Improving Computational Efficiency of Prediction in Model-based Prognostics Using the Unscented Transform

Matthew Daigle [1] and Kai Goebel [2]

[1] *University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035, USA*
matthew.j.daigle@nasa.gov
[2] *NASA Ames Research Center, Moffett Field, CA, 94035, USA*
kai.goebel@nasa.gov

ABSTRACT

Model-based prognostics captures system knowledge in the form of physics-based models of components, and how they fail, in order to obtain accurate predictions of end of life (EOL). EOL is predicted based on the estimated current state distribution of a component and expected profiles of future usage. In general, this requires simulations of the component using the underlying models. In this paper, we develop a simulation-based prediction methodology that achieves computational efficiency by performing only the minimal number of simulations needed in order to accurately approximate the mean and variance of the complete EOL distribution. This is performed through the use of the unscented transform, which predicts the means and covariances of a distribution passed through a nonlinear transformation. In this case, the EOL simulation acts as that nonlinear transformation. In this paper, we review the unscented transform, and describe how this concept is applied to efficient EOL prediction. As a case study, we develop a physics-based model of a solenoid valve, and perform simulation experiments to demonstrate improved computational efficiency without sacrificing prediction accuracy.

1. INTRODUCTION

Prognostics is an essential technology for improving system safety, reliability, and availability. Prognostics deals with determining the health state of components, and projecting the evolution of the health into the future to make *end of life* (EOL) and *remaining useful life* (RUL) predictions. Model-based prognostics approaches perform these tasks with the aid of a model that captures knowledge about the system, its components, and their failures, typically in the form of a physics-based model that is derived from first principles (Roemer, Byington, Kacprzynski, & Vachtsevanos, 2005; Byington, Watson, Edwards, & Stoelting, 2004; Saha & Goebel, 2009; Daigle & Goebel, 2010).

The expression of confidence in a prediction provides important information to a decision maker. It is therefore critical to properly represent and process various sources of uncertainty. EOL and RUL can then be, for example, embodied as probability distributions. These distributions are often dominated by the uncertainty of future usage. For the system considered here, we assume a single trajectory of future usage, which, for a given fault mode, makes the distribution unimodal (but not necessarily Gaussian). In this case, the means and variances of these distributions are the most important and useful pieces of information, as they provide information on both the accuracy and spread of the prediction. Often, the EOL distribution is obtained starting with a distribution describing the current state of the system, and propagating that distribution forward to EOL. If the representation of the distribution is sample-based, as with particle filters, then this is straightforward, otherwise, in general, a sample-based representation is needed, as often an analytical solution is unavailable or intractable. Prediction is then performed by simulating each sample forward to EOL. However, this task can be computationally prohibitive due to the large number of samples often needed to accurately represent the state distribution.

In this paper, we develop a novel method to increase the efficiency of the prediction step. We do this using the unscented transform (Julier & Uhlmann, 1997), which is a method to predict the mean and covariance of a distribution that undergoes a nonlinear transformation. In this case, the nonlinear transformation is the simulation to EOL. The unscented transform approximates the given distribution with deterministically selected samples, which are then transformed, and the mean and covariance of the EOL distribution may be computed from the transformed samples. Effectively, only the minimal amount of simulations are being performed, and the samples are chosen in such a way that the predicted mean and covariance closely approximate the mean and covariance obtained by transforming the entire distribution, thus achieving the same result at a fraction of the computational cost, both in time and memory. Since prediction is the main goal of prognostics, computationally efficient

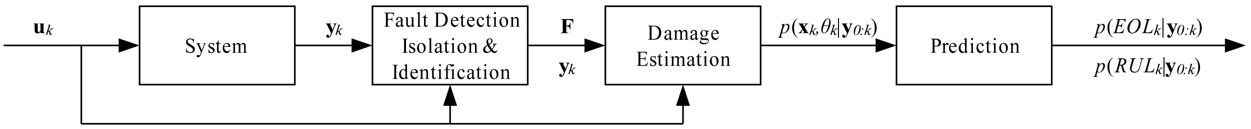
$\mathbf{u}_k$
System
$\mathbf{y}_k$
Fault Detection Isolation & Identification
$\mathbf{F}$
$\mathbf{y}_k$
Damage Estimation
$p(\mathbf{x}_k, \theta_k | \mathbf{y}_{0:k})$
Prediction
$p(EOL_k | \mathbf{y}_{0:k})$
$p(RUL_k | \mathbf{y}_{0:k})$

Figure 1: Prognostics architecture.

prediction is of utmost importance. Efficient prediction methods take less time, so, therefore, more predictions can be made at a faster rate.

We review the common forms of the unscented transform, and develop the new prediction methodology as part of our model-based prognostics framework (Daigle & Goebel, 2009, 2010). As a case study, we construct a detailed physics-based model of a solenoid valve that includes models of different damage mechanisms and their progression. Solenoid valves have application in many domains, and reliable performance of these valves is crucial to many complex systems (Tansel, Perotti, Yenilmez, & Chen, 2005). We run a set of simulation-based prognostics experiments, using the solenoid valve model, to demonstrate the application of the new prediction methodology and compare it to the baseline approach.

The paper is organized as follows. Section 2. describes the prognostics approach. Section 3. presents the modeling methodology and develops the model of the solenoid valve. Section 4. discusses the damage estimation approach. Section 5. overviews the unscented transform and develops the new prediction procedure. Section 6. presents comprehensive simulation experiments applying the framework to the solenoid valve case study. Section 7. concludes the paper.

2. PROGNOSTICS APPROACH

The problem of prognostics is to predict the EOL and/or the RUL of a component. In this section, we first formally define the problem of model-based prognostics. We then describe a general model-based architecture within which a prognostics solution may be implemented.

2.1 Problem Formulation

In a general model-based prognostics approach, the system model may be given by

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{v}(t))\\ \mathbf{y}(t) &= \mathbf{h}(t, \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{u}(t), \mathbf{n}(t)),\end{aligned}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$ is the parameter vector, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(t) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation. The parameters $\boldsymbol{\theta}(t)$ evolve by some unknown process, but, in practice, are typically considered to be constant.

Our goal is to predict EOL at a given time point t_P using the discrete sequence of observations up to time t_P, denoted as $\mathbf{y}_{0:t_P}$. EOL is defined as the time point at which the component no longer meets a functional requirement (e.g., a valve does not open in the required amount of time). This point is often linked to a damage threshold, beyond which the component fails to function properly. In general, we may express this threshold as a function of the system state and parameters, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t))$, which determines whether EOL has been reached, where

$$T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = \begin{cases} 1, & \text{if EOL is reached} \\ 0, & \text{otherwise.} \end{cases}$$

Using this function, we can formally define EOL with

$$EOL(t_P) \triangleq \operatorname*{arg\,min}_{t \geq t_P} T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1,$$

and RUL with

$$RUL(t_P) \triangleq EOL(t_P) - t_P.$$

Due to the many sources of uncertainty that exist in the prediction problem, it is much more useful to compute a probability distribution of the EOL or RUL, rather than a single prediction point. The goal, then, is to compute, at time t_P, $p(EOL(t_p)|\mathbf{y}_{0:t_P})$ or $p(RUL(t_P)|\mathbf{y}_{0:t_P})$.

2.2 Prognostics Architecture

We adopt a model-based approach, wherein we develop detailed physics-based models of components and systems that include descriptions of how fault parameters evolve in time. These models depend on unknown and possibly time-varying wear parameters, $\boldsymbol{\theta}(t)$. Therefore, our solution to the prognostics problem takes the perspective of joint state-parameter estimation. In discrete time k, we estimate $\mathbf{x}_k$ and $\boldsymbol{\theta}_k$, and use these estimates to predict EOL and RUL at desired time points. Using $p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P}|\mathbf{y}_{0:k_P})$ at prediction time k_P, we compute $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$.

