A Model-Based Prognostics Approach Applied to Pneumatic Valves*

Matthew J. 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

Within the area of systems health management, the task of prognostics centers on predicting when components will fail. Model-based prognostics exploits domain knowledge of the system, its components, and how they fail by casting the underlying physical phenomena in a physics-based model that is derived from first principles. Uncertainty cannot be avoided in prediction, therefore, algorithms are employed that help in managing these uncertainties. The particle filtering algorithm has become a popular choice for model-based prognostics due to its wide applicability, ease of implementation, and support for uncertainty management. We develop a general model-based prognostics methodology within a robust probabilistic framework using particle filters. As a case study, we consider a pneumatic valve from the Space Shuttle cryogenic refueling system. We develop a detailed physics-based model of the pneumatic valve, and perform comprehensive simulation experiments to illustrate our prognostics approach and evaluate its effectiveness and robustness. The approach is demonstrated using historical pneumatic valve data from the refueling system.

1 INTRODUCTION

Prognostics is concerned with determining the health of system components and making *end of life* (EOL) and *remaining useful life* (RUL) predictions. It is a key enabling technology for condition-based maintenance, and serves to increase system availability, reliability, and safety by enabling timely maintenance decisions to be made. As with diagnostics, prognostics methods are typically categorized as either model-based or data-driven. Data-driven approaches, rather than taking advantage of system and domain knowledge, instead utilize large amounts of run-to-failure data that are used to train machine learning algorithms to identify trends and determine EOL and RUL (Schwabacher, 2005). However, such data is often difficult to acquire. In contrast, model-based approaches exploit domain knowledge of the system, its components, and how they fail, in the form of models, in order to provide EOL and RUL predictions (Roemer, Byington, Kacprzynski, & Vachtsevanos, 2005; Byington, Watson, Edwards, & Stoelting, 2004; Saha & Goebel, 2009; Daigle & Goebel, 2010b; Luo, Pattipati, Qiao, & Chigusa, 2008). The underlying physical phenomena are captured in a physics-based model that is derived from first principles, therefore, model-based approaches can provide EOL and RUL estimates that are much more accurate and precise than data-driven approaches, if the models are correct.

In this paper, we develop a general model-based prognostics framework using particle filters, based on preliminary work presented in (Daigle & Goebel, 2009, 2010b). Particle filters are nonlinear state observers that approximate the posterior state distribution as a set of discrete, weighted samples. They have become a popular methodology in the context of prognostics, where they are used for joint state-parameter estimation, e.g., (Saha & Goebel, 2009; Orchard, 2007; Daigle & Goebel, 2009, 2010b). Although suboptimal, the advantage of particle filters is that they can be applied to systems which may be nonlinear and have non-Gaussian noise terms, where optimal solutions are unavailable or intractable. Further, because they are based on probability distributions, they help in managing the uncertainty that may arise from a number of sources in prognostics.

As a case study, we consider a pneumatic valve from the Space Shuttle cryogenic refueling system. We construct a detailed physics-based model of a pneumatic valve that in-

*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).

cludes models of different damage mechanisms. We run a comprehensive set of prognostics experiments in simulation to demonstrate the approach and establish that prognostics may be performed for pneumatic valves using only discrete position sensors. Further, we demonstrate the approach using historical pneumatic valve data from the refueling system.

The paper is organized as follows. Section 2 formally defines the prognostics problem and describes the computational architecture. Section 3 presents the modeling methodology and develops the model of the pneumatic valve. Section 4 discusses the damage estimation approach using particle filters, and Section 5 provides the prediction algorithm. Section 6 presents simulation results and the demonstration of the approach on real data. Section 7 discusses related work. Section 8 concludes the paper.

2 PROGNOSTICS APPROACH

The problem of prognostics is predicting 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

We assume the system may be described by

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

where $t \in \mathbb{R}$ is the continuous time variable, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ 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. This representation considers a general nonlinear model with no restrictions on the functional forms of $\mathbf{f}$ or $\mathbf{h}$. Further, the noise terms may be coupled in a nonlinear way with the states and parameters. The parameters $\boldsymbol{\theta}(t)$ evolve in an unknown way. In practice, they are typically considered to be constant, but may in fact be time-varying.

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 one of a set of functional requirements (e.g., a valve does not open in the required amount of time). These requirements may be expressed using a threshold, beyond which we say the component has failed. 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 the system has failed, where $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if a requirement is violated, and 0 otherwise. Using this function, we can formally define EOL with

$$EOL(t_P) \triangleq \inf\{t \in \mathbb{R} : t \geq t_P \wedge T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1\},$$

and RUL with

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

In practice, many sources of uncertainty exist that unfold into the prediction. Uncertainty in the initial state, uncertainty in the model, process noise (i.e., $\mathbf{v}(t)$), and sensor noise (i.e., $\mathbf{n}(t)$) result in an uncertain estimate of $(\mathbf{x}(t), \boldsymbol{\theta}(t))$. In predicting from this uncertain state, modeling errors, process noise, and uncertainty in the future inputs of the system further add to the overall uncertainty. At best, then, we can only 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, $p(EOL(t_p)|\mathbf{y}_{0:t_P})$ and/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 faults evolve in time. These models depend on unknown and possibly time-varying parameters. 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.

We employ the prognostics architecture in Fig. 1. The system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. Prognostics may begin at $k = 0$, with the damage estimation module determining 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 the joint state-parameter distribution, along with hypothesized future inputs, to compute EOL and RUL as probability distributions $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$ at given prediction times k_P. In parallel, a fault detection, isolation, and identification (FDII) module may be used to determine which damage mechanisms are active, represented as a fault set $\mathbf{F}$. The damage estimation module may then use this result to limit the space of parameters that must be estimated. Alternatively, prognostics may begin only when diagnostics has completed. In this paper, we assume prognostics is not aided by FDII.

3 MODELING METHODOLOGY

To implement a model-based prognostics approach, detailed physics-based models of the component are necessary. Such models must not only describe the nominal behavior of the component, but also the faulty behavior. Further, they must describe how faults, or damage, progress in time. It is with these models that prediction may be performed.

To construct models in this paradigm, the first step is to develop a nominal model based on a first principles, physical understanding of the system. The next step is to identify the variables or parameters that are representative of faults, which we call damage variables, $\mathbf{d}(t)$, e.g., a friction parameter, or the size of a leak. Since these variables change dynamically,

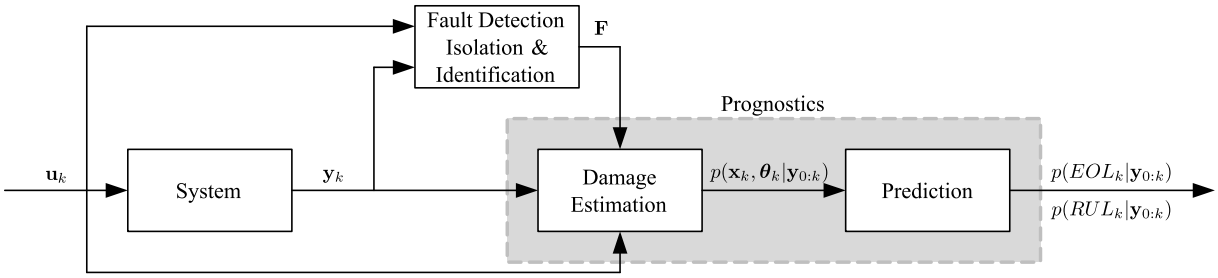


Figure 1. Prognostics architecture

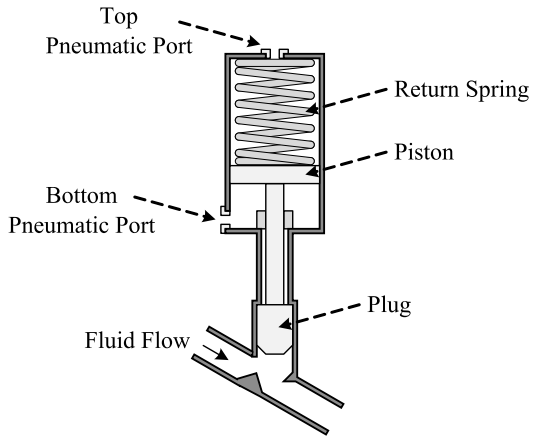


Figure 2. Pneumatic valve

they form part of the state vector, i.e., $\mathbf{d}(t) \subseteq \mathbf{x}(t)$. The final step is to develop models for how these variables change in time, i.e., how the damage progresses or evolves, which we call *damage progression functions*. These models are typically dependent on other states of the system, and, therefore, dependent on the system inputs. They augment the state equation $\mathbf{f}$. These functions are parameterized by unknown parameters, which we call *wear parameters*, $\mathbf{w}(t) \subseteq \boldsymbol{\theta}(t)$.

