Model Adaptation for Prognostics in a Particle Filtering Framework

Bhaskar Saha[1] and Kai Goebel[2]

[1]*Mission Critical Technologies, Inc. (NASA ARC), 2041 Rosecrans Avenue, Suite 220, El Segundo, CA 90245*
bhaskar.saha@nasa.gov

[2]*NASA Ames Research Center, Moffett Field, CA 95134, USA*
kai.goebel@nasa.gov

ABSTRACT

One of the key motivating factors for using particle filters for prognostics is the ability to include model parameters as part of the state vector to be estimated. This performs model adaptation in conjunction with state tracking, and thus, produces a tuned model that can used for long term predictions. This feature of particle filters works in most part due to the fact that they are not subject to the "curse of dimensionality", i.e. the exponential growth of computational complexity with state dimension. However, in practice, this property holds for "well-designed" particle filters only as dimensionality increases. This paper explores the notion of wellness of design in the context of predicting remaining useful life for individual discharge cycles of Li-ion batteries. Prognostic metrics are used to analyze the tradeoff between different model designs and prediction performance. Results demonstrate how sensitivity analysis may be used to arrive at a well-designed prognostic model that can take advantage of the model adaptation properties of a particle filter.

1. INTRODUCTION

The field of system health management (SHM) is undergoing a paradigm shift from the reliability driven maintenance strategies that relied on metrics like mean-time-to-failure (MTTF), to more proactive condition-based maintenance (CBM) strategies that estimate the remaining useful life (RUL) specific to the system under consideration. This results in more efficient performance, longer system life, as well as reduction in costs from unscheduled maintenance due to unforeseen failures. The applicability of this methodology that was once pioneered by the aerospace and the defense industry now ranges far and wide from green buildings to electric cars to consumer electronics.

The trigger for this evolution has been the concept of *prognostics* and the need to integrate it into the operations and maintenance decisioning process. The definition of what constitutes prognostics is still an open discussion in the SHM community, but for the purposes of this paper, we will define it to be *the process by which the evolution of a system variable or vector indicating its health is tracked over time under current and proposed future usage, until its value no longer falls within the limits set forth by the system specifications*. This somewhat broadens the definition set forth by Saxena *et al*. (2008), where prognostics is triggered by a diagnostic routine, and the detected failure precursor is tracked through time until a predefined end-of-life (EOL) threshold is reached. Other applications may include predicting nominal wear or intermediate cycle-life as discussed in the case of rechargeable batteries by Saha & Goebel (2009).

Prognostic approaches can be broadly classified into two categories: *data-driven* and *model-based*. Data-driven techniques mainly exploit evolution trends of the tracked variable observed from training or archived data under similar operational conditions. Although, they circumvent the need for domain expertise and model development both of which cost time and money, they lead to the problem of data availability and integrity. In most cases, little data is collected from

engineered systems in use. This may not be true for aerospace applications, but even when there is data, very little of it is actually collected under faulty conditions. Accelerated aging tests are even more rare since most systems are either too costly to run to failure, or take too long to do so. Additionally, there are problems with sensor bias and drift, and in some cases, outright failure.

This motivates the development of model-based techniques where domain expertise may be brought to bear. However, most high fidelity models are too computationally intractable to be run in an online environment that can be integrated with the decisioning process. Consequently, there is a need for a model-based prognostic framework that can track the nonlinear dynamics of system health while using a lower-order system representation. The *Particle Filter* (PF) introduced by Gordon *et al*. (1993) is an elegant solution to this need. PFs are a novel class of nonlinear filtering methods that combine *Bayesian learning* techniques with *importance sampling* to provide good state tracking performance. Additionally, model parameters can be included as a part of the state vector to be tracked, thus performing model adaptation in conjunction with state estimation. The model, thus tuned during the tracking phase, can then be propagated subject to expected future use to give long-term prognosis.

2. BACKGROUND

Nonlinear filtering has been an active topic of research for the last several decades in the statistical and engineering community (Jazwinski, 1970). The core problem is to sequentially estimate the state of a dynamic system $\mathbf{x}_k \in \mathbb{R}^{n_x}$, where $\mathbb{R}$ is the set of real numbers and n_x is the dimension of the state vector, using a time-sequence of noisy measurements $\mathbf{z}_k \in \mathbb{R}^{n_z}$, where n_z is the dimension of the measurement vector (Ristic *et al*., 2004). The time index $k \in \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers, is assigned to the continuous-time instant t_k. Thus the state evolution model and the measurement equation may be expressed as:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \omega_{k-1}) \quad (1)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \nu_k) \quad (2)$$

where, $\mathbf{f}$ and $\mathbf{h}$ are known nonlinear functions, and ω and ν represent process and measurement noise sequences, possibly non-Gaussian, whose statistics are known. It is desired to obtain the filtered estimates of $\mathbf{x}_k$ from all available measurements $\mathbf{Z}_k \equiv \{\mathbf{z}_i,\ i = 1,\ldots,k\}$ up to t_k, which, from a Bayesian perspective, amounts to constructing the *posterior* pdf (probability density function) $p(\mathbf{x}_k|\mathbf{Z}_k)$. Once the *initial* density $p(\mathbf{x}_0) \equiv p(\mathbf{x}_0|\mathbf{Z}_0)$ is determined, the pdf may be obtained recursively using the prediction and update steps shown in Eqs. (1) and (2).