We employ the prognostics architecture in Fig. 1 (Daigle & Goebel, 2010). The system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The fault detection, isolation, and identification (FDII) module determines a fault set $\mathbf{F}$, which is used by the damage estimation module to determine estimates of the states and unknown parameters, represented as a probability distribution $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k})$. The prediction module uses this distribution, along with hypothesized future inputs, to compute EOL and RUL as probability distributions $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$. In this paper, we focus on the damage estimation and prediction modules, and assume a solution to FDII.

3. SOLENOID VALVE MODELING

We apply our prognostics approach to a solenoid valve, and develop a physics-based model of its nominal and faulty behavior. A typical three-way, two-position solenoid valve for controlling gas flow is shown in Fig. 2. The valve is held in its de-energized position by the return spring, as shown in the figure. In this position, gas is

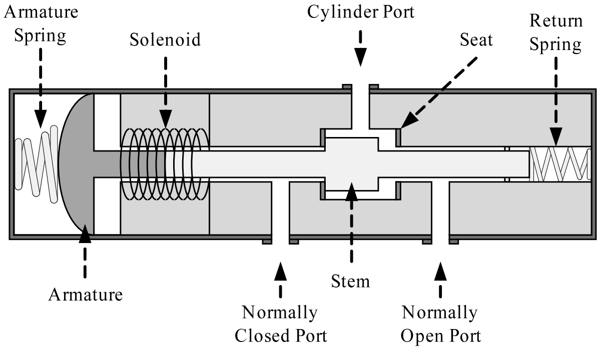

Figure 2: Three-way two-position solenoid valve.

allowed to pass between the normally open port and the cylinder port. To energize the valve, a voltage is applied to the solenoid, which produces an electromagnetic force that moves the valve stem towards its energized position until it contacts the seat. In this position, gas is allowed to pass between the normally closed port and the cylinder port. We refer to the de-energized position as the closed position, and the energized position as the open position.

The state $\mathbf{x}$ of the solenoid valve is given by

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ v(t) \\ i(t) \end{bmatrix},$$

where $x(t)$ is the valve position, $v(t)$ is the valve velocity, and $i(t)$ is the solenoid current. We define $x = 0$ as the position of the valve when in the closed (de-energized) position, and $x = L_s$ as the position of the valve when in the open (energized) position, where L_s is the length of the valve stroke.

The position derivative is given by $v(t)$, and the velocity derivative is determined from the forces acting on the stem:

$$\frac{dv(t)}{dt} = \frac{1}{m}\left(F_e(t) - k(x(t) - x_o) - rv(t) - F_c(t)\right),$$

where $F_e(t)$ is the electromagnetic force, k is the return spring constant and x_o is the amount of spring compression when the valve is in the closed position (where we lump the armature and return spring into a single spring), r is the kinetic friction coefficient, and $F_c(t)$ is the contact force with the seat, which may be described by

$$F_c(t) = \begin{cases} k_c(-x), & \text{if } x < 0, \\ 0, & \text{if } 0 \leq x \leq L_s, \\ -k_c(x - L_s), & \text{if } x > L_s, \end{cases}$$

where k_c is the (large) spring constant associated with the flexible seats. In general, we may also consider forces from the gas flowing through the valve, however, here, we assume a balanced design in which the pressure forces always cancel.

The solenoid force is given by

$$F_e(t) = \frac{1}{2} i(t)^2 \frac{\partial L(x)}{\partial x},$$

where $L(x)$ is the inductance of the solenoid (Lyshevski, Sinha, & Seger, 1999; Rahman, Cheung, & Lim, 1996). The force acts to decrease the reluctance of the magnetic circuit by decreasing the air gap, which is a function of x, thus acting to open the valve. The solenoid current is described by

$$\frac{di(t)}{dt} = \frac{1}{L(x)}\left(u(t) - Ri(t) - i(t)\frac{\partial L(x)}{\partial x} v(t)\right),$$

where $u(t)$ is the applied voltage, and R is the coil resistance (Lyshevski et al., 1999; Rahman et al., 1996; Szente & Vad, 2001). The voltage $u(t)$ is the only external input considered here, i.e.,

$$\mathbf{u}(t) = [u(t)].$$

The inductance of a solenoid is given by

$$L(x) = \frac{N^2}{\mathcal{R}(x)},$$

where N is the number of wire turns in the coil, and $\mathcal{R}$ is the reluctance of the magnetic circuit. In general, reluctance is given by

$$\mathcal{R} = \frac{l}{\mu A},$$

where l is the length of the magnetic circuit, A is the cross-sectional area of the circuit, and μ is the magnetic permeability of the material. If we define the maximum air gap as g_0, then the actual air gap is given by $g_0 - x$. The reluctance depends on the geometry of the solenoid. We may assume a typical geometry in which reluctance is described by

$$\mathcal{R}(x) = \frac{l_c}{\mu_c A_c} + \frac{g_0 - x}{\mu_0 A_g},$$

where the c subscript denotes lumped parameters for the core and armature, μ_0 is the permeability of air, and A_g is the effective cross-sectional area of the air gap (Lyshevski et al., 1999). Therefore, the inductance is given by

$$L(x) = \frac{N^2 \mu_0 A_g \mu_c A_c}{\mu_0 A_g l_c + \mu_c A_c (g_0 - x)},$$

and its derivative with respect to x is

$$\frac{\partial L(x)}{\partial x} = \frac{N^2 \mu_0 A_g \mu_c^2 A_c^2}{\left(\mu_0 A_g l_c + \mu_c A_c (g_0 - x)\right)^2}.$$

We select our complete measurement vector as

$$\mathbf{y}(t) = \begin{bmatrix} x(t) \\ i(t) \\ open(t) \\ closed(t) \end{bmatrix},$$

The $open(t)$ and $closed(t)$ measurements are discrete sensors which output 1 if the valve is in the fully opened or fully closed state:

$$open(t) = \begin{cases} 1, & \text{if } x(t) \geq L_s \\ 0, & \text{otherwise} \end{cases}$$

$$closed(t) = \begin{cases} 1, & \text{if } x(t) \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

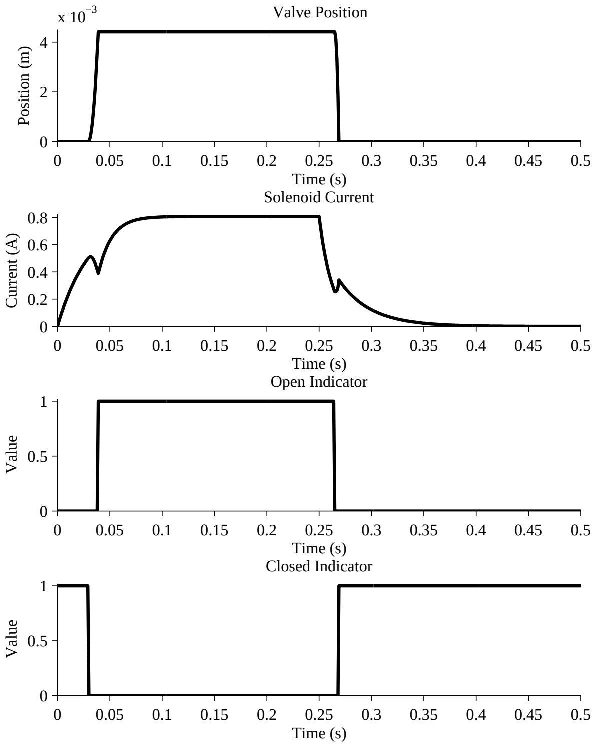

Figure 3: Nominal solenoid valve operation

Fig. 3 shows a nominal valve cycle. The valve is commanded to open at 0 s. The current and magnetic field build up in the solenoid, and soon, enough force is produced to overcome friction and the return spring. As the valve moves, the $i(t)\dfrac{\partial L(x)}{\partial x}v(t)$ term begins to dominate, causing the current to decrease. When the valve opens against the seat and stops moving, the current increases again, resulting in the cusp observed in the current just before 0.05 s. The current then increases to its steady state, determined by the applied voltage and the coil resistance. At 0.25 seconds, the valve is commanded to close by removing the applied voltage. The current drains out of the solenoid, and soon, the electromagnetic force is no longer strong enough to keep the valve in place. The valve begins to close, and, as the $i(t)\dfrac{\partial L(x)}{\partial x}v(t)$ term again comes to dominate, the current increases briefly until the valve fully closes and $v(t)$ becomes 0, resulting in another cusp. The current then decreases smoothly to 0.