In the remainder of this section, we apply this modeling framework to a pneumatic valve, which serves as the case study for this paper.

3.1 Pneumatic Valve Modeling

Pneumatic valves are gas-actuated valves used in many domains. A normally-closed valve with a linear cylinder actuator is illustrated in Fig. 2. The valve is opened by filling the chamber below the piston with gas up to the supply pressure, and evacuating the chamber above the piston down to atmospheric pressure. The valve is closed by filling the chamber above the piston, and evacuating the chamber below the piston. The return spring ensures that when pressure is lost, the valve will close due to the force exerted by the return spring, hence it is a normally-closed valve.

We develop a physics model of the valve based on mass and energy balances. The system state for the nominal model includes the position of the valve, $x(t)$, the velocity of the valve, $v(t)$, the mass of the gas in the volume above the piston, $m_t(t)$, and the mass of the gas in the volume below the piston, $m_b(t)$:

$$\mathbf{x}(t) = \begin{bmatrix} x(t) & v(t) & m_t(t) & m_b(t) \end{bmatrix}^T.$$

The position is defined as $x = 0$ when the valve is fully closed. The stroke length of the valve is denoted by L_s; when the valve is fully open its position $x = L_s$.

The derivatives of the states are described by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} v(t) & a(t) & f_t(t) & f_b(t) \end{bmatrix}^T,$$

where $a(t)$ is the valve acceleration, and $f_t(t)$ and $f_b(t)$ are the mass flows going into the top and bottom pneumatic ports, respectively.

The inputs are considered to be

$$\mathbf{u}(t) = \begin{bmatrix} p_l(t) & p_r(t) & u_t(t) & u_b(t) \end{bmatrix}^T,$$

where $p_l(t)$ and $p_r(t)$ are the fluid pressures on the left and right side of the plug, respectively, and $u_t(t)$ and $u_b(t)$ are the input pressures to the top and bottom pneumatic ports. These pressures will alternate between the supply pressure and atmospheric pressure depending on the commanded valve position.

The acceleration is defined by the combined mass of the piston and plug, m, and the sum of forces acting on the valve, which includes the forces from the pneumatic gas, $(p_b(t) - p_t(t))A_p$, where $p_b(t)$ and $p_t(t)$ are the gas pressures on the bottom and the top of the piston, respectively, and A_p is the surface area of the piston; the forces from the fluid flowing through the valve, $(p_r(t) - p_l(t))A_v$, where A_v is the area of the valve contacting the fluid; the weight of the moving parts of the valve, $-mg$, where g is the acceleration due to gravity; the spring force, $-k(x(t) - x_o)$, where k is the spring constant and x_o is the amount of spring compression when the valve is closed; friction, $-rv(t)$, where r is the coefficient of kinetic friction, and the contact forces $F_c(t)$ at the boundaries of the valve motion,

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

where k_c is the (large) spring constant associated with the flexible seals. Overall, the acceleration term is defined by

$$a(t) = \frac{1}{m}\Big[(p_b(t) - p_t(t))A_p + (p_r(t) - p_l(t))A_v - mg - k(x(t) - x_o) - rv(t) + F_c(t)\Big].$$

The pressures $p_t(t)$ and $p_b(t)$ are calculated as:

$$p_t(t) = \frac{m_t(t)R_gT}{V_{t_0} + A_p(L_s - x(t))}$$
$$p_b(t) = \frac{m_b(t)R_gT}{V_{b_0} + A_p x(t)},$$

where we assume an isothermal process in which the (ideal) gas temperature is constant at T, R_g is the gas constant for the pneumatic gas, and V_{t_0} and V_{b_0} are the minimum gas volumes for the gas chambers above and below the piston, respectively. The gas flows are given by:

$$f_t(t) = f_g(p_t(t), u_t(t))$$
$$f_b(t) = f_g(p_b(t), u_b(t)),$$

where f_g defines gas flow through an orifice for choked and non-choked flow conditions (Perry & Green, 2007). Non-choked flow for $p_1 \geq p_2$ is given by $f_{g,nc}(p_1, p_2) =$

$$C_s A_s p_1 \sqrt{\frac{\gamma}{ZR_gT}\left(\frac{2}{\gamma - 1}\right)\left(\left(\frac{p_2}{p_1}\right)^{\frac{2}{\gamma}} - \left(\frac{p_2}{p_1}\right)^{\frac{\gamma+1}{\gamma}}\right)},$$

where γ is the ratio of specific heats, Z is the gas compressibility factor, C_s is the flow coefficient, and A_s is the orifice area. Choked flow for $p_1 \geq p_2$ is given by

$$f_{g,c}(p_1, p_2) = C_s A_s p_1 \sqrt{\frac{\gamma}{ZR_gT}\left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma+1}{\gamma-1}}}.$$

Choked flow occurs when the upstream to downstream pressure ratio exceeds $\left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)}$. The overall gas flow equation is then given by

$$f_g(p_1, p_2) = \begin{cases} f_{g,nc}(p_1, p_2) & \text{if } p_1 \geq p_2 \\ & \text{and } \frac{p_1}{p_2} < \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ f_{g,c}(p_1, p_2) & \text{if } p_1 \geq p_2 \\ & \text{and } \frac{p_1}{p_2} \geq \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ -f_{g,nc}(p_2, p_1) & \text{if } p_2 > p_1 \\ & \text{and } \frac{p_2}{p_1} < \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \\ -f_{g,c}(p_2, p_1) & \text{if } p_2 > p_1 \\ & \text{and } \frac{p_2}{p_1} \geq \left(\frac{\gamma+1}{2}\right)^{\frac{\gamma}{(\gamma-1)}}, \end{cases}.$$

We select our complete measurement vector as

$$\mathbf{y}(t) = \begin{bmatrix} x(t) & p_t(t) & p_b(t) & f_v(t) & open(t) & closed(t) \end{bmatrix}^T.$$

Here, f_v is the fluid flow through the valve, given by

$$f_v(t) = \frac{x(t)}{L_s} C_v A_v \sqrt{\tfrac{2}{\rho}\,|p_{fl} - p_{fr}|}\,\text{sign}(p_{fl} - p_{fr}),$$

where C_v is the (dimensionless) flow coefficient of the valve, ρ is the liquid density, and we assume a linear flow characteristic for the valve. The $open(t)$ and $closed(t)$ signals are from discrete sensors that output 1 if the valve is in the fully open or fully closed state:

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

Fig. 3 shows a nominal valve cycle. The valve is commanded to open at 0 s. The top pneumatic port opens to atmosphere and the bottom port opens to the supply pressure (approximately 5.3 MPa, or 750 psig). When the force on the underside of the piston is large enough to overcome the return spring, friction, and the gas force on the top of the piston, the valve begins to move upward as the pneumatic gas continues to flow into and out of the valve actuator. At about 8 s, the valve is completely open. The valve is commanded to close at 15 s. The bottom pneumatic port opens to atmosphere and the top port opens to the supply pressure. When the force balance becomes negative, the valve starts to move downward, and completely closes at around 20 s. The valve closes faster than it opens due to the return spring.

3.2 Damage Modeling

With the nominal model defined, we may now determine which damage mechanisms are relevant, and what model parameters change as a result of the damage. From valve documentation and historical maintenance records, we have identified a set of faults that, although not exhaustive, contains the most important and most often observed faults that affect valve functionality and lead to EOL. The set of faults includes friction damage, spring damage, internal valve leaks, and external valve leaks. The functional requirements on the valve that define T_{EOL} are that it opens and closes within given timing limits, and that it fully closes upon loss of actuating pressure. In this example, we use 15 s for both the opening and closing time limits.