Let us say that at time t_{k-1} we have the pdf $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$. In the prediction step the system model in Eq. (1) is used to obtain the *prior* pdf at time t_k via the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} . \quad (3)$$

Assuming a first-order Markov process, $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$, which may be determined from Eq. (1) and the known statistics of ω_{k-1}. Equation (3) thus reduces to:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} . \quad (4)$$

At time t_k when the measurement $\mathbf{z}_k$ is received, the prior pdf is updated using Bayes' rule as follows:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= p(\mathbf{x}_{k-1}|\mathbf{z}_k, \mathbf{Z}_{k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{k-1}) p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} . \end{aligned} \quad (5)$$

The last step of Eq. (5) assumes that the measurements are independent of each other such that $\mathbf{z}_k$ only depends upon $\mathbf{x}_k$. The normalizing constant in the denominator can be represented in terms of the *likelihood* function $p(\mathbf{z}_k|\mathbf{x}_k)$, defined by Eq. (2) and the known statistics of ν, as follows:

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1}) d\mathbf{x}_k . \quad (6)$$

Substituting Eq. (6) into Eq. (5), we can express the *posterior* pdf obtained after the update step as:

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{\int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1}) d\mathbf{x}_k} . \quad (7)$$

The recurrence relations in Eqs. (4) and (7) form the basis for computing the optimal Bayesian estimate. However, these integrals are rarely ever analytical in

nature, thus leading to the need for sub-optimal filters like particle filters. PFs evaluate these integrals by performing Monte Carlo (MC) integration, which is the basis for all sequential Monte Carlo (SMC) estimation methods. Noting the fact that $\int p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1} = \int p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k = 1$, both the integrals in Eqs. (4) and (7) can expressed in the form of:

$$I = \int \Phi(\mathbf{x})\pi(\mathbf{x})d\mathbf{x} \tag{8}$$

where, $\pi(\mathbf{x})$ is of the form $p(\mathbf{x}_{k-l}|\mathbf{Z}_{k-1})$, $l = 0$ or 1, satisfying the pdf properties $\pi(\mathbf{x}) \geq 0$ and $\int \pi(\mathbf{x})d\mathbf{x} = 1$. $\Phi(\mathbf{x})$ may be derived from Eqs. (1) and (2) for Eqs. (4) and (7) respectively. The MC estimate of this integral can expressed as the mean of $N >> 1$ samples $\{\mathbf{x}^i; i = 1,\ldots,N\}$:

$$I_N = \frac{1}{N}\sum_{i=1}^{N}\Phi(\mathbf{x}^i). \tag{9}$$

Assuming independent samples, $\boldsymbol{I}_N$ is an unbiased estimate and, according to the *law of large numbers*, will converge to $\boldsymbol{I}$. Given the fact that in our case $\Phi(\mathbf{x})$ is a pdf constrained within the values of 0 and 1, its variance $\sigma^2 = \int(\Phi(\mathbf{x})-\boldsymbol{I})^2\pi(\mathbf{x})d\mathbf{x}$ is also finite. This means that applying the *central limit theorem* the estimation error can be said to converge as:

$$\lim_{N\to\infty}\sqrt{N}(I_N - I) \sim \mathcal{N}(0,\sigma^2) \tag{10}$$

where $\mathcal{N}(0,\sigma^2)$ denotes a normal distribution with zero mean and variance σ^2. The MC estimate error, $e = \boldsymbol{I}_N - \boldsymbol{I}$, is of the order of $O(N^{1/2})$, which means that the rate of convergence is dependent on the number of particles N, but not the dimension of the state, n_x (Ristic *et al.*, 2004). This leads to the notion that PFs are not subject to the *curse of dimensionality* like other nonlinear filters.

The phrase "curse of dimensionality" was coined by Richard Bellman (1957) more than half a century ago to denote the exponential increase in computational complexity in nonlinear filters as a function of the state dimension n_x. Daum (2005) in his tutorial on nonlinear filters discusses this aspect of particle filters. He states that "It has been asserted that PFs avoid the curse of dimensionality, but this is generally incorrect. Well designed PFs with good proposal densities sometimes avoid the curse of dimensionality, but not otherwise." Figure 1 and Figure 2, reprinted from (Daum, 2005), show the comparison between the median dimensionless error for good and poor proposal densities respectively evaluated over a chosen nonlinear filtering problem with "vaguely Gaussian" conditional densities (Daum & Huang, 2003).

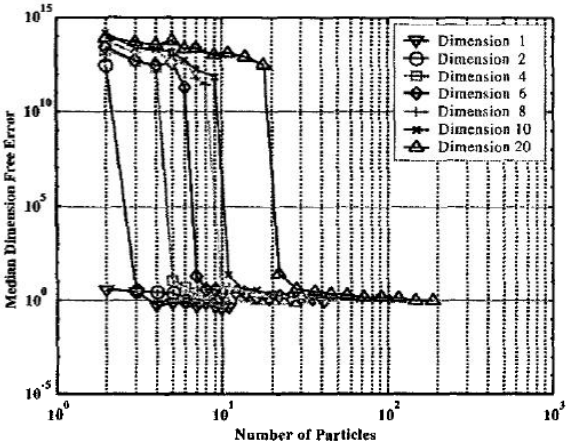

Figure 1. Dimension free error vs. number of particles for PF with good proposal density (Daum, 2005).