3.1 Damage Modeling

In our modeling methodology, the nominal model is extended with damage models. These models describe how parameters associated with the degree of valve damage progress in time, and allow us to make predictions of damage progression. From valve documentation and historical maintenance records, we have identified the most relevant faults for prognostics. The set of faults includes friction damage, spring damage, and the accumulation of debris on the valve seats.

A common damage mechanism present in valves is sliding wear (Daigle & Goebel, 2009). The equation for sliding wear takes on the following form:

$$\dot{V}(t) = w|F(t)v(t)|,$$

where $V(t)$ is the wear volume, w is the wear coefficient (which depends on material properties such as hardness), $F(t)$ is the sliding force, and $v(t)$ is the sliding velocity (Hutchings, 1992). Friction will increase linearly with sliding wear, because the contact area between the sliding bodies becomes greater as surface asperities wear down (Hutchings, 1992). We characterize friction damage by a change in the friction coefficient, and model the damage progression in a form similar to sliding wear (Daigle & Goebel, 2009):

$$\dot{r}(t) = w_r|F_f(t)v(t)|$$

where w_r is the wear coefficient, and $F_f(t)$ is the friction force defined previously. The friction parameter only grows when the valve is moving, so, the friction parameter evolves in a step-wise fashion, with damage only occurring during the valve's opening and closing motions. As the friction parameter increases, the friction force increases, further increasing the rate at which the friction parameter grows, resulting in a damage progression similar to an exponential when viewed at large time scales. We define r^+ as the largest value of the friction coefficient at which the valve still actuates in the required time. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if $r(t) > r^+$.

We assume a similar equation form for spring damage (Daigle & Goebel, 2009):

$$\dot{k}(t) = -w_k|F_s(t)v(t)|,$$

where w_k is the spring wear coefficient and $F_s(t)$ is the spring force. The more the spring is used, the weaker it becomes, characterized by the change in the spring constant. As with friction damage, the spring constant only decreases when the valve moves. As the spring becomes damaged, the spring force will decrease, and so the rate at which spring damage occurs will also decrease. We define k^- as the smallest value of the spring constant at which the valve still closes in the required time. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if $k(t) < k^-$.

Another failure relates to the accumulation of particulate matter and other forms of debris at the seats. As debris builds up, it impedes the valve's travel and prevents the valve from fully opening or closing, which, in turn, causes leaks through the valve. We assume that the accumulation of debris is due to sliding wear. It results in a change in the boundary conditions of the valve motion. We define L_c as the boundary when the valve is in the closed position (nominally 0, where $L_c \geq 0$), and $L_s - L_o$ as the boundary when in the open position (nominally L_s, where $L_o \geq 0$). We assume that the rates of change of the offsets L_c and L_o grow proportionally to sliding wear:

$$\dot{L}_c(t) = w_c|F_f(t)v(t)|$$
$$\dot{L}_o(t) = w_o|F_f(t)v(t)|.$$

We define L_c^+ and L_o^+ as the largest allowable values of the offsets. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if $L_c(t) > L_c^+$ or $L_o(t) > L_o^+$.

Algorithm 1 SIR Filter

```
**Inputs:** $\{(\mathbf{x}^i_{k-1}, \boldsymbol{\theta}^i_{k-1}), w^i_{k-1}\}^N_{i=1}, \mathbf{u}_{k-1:k}, \mathbf{y}_k$
**Outputs:** $\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1}$
**for** $i = 1$ **to** $N$ **do**
    $\boldsymbol{\theta}^i_k \sim p(\boldsymbol{\theta}_k | \boldsymbol{\theta}^i_{k-1})$
    $\mathbf{x}^i_k \sim p(\mathbf{x}_k | \mathbf{x}^i_{k-1}, \boldsymbol{\theta}^i_{k-1}, \mathbf{u}_{k-1})$
    $w^i_k \leftarrow p(\mathbf{y}_k | \mathbf{x}^i_k, \boldsymbol{\theta}^i_k, \mathbf{u}_k)$
**end for**
$W \leftarrow \sum_{i=1}^{N} w^i_k$
**for** $i = 1$ to $N$ **do**
    $w^i_k \leftarrow w^i_k / W$
**end for**
$\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1} \leftarrow \texttt{Resample}(\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1})$
```

4. DAMAGE ESTIMATION

In the model-based paradigm, damage estimation reduces to joint state-parameter estimation, i.e., computation of $p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k})$. A general solution to this problem is the *particle filter*, which may be directly applied to nonlinear systems with non-Gaussian noise terms. Particle filters offer approximate (suboptimal) solutions for systems where optimal solutions are unavailable or intractable (Arulampalam, Maskell, Gordon, & Clapp, 2002; Cappe, Godsill, & Moulines, 2007). In particle filters, the state distribution is approximated by a set of discrete weighted samples, called *particles*. As the number of particles is increased, performance increases and the optimal solution is approached.

With particle filters, the particle approximation to the state distribution is given by

$$\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1},$$

where N denotes the number of particles, and for particle i, $\mathbf{x}^i_k$ denotes the state vector estimate, $\boldsymbol{\theta}^i_k$ denotes the parameter vector estimate, and w^i_k denotes the weight. The posterior density is approximated by

$$p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k}) \approx \sum_{i=1}^{N} w^i_k \delta_{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k)}(d\mathbf{x}_k d\boldsymbol{\theta}_k),$$

where $\delta_{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k)}(d\mathbf{x}_k d\boldsymbol{\theta}_k)$ denotes the Dirac delta function located at $(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k)$.

We employ the sampling importance resampling (SIR) particle filter, and implement the resampling step using systematic resampling (Kitagawa, 1996). The pseudocode for a single step of the SIR filter is shown as Algorithm 1. Each particle is propagated forward to time k by first sampling new parameter values and sampling new states. The particle weight is assigned using $\mathbf{y}_k$. The weights are then normalized, followed by the resampling step[1].

Here, the parameters $\boldsymbol{\theta}_k$ evolve by some unknown process that is independent of the state $\mathbf{x}_k$. However, we need to assign some type of evolution to the parameters. The typical solution is to use a random walk, i.e., for parameter θ, $\theta_k = \theta_{k-1} + \xi_{k-1}$, where ξ_{k-1} is typically Gaussian noise. With this type of evolution, the particles generated with parameter values closest to the true values should be assigned higher weight, thus allowing the particle filter to converge to the true values. The selected variance of the random walk noise determines both the rate of this convergence and the estimation performance once convergence is achieved.

Note that in a particle filter, a certain amount of sensor noise must be assumed, but, in practice, the discrete position sensors (*open* and *closed*) have no noise, therefore, a small amount of noise must be assumed within the particle filter for those sensors.