One damage mechanism present in valves is sliding wear. The equation for sliding wear takes on the following form (Hutchings, 1992):

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

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. Friction will increase linearly with sliding wear, because the

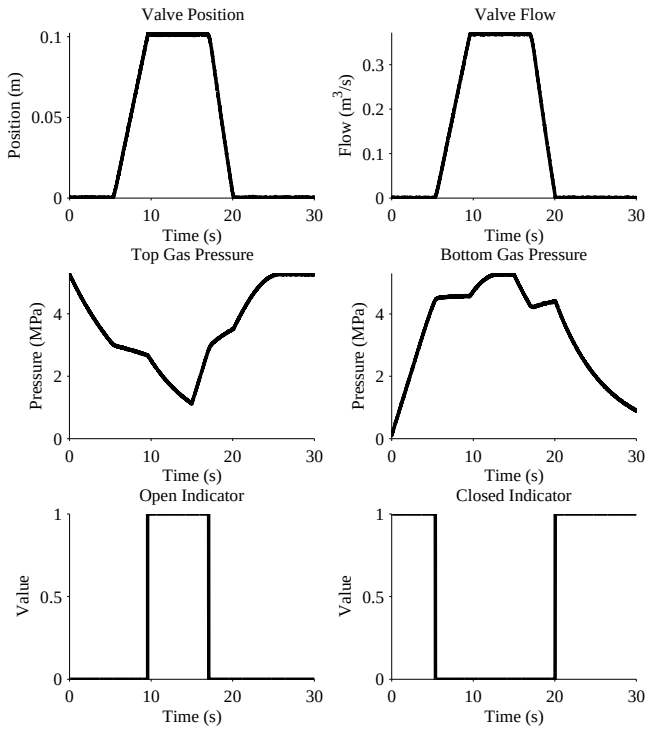


Figure 3. Nominal valve operation

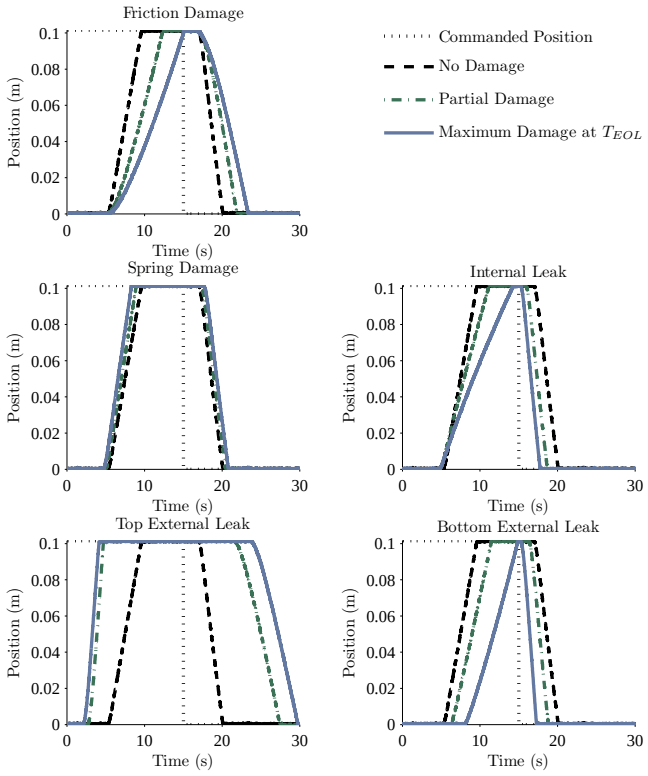


Figure 4. Valve operation with damage

contact area between the sliding bodies becomes greater as surface asperities wear down (Hutchings, 1992). Lubrication between the sliding bodies can also degrade over time. We therefore characterize friction damage as change in the friction coefficient, and model the damage progression in a form similar to sliding wear:

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

where w_r is the wear coefficient, and $F_f(t)$ is the friction force defined in the previous subsection. The friction coefficient only grows when the valve is moving, as only then can sliding wear occur. Therefore, the friction coefficient evolves in a step-wise fashion, with damage only growing during the opening and closing motions. As the friction coefficient increases, the friction force increases, further increasing the rate at which the friction coefficient grows. This results in a damage progression that is similar to an exponential when viewed at large time scales.

Fig. 4 shows the effect of an increase in friction on the valve cycle. We define r^+ as the maximum value of the friction parameter at which the valve still behaves within the timing limits. Above this value, the friction force becomes large enough that the valve cannot open within the 15 s limit, as shown in Fig. 4. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if $r(t) > r^+$.

We assume a similar equation form for spring damage:

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

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. Like with friction damage, the spring constant decreases only 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.

Fig. 4 shows the effect of a decrease in the spring constant on the valve cycle. In normal operation, without the spring tending the valve to close, the valve will open faster and close slower. The spring can actually fail completely, and the valve would still open and close in time, however, the spring must be strong enough to close the valve against system pressure when the actuating pressure is lost. So, it is the loss of this function leads to T_{EOL} for spring damage, and we define k^- as the smallest value of k at which the valve will still fully close upon loss of supply pressure. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ also if $k(t) < k^-$.

An internal leak in the valve can appear at the seal surrounding the piston as a result of sliding wear. The pneumatic gas is then able to flow between the volumes above and below the piston, decreasing the response time of the valve. We parameterize this leak by its equivalent orifice area, $A_i(t)$, described by

$$\dot{A}_i(t) = w_i |F_f(t) v(t)|,$$

where w_i is the wear coefficient. The mass flow at the leak,

$f_i(t)$, is computed using the gas flow equation described earlier:

$$f_i(t) = f_g(p_t(t), p_b(t)),$$

where positive flow denotes flow from the top volume to the bottom volume. The term is subtracted from the $f_t(t)$ equation and added to the $f_b(t)$ equation.

As sliding wear occurs, the leak size increases. As with the friction and spring damage, the internal leak only grows when the valve is moving. Over large time scales, the internal leak appears to grow linearly with the valve cycles, since the friction force does not change much as the leak size grows.

The presence of an internal leak makes it more difficult to actuate the valve, because it causes gas to flow into the lower pressure volume that is being evacuated and out of the higher pressure volume that is being filled. We define A_i^+ as the maximum internal leak area at which the valve opens and closes within the functional limits. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ also if $A_i(t) > A_i^+$. Fig. 4 shows the effect of an internal leak on the valve cycle.

External leaks may also form, most likely at the actuator connections to the pneumatic gas supply, due to corrosion and other environmental factors. Without knowledge of how the leak size progresses, we assume the growth of the area of the leak holes, $A_{e,t}(t)$ and $A_{e,b}(t)$, is linear:

$$\dot{A}_{e,t}(t) = w_{e,t}$$
$$\dot{A}_{e,b}(t) = w_{e,b},$$

where the t and b subscripts denote a leak at the top and bottom pneumatic ports, respectively, and $w_{e,t}$ and $w_{e,b}$ are the wear coefficients. The leak flows are given by

$$f_{e,t}(t) = f_g(p_t(t), p_{atm}(t))$$
$$f_{e,b}(t) = f_g(p_b(t), p_{atm}(t)),$$

where p_{atm} is atmospheric pressure. The $f_{e,t}(t)$ term is subtracted from the $f_t(t)$ equation, and the $f_{e,b}(t)$ term is subtracted from the $f_b(t)$ equation.

Note that this damage progression is independent of the valve inputs. Fig. 4 shows the effects of top and bottom external leaks on the valve cycle. The effect of the formation of a leak at the top pneumatic port is that it becomes easier to open the valve but more difficult to close it. Conversely, the effect of a leak at the bottom pneumatic port is that it becomes more difficult to open but easier to close the valve. We define the maximum leak hole areas at which the valve still opens and closes within the functional limits as $A_{e,t}^+$ and $A_{e,b}^+$. So, $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ also if $A_{e,t}(t) > A_{e,t}^+$ or $A_{e,b}(t) > A_{e,b}^+$. Alternatively, maximum allowable leakage rates may define EOL.