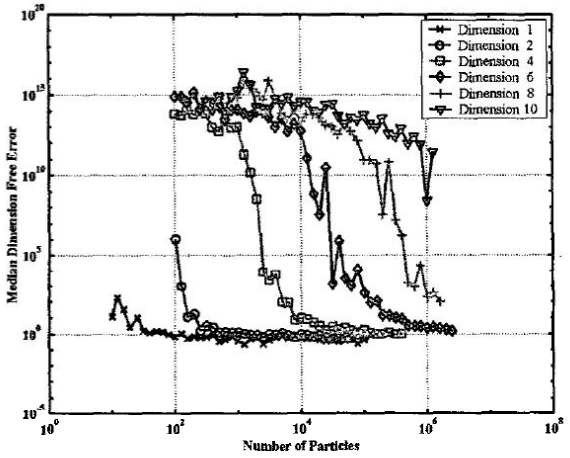

Figure 2. Dimension free error vs. number of particles for PF with poor proposal density (Daum, 2005).

It can be seen from the figures that for a state vector of dimension 8 i.e., $n_x = 8$, the PF with the poor proposal density achieves the same error level with about 10^6 particles that a PF with good proposal density achieves with 10 particles. This discrepancy gets exponentially higher as the dimensionality of the state vector increases linearly, clearly showing that the PF performance does not always escape the curse of dimensionality. Further discussion on this topic can be found in (Daum & Huang, 2003).

The theoretical basis behind the particle filter escaping the curse of dimensionality is that the proposal density

considered, given by the samples $\{\mathbf{x}^i;\ i = 1,\ldots,N\}$, come from the regions of the state space that are important for the pdf integration results in Eqs. (4) and (7). However, it is usually not possible to sample effectively from the posterior distribution $\pi(\mathbf{x})$ being multivariate, non-parametric and, in most cases, unknown beyond a proportionality constant (Ristic *et al*., 2004). In the case of the prognostic problem, even though the system health vector to be tracked may not be high dimensional, the incorporation of model parameters into the state vector, in order to track the non-stationarity of the system model, adds extra dimensions (Saha & Goebel, 2009). Thus, model adaptation that facilitates good prognosis necessitates a good choice of proposal density.

3. THE PROGNOSTICS FRAMEWORK

Before we investigate the issues with model adaptation, let us take a step back and look at how prognostics is performed in the PF framework. The framework has been described before (Saha *et al*., 2009), however, some basic elements are reproduced below in order to set the context. Particle methods assume that the state equations can be modeled as a first order Markov process with additive noise and conditionally independent outputs. Under these assumptions Eqs. (1) and (2) become:

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \omega_{k-1} \tag{11}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \nu_k\,. \tag{12}$$

As mentioned in (Daum, 2005) there are several flavors of PFs. Analyzing all is not within the scope of this paper. Here we shall focus on *Sampling Importance Resampling* (SIR), which is a very commonly used particle filtering algorithm that approximates the posterior filtering distribution denoted as $p(\mathbf{x}_k|\mathbf{Z}_k)$ by a set of N weighted particles $\{\langle x_p^i, w_p^i\rangle;\ i = 1,\ldots,N\}$ sampled from a distribution $q(\mathbf{x})$ that is "similar" to $\pi(\mathbf{x})$, i.e., $\pi(\mathbf{x}) > 0 \Rightarrow q(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$. The *importance weights* w_k^i are normalized in the following way:

$$w_k^i = \frac{\pi(\mathbf{x}_k^i)/q(\mathbf{x}_k^i)}{\sum_{j=1}^{N}\pi(\mathbf{x}_k^j)/q(\mathbf{x}_k^j)} \tag{13}$$

such that $\Sigma_i w_k^i = 1$, and the posterior distribution can be approximated as:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \tag{14}$$

Using the model in Eq. (11) the prediction step from Eq. (4) becomes:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) \approx \sum_{i=1}^{N} w_{k-1}^i \mathbf{f}_{k-1}(\mathbf{x}_{k-1}^i). \tag{15}$$

The weights are updated according to the relation:

$$\overline{w}_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)\,p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\mathbf{z}_k)}, \tag{16}$$

$$w_k^i = \frac{\overline{w}_k^i}{\sum_{j=1}^{N}\overline{w}_k^i}\,. \tag{17}$$

Resampling is used to avoid the problem of degeneracy of the PF algorithm, i.e., avoiding the situation that all but a few of the importance weights are close to zero. If the weights degenerate, we not only have a very poor representation of the system state, but we also spend valuable computing resources on unimportant calculations. More details on this are provided in (Saha *et al*., 2009). The basic logical flowchart is shown in Figure 3.