5. PREDICTION

Prediction is initiated at a given time k_P. Using the current joint state-parameter estimate, $p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P} | \mathbf{y}_{0:k_P})$, which represents the most up-to-date knowledge of the system at time k_P, the goal is to compute $p(EOL_{k_P} | \mathbf{y}_{0:k_P})$ and $p(RUL_{k_P} | \mathbf{y}_{0:k_P})$. As discussed in Section 4., the particle filter computes

$$p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w^i_{k_P} \delta_{(\mathbf{x}^i_{k_P}, \boldsymbol{\theta}^i_{k_P})}(d\mathbf{x}_{k_P} d\boldsymbol{\theta}_{k_P}).$$

We can approximate a prediction distribution n steps forward as (Doucet, Godsill, & Andrieu, 2000)

$$p(\mathbf{x}_{k_P+n}, \boldsymbol{\theta}_{k_P+n} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w^i_{k_P} \delta_{(\mathbf{x}^i_{k_P+n}, \boldsymbol{\theta}^i_{k_P+n})}(d\mathbf{x}_{k_P+n} d\boldsymbol{\theta}_{k_P+n}).$$

So, for a particle i propagated n steps forward without new data, we may take its weight as $w^i_{k_P}$. Similarly, we can approximate the EOL as

$$p(EOL_{k_P} | \mathbf{y}_{0:k_P}) \approx \sum_{i=1}^{N} w^i_{k_P} \delta_{EOL^i_{k_P}}(dEOL_{k_P}).$$

To compute EOL, then, we propagate each particle forward to its own EOL and use that particle's weight at k_P for the weight of its EOL prediction.

If an analytical solution exists for the prediction, this may be directly used to obtain the prediction from the state-parameter distribution. An analytical solution is rarely available, so the general approach to solving the prediction problem is through simulation. Each particle is simulated forward to EOL to obtain the complete EOL distribution. The pseudocode for the baseline prediction procedure is given as Algorithm 2 (Daigle & Goebel, 2010). Each particle i is propagated forward until $T_{EOL}(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k)$ evaluates to 1; at this point EOL has been reached for this particle.

Note that, in general, we may sample new parameter values $\boldsymbol{\theta}$, however, the noise considered here should typically be considerably less than the noise used for the random walk during the estimation phase, as we usually assume these parameters are either constant or only exhibit very small deviations. Note also that prediction requires hypothesizing future inputs of the system, $\hat{\mathbf{u}}_k$, because damage progression is rarely independent of the system inputs. For this reason the inputs must be chosen carefully. Here, we assume only a single future input

[1] Pseudocode for the systematic resampling algorithm is provided in (Arulampalam et al., 2002).

Algorithm 2 EOL Prediction

```
**Inputs:** $\{(\mathbf{x}^i_{k_P}, \boldsymbol{\theta}^i_{k_P}), w^i_{k_P}\}^N_{i=1}$
**Outputs:** $\{EOL^i_{k_P}, w^i_{k_P}\}^N_{i=1}$
**for** $i = 1$ **to** $N$ **do**
    $k \leftarrow k_P$
    $\mathbf{x}^i_k \leftarrow \mathbf{x}^i_{k_P}$
    $\boldsymbol{\theta}^i_k \leftarrow \boldsymbol{\theta}^i_{k_P}$
    **while** $T_{EOL}(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k) = 0$ **do**
        Predict $\hat{\mathbf{u}}_k$
        $\boldsymbol{\theta}^i_{k+1} \sim p(\boldsymbol{\theta}_{k+1}|\boldsymbol{\theta}^i_k)$
        $\mathbf{x}^i_{k+1} \sim p(\mathbf{x}_{k+1}|\mathbf{x}^i_k, \boldsymbol{\theta}^i_k, \hat{\mathbf{u}}_k)$
        $k \leftarrow k + 1$
        $\mathbf{x}^i_k \leftarrow \mathbf{x}^i_{k+1}$
        $\boldsymbol{\theta}^i_k \leftarrow \boldsymbol{\theta}^i_{k+1}$
    **end while**
    $EOL^i_{k_P} \leftarrow k$
**end for**
```

trajectory, i.e., $\hat{\mathbf{u}}_k$ is defined uniquely for all values of k. This is a practical assumption for the solenoid valve, because the valve is always fully opened or fully closed, and a single voltage value $u(t)$ is consistently applied for opening the valve. Since damage occurs only when the valve is moving, then for the purposes of prediction, we may produce an input sequence that represents a full valve cycle (e.g., that of Fig. 3) repeated indefinitely, and, using this, we may obtain EOL and RUL predictions in the number of valve cycles.

5.1 Computationally Efficient Prediction

The computational complexity of the prediction procedure presented as Algorithm 2 is linear in the number of particles, however, each particle may take a variable amount of time to simulate to EOL. Particles that predict quickly progressing wear will complete quickly, while particles that predict slowly progressing wear will complete slowly, because many more simulation steps will be needed to reach EOL. This problem is exacerbated with models that require very small sampling periods. In fact, particles with very poor wear parameter estimates, i.e., close to 0, which correspond to very large EOL predictions, may take an exceedingly long time. Also, these particles may correspond to outliers, and, as such, contribute little to the prediction distribution.

The only way to reduce the computational effort is to reduce the number of particles that are used in the prediction step. One approach is to randomly select an arbitrary number of particles from the original distribution, but the statistics of the original distribution may not be preserved. A better approach is to sample from the distribution in such a way that the important statistical information is preserved, and the EOL distribution computed from this limited sample set closely approximates the statistical properties of the EOL distribution computed from the complete set of samples.

The unscented transform solves this problem. It takes a random variable $\mathbf{x} \in \mathbb{R}^{n_x}$, with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$, which is related to a second random variable $\mathbf{y}$ by some nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$, and computes the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_{yy}$ using a (small) set of *deterministically* selected weighted samples, called *sigma points* (Julier & Uhlmann, 1997). For the task of EOL prediction, $\mathbf{x}$ is simply the joint state-parameter distribution represented by $\{(\mathbf{x}^i_{k_P}, \boldsymbol{\theta}^i_{k_P}), w^i_{k_P}\}^N_{i=1}$, $\mathbf{g}$ is the function that computes EOL (i.e., simulates a particle to EOL), and $\mathbf{y}$ is the EOL. The required mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ may be computed from the particle distributions using the formulas for weighted mean and weighted covariance.

The statistics of $\mathbf{y}$ are computed by selecting a set of weighted sigma points from $\mathbf{x}$, where $\boldsymbol{\mathcal{X}}_i$ denotes the ith point and w_i denotes its weight. The sigma points are always chosen such that the mean and covariance match those of the original distribution, $\bar{\mathbf{x}}$ and $\mathbf{P}_{xx}$. Each sigma point is passed through $\mathbf{g}$ to obtain new sigma points $\boldsymbol{\mathcal{Y}}$, i.e.,

$$\boldsymbol{\mathcal{Y}}_i = \mathbf{g}(\boldsymbol{\mathcal{X}}_i)$$

with mean and covariance calculated as

$$\bar{\mathbf{y}} = \sum_i w_i \boldsymbol{\mathcal{Y}}_i$$

$$\mathbf{P}_{yy} = \sum_i w_i (\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}})(\boldsymbol{\mathcal{Y}}_i - \bar{\mathbf{y}})^T.$$

The underlying idea of the unscented transform is that it is easier to approximate the distribution $\mathbf{x}$ than to approximate the nonlinear function $\mathbf{g}$. This is the idea behind the unscented Kalman filter, where the unscented transform is exploited for nonlinear state estimation (Julier & Uhlmann, 1997, 2004). At each step, the unscented transform is applied to the state estimate and is used for a single step prediction. In contrast, here, we apply the transform to the state-parameter distribution at given single time point k_P, and use this for multi-step predictions to EOL.

