

# Prognostics for Ground Support Systems: Case Study on Pneumatic Valves

Matthew Daigle\*

*University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035, USA*

Kai Goebel†

*NASA Ames Research Center, Moffett Field, CA, 94035, USA*

Prognostics technologies determine the health (or damage) state of a component or sub-system, and make end of life (EOL) and remaining useful life (RUL) predictions. Such information enables system operators to make informed maintenance decisions and streamline operational and mission-level activities. We develop a model-based prognostics methodology for pneumatic valves used in ground support equipment for cryogenic propellant loading operations. These valves are used to control the flow of propellant, so failures may have a significant impact on launch availability. Therefore, correctly predicting when valves will fail enables timely maintenance that avoids launch delays and aborts