P} w^i_k = 1.$$  

The weight update is given by

$$w^i_k = w^i_{k-1} \frac{p(y_k | x_k) \cdot p(x_k | x_{k-1})}{\pi(x_k | x_{0:k-1}, y_{1:k})},$$

where the importance distribution $\pi(x_k | x_{0:k-1}, y_{1:k})$ is approximated as $p(x_k | x_{k-1})$.

The procedure of the basic particle filter is as follows.

1. **Initialization.** $k = 0$ is set, $N$ particles are sampled and represented as $\{x^0_i, i = 1, 2, \ldots, N\}$ from prior distribution $p(x_0)$, and each weight of the sample is set to $1/N$ as the initial weight.

2. **Importance Sampling.** $k = 1$ is set and the state of particles is updated using the importance distribution which can be represented as $x^i_k \sim p(x_k | x^i_{0:k-1}, y_{0:k})$, $i = 1, 2, \ldots, N$.

   The importance weights are then calculated according to (6) and the weights are normalized as $w^i_k = w^i_k / \sum_{i=1}^{N} w^i_k$.

3. **Resampling.** Ahead of time, a threshold number of effective particles is set as $N_{\text{threshold}}$. The number of effective particles are calculated and then denoted as $N_{\text{eff}} = 1/\sum_{i=1}^{N} (w^i_k)^2$. If $N_{\text{eff}} < N_{\text{threshold}}$, the particles are resampled; that is, the weighted particles $\{x^i_{0:k}, w^i_{1:k}\}$ are mapped to become equal to the weighted particles $\{x^i_{0:k}, N^{-1}w^i_{1:k}\}$.

4. **Prediction.** $x^i_{k+1}$ is obtained using the state equations (4). We set $k = k + 1$ and then proceed to Step (2) [5, 19].

   The regularized particle filter is an improved version of the basic PF. By means of resampling from a continuous approximation instead of resampling from a discrete approximation of the posterior density $p(x_k | z_k)$; RPF exactly avoids the problem of basic PF. In this study, we do not discuss RPF further; detailed information about RPF could be obtained in [13, 15].

### 2.3. AFSA-Based Particle Filter

Introducing the concept of AFSA into basic PF aims at avoiding the phenomenon of particle degeneracy. The combination is applied in the selection of the objective function and the adjustment of the importance weights. AFSA is introduced to improve the process of importance sampling in the basic particle filter by driving the particles move to the high likelihood areas. In this study, we consider the objective function as the posterior PDF. We refer to the new method as AFSA-PF. In AFSA-PF, the number of fish is the same as the number of particles.

Likelihood function is defined as the objective function $Y$, which is also the posterior PDF:

$$p(y_k | x^j_k) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left[ -\frac{1}{2\sigma^2} (y_k - y^j_{k(k-1)})^2 \right],$$

where $y_k$ is the latest observed value and $y^j_{k(k-1)}$ is the prediction of the observed value of the last moment.

The main process of AFSA-PF is based on the process of PF as described in Section 2.2. The difference between AFSA-PF and the basic PF lies in importance sampling, as shown in Figure 1. The detailed process of AFSA-PF is described as follows.

1. **Parameter Initialization.** The initial states of particles and some parameters (number of particles, process noise, measurement noise, etc.) need to be determined, similar to Step (1) in Section 2.2.
(2) **Importance Resampling.** The latest observed value is input to the AFSA, together with the corresponding particle state $X_i$ that acts as the initial position of the AF. The fitness value $p(y_k^i \mid x_k^i)$, calculated using (7), acts as the objective function $Y$ in (2).

The position $X_i$ of each AF is continuously updated based on the preying and swarming behaviors, as described in (2) and (3). The iteration process terminates once a predetermined iteration number is attained or the variation of the AF is less than a specified threshold.