Several methods exist for selecting sigma points. In the following sections, we briefly review three common unscented transforms, and compare their fidelity on an example EOL prediction problem. Detailed performance results will be presented in Section 6. for fault prognosis of the solenoid valve.

Symmetric Unscented Transform

In the symmetric unscented transform, $2n_x + 1$ sigma points are selected symmetrically about the mean in the following way (Julier & Uhlmann, 2004):

$$w_i = \begin{cases} \dfrac{\kappa}{(n_x + \kappa)}, & i = 0 \\ \dfrac{1}{2(n_x + \kappa)}, & i = 1, \ldots, 2n_x \end{cases}$$

$$\boldsymbol{\mathcal{X}}_i = \begin{cases} \bar{\mathbf{x}}, & i = 0 \\ \bar{\mathbf{x}} + \left(\sqrt{(n_x + \kappa)\mathbf{P}_{xx}}\right)_i, & i = 1, \ldots, n_x \\ \bar{\mathbf{x}} - \left(\sqrt{(n_x + \kappa)\mathbf{P}_{xx}}\right)_i, & i = n_x + 1, \ldots, 2n_x \end{cases},$$

where $\left(\sqrt{(n_x + \kappa)\mathbf{P}_{xx}}\right)_i$ refers to the ith column of the matrix square root of $(n_x + \kappa)\mathbf{P}_{xx}$ (e.g., computed using the Cholesky decomposition). The number κ is a free parameter that can be used to tune the higher order moments of the distribution. If $\mathbf{x}$ is assumed Gaussian, then selecting $\kappa = 3 - n_x$ is recommended (Julier & Uhlmann, 1997). A smaller value of κ will bring the sigma points closer together. Note that the sigma point weights do not directly represent probabilities, so are not restricted to the interval $[0, 1]$.

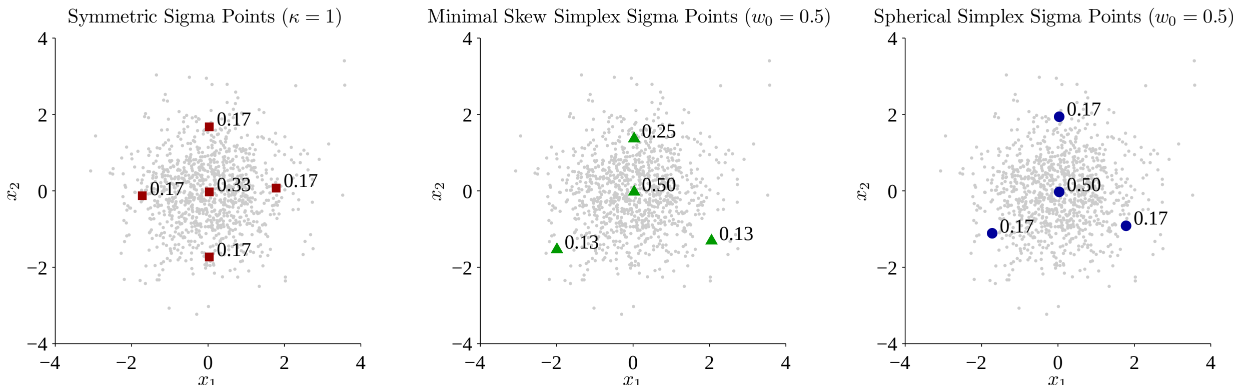

Figure 4: Sigma point locations and weights for a two-dimensional random variable $\mathbf{x}$ with $\bar{\mathbf{x}} = 0$ and $\mathbf{P}_{xx} = \mathbf{I}$. The gray dots represent the random samples, and the markers represent the sigma points, labeled with their weights.

Minimal Skew Simplex Unscented Transform

The symmetric unscented transform uses $2n_x + 1$ sigma points, however, it is possible to reduce the number of points to $n_x + 2$, while still capturing the first two moments of the distribution, thus reducing the amount of computation. The minimal skew sigma points are such a set, and satisfy an additional constraint in which the skew (third moment) is minimized, which reduces the average error for a symmetric distribution (Julier & Uhlmann, 2002).

The minimal skew sigma points are selected in a constructive manner, first by choosing the set of points for $n_x = 1$, and then increasing n_x by one until the full dimension is reached. The procedure for selecting sigma points for dimension n_x for a distribution with mean 0 and $\mathbf{P}_{xx} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, is as follows (Julier & Uhlmann, 2002). First, the weight of the 0th sigma point is selected freely as

$$w_0 \in [0, 1].$$

The remaining weights are computed using

$$w_i = \begin{cases} \dfrac{1 - w_0}{2^{n_x}}, & i = 1, 2 \\ 2^{i-2} w_1, & i = 3, \dots, n_x + 1 \end{cases}$$

For the initial dimension size $j = 1$, where $\boldsymbol{\mathcal{X}}_i^j$ refers to the ith sigma point for the jth dimensional space, the sigma points are initialized as

$$\boldsymbol{\mathcal{X}}_0^1 = 0$$
$$\boldsymbol{\mathcal{X}}_1^1 = -\frac{1}{\sqrt{2w_1}}$$
$$\boldsymbol{\mathcal{X}}_2^1 = \frac{1}{\sqrt{2w_1}}.$$

Expanding up to higher dimensions $j = 2, \dots, n_x$, the higher-dimensional sigma points are recursively defined as

$$\boldsymbol{\mathcal{X}}_i^j = \begin{cases} \begin{bmatrix} \boldsymbol{\mathcal{X}}_0^{j-1} \\ 0 \end{bmatrix}, & i = 0 \\ \begin{bmatrix} \boldsymbol{\mathcal{X}}_i^{j-1} \\ -\frac{1}{\sqrt{2w_j}} \end{bmatrix}, & i = 1, \dots, j \\ \begin{bmatrix} \mathbf{0}_j \\ \frac{1}{\sqrt{2w_j}} \end{bmatrix}, & i = j + 1 \end{cases},$$

where $\mathbf{0}_j$ is a vector of j zeros. The points form a simplex (a generalization of the triangle to arbitrary dimensions) centered about the origin, with an additional point located at the origin ($\boldsymbol{\mathcal{X}}_0$).

The sigma points may then be transformed to those for mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ using

$$\boldsymbol{\mathcal{X}}_i' = \bar{\mathbf{x}} + \sqrt{\mathbf{P}_{xx}} \boldsymbol{\mathcal{X}}_i,$$

where $\sqrt{\mathbf{P}_{xx}}$ is the matrix square root of $\mathbf{P}_{xx}$. The transformed sigma points $\boldsymbol{\mathcal{Y}}$ and its statistics are computed as in the basic unscented transform, using $\boldsymbol{\mathcal{X}}_i'$.

Spherical Simplex Unscented Transform

The problem identified with the skew simplex set of sigma points is that the weights vary by a factor of 2^{n_x} and the point coordinates vary by a factor of $2^{n_x/2}$, so with large values of n_x, numerical problems may arise (Julier, 2003). The spherical simplex points still use only $n_x + 2$ points, but overcome this issue, placing the sigma points on a hypersphere centered at the origin, with the 0th sigma point located at the origin. These points are constructed in a similar fashion to the minimal skew sigma points as follows (Julier, 2003).