So, the damage variables are given by

$$\mathbf{d}(t) = \begin{bmatrix} r(t) & k(t) & A_i(t) & A_{e,t}(t) & A_{e,b}(t) \end{bmatrix}^T,$$

and the complete state vector of the extended valve model becomes

$$\mathbf{x}(t) = \begin{bmatrix} x(t)v(t) \\ m_t(t) \\ m_b(t) \\ r(t) \\ k(t) \\ A_i(t) \\ A_{e,t}(t) \\ A_{e,b}(t) \end{bmatrix}.$$

The wear parameters form the unknown parameter vector, i.e.,

$$\mathbf{w}(t) = \boldsymbol{\theta}(t) = \begin{bmatrix} w_r(t) \\ w_k(t) \\ w_i(t) \\ w_{e,t}(t) \\ w_{e,b}(t) \end{bmatrix}.$$

4 DAMAGE ESTIMATION

In the model-based paradigm, damage estimation reduces to joint state-parameter estimation, i.e., computation of $p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k})$. The typical approach to state estimation is to use a state observer, which helps to compensate for the effects of process and sensor noise on the state estimate. Joint state-parameter estimation is traditionally performed using a state observer by augmenting the state vector with the parameter vector. In this way, both states and parameters are simultaneously estimated.

For linear systems with additive Gaussian noise terms, the Kalman filter is appropriate. For nonlinear systems with additive Gaussian noise terms, the extended Kalman filter or unscented Kalman filter (Julier & Uhlmann, 1997) may be used. However, for nonlinear systems with non-Gaussian noise terms, *particle filters* are best suited, and offer approximate (suboptimal) solutions to the state estimation problem for such 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, accuracy increases and the optimal solution is approached. In addition, particle filters are straightforward to implement, and computational complexity may be controlled by increasing or decreasing the number of particles, with respect to the desired estimation performance. Further, the discrete position sensors cannot be directly handled by other filtering algorithms. For these reasons, we select particle filters for our model-based prognostics framework.

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

$$\{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i), w_k^i\}_{i=1}^N,$$

where N denotes the number of particles, and for particle i, $\mathbf{x}_k^i$ denotes the state estimates, $\boldsymbol{\theta}_k^i$ denotes the parameter es-

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

timates, 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 i is propagated forward to time k by first sampling new parameter values. Here, the parameters $\boldsymbol{\theta}_k$ evolve by some unknown random process that is independent of the state $\mathbf{x}_k$. To perform parameter estimation within a particle filter framework, 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. After sampling parameter values from the selected distribution, new states are sampled by applying the state equation $\mathbf{f}$ to $(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k)$ with process noise $\mathbf{v}(t)$ sampled from its assumed distribution. The particle weight is assigned using $\mathbf{y}_k$. Specifically, the output equation $\mathbf{h}$ is applied to $(\mathbf{x}^i_{k+1}, \boldsymbol{\theta}^i_{k+1})$ and the likelihood of the corresponding output is computed using the assumed probability density function of the sensor noise. The weights are then normalized, followed by the resampling step.

Pseudocode for the resampling step is given as Algorithm 2. First, the cumulative distribution function (CDF) of the weights is computed as $\mathbf{c}$. A starting point is drawn from the uniform distribution. The algorithm then moves along the CDF and a new particle is resampled from the old particle set. The weights of the resampled particles are all assigned to be the same. The resampling step is necessary to avoid *degeneracy*, in which all but a few particles have negligible weight. If this occurs, then the state distribution is very poorly represented, and computation is wasted on particles that do not contribute to the approximation. Resampling based on the CDF avoids this problem by multiplying particles with high weight and dropping particles with low weight. Here, we resample at each time step, but the frequency of resampling may be reduced by computing the effective number of particles, and only resampling when this number falls below a given threshold (Arulampalam et al., 2002).

Algorithm 2 Systematic Resampling

Inputs: $\{(\mathbf{x}^i_k, \boldsymbol{\theta}^i_k), w^i_k\}^N_{i=1}$
Outputs: $\{(\mathbf{x}^j_k, \boldsymbol{\theta}^j_k), w^j_k\}^N_{j=1}$
$\mathbf{c}_1 \leftarrow 0$
for $i = 2$ **to** N **do**
  $\mathbf{c}_i \leftarrow \mathbf{c}_{i-1} + w^i_k$
end for
$i \leftarrow 1$
$u_1 \sim \mathcal{U}(0, 1/N)$
for $j = 1$ **to** N **do**
  $u \leftarrow u_1 + (j+1)/N$
  while $u > \mathbf{c}_i$ **do**
    $i \leftarrow i + 1$
  end while
  $\mathbf{x}^j_k \leftarrow \mathbf{x}^i_k$
  $\boldsymbol{\theta}^j_k \leftarrow \boldsymbol{\theta}^i_k$
  $w^j_k = 1/N$
end for

During the sampling step, particles are generated with parameter values that will be different from the initial guesses for the unknown parameters. The particles with parameter values closest to the true values should match the outputs better, and therefore be assigned higher weight. Resampling will cause more particles to be generated around these better values. As this process is repeated over time, the particle filter converges to the true values. The selected variance of the random walk noise must be large enough so as to allow convergence in a reasonable amount of time, but small enough such that when convergence is reached, the parameter can be tracked smoothly. Fig. 5 illustrates the effect of the variance value on estimation performance shown in an accuracy-precision plot. The horizontal axis provides estimation accuracy as computed using percent root mean square error (PRMSE) of the unknown wear parameter, $w_{e,b}$, and the vertical axis provides the variability of the wear parameter distribution as computed using relative median absolute deviation (RMAD) (performance metrics are defined in the Appendix). The optimal point is located at the origin. As the random walk variance of $w_{e,b}$ is decreased, both PRMSE and RMAD become smaller, improving in both performance dimensions. As the random walk variance is decreased beyond 5×10^{-20}, however, the wear parameter estimate does not always converge and cannot be tracked. Since the parameter values are unknown, an optimal value cannot be determined a priori. So, selecting an appropriate value can be difficult, but knowledge of the correct order of magnitude of the parameter is helpful. Additionally, correction loop meth-

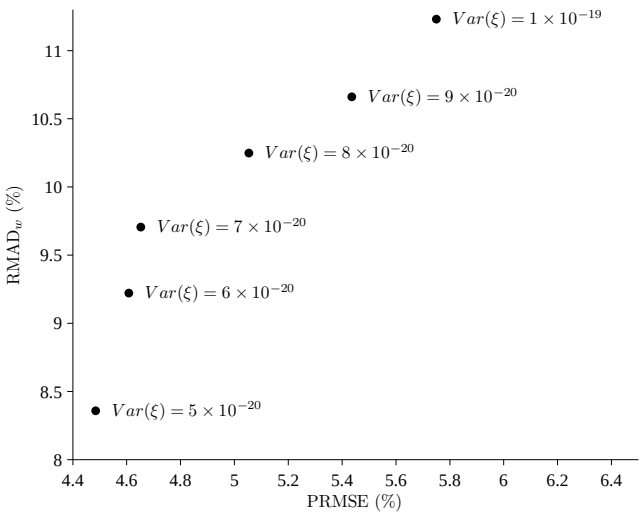


Figure 5. Estimation performance for a bottom external leak, with $N = 500$, $M = \{x, f, p_t, p_b\}$, and varying random walk variance

ods can be used to tune this value online as a function of performance (Orchard, Kacprzynski, Goebel, Saha, & Vachtsevanos, 2008; Saha & Goebel, 2011; Daigle & Goebel, 2011). If the unknown parameters may be assumed to be constant, then other approaches can be employed to improve estimates and offset the increase in covariance contributed by the random walk (Liu & West, 2001; Clapp & Godsill, 1999).

Note that the discrete position sensors (*open* and *closed*), in reality, have no noise, but a certain amount of sensor noise must be assumed for sensors within the particle filter framework. Therefore, within the algorithm, we assume some sensor noise is present for these sensors.

5 PREDICTION

Prediction is initiated at a given time k_P. Using the current state estimate, $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k})$ the goal is to compute $p(EOL_{k_P}|\mathbf{y}_{0:k_P})$ and $p(RUL_{k_P}|\mathbf{y}_{0:k_P})$. The particle filter computes

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

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

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

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

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

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.

Algorithm 3 EOL Prediction

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

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.

The pseudocode for the prediction procedure is given as Algorithm 3 (Daigle & Goebel, 2010b). Each particle i is propagated forward until $T_{EOL}(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)$ evaluates to 1; at this point EOL has been reached for this particle. The complexity of the algorithm is variable, as each particle must be simulated forward until its individual EOL, which is different for each particle.