(3) **State Prediction.** After the update process for all the AFs, the optimal position $X_i$ can be obtained, which is input to (4) as $x_{k-1}$ to acquire the next particle state $x_k$. Then, we set $k = k + 1$ and repeat Steps (2–3) [19].

In AFSA-PF, PF is responsible for providing new observed values to the state estimate, and the AFSA is in charge of driving the particles toward the high-likelihood areas, which solves the problem that only a small fraction of the particles are in the high-likelihood areas and thus maintaining the diversity of the particles.

In order to explain the process of AFSA-PF more intuitively, Figure 1 is shown below.

### 3. RUL Prediction for Li-Ion Batteries

To avoid the phenomenon of particle degeneracy and obtain a predicted result with better accuracy and higher precision, we adopt AFSA-PF. As mentioned before in the Introduction, RPF is an effective method to avoid the degeneracy; thus, in
In this study, we compare AFSA-PF and RPF based on accuracy and precision. Based on the principle of PF, a state equation for Li-ion batteries is required in both AFSA-PF and RPF. However, a number of unknown parameters always exist in the state equation to be estimated. Although PF can update the parameters when predicting the RUL, this momentary updating of parameters is extremely time-consuming, which impedes the popularization of AFSA-PF and RPF in engineering applications. Moreover, the objects we predict are almost degenerated over time; thus, a considerable amount of history data could be accumulated under normal conditions. Given this condition, by applying a number of history data, we can determine the unknown parameters before predicting RUL using AFSA-PF or RPF to obtain time-saving predictions without updating parameters momentarily. The procedures of the experiments for predicting the RUL of Li-ion batteries are shown in Figure 2.

3.1. Li-Ion Battery Data Set. To verify the performance of the AFSA-PF algorithm in predicting the RUL of Li-ion batteries, we adopt the Li-ion batteries data set from NASA AMES Center [5, 25]. The battery data set is from the NASA PCoE. The Li-ion batteries are tested under certain conditions (with the temperature $+23\,^\circ C$) and the test ending life is set as 70% of the rated capacity to measure the capacity degradation. Detailed information about the data set can be found in [26]. In this study, the capacity is considered as the health indicator of the degradation of the Li-ion battery. We set 70% of the rated capacity as the failure threshold [27, 28]. For the selected data set, the rated capacity is $2\,Ah$ and the failure threshold $U$ is set as $1.38\,Ah$, which means that when the capacity degenerates to $1.38\,Ah$, the experiment will be stopped.

The data set includes the capacity of four batteries, that is, Batteries number 05, number 06, number 07, and number 18. The Li-ion battery capacity degradation is shown in Figure 3.

The detailed descriptions are as follows.

First step is the data preprocessing. An appropriate health indicator is selected and the failure threshold is determined. The data selected for verifying the effectiveness of AFSA-PF and RPF should be sufficient to surpass the failure threshold. Here, we select the capacity of Li-ion batteries as the health indicator.

Second, an appropriate state equation is ensured to be available.

Third, based on the analysis of the data, part of the whole data is truncated to be the history data which are used to determine the unknown parameters in the state equation.

Fourth, the parameters in PF, RPF, and AFSA-PF are set, such as the number of particles, the covariance of the process noise, and the covariance of the measurement noise.

Finally, the RUL of the Li-ion batteries is predicted using PF, RPF, and AFSA-PF, and the results of these methods are then compared.
The empirical degradation model can be described by the following equation:

\[ C_{k+1} = \eta C_k + \beta_1 \exp\left(-\frac{\beta_2}{\Delta t_k}\right), \tag{8} \]

where \( C_k \) denotes the charge capacity of the \( k \)th cycle, \( \eta \) denotes the coulombic efficiency, \( \beta_1 \) and \( \beta_2 \) denote the unknown parameters to be estimated, and \( \Delta t_k \) denotes the rest period between cycles \( k \) and \( k + 1 \). In this study, we set \( \eta = 0.997 \) [26] and \( \Delta t_k = 1 \).