First, the weight of the 0th sigma point is selected freely as

$$w_0 \in [0, 1].$$

The remaining weights are computed using

$$w_i = \frac{1 - w_0}{n + 1}.$$

The sigma points for dimensional space $j = 1$ are initialized again as

$$\boldsymbol{\mathcal{X}}_0^1 = 0$$
$$\boldsymbol{\mathcal{X}}_1^1 = -\frac{1}{\sqrt{2w_1}}$$
$$\boldsymbol{\mathcal{X}}_2^1 = \frac{1}{\sqrt{2w_1}}.$$

Expanding up to higher dimensions $j = 2, \ldots, n_x$, the higher-dimensional sigma points are recursively defined as

$$\boldsymbol{\mathcal{X}}_i^j = \begin{cases} \begin{bmatrix} \boldsymbol{\mathcal{X}}_0^{j-1} \\ 0 \end{bmatrix}, & i = 0 \\ \begin{bmatrix} \boldsymbol{\mathcal{X}}_i^{j-1} \\ -\frac{1}{\sqrt{j(j+1)w_1}} \end{bmatrix}, & i = 1, \ldots, j \\ \begin{bmatrix} \mathbf{0}_j \\ \frac{j}{\sqrt{j(j+1)w_1}} \end{bmatrix}, & i = j+1 \end{cases},$$

where $\mathbf{0}_j$ is a vector of j zeros. These points may then be transformed for mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ as before.

Fig. 4 compares the location and weights for a two-dimensional random variable for the three different transforms. The mean of the random variable is 0, and the covariance is $\mathbf{I}$.

Improved Prediction Procedure

The improved prediction procedure uses Algorithm 2, only instead of the inputs being the particles and their weights, the inputs become the sigma points and their weights computed from the particle distribution at time k_P, with a suitable sigma point selection algorithm. Fig. 5 shows an example output of the prediction procedure for spring damage, using the full particle distribution ($N = 100$). Each particle creates a predicted trajectory, and determines a single EOL prediction. These individual predictions then form the complete prediction distribution.

The predicted EOLs based on simulating the sigma points for the different selection algorithms reviewed here are shown in Fig. 6. The free parameters were selected by hand in this particular example. The predicted EOL means and variances are shown in the figure. Recall that the aim is to approximate the full state-parameter distribution using a small set of samples, such that, when transformed to EOL, accurately predict the mean and variance of the EOL distribution computed using the full distribution. An under- or overapproximation of either statistic is undesirable, as it misrepresents the EOL corresponding to the current belief state. For this example, in each case, the mean EOL predicted with the sigma points matches the mean from the full distribution within 0.2% error. The variances are less accurate, with around 6% error for the symmetric and simplex sigma points, and around 16% error for the spherical sigma points. The error in predicted variance may be improved with better selection of the free parameters. An improvement in computation time of 18-24$\%$ was observed for the sigma point method. In this case $n_x = 5$, so the symmetric set has 11 sigma points and the simplex methods have 7 points, so there is also a very significant improvement in memory requirements for prediction.

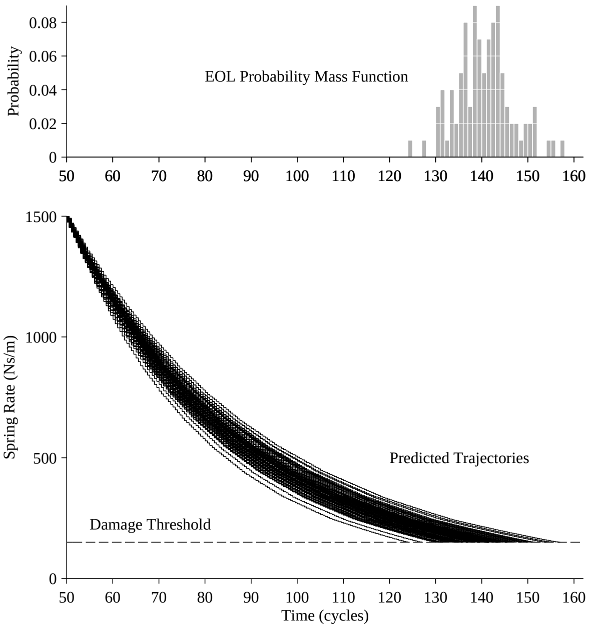

Figure 5: Predicted trajectories and EOL distribution for spring damage.

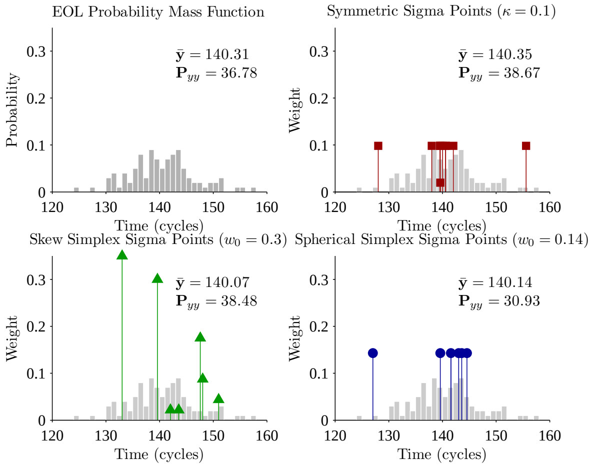

Figure 6: EOL predictions based on sigma points.

6. RESULTS

In this section, we evaluate the prognostics performance for the different prediction methods. In each case, we predict using the full particle distribution, the symmetric sigma points, the minimal skew simplex sigma points, and the spherical simplex sigma points, in order to compare the accuracy, precision, and computational cost of the prediction.

Estimation accuracy is evaluated using percentage root mean square error (PRMSE), which expresses relative estimation accuracy as a percentage:

$$\text{PRMSE} = 100\sqrt{\text{Mean}_k\left[\left(\frac{\hat{w}_k - w_k^*}{w_k^*}\right)^2\right]},$$

Table 1: Prognostics Performance Results for $N = 500$ and $M = \{x, i, open, closed\}$

Fault	PRMSE	$\overline{\text{RSD}}_w$	$\overline{\text{RA}}$				$\overline{\text{RSD}}_{RUL}$				$\overline{T}_{cpu}$			
			Full	Sym.	Skew	Sph.	Full	Sym.	Skew	Sph.	Max	Sym.	Skew	Sph.
r	3.72	26.09	94.39	94.49	93.90	94.48	19.89	18.40	21.35	18.48	72.98	57.39	53.91	60.21
k	3.90	14.74	96.19	96.20	96.12	96.17	15.21	14.27	15.35	14.42	61.46	58.05	55.85	58.06
L_c	5.01	18.01	93.62	93.85	93.82	93.77	22.65	20.18	22.35	21.63	77.22	63.32	61.38	61.41
L_o	3.05	17.89	95.48	95.59	95.12	95.36	16.95	16.20	19.30	18.61	67.93	56.70	52.69	53.56

where $\hat{w}_k$ denotes the estimated wear parameter value at time k, w_k^* denotes the true wear parameter value at k, and Mean$_k$ denotes the mean over all values of k. Estimation spread is calculated using relative standard deviation (RSD), computed for the wear parameter distribution at each prediction point (every 10 cycles), and averaged over all prediction points. The average is denoted as $\overline{\text{RSD}}$. In computing both PRMSE and RSD, we ignore the initial time period associated with estimation convergence. Convergence of the wear parameter estimate, C_w, is computed based on the definition of the convergence metric described in (Saxena et al., 2008), where the convergence of a curve is expressed as the distance from the origin to the centroid under the curve (a shorter distance is better). We use the absolute error of the hidden parameter estimate as the curve.