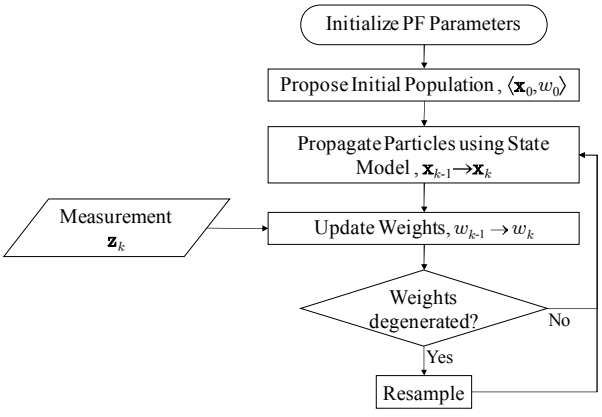

Figure 3. Particle filtering flowchart.

During prognosis this tracking routine is run until a long-term prediction is required, say at time t_p, at which point Eq. (11) will be used to propagate the posterior pdf given by $\{\langle x_p^i, w_p^i\rangle;\ i = 1,\ldots,N\}$ until $\mathbf{x}^i$ fails to meet the system specifications at time t_{EOL}^i. The RUL pdf, i.e., the distribution $p(t_{\text{EOL}}^i - t_p)$, is given by the

distribution of w_p^i. Figure 4 shows the flow diagram of the prediction process.

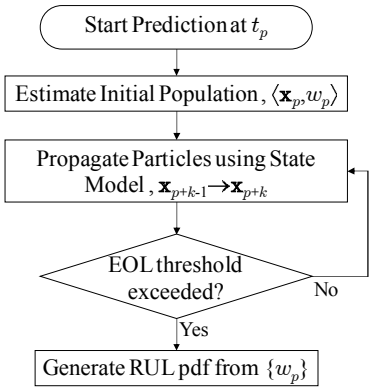

Figure 4. Prediction flowchart.

4. MODEL ADAPTATION

Now that the PF prognostic framework has been set up, let us investigate how we can take advantage of it to perform model adaptation online. For most engineered systems models for nominal operation are available, but true prognostic models like Arrhenius model or Paris' law are comparatively rare. As mentioned before, developing these models require extensive destructive testing which may not be possible in many cases. In some cases, testing may be done on subscale systems, but there may be difficulty in generalizing the models learned. Additionally, the parameter values of these models are often system specific, and thus need to be re-learned for every new application. The PF framework described above can help in these cases by adapting the prognostic/aging model in an online fashion.

For the purposes of this paper we shall assume that the system health state is 1-dimensional, given by x_k, and the state evolution model $\mathbf{f}$ and the measurement model $\mathbf{h}$ are stationary in nature with known noise distributions ω and ν respectively. Additionally, we also assume that the parameter values of $\mathbf{h}$ are known. This assumption can be relaxed in a more generic approach. Indeed, considering a non-stationary measurement model can be used to account for progressive degradation in sensors caused by corrosion, fatigue, wear, etc. The parameters of $\mathbf{f}$, denoted by $\alpha_k = \{\alpha_{j,k};\ j = 1,\ldots,n_{\mathbf{f}}\}$, $n_{\mathbf{f}} \in \mathbb{N}$, are combined with x_k to give the state vector $\mathbf{x}_k = [x_k\ \alpha_k]^T$, where T represents the transpose of a vector or matrix. Equations (11) and (12) can then be rewritten as:

$$x_k = \mathbf{f}(x_{k-1}, \alpha_{k-1}) + \omega_{k-1} \tag{18}$$

$$z_k = \mathbf{h}(x_k) + \nu_k . \tag{19}$$

The issue now is to formulate the state equations for α_k. One easy solution is to pick a *Gaussian random walk* such that:

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k-1} \tag{20}$$

where $\omega_{j,k-1}$ is drawn from a normal distribution, $\mathcal{N}(0,\sigma_j^2)$, with zero mean and variance σ_j^2. Given a suitable starting point $\alpha_{j,0}$, and variance σ_j^2, the PF estimate will converge to the actual parameter value $\bar{\alpha}_j$, according to the *law of large numbers*. In this way, we appear to have introduced model adaptation into the PF framework, adding $n_{\mathbf{f}}$ extra dimensions, yet achieving convergence without incurring the curse of dimensionality.

The notion of a good proposal density, though, comes into play in the choice of the values of $\alpha_{j,0}$ and σ_j^2. If the initial estimate $\alpha_{j,0}$ is far from the actual value and the variance σ_j^2 is small, then the filter may take a large number of steps to converge, if at all. The variance value may be chosen to be higher in order to cover more state-space, but that can also delay convergence. One way to counter this is to make the noise variance itself a state variable that increases if the associated weight is lower than a preset threshold, i.e., the estimated parameter value is far from the true value, and vice-versa. Equation (20) then may be rewritten as:

$$\alpha_{j,k} = \alpha_{j,k-1} + \omega_{j,k-1};\ \ \omega_{j,k-1} \sim \mathcal{N}\left(0,\sigma_{j,k-1}^2\right), \tag{21}$$

$$\sigma_{j,k} = c_{j,k}.\sigma_{j,k-1}; \begin{cases} c_{j,k} < 1, \text{if } w_{k-1} > w_{\text{th}}, \\ c_{j,k} = 1, \text{if } w_{k-1} = w_{\text{th}}, \\ c_{j,k} > 1, \text{if } w_{k-1} < w_{\text{th}}. \end{cases} \tag{22}$$

The multiplier $c_{j,k}$, is a positive valued real number, while the threshold w_{th} is some value in the interval (0, 1). The intent is to increase the search space when the error is high and tightening the search when we are close to the target. Note that although this produces a better proposal density, it introduces a further $n_{\mathbf{f}}$ dimensions to the state vector.