Prediction requires hypothesizing future inputs of the system, $\hat{\mathbf{u}}_k$, because damage progression is rarely independent of the system inputs. In general, these inputs are unknown and must be chosen in order to satisfy both operational constraints and the type of prediction that is required. For the valve, we assume input-independence for the external leaks, as described by their damage models, but not for the remaining faults. For the valve, the problem of hypothesizing inputs is simplified. Each valve cycle corresponds to the same set of inputs, except for the fluid pressures p_L and p_R. However, these pressures can safely be assumed to be constant in our application domain, and, further, they have an almost negligible effect on valve behavior because the forces they produce are very small compared to the other forces acting on the valve. So, the future inputs for each valve cycle are deterministic. We can simply provide repeated valve cycles as input, and the prediction step will determine after how many valve cycles EOL will be reached for each particle.

Fig. 6 shows results from the prediction of friction damage for $N = 100$. Initially, the particles have a very tight distribution of friction parameter values, but the distribution of the wear parameter, w_r, is relatively large. As a result, the individual

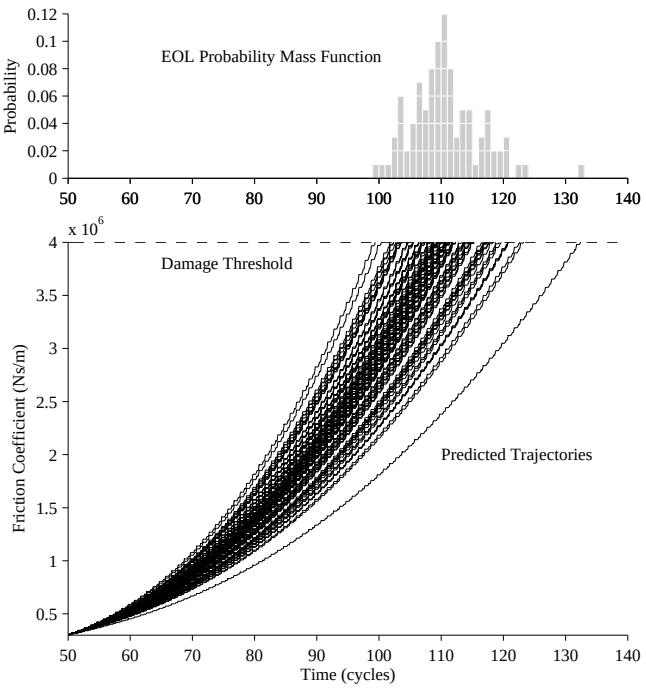


Figure 6. Prediction of friction damage

Parameter	Value
g	9.8 m/s^2
p_{atm}	1.01×10^5 Pa
p_{supply}	5.27×10^6 Pa
m	50 kg
r	6.00×10^3 Ns/m
k	4.80×10^4 N/m
x_o	2.54×10^{-1} m
L_s	3.81×10^{-2} m
A_p	8.10×10^{-3} m^2
V_{t_0}	8.11×10^{-4} m^3
V_{b_0}	8.11×10^{-4} m^3
ρ	7.10×10^2 kg/m^3
A_v	5.07×10^{-2} m^2
C_v	4.36×10^{-1}
R_g	2.96×10^2 J/K/kg
T	293 K
γ	1.4
Z	1
A_s	1.00×10^{-5} m^2
C_s	0.62

Table 1. Nominal valve parameters

trajectories for the different wear parameter values are easily distinguishable as EOL is approached. The different EOL values along with particle weights form an EOL distribution approximated by the probability mass function shown in the figure.

6 RESULTS

We apply the prognostics framework to a pneumatic valve of the Space Shuttle cryogenic refueling system. This system transfers cryogenic propellant (liquid hydrogen) from a storage tank to the vehicle tank through a network of pipes and valves. We focus on one of the transfer line valves in this system. In this section, we first develop and validate the model for the selected valve. Then, since only the discrete $open$ and $closed$ sensors are available for the valve, we establish, through a set of simulation-based experiments, that prognostics can still be performed using only those sensors. We then apply the prognostics approach to historical valve degradation data. Since proactive maintenance was performed early, no complete run-to-failure data exist. We augmented the missing portion with simulated data.

6.1 Model Validation

For the selected pneumatic valve, we identified model parameters using component specifications and estimated the remaining parameters (m, r, k, x_o, and C_s) from nominal data. The model parameter values are shown in Table 1. Note that the pneumatic gas is nitrogen.

Nominally, this valve opens in a little over 1 s and closes within 0.2 s. The valve has an opening time requirement of 5 s and a closing time requirement of 4 s, and these define functional thresholds for EOL. The valve is considered to have failed when these timing limits are exceeded. Further, the valve must fully close when actuating pressure is lost.

Due to the high safety margin of the fueling system, components must always meet tight operational constraints, and components are never intentionally run to failure. We focus on a particular time frame of historical data, consisting of 7 valve cycles, where a clear trend in performance degradation was observed before a maintenance action took place. We analyze this particular data set to determine the dominant damage mode and validate the corresponding damage model.

The obtained valve data includes only the $open$ and $closed$ sensors. Along with valve open/close commands, we may extract the following timing information, shown in Fig. 7:

1. $t_{open,1}$, defined as the difference between the time when the valve is commanded to open and it starts to move open,
2. $t_{open,2}$, defined as the difference between the time when the valve starts to move open and it fully opens,
3. $t_{open} = t_{open,1} + t_{open,2}$, which is the total time for the valve to open,
4. $t_{close,1}$, defined as the difference between the time when the valve is commanded to close and it starts to move closed,
5. $t_{close,2}$, defined as the difference between the time when the valve starts to move closed and it fully closes,

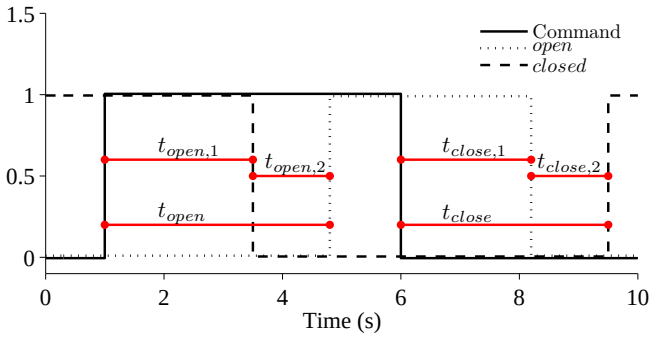


Figure 7. Pneumatic valve timing diagram. A command of 1 opens the valve, and a command of 0 closes the valve.

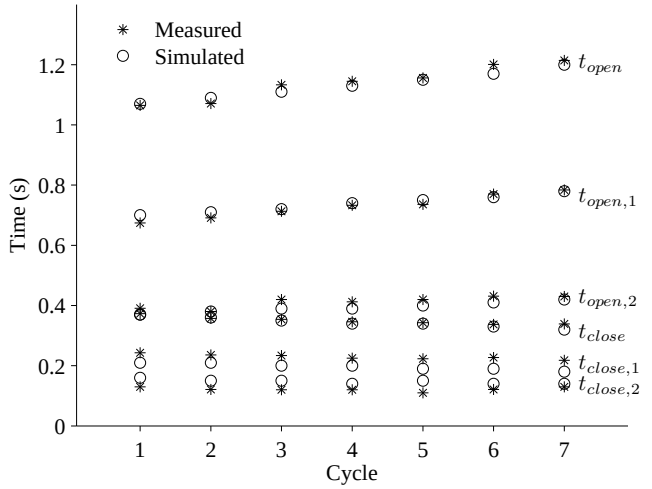


Figure 8. Pneumatic valve model comparison for growth of bottom external leak

6. $t_{close} = t_{close,1} + t_{close,2}$, which is the total time for the valve to close.

We plot these values for the degrading valve in Fig. 8. We note trends in each of the timing variables. Most importantly, the opening time increases, while the closing time decreases. From Fig. 4 (Section 3), we know that only two fault modes may cause this behavior, namely, the internal leak, and the bottom external leak. We may distinguish between these two damage modes by looking at $t_{open,1}$. For the internal leak, a slight decrease in this value is expected as the damage progresses, but for the bottom external leak, a significant increase in this value is expected. According to the data, this value increases, therefore, the bottom external leak is the only consistent damage mode, and must be dominant. Fig. 8 shows simulated timing values plotted alongside the measured timing values, with a wear rate of $w_{e,b} = 3.0 \times 10^{-7}$ m^2/cycle, which best matched the data. The simulated data captures all the trends of the real data, and the simulated timing values are all fairly close to the real values at each cycle.