For a given prediction point k_P, we compute measures of accuracy and spread. For accuracy, we use the relative accuracy (RA) metric (Saxena, Celaya, Saha, Saha, & Goebel, 2009):

$$\text{RA}_{k_P} = 100\left(1 - \frac{|RUL_{k_P}^* - \text{Mean}_i(RUL_{k_P}^i)|}{RUL_{k_P}^*}\right).$$

We calculate prediction spread using RSD, which we denote as RSD$_{RUL}$. Both RA and RSD are averaged over all prediction points starting from the prediction at which a prognostics horizon (RA within a specified bound) is first reached (denoted using $\overline{\text{RA}}$ and $\overline{\text{RSD}}_{RUL}$).

In order to measure the computational performance, at each prediction point we measure the time taken for the prediction to be completed, $t_{cpu}(k_P)$. For a given prediction method, we then compute the percent improvement over the time for the full distribution, $t_{cpu}^{full}(k_P)$, defined as

$$T_{cpu} = 100\frac{|t_{cpu}^{full}(k_P) - t_{cpu}(k_P)|}{t_{cpu}^{full}(k_P)}.$$

This metric is then averaged over all k_P, denoted as $\overline{T}_{cpu}$, to summarize percent improvement over the entire experiment. We characterize the maximum possible performance increase by computing $\overline{T}_{cpu}$ for the prediction using a single point representing the mean of the state-parameter distribution. This performance can be achieved with the sigma point method by selecting a small enough value of κ or w_0 such that all the sigma points are concentrated on the mean, however, this would result in a vast underapproximation to the variance.

We consider the case where only a single damage mode is actively progressing. Table 1 shows the performance for each fault for $N = 500$, and taking the complete measurement set M. The random walk variances were chosen as fixed values assuming that the orders of magnitude of the wear parameters were known. Overall, the unknown wear parameter can be estimated well. The desired outcome is that the computed RA and RSD$_{RUL}$ using the sigma point methods closely approximate those produced using the full distribution. In the case of RA, the sigma point methods are within 0.5% of the RA calculated using the full distribution, meaning that the means of the distribution (from which RA is calculated) are predicted well. Larger differences are observed when comparing RSD$_{RUL}$, as in most cases the sigma point methods underapproximate the variance, with a worst-case error of 6-14%. The accuracy of variance prediction depends on the selected values of the free parameters of the unscented transforms, as correctly selected values will lead to better approximations. In this case, we selected the suggested value of κ for the symmetric sigma points, and this seemed to work well in all cases. For the minimal skew simplex sigma points, we chose $w_0 = -1$ for r, L_c, and L_o, and $w_0 = 0.1$ for k. For the spherical simplex sigma points, we chose $w_0 = 0.1$ for k, L_c, and L_o, and $w_0 = -1$ for r. These values were selected manually.

Over 50% improvement in computation time was observed in all cases, coming within 75-90% of the maximum possible improvement. Further, only a fraction of the samples are used in the sigma point methods, saving significantly on memory. At the selected prediction points, the valve is in a closed state, and the effective n_x is only 3 (i.e., only 3 of the states have different values between particles). Therefore, for the symmetric unscented transform only $2n_x + 1 = 7$ sigma points are required, and only $n_x + 2 = 5$ are required for the minimal skew simplex and spherical simplex sigma points. For $N = 500$ this is an improvement of over 98% in memory usage.

To explore further, we focus on the case of spring damage, and vary the number of particles and the measurement set. Table 2 shows the estimation results. Overall, the unknown wear parameter can be estimated well with both $N = 100$ and $N = 500$. This is also true when the measurement set is varied, in fact, using only the open and closed indicators, prognostics can still be performed, but at the cost of a wider variance in the prediction and slower convergence. With more particles, PRMSE improves and RSD generally increases slightly. Convergence is somewhat better with fewer particles as the filter tends to be more aggressive, whereas additional particles smooth the behavior. Of course, with too few particles, convergence may not occur, therefore a reasonably large N must usually be chosen.

Table 3 shows the prediction performance. RA is

Table 3: Comparison of Prognostics Prediction Methods for Spring Damage

M	N	$\overline{\text{RA}}$				$\overline{\text{RSD}}_{RUL}$				$\overline{T}_{cpu}$			
		Full	Sym.	Skew	Sph.	Full	Sym.	Skew	Sph.	Max	Sym.	Skew	Sph.
$\{x, i, open, closed\}$	100	94.59	94.64	94.59	94.56	15.01	13.88	14.96	14.03	48.08	42.98	23.81	29.98
$\{x, i, open, closed\}$	500	96.19	96.20	96.12	96.17	15.21	14.27	15.35	14.42	61.46	58.05	55.85	58.06
$\{x, i\}$	100	93.96	94.00	93.88	94.00	15.53	14.73	15.81	14.90	45.49	38.98	22.49	26.02
$\{x, i\}$	500	97.16	97.18	97.16	97.22	15.39	14.30	15.46	14.48	62.39	58.97	56.40	58.81
$\{x\}$	100	93.68	93.82	93.65	93.74	18.17	16.58	17.83	16.79	50.89	42.83	27.79	31.37
$\{x\}$	500	94.85	95.02	94.72	94.99	18.25	16.68	18.13	16.93	67.39	62.92	60.81	63.09
$\{i\}$	100	93.34	93.34	93.13	93.31	16.65	15.51	17.23	16.22	47.01	37.62	22.19	25.65
$\{i\}$	500	94.61	94.77	94.49	94.77	18.32	16.83	18.34	17.14	65.25	60.80	57.84	60.52
$\{open, closed\}$	100	94.46	94.74	94.28	94.65	22.84	20.13	22.83	21.48	48.95	38.53	16.73	36.78
$\{open, closed\}$	500	94.20	94.58	94.14	94.40	23.62	20.50	23.25	21.55	74.61	69.16	66.20	68.29

Table 2: Damage Estimation Performance for Spring Damage

M	N	PRMSE	$\overline{\text{RSD}}_w$	C_w
$\{x, i, open, closed\}$	100	4.62	14.81	32.90
$\{x, i, open, closed\}$	500	3.90	14.74	32.31
$\{x, i\}$	100	5.83	14.58	33.79
$\{x, i\}$	500	3.87	14.93	34.37
$\{x\}$	100	3.99	16.44	38.13
$\{x\}$	500	3.10	16.83	35.39
$\{i\}$	100	5.16	15.74	40.72
$\{i\}$	500	4.61	16.45	34.90
$\{open, closed\}$	100	3.47	21.47	42.55
$\{open, closed\}$	500	3.00	21.89	42.35

estimated within similar error bounds as in Table 1. Again, the sigma point methods usually underapproximate RSD$_{RUL}$. With fewer particles, the gain in computational efficiency is smaller, as expected, but gains of 20-40% are still observed with $N = 100$, coming within 30-90% of the maximum possible increase, and the memory usage improves by at least 93% (i.e., 100 particles compared to at most 7 sigma points). Notice also that for the cases where RSD$_{RUL}$ is larger, such as with $M = \{open, closed\}$, the savings are even greater, i.e., the wider the full particle distribution, the more of a savings the sigma point methods can offer.

Overall, these results demonstrate that prediction can be achieved much more efficiently with limited deviations in prediction performance. The symmetric sigma points seemed to provide the largest improvement in time efficiency, but underapproximated the variance the most. These two effects are interrelated. The smallest values of the wear parameter in the full distribution contribute most to the time cost, so sigma points concentrated more towards the mean take less time to simulate. The minimal skew simplex sigma points came closest to the variance of the true distribution, but usually with the smallest time improvement.