5. SENSITIVITY ANALYSIS

It is quickly evident that it is not feasible to take this approach for all the parameters of a sufficiently high-order model. This motivates the use of sensitivity analysis techniques (SA) to determine the more sensitive parameters that need to be estimated online.

SA is essentially a methodology for systematically changing parameters in a model to determine the effects on the model output. There are several methods to perform SA like local derivatives (Cacuci, 2003), sampling (Helton *et al*., 2006), Monte Carlo sampling (Saltelli et al., 2004), etc. Depending upon the form of the system model any of these methods may be used assess which parameters to target.

In this paper, we assume that the model function $\mathbf{f}$ in Eq. (18) is differentiable, i.e., we can compute $\partial\mathbf{f}/\partial\alpha_j$, time index k dropped for the sake of generality, at any point in the state space defined by $\mathbf{x}_k = [x_k \ \alpha_k]^T$. If the partial derivative is positive, then the value of the function increases with an increase in the parameter value and vice-versa. The magnitude of the derivative indicates the degree to which the parameter affects the output of $\mathbf{f}$. This allows us to choose the parameters to estimate online. For example consider the function:

$$\mathbf{f}(x) = \alpha_1 . \exp(\alpha_2 x) \tag{23}$$

where α_1 and α_2 are the function parameters. Then the partial derivatives are given by:

$$\frac{\partial \mathbf{f}}{\partial \alpha_1} = \exp(\alpha_2 x), \tag{24}$$

$$\frac{\partial \mathbf{f}}{\partial \alpha_2} = \alpha_1 x . \exp(\alpha_2 x). \tag{25}$$

Figure 5 shows the sensitivity analysis of $\mathbf{f}(x)$ due to 10% variation in parameters α_1 and α_2 around the value 10, with x = 1.

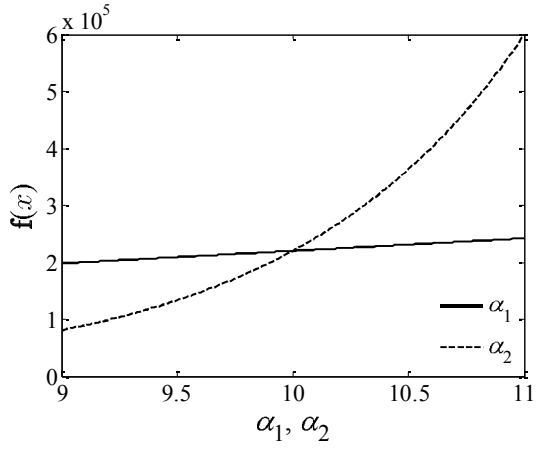

Figure 5. Effect on $\mathbf{f}(x)$ due to 10% variation in parameters α_1 and α_2.

As expected in this simple example, the output of the function is more sensitive to similar variations in the exponential coefficient α_2 than the multiplier α_1, almost by an order of magnitude. Depending on the desired estimation accuracy, α_2 makes a better candidate for online identification than α_1.

Another possibility to note is to replace the random walk model for parameter identification by one that takes into account how a change in the parameter value affects the model output. A similar concept has been applied by Orchard et al., (2009), where they incorporate information from the short term prediction error back into the estimation routine to improve PF performance for both state estimation and prediction. In the case of our example we can construct a similar framework by considering the posterior state error:

$$e_k^i = x_k^i - \sum_{i=1}^{N} w_k^i x_k^i \ . \tag{26}$$

If e_k^i is positive then the parameters that have a positive local partial derivative need to be reduced and those with a negative one need to be increased. The opposite holds true if e_k^i is negative. The amount by which the parameters need to be reduced or increased also depends on the magnitude of the local partial derivative. The higher the magnitude, the smaller steps we take in order to prevent instability while approaching the true value. We can formalize this notion in the following way (the particle index i has been dropped for the sake of generality):

$$\alpha_{jk} = \alpha_{j,k-1} + C_{jk} + \omega_{j,k-1}; \ \ \omega_{j,k-1} \sim \mathcal{N}(0, \sigma_j^2), \tag{27}$$

$$\begin{aligned} C_{jk} &\propto -e_k \ , \\ &\propto \left. \frac{\partial \mathbf{f}}{\partial \alpha_{jk}} \right|_{\mathbf{x}_k} \ , \\ &= -K . \frac{e_k}{\left. \partial \mathbf{f} / \partial \alpha_{jk} \right|_{\mathbf{x}_k}} . \end{aligned} \tag{28}$$

Note that in this model adaptation scenario we are not adding the noise variance parameter to the state vector since the search process is directed and not random as discussed in the precious section.