6.2 Measurement Analysis

Since only the discrete *open* and *closed* sensors are available, we must first establish that prognostics can still be performed satisfactorily in this case. To investigate this issue, we performed a number of simulation experiments for each fault under different measurement sets. In each case, we used $N = 500$ particles, and tuned the random walk variances assuming that the order of magnitude of the wear parameters was known. Wear parameter values were selected so that EOL occurred near 100 cycles. For each fault and measurement set, 5 experiments were performed. Table 2 presents averaged results over these experiments. The performance metrics are defined in the Appendix. Estimation is evaluated based on the estimate of the unknown wear parameter, as it is this estimate that most greatly influences subsequent prognostics performance, since the future valve usage is well-defined. Estimation accuracy is calculated using the percentage root mean square error (PRMSE), where we ignore the first 10 cycles, which are associated with convergence. We calculate the spread using the relative median absolute deviation (RMAD), which is a robust measure of statistical dispersion. We compute also convergence of the wear parameter estimate, denoted as C_w, over the first 3 cycles. For a prediction point k_P, we compute measures of accuracy and precision. For accuracy, we use the relative accuracy (RA) metric (Saxena, Celaya, Saha, Saha, & Goebel, 2010). We use $\overline{\text{RA}}$ to denote the RA averaged over all prediction points. We calculate prediction spread using RMAD, which we denote as RMAD_{RUL}. We use $\overline{\text{RMAD}}_{RUL}$ to denote RMAD_{RUL} averaged over all prediction points.

From Table 2, it is clear that prognostics can be performed successfully with all measurement sets, including using only the discrete sensors *open* and *closed*. Because of the loss of information in going from the continuous sensors to the discrete sensors, the state distribution has a much wider variance, usually increasing by around 50%. In turn, the wider variance causes the estimate to smooth out, and, as a result, PRMSE is lowered, and this corresponds to increases in RA, usually around 0.5%, however this is at the cost of a less confident prediction, as shown by RMAD_{RUL}. Further, convergence of the wear parameter estimate is slower using only the discrete sensors.

The results show that, if it is feasible to add the continuous sensors, prognostics performance can be improved significantly. The increase in information provided by these sensors leads to more confident predictions. Different sensors are more useful for different faults, so additional sensors may be selected based on which faults are more likely to appear, or more critical. For friction and spring damage, the position sensor is more useful than the pressure sensors, as shown by the decreased spread when $M = \{x\}$ compared to when $M = \{p_t, p_b\}$ for these faults. In contrast, the pressure sensors are more valuable for the leak faults.

Fault	M	PRMSE$_w$	$\overline{\text{RMAD}}_w$	C_w	$\overline{\text{RA}}_{RUL}$	$\overline{\text{RMAD}}_{RUL}$
Friction	$\{x, f, p_t, p_b\}$	4.00	8.07	30.72	96.59	8.64
	$\{x\}$	3.92	9.38	28.79	97.24	8.62
	$\{p_t, p_b\}$	3.58	10.11	34.90	96.27	10.17
	$\{open, closed\}$	3.52	13.53	44.68	97.38	11.68
Spring Rate	$\{x, f, p_t, p_b\}$	5.58	10.87	32.70	94.91	11.04
	$\{x\}$	5.84	12.36	36.07	94.80	12.43
	$\{p_t, p_b\}$	5.47	13.14	31.74	94.51	13.09
	$\{open, closed\}$	5.10	16.58	43.50	94.58	16.40
Internal Leak	$\{x, f, p_t, p_b\}$	6.00	10.33	25.88	95.26	10.82
	$\{x\}$	5.84	13.81	21.40	95.29	14.37
	$\{p_t, p_b\}$	5.39	10.52	27.59	94.74	10.69
	$\{open, closed\}$	4.51	17.42	33.49	95.94	17.06
Top External Leak	$\{x, f, p_t, p_b\}$	5.33	9.27	21.54	95.73	9.32
	$\{x\}$	6.78	10.99	17.30	94.72	11.06
	$\{p_t, p_b\}$	4.00	10.04	20.83	96.87	10.12
	$\{open, closed\}$	5.14	14.11	18.82	95.94	14.24
Bottom External Leak	$\{x, f, p_t, p_b\}$	5.52	11.21	25.29	95.57	11.13
	$\{x\}$	6.02	14.49	25.15	95.04	14.22
	$\{p_t, p_b\}$	5.13	11.36	24.45	95.64	11.27
	$\{open, closed\}$	5.07	18.59	34.12	95.69	18.14

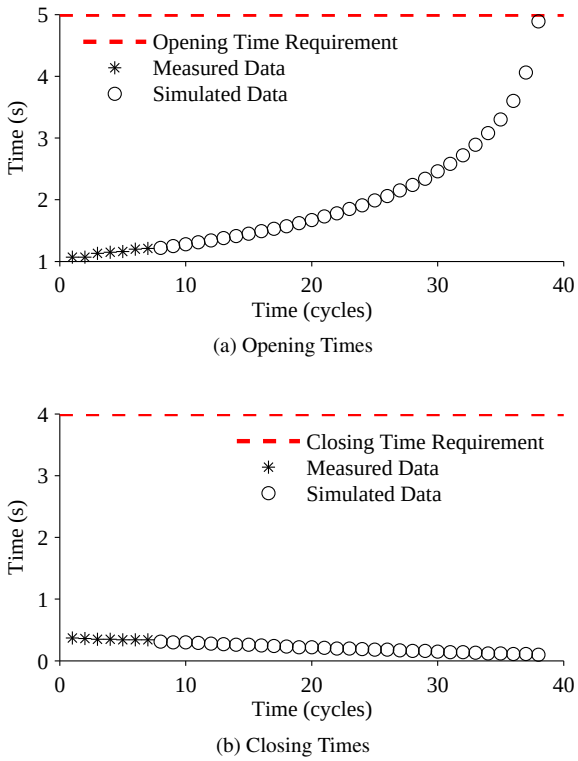
Table 2. Prognostics performance for different measurement sets

The valve considered here only sees discrete operation, where the valve is either fully closed or fully opened. In this case, the discrete open and closed sensors provide enough information for prognostics. For situations where the valve's position is controlled in a continuous manner, a continuous position sensor is necessary for prognostics. For such valves, this sensor is usually already available for feedback to the position controller.

6.3 Demonstration of the Approach

Given that prognostics may still be performed using only the *open* and *closed* sensors, we now apply the prognostic algorithms to the historical data. Because the actual data consisted of only 7 valve cycles before maintenance was performed, we extend the data with simulated valve data, based on the identified bottom external leak damage mode with $w_{e.b} = 3.0 \times 10^{-7}$ m^2/cycle, up to EOL at 38 cycles, at which the valve fails to open within the 5 second limit. Fig. 9 shows the valve timing data used for the demonstration. It is important to note that, even though the trend in valve opening time appears linear over the first 10 to 15 cycles, the overall trend is clearly nonlinear. This behavior is captured by the model, and serves as a justification of a model-based approach as opposed to simple trending strategies. Projecting a linear trend out from the 10th cycle would produce a severe overestimate of the true RUL.