The biggest practical difficulty in applying this method is the selection of the free parameters, as this relates to performance. Using the symmetric sigma points with the suggested value of κ seemed to always work well, requiring no further tuning. There is no heuristic available in the literature for the minimal skew simplex and spherical simplex sigma points. In order to examine the sensitivity of the selected value of the free parameter on performance, we varied the value of w_0 for the minimal skew simplex sigma points for the case of spring damage with $N = 500$ and $M = \{x, i, open, closed\}$. According to Table 1, the full distribution achieves $\overline{\text{RA}} = 96.19\%$ and $\overline{\text{RSD}}_{RUL} = 15.21$ cycles. Fig. 7 illustrates how these metrics vary over the selected range of w_0. In general, a large value of w_0 will spread out the sigma points, and therefore increase the predicted variance. For $w_0 \in [-1.0, 0.5]$, RA and RSD have little deviation from the desired values produced by the full distribution, so any value within this range will result in an acceptable approximation. But, for $w_0 \in (0.5, 0.9]$, the approximated RSD begins to increase significantly, and RA decreases with it at a much smaller rate.

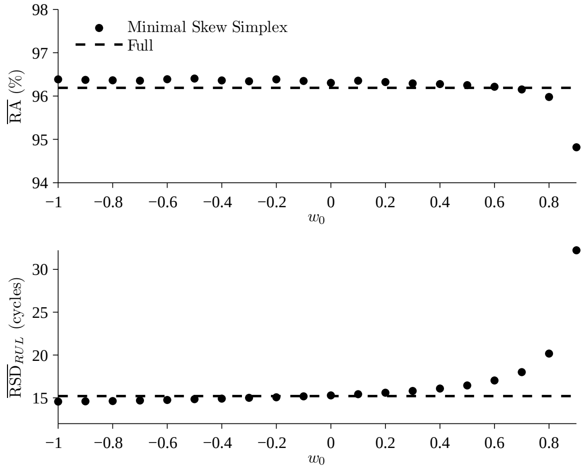

Figure 7: Prognostics performance for different values of w_0 for the minimal skew simplex sigma points, for k with $N = 500$ and $M = \{x, i, open, closed\}$.

7. CONCLUSIONS

In this paper, we developed a computationally efficient prediction scheme for model-based prognostics based on the unscented transform. The unscented transform allows the statistics of a distribution passed through a nonlinear transformation to be predicted using a minimal set of deterministically selected samples. Applying this to the prognostics problem, we are able to predict the mean and variance of the EOL accurately, and with improved computational efficiency and significantly reduced memory costs.

Particle filtering approaches have become a popular choice for model-based prognostics (e.g., (Saha & Goebel, 2009; Abbas, Ferri, Orchard, & Vachtsevanos, 2007)). The most significant disadvantage is the computational complexity, as usually a large number of particles are needed for accurate estimation, and, subsequently, prediction. A related approach to efficient prediction is described in (Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008), however, in this approach, random sampling with a smaller number of particles is advocated. As described in Section 5., a large number of randomly selected particles are needed to correctly approximate the statistics of the prediction based on the full particle set, so a significant number of particles would still be needed. However, the method described in this paper selects only the minimal number of points necessary to capture those statistics. This number is dependent only on the dimension of the state space. This approach is also applicable when a technique other than particle filters is used for the estimation task, as long as the method provides a state distribution which is to be propagated forward to EOL. Note that if the unscented Kalman filter is used, the sigma points are already available for prediction.

The unscented transform is not limited to Gaussian distributions, but the prediction method based on it is useful only when the mean and variance of the EOL distribution are meaningful statistics. For example, for a multi-modal distribution, a single mean and variance are not meaningful. This could be the case when multiple future input trajectories are considered. In this case, each mode is associated with one of these trajectories, and each may be defined by a mean and variance. Therefore, the method could be applied to each case individually to obtain the means and variances of the different modes.

As part of future work, it is important to determine strategies for selecting the free parameters of the different unscented transforms, as this is the main hurdle to practical implementation. As discussed in Section 6., the suggested heuristic for selecting κ for the symmetric sigma points worked well. For the remaining methods, selection of w_0 may be difficult, although a value between -1 and 0.1 worked well here. Further, scaling of the sigma points can also be performed, which introduces additional free parameters (Julier, 2002). Bringing the sigma points closer together will speed up computation, but it should be ensured that the variance estimate remains accurate. A detailed analysis over this parameter space is necessary to suggest useful heuristics in the context of prognostics. One may then envision automatic methods to tune the free parameters to achieve the desired spread in sigma points to correctly approximate EOL mean and variance. Extensions of the unscented transform to prediction of higher-order moments such as skew and kurtosis may also be useful for prognostics.

ACKNOWLEDGMENTS

The funding for this work was provided by the NASA Fault Detection, Isolation, and Recovery (FDIR) project under the Exploration Technology and Development Program (ETDP) of the Exploration Systems Mission Directorate (ESMD).

REFERENCES

Abbas, M., Ferri, A. A., Orchard, M. E., & Vachtsevanos, G. J. (2007). An intelligent diagnostic/prognostic framework for automotive electrical systems. In *2007 IEEE Intelligent Vehicles Symposium* (pp. 352–357).

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Byington, C. S., Watson, M., Edwards, D., & Stoelting, P. (2004, March). A model-based approach to prognostics and health management for flight control actuators. In *Proceedings of the 2004 IEEE Aerospace Conference* (Vol. 6, pp. 3551–3562).

Cappe, O., Godsill, S. J., & Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, *95*(5), 899.

Daigle, M., & Goebel, K. (2009, September). Model-based prognostics with fixed-lag particle filters. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009.*

Daigle, M., & Goebel, K. (2010, March). Model-based prognostics under limited sensing. In *Proceedings of the 2010 IEEE Aerospace Conference.*

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*, 197–208.

Hutchings, I. M. (1992). *Tribology: friction and wear of engineering materials*. CRC Press.

Julier, S. J. (2002, November). The scaled unscented transformation. In *Proceedings of the 2002 American Control Conference* (Vol. 6, pp. 4555–4559).

Julier, S. J. (2003, June). The spherical simplex unscented transformation. In *Proceedings of the 2003 American Control Conference* (Vol. 3, p. 2430-2434).

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls* (pp. 182–193).

Julier, S. J., & Uhlmann, J. K. (2002, November). Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In *Proceedings of the 2002 American Control Conference* (Vol. 2, pp. 887–892).

Julier, S. J., & Uhlmann, J. K. (2004, March). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, *5*(1), 1–25.

Lyshevski, S. E., Sinha, A. S. C., & Seger, J. P. (1999, June). Modeling and control of turbocharged diesels for medium and heavy vehicles. In *Proceedings of the American Control Conference.*

Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008, October). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *Proceedings of International Conference on Prognostics and Health Management.*

Rahman, M. F., Cheung, N. C., & Lim, K. W. (1996, September). Modelling of a nonlinear solenoid towards the development of a proportional actuator. In *Proceedings of the 5th International Conference on Modelling and Simulation of Electrical Machines, Convertors, and Systems, ELECTRIMACS* (p. 121-128).

Roemer, M., Byington, C., Kacprzynski, G., & Vachtsevanos, G. (2005). An overview of selected prognostic technologies with reference to an integrated PHM architecture. In *Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace.*

Saha, B., & Goebel, K. (2009, September). 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.*

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., et al. (2008, Oct). Metrics for evaluating performance of prognostic techniques. In *International Conference on Prognostics and Health Management 2008.*

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009, September). On Applying the Prognostic Performance Metrics. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009.*

Szente, V., & Vad, J. (2001). Computational and experimental investigation on solenoid valve dynamics. In *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (Vol. 1, p. 618 -623).

Tansel, I. N., Perotti, J. M., Yenilmez, A., & Chen, P. (2005). Valve health monitoring with wavelet transformation and neural networks (WT-NN). In *2005 ICSC Congress on Computational Intelligence Methods and Applications.*