6. PREDICTING BATTERY DISCHARGE

The application example chosen to investigate the notions described above is the discharge of Lithium-ion rechargeable batteries. The electro-chemistry behind the process as well as the model derivation has been

discussed in detail in (Saha & Goebel, 2009). Some information is repeated here to maintain readability. For the empirical charge depletion model considered here, we express the output voltage $E(t_k)$ of the cell in terms of the effects of the changes in the internal parameters, as shown below:

$$E(t_k) = E^{\circ} - \Delta E_{sd}(t_k) - \Delta E_{rd}(t_k) - \Delta E_{mt}(t_k) \quad (29)$$

where E° is the Gibb's free energy of the cell, ΔE_{sd} is the drop due to self-discharge, ΔE_{rd} is the drop due to cell reactant depletion and ΔE_{mt} denotes the voltage drop due to internal resistance to mass transfer (diffusion of ions). These individual effects are modeled as:

$$\Delta E_{sd}(t_k) = \alpha_{1,k}.\exp(-\alpha_{2,k}/t_k), \quad (30)$$

$$\Delta E_{rd}(t_k) = \alpha_{3,k}.\exp(\alpha_{4,k}t_k), \quad (31)$$

$$\Delta E_{mt}(t_k) = \Delta E_{init} - \alpha_{5,k}t_k . \quad (32)$$

where ΔE_{init} is the initial voltage drop when current flows through the internal resistance of the cell at the start of the discharge cycle, and $\alpha_k = \{\alpha_{j,k};\ j = 1,\ldots,5\}$ represents the set of model parameters to be estimated. Figure 6 shows how the different voltage drop components defined in Eqns. (30)–(32) combine to give the typical constant current Li-ion discharge profile.

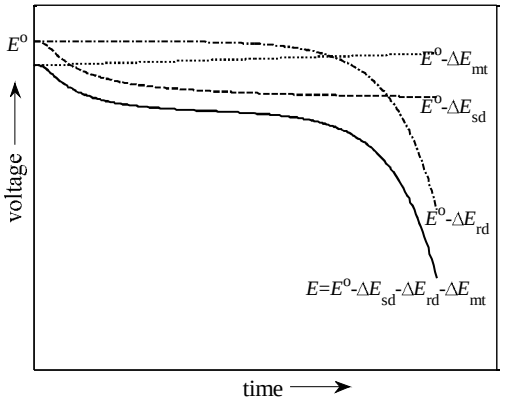

Figure 6. Decomposition of the Li-ion discharge profile in to different components (Saha & Goebel, 2009).

The problem is to predict the end-of-discharge (EOD), i.e., the time instant t_{EOD} when the state x denoting the cell voltage E reaches the threshold level of 2.7 V. The PF representation of this problem is given by:

$$x_k = x_{k-1} - \{\alpha_{1,k-1}\alpha_{2,k-1}\exp(-\alpha_{2,k-1}/t_{k-1})t_{k-1}^{\ 2} - \alpha_{3,k-1}\alpha_{4,k-1}\exp(\alpha_{4,k-1}t_{k-1}) - \alpha_{5,k-1}\}(t_k - t_{k-1}) + \omega_{k-1} \quad (33)$$

$$z_k = x_k + v_k . \quad (34)$$

This is a 6 dimensional state vector with 1 dimension being the system health indicator (cell voltage) and the other dimensions coming from the model parameters.

This is a sufficiently complex problem to investigate the PF-based model adaptation techniques described in the paper, since the critical health variable, battery voltage, is dependent on multiple simultaneous internal processes that are not independently observable. Additionally, the voltage undergoes a very steep and nonlinear transformation near the EOD threshold, as shown in Figure 6, which is difficult to predict early on. For simple voltage tracking purposes, a *random walk* model over the cell voltage, i.e. $E(t_{k+1}) = E(t_k) + \omega_{k-1}$, is enough, but when the voltage trajectory needs to be predicted on the basis of present estimates, then accurate estimates of the underlying model parameters are indispensible. This point is illustrated in Figure 7, which shows that a 10% error in estimating the model parameters $\{\alpha_j;\ j = 1,\ldots,5\}$ can lead to a 15 minute error in determining the remaining battery life.

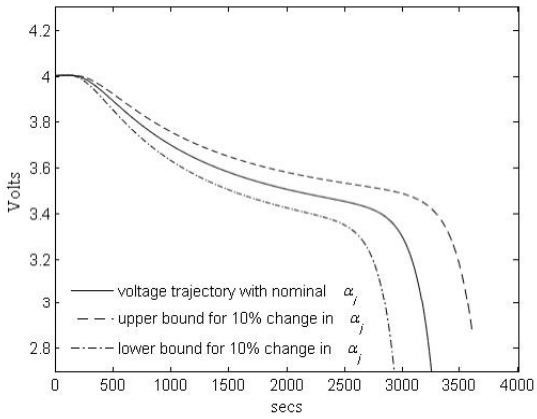

Figure 7. Li-ion discharge trajectories with changes in model parameter estimates.