To demonstrate a general solution, we allowed the particle



(a) Opening Times

(b) Closing Times

Figure 9. Real and simulated valve timing

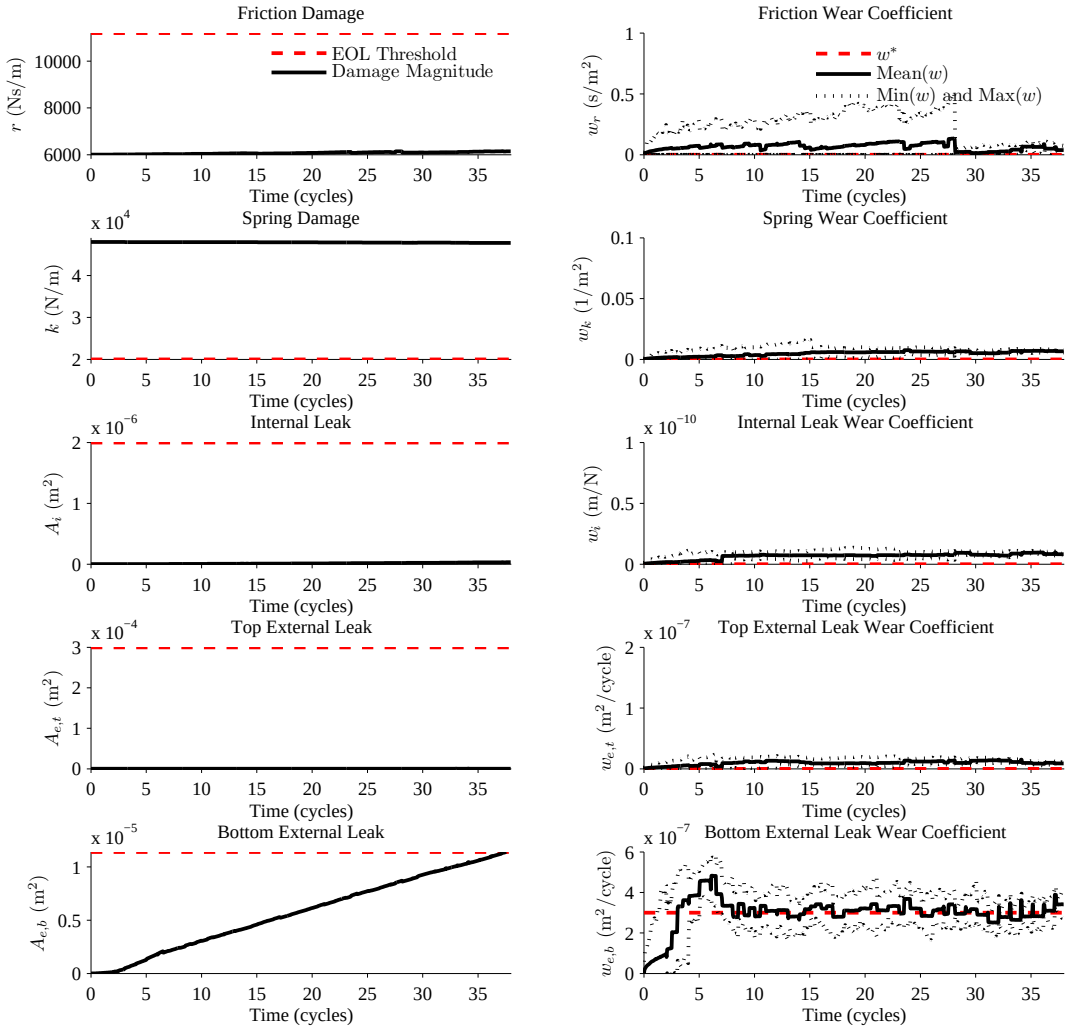


Figure 10. Estimation of pneumatic valve damage modes

filter to jointly estimate all the damage modes of the valve. Because of the large state space, $N = 2000$ was used. The algorithm should converge to correct wear rates for each damage mode, and, in this case, identify the bottom external leak as the dominant damage mechanism. Estimation results are shown in Fig. 10. The algorithm correctly identifies the bottom external leak as the dominant damage mode, since it is the only damage mode for which the estimated damage variable is significantly increasing. During the real data portion, the algorithm is still converging, and by the time the real data portion has ended, the algorithm estimates the wear parameter within 30% of the true value, and the mean RUL prediction is within 25% of the true RUL. The estimated value of the RUL at that time point is less than the true value, resulting in a conservative estimate. By 10 cycles, the mean RUL prediction has improved to within 7% of the true RUL.

The prognostics performance is summarized in Table 3. All times are given in cycles. RUL^* denotes the true RUL value. We show the predictions starting at 5 cycles. At this point, the particle filter is still converging, but the RUL is predicted with an accuracy of 80%. Afterwards, RA ranges from 87 to 97%. The RMAD of the prediction varies from around 5 to 7%. So, RUL is predicted with reasonable accuracy, and the predictions are fairly confident. It is also useful to inspect the RUL prediction at a selected level of confidence. Fig. 11 shows the predictions at the 99% confidence level, i.e., the value at which 99% of the prediction distribution is greater than or equal to that value. The predictions at this confidence level always lie below the true RUL, therefore, they serve as conservative estimates upon which risk-averse decisions can be made.

Although encouraging, the results presented here represent only a limited validation of the overall approach, since the historical data covers only a single fault mode of the five considered, and extends only to 7 valve cycles. A more complete validation requires a more flexible testbed, such as one that

t_P	RUL^*	$\overline{RUL}$	RA_{RUL}	RMAD_{RUL}
5.0	33.00	26.60	80.60	7.28
7.5	30.50	27.53	90.27	5.49
10.0	28.00	26.04	92.99	7.65
12.5	25.50	24.42	95.77	5.96
15.0	23.00	21.75	94.57	7.61
17.5	20.50	21.84	93.46	6.04
20.0	18.00	16.55	91.93	6.52
22.5	15.50	13.53	87.29	6.63
25.0	13.00	11.64	89.52	7.87
27.5	10.50	10.15	96.71	6.90
30.0	8.00	6.78	84.74	5.14
32.5	5.50	5.34	97.11	4.81
35.0	3.00	2.93	97.69	7.16

Table 3. Pneumatic valve prognostics performance

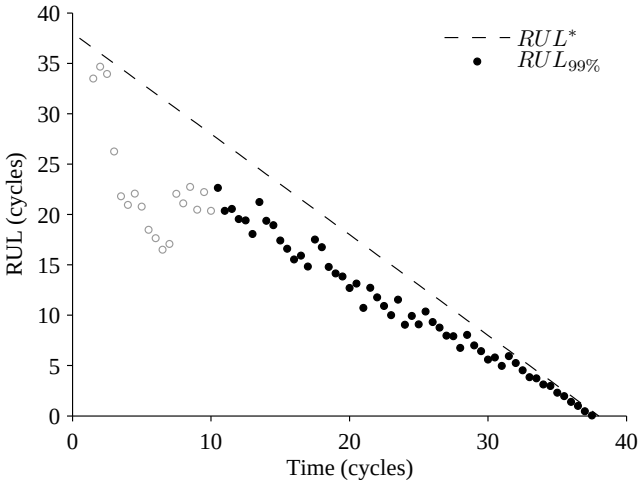


Figure 11. RUL predictions at the 99% confidence level

would allow the injection of faults and/or valves to be run to failure.

7 Related Work

Very little work has been done in valve prognostics. A health monitoring application for a pneumatic valve in a pressure control system is considered in (Gomes, Ferreira, Cabral, Glavão, & Yoneyama, 2010). Friction damage, spring aging, and internal leaks are also identified as common failure modes in their application. The distribution of pressure signals is monitored using a probability integral transform, and a measure of dissimilarity from a baseline distribution serves as a health index. The approach is purely data-driven, relying on a nominal distribution of pressure signals to serve as a baseline, and comparing future pressure distributions to the baseline. No identification of the form of damage present is performed, and prediction is not performed either.

Model-based prognostic techniques have been applied to other systems, and particle filtering techniques have been particularly popular. In (Saha & Goebel, 2009), a particle filtering approach is applied to end of discharge and end of life prediction for lithium-ion batteries. In (Orchard et al., 2008), a particle filter-based diagnosis and prognosis framework is applied to crack growth in aircraft components. They also implement outer correction loops to control the variance of the random walk of the unknown parameters based on prediction error. This type of approach is preferred to the fixed variance method employed in this paper, however, the approach presented in (Orchard et al., 2008) is only applicable when a feature can be derived that is a direct function of a single fault dimension, which is not the case with valves. In (Abbas, Ferri, Orchard, & Vachtsevanos, 2007), the authors apply a particle filter-based prognosis method to prediction of battery grid corrosion. A physics-based methodology to prediction of aircraft engine bearing spalls is developed in (Bolander, Qiu, Eklund, Hindle, & Rosenfeld, 2010). Particle filters are used for estimation of spall size, using diagnostic information from vibration and oil debris sensors in the update step. Prediction is performed using anticipated future load and speed, and the approach was evaluated in a physical testbed. Although the size of our state-parameter vector is much larger than those considered in these previous works, our particle filter-based approach does not differ from these previous approaches in any fundamental way.