### 3.3. Parameter Settings for Predicting the RUL of Li-Ion Batteries

To integrate the model-based method with the data-driven method, an empirical degradation model is introduced as (8), and the unknown parameters, \( \beta_1 \) and \( \beta_2 \), are identified using PF, which is an efficient method to estimate unknown parameters in nonlinear physical models with non-Gaussian noise [32]. Theoretically, the unknown parameters can be real-time updated using PF as a part of the prognostic process. However, in real engineering applications, the capacity degradation of Li-ion batteries is a gradual decay process, and the model parameters change slowly during the full life cycle. Therefore, it seems to be time-consuming if a real-time updating of the unknown parameters based on history data is adopted. In this study, we use the history data to identify the parameters before the AFSA-PF prediction instead of real-time parameter updating. The detailed real-time parameter updating process can refer to [32].

To avoid the occasionality of the validity of the experiments, we provide the RUL estimation results of Batteries number 05, number 06, and number 18, which, respectively, contain 167, 168, and 132 data points, and the real-life times are 127, 112, and 100 cycles, respectively. Among the three batteries, the shortest life is 100 cycles. Thus, we consider the previous 40, 50, 60, 70, and 80 data samples, respectively, as the history data to determine the unknown parameters \( \beta_1 \) and \( \beta_2 \) and then to predict the RUL of the Li-ion batteries. To illustrate the effect of the proposed method without taking up too much space, we consider an example in which the unknown parameters are determined by the previous 60 data samples.

In the selected exemplificative experiment, we use the former 60 data samples of Battery number 05 to estimate the parameters. Given the randomness in the training algorithm, we typically cannot obtain exactly the same parameter values each time. Thus, in this study, we train the PF 25 times, and finally set the average of the whole estimated results, \( \beta_1 = -0.8 \) and \( \beta_2 = 6 \), as the determined parameters for future use. Given that the three batteries are of the same type and tested under the same conditions, we assume that the obtained state equation is suitable for all of them, which will be verified in the next section.

As mentioned above, the state equation is established using (8) with \( \beta_1 = -0.8 \) and \( \beta_2 = 6 \); the observation equation is set as \( z_{k+1} = C_{k+1} + \nu_k \); and the objective function is set using (7). The parameters are set as follows: the number of particles \( N = 200 \), the state initial values of the battery capacity are the first data in the data set of Batteries number 05, number 06, and number 18, the covariance of the process noise \( \sigma_w = 0.0001 \), and the covariance of the measurement noise \( \sigma_z = 0.0001 \). Noise is considered as the Gaussian white noise to make the experiments closer to the ideal situation. The parameters of the AFSA are set as follows: \( \tau_1 = R = 0.0001, s = 0.00008, v = 0.1, \delta = 0.1 \), the number of iterations is 50, and the number of the AF is 200.

### 4. Results and Discussion

As mentioned in Section 3.3, after all the necessary parameters are set, we predict the RUL of Batteries number 05, number 06, and number 18 using PF, RPF, and AFSA-PF. The prediction results and the PDF distributions of the results are shown in Figures 4–9. In Figures 4, 6, and 8, the blue, red, and green curves before the prediction starting point, respectively, represent the tracking curves of PF, RPF, and AFSA-PF. The tracking curves and the real degradation curve are almost coincident in all the three figures, which demonstrates the effectiveness of the applied empirical degradation model and that the parameters in the state equation determined using the data for Battery number 5 are suitable for the other batteries, which is of great importance in engineering applications.

To analyze the prediction results explicitly, we discuss the RUL prediction results and PDF distributions. Figures 4, 6, and 8 show that the green curve has the best RUL prediction result, which is generated by AFSA-PF. Then the red curve has the second-best result, which is generated by RPF. The blue curve generated by PF comes in last. A summary for the prediction results is shown in Table 1. By comparison, AFSA-PF is obviously superior to PF and RPF, and RPF is better than PF, which is consistent with the principles mentioned before.