7. RESULTS

The suitability of using the proposed model adaptation routines, described in Sections 4 and 5, for EOD prediction is measured using the α–λ metric defined in (Saxena *et al.*, 2008). Multiple predictions are made as the battery progressively discharges at a constant current of 2 A. The data have been collected from a

custom built battery prognostics testbed at the NASA Ames Prognostics Center of Excellence (PCoE). An example of the PF prediction output based on 50 particles is shown in Figure 8. The prediction points are denoted by stars in blue. The EOD pdfs overlap as shown on the bottom right with the earlier predictions more faded than the newer ones.

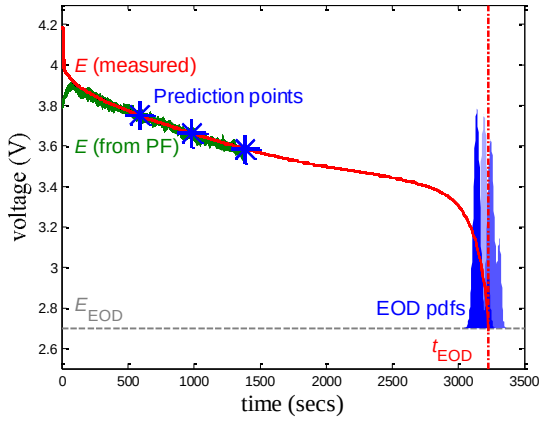

Figure 8. EOD prediction (Saha & Goebel, 2009).

Three different model adaptation routines have been tried:

- *Type A* – the parameters are adapted according to the Gaussian random walk model described in Eq. (20).
- *Type B* – the parameters are adapted based on the noise variance variation strategy described in Eqs. (21) and (22). The threshold w_{th} is chosen to be 0.5.
- *Type C* – the parameters are adapted according to the sensitivity analysis based strategy described in Eqs. (27) and (28). The proportionality factor K is chosen to be 10^5.

For each type of model adaptation 10 EOD prediction runs are conducted each including 13 predictions performed at predetermined time instants. The number of particles is 50 in all cases. The initial population $\langle \mathbf{x}_0^i, w_0^i \rangle$ is also the same for all runs, with w_0^i = 1/50. The initial values of the parameters have been learned from discharge runs at 4 A in order to test the model adaptation performance. Figure 9 shows an example of the variation in parameter values at different discharge levels.

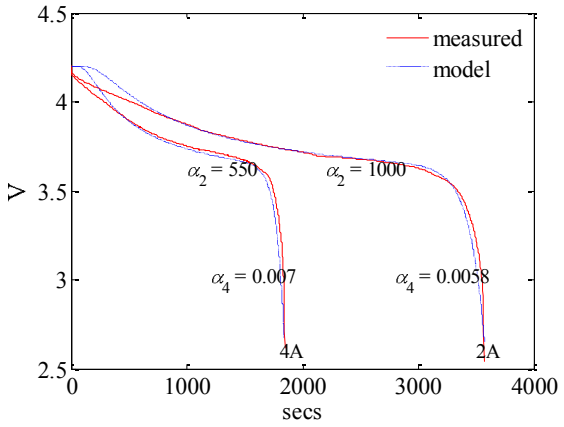

Figure 9. Difference in parameter values for different load currents.

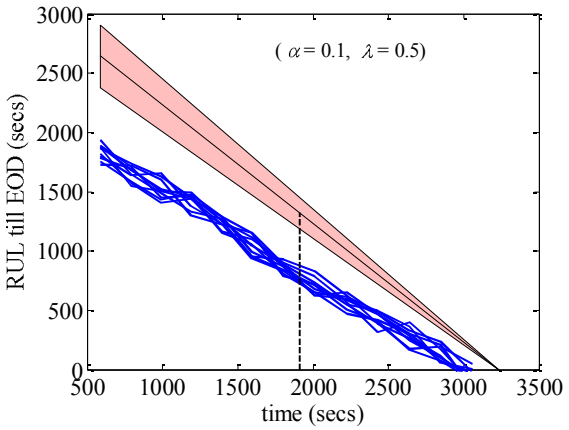

Figure 10. Prognostic performance of model adaptation *type A*.

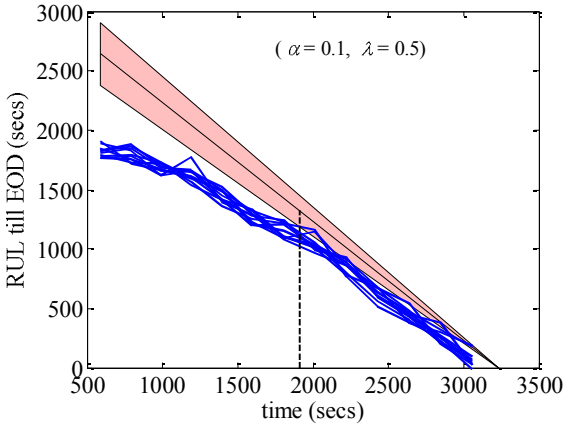

Figure 11. Prognostic performance of model adaptation *type B*.