Other filtering techniques may also be used. In (Chelidze, 2002), the system is modeled in a hierarchical fashion, with a fast-time observable subsystem coupled to a slow-time damage subsystem, where damage is tracked using a Kalman filter. The approach is applied to tracking open circuit battery voltages in a vibrating beam testbed using only beam displacement measurements. In a related approach, a model-based prognosis methodology is developed in (Luo et al., 2008) using an interacting multiple model filter for state-parameter estimation and prediction. The approach is applied to prognosis of fatigue cracks in a simulation of an automotive suspension system.

8 Conclusions

In this paper, we developed a model-based prognostics approach using particle filters for joint state-parameter estimation. The estimated damage state of the valve is propagated forward in time to predict EOL and form EOL and RUL distributions. We applied the approach to a pneumatic valve, developing a detailed physics-based model that included damage progression models. Through simulation experiments, we established that prognostics could be performed successfully using only the discrete position sensors of the valve, and applied the approach to real valve data under this constraint. The results here demonstrated the effectiveness of a model-based approach, and gave insight into the selection of sensors for

valve prognostics.

Here, we advocate a model-based approach, where performance will depend greatly on the accuracy of the model provided. Clearly, developing such models is a key obstacle. The model developed here is suitable for a normally-closed pneumatic valve with a piston-based actuation mechanism. With small modifications, the model may also be used for normally-open valves or valves that use a diaphragm mechanism. With other valve types, the actuation mechanism changes, so a new model must be developed. However, there will be several commonalities between the different models. For instance, in past work, we have applied similar modeling techniques to a solenoid valve (Daigle & Goebel, 2010a). Here, damage modes for the spring and friction are also present, and the same damage progression models are used. The damage models can also be translated into a forms for rotational motion, as in (Daigle & Goebel, 2011), where a modified friction damage progression is used in a centrifugal pump model. Of course, one must also consider the cost of developing such detailed models against the corresponding prognostics performance achieved, and this is the subject of recent work (Daigle et al., 2011).

The valves considered here are always operated in the same way; they fully open, then fully close. Therefore, future valve inputs can be easily hypothesized, and in this paper we assumed a single known future input trajectory to provide EOL and RUL in the number of valve cycles. In general, the future inputs may not be well-known, and different sets or distributions of possible inputs must be considered. In future work, we will extend the approach to these cases. Further, although the algorithm assumes all damage modes may be progressing simultaneously, the results here cover cases where only a single damage mode is progressing. Recent work considers the case when multiple damage modes are active for a centrifugal pump simulation (Daigle & Goebel, 2011), and investigating this for pneumatic valves, especially with limited sensing, where fault masking becomes an issue, is considered as future work. Applying the approach to additional testbeds is also part of future work.

APPENDIX

Performance Metrics

Estimation performance is evaluated based on the estimate of the unknown wear parameter. Estimation accuracy is calculated using the percentage root mean square error (PRMSE), which expresses relative estimation accuracy as a percentage:

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

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. In computing PRMSE, we typically ignore the initial portion of data associated with convergence.

We calculate the spread using the relative median absolute deviation (RMAD), which expresses the spread relative to the median as a percentage:

$$\text{RMAD}(X) = 100\frac{\text{Median}_i\left(|X_i - \text{Median}_j(X_j)|\right)}{\text{Median}_j(X_j)},$$

where X is a data set and X_i is an element of that set. For estimation spread, for time k we use the distribution of wear parameter values given by the particle set at k as the data set. We summarize RMAD over an experiment by averaging RMAD over all k. We denote the average RMAD over multiple k using $\overline{\text{RMAD}}_w$:

$$\overline{\text{RMAD}}_w = \text{Mean}_k(\text{RMAD}_{w,k}),$$

where $\text{RMAD}_{w,k}$ denotes the RMAD of the wear parameter at time k.

The final estimation metric is convergence of the wear parameter estimate, denoted as C_w. We use the definition of the convergence metric described in (Saxena et al., 2010), where the convergence of a curve is expressed as the distance from the origin to the centroid under the curve (where a shorter distance is better). We use the absolute error of the hidden parameter estimate as the curve. We compute convergence only over the initial convergence period, so that errors after convergence do not contribute. The same time window must be used consistently over all experiments to properly compare.

For a prediction point k_P, we compute measures of accuracy and precision. For accuracy, we use the relative accuracy (RA) metric (Saxena et al., 2010), computed as a percentage:

$$\text{RA}_{k_P} = 100\left(1 - \frac{|RUL^*_{k_P} - \text{Median}_i(RUL^i_{k_P})|}{RUL^*_{k_P}}\right),$$

where $RUL^*_{k_P}$ is the true RUL at time k_P. Here, we have chosen the median as a robust measure of central tendency of the distribution. We use $\overline{\text{RA}}$ to denote the averaged RA over all prediction points. We calculate prediction spread using RMAD, which we denote as RMAD_{RUL}. We average RMAD_{RUL} over all prediction points to obtain a single number representing prediction spread over a single experiment, denoted using $\overline{\text{RMAD}}_{RUL}$.

NOMENCLATURE

EOL	end of life
RUL	remaining useful life
t	time (continuous)
k	time (discrete)
t_P or k_P	time of prediction
$\mathbf{x}$	state vector
$\boldsymbol{\theta}$	parameter vector
$\mathbf{v}$	process noise vector
$\mathbf{y}$	output vector
$\mathbf{n}$	measurement noise vector
T_{EOL}	EOL threshold
PRMSE	percent root mean square error
RMAD	relative median absolute deviation
RA	relative accuracy

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 on-line nonlinear/non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, *50*(2), 174–188.

Bolander, N., Qiu, H., Eklund, N., Hindle, E., & Rosenfeld, T. (2010, October). Physics-based remaining useful life prediction for aircraft engine bearing prognosis. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2010.*

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.

Chelidze, D. (2002). Multimode damage tracking and failure prognosis in electromechanical system. In *Proceedings of the SPIE Conference* (Vol. 4733, pp. 1–12).

Clapp, T. C., & Godsill, S. J. (1999). Fixed-lag smoothing using sequential importance sampling. *Bayesian Statistics IV*, 743–752.

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. (2010a, October). Improving Computational Efficiency of Prediction in Model-based Prognostics Using the Unscented Transform. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2010.*

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

Daigle, M., & Goebel, K. (2011, March). Multiple damage progression paths in model-based prognostics. In *Proceedings of the 2011 IEEE Aerospace Conference.*

Daigle, M., Roychoudhury, I., Narasimhan, S., Saha, S., Saha, B., & Goebel, K. (2011, September). Investigating the Effect of Damage Progression Model Choice on Prognostics Performance. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011.*

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

Gomes, J. P. P., Ferreira, B. C., Cabral, D., Glavão, R. K. H., & Yoneyama, T. (2010, October). Health monitoring of a pneumatic valve using a PIT based technique. In *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2010.*

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

Julier, S., & Uhlmann, J. (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).

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.

Liu, J., & West, M. (2001). Combined parameter and state estimation in simulation-based filtering. *Sequential Monte Carlo methods in Practice*, 197–223.

Luo, J., Pattipati, K. R., Qiao, L., & Chigusa, S. (2008, September). Model-Based Prognostic Techniques Applied to a Suspension System. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *38*(5), 1156 -1168.

Orchard, M. (2007). *A Particle Filtering-based Framework for On-line Fault Diagnosis and Failure Prognosis*. Unpublished doctoral dissertation, Georgia Institute of Technology.

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.*

Perry, R., & Green, D. (2007). *Perry's chemical engineers' handbook*. McGraw-Hill Professional.

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 Manage-*

ment 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.*

Saha, B., & Goebel, K. (2011). Model Adaptation for Prognostics in a Particle Filtering Framework. *International Journal of Prognostics and Health Management*.

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for Offline Evaluation of Prognostic Performance. *International Journal of Prognostics and Health Management (IJPHM)*, *1*.

Schwabacher, M. (2005). A survey of data-driven prognostics. In *Proceedings of the AIAA Infotech@Aerospace Conference.*