As shown in Figures 5, 7, and 9, the distributions of AFSA-PF prediction results are narrower and taller than both that...
Table 1: Prediction results of PF, RPF, and AFSA-PF.

<table>
<thead>
<tr>
<th>Battery number</th>
<th>Prediction method</th>
<th>Prediction result (cycle)</th>
<th>Absolute error (cycle)</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number 05</td>
<td>PF</td>
<td>113</td>
<td>14</td>
<td>11.02%</td>
</tr>
<tr>
<td></td>
<td>RPF</td>
<td>119</td>
<td>8</td>
<td>6.30%</td>
</tr>
<tr>
<td></td>
<td>AFSA-PF</td>
<td>123</td>
<td>4</td>
<td>3.15%</td>
</tr>
<tr>
<td>Number 06</td>
<td>PF</td>
<td>97</td>
<td>15</td>
<td>13.40%</td>
</tr>
<tr>
<td></td>
<td>RPF</td>
<td>105</td>
<td>7</td>
<td>6.25%</td>
</tr>
<tr>
<td></td>
<td>AFSA-PF</td>
<td>110</td>
<td>2</td>
<td>1.79%</td>
</tr>
<tr>
<td>Number 18</td>
<td>PF</td>
<td>86</td>
<td>14</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>RPF</td>
<td>92</td>
<td>8</td>
<td>8%</td>
</tr>
<tr>
<td></td>
<td>AFSA-PF</td>
<td>97</td>
<td>3</td>
<td>3%</td>
</tr>
</tbody>
</table>

The results demonstrate that AFSA-PF is indeed an effective approach in predicting the RUL of the Li-ion batteries. As expected, we have obtained better results from AFSA-PF than PF, and RPF because in AFSA-PF, traditional resampling methods are replaced by the preying and swarming behaviors of AFSA. In this approach, all the artificial fish make an utmost effort to find the optimal result freely, thereby driving the particles to the high-likelihood domain and in turn, improves the distribution of the particles and increases the accuracy and precision.

5. Conclusions and Future Works

This study uses an intelligent prognostic approach based on AFSA and PF to predict the RUL of Li-ion batteries. The main contribution of this research are as follows: (1) successfully applying AFSA-PF to predict the RUL of Li-ion batteries, (2) successfully illustrating the availability of combining model-based and data-driven methods in RUL prediction, and (3) in terms of predicting the RUL of Li-ion batteries, making a detailed comparison between AFSA-PF and RPF, as RPF is regarded as an effective method of preventing particle degeneracy. The experimental results indicate that the proposed AFSA-PF method is suitable for predicting the
RUL of Li-ion batteries. Compared with PF and RPF, AFSA-PF is more accurate and precise. The proposed method shows promising application in other kinds of batteries.

Our future research will focus on two problems. First, in the study, we noticed that the parameters in AFSA have a major influence on optimization. Selecting appropriate parameters for different optimization functions under different required precisions is difficult yet vitally important. Second, more research should be carried on solving the time-consuming problem in real-time prediction, because the introduction of intelligent thought increases algorithm complexity; moreover, how to realize real-time updating parameters without consuming too much resource is a challenge. AFSA itself is a rapid optimization method, but when combined with PF, the speed is slowed down near to that of RPF, which limits the application of both AFSA-PF and RPF in engineering. Fortunately, the AFSA method is so flexible that it can be improved for better timeliness.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

---

**Figure 7:** PDF distribution of the prediction result for Battery number 06.

**Figure 8:** Battery RUL prediction results using basic PF, RPF, and AFSA-PF (Battery number 18).

**Figure 9:** PDF distribution of the prediction result for Battery number 18.
Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grants nos. 61074083 and 51105019) and the Technology Foundation Program of National Defense (Grant no. Z132013B002).

References