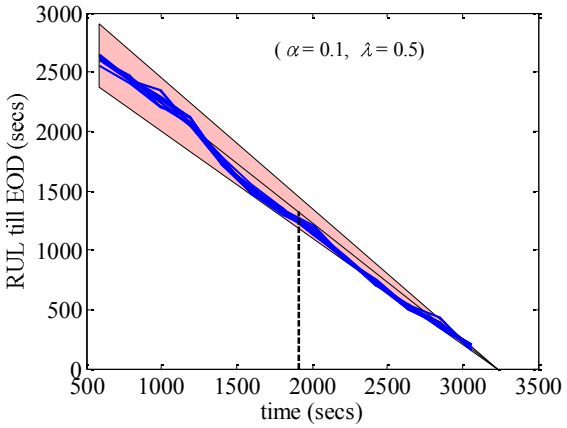

Figure 12. Prognostic performance of model adaptation *type C*.

Figures 10 – 12 show the prognostic performance of the 10 prediction runs of each model adaptation type. As can be seen from Figure 10 the noise variance selected for the model in insufficient to overcome the error between the initial parameter population and the true value.

Figure 11 shows that the noise variance adaptation routine is capable of achieving convergence although it takes up almost half of RUL from the point of prediction to EOD. The SA based adaptation routine performs the best with convergence within 10% (α = 0.1) throughout the prediction horizon as shown in Figure 12, i.e., the model adaptation takes place within the first 500 secs of the discharge. The multiple runs allow us to have some statistical confidence in these results.

Overall, if prognostic performance is evaluated at the 50% mark of the full prediction horizon (λ = 0.5) then only *type C* meets the 10% error performance criterion. In the context of decision making, this prediction can be used to take corrective actions with more than 20 mins remaining. For battery applications, such corrective actions could include altering the load to match the desired battery life.

8. CONCLUSION

In summary, this paper investigates the possibility of performing model adaptation in a PF framework without incurring the curse of dimensionality. It has been shown how various strategies may be used to adapt model parameters online in order to tune the state model for RUL predictions. The feasibility of doing this without incurring the curse of dimensionality has been demonstrated by the application of sensitivity analysis techniques.

However, the analysis performed in this paper is still preliminary in nature since the effects of the initial populations and the priors chosen for the noise variances have not been investigated. Additionally, theoretical analysis of PF convergence bounds while using model adaptation techniques is necessary for the adoption of these methods into Prognostic Health Management (PHM) practice, and will be tackled in future papers.

ACKNOWLEDGEMENT

The funding for this work was provided by the NASA Integrated Vehicle Health Management (IVHM) project under the Aviation Safety Program of the Aeronautics Research Mission Directorate (ARMD).

REFERENCES

Bellman, R.E. (1957). *Dynamic Programming*, Princeton University Press, Princeton, NJ.

Cacuci, D. G. (2003). *Sensitivity and Uncertainty Analysis: Theory, Volume I*, Chapman & Hall.

Daum, F. E. (2005). Nonlinear Filters: Beyond the Kalman Filter, *IEEE A&E Systems Magazine*, vol. 20, no. 8, pp. 57-69.

Daum, F. E. & Huang, J. (2003). Curse of Dimensionality and Particle Filters, *in Proceedings of IEEE Conference on Aerospace*, Big Sky, MT.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107-113.

Helton, J. C., Johnson, J. D., Salaberry, C. J. & Storlie, C. B. (2006). Survey of sampling based methods for uncertainty and sensitivity analysis, *Reliability Engineering and System Safety*, vol. 91, pp. 1175–1209.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*, Academic Press, N. Y.

Orchard, M., Tobar, F. & Vachtsevanos, G.. (2009). Outer Feedback Correction Loops in Particle Filtering-based Prognostic Algorithms: Statistical Performance Comparison, *Studies in Informatics and Control*, vol. 18, issue 4, pp. 295-304.

Ristic, B., Arulampalam, S. & Gordon, N. (2004). *Beyond the Kalman Filter*, Artech House.

Saha, B. & Goebel, K. (2009). Modeling Li-ion Battery Capacity Depletion in a Particle Filtering Framework, *in Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009*, San Diego, CA.

Saha, B., Goebel, K., Poll, S. & Christophersen, J. (2009). Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework, *IEEE Transactions on Instrumentation and Measurement*, vol.58, no.2, pp. 291-296.

Saltelli, A., Tarantola, S., Campolongo, F. & Ratto, M. (2004). *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, John Wiley and Sons.

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S. & Schwabacher, M. (2008). Metrics for Evaluating Performance of Prognostic Techniques, *in Proceedings of Intl. Conf. on Prognostics and Health Management*, Denver, CO.

Bhaskar Saha received his Ph.D. from the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA, USA in 2008. He received his M.S. also from the same school and his B. Tech. (Bachelor of Technology) degree from the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, India. He is currently a Research Scientist with Mission Critical Technologies at the Prognostics Center of Excellence, NASA Ames Research Center. His research is focused on applying various classification, regression and state estimation techniques for predicting remaining useful life of systems and their components, as well as developing hardware-in-the-loop testbeds and prognostic metrics to evaluate their performance. He has been an IEEE member since 2008 and has published several papers on these topics.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics & Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the Associate Principal Investigator for Prognostics of NASA's Integrated Vehicle Health Management Program. He worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds eleven patents and has published more than 100 papers in the area of systems health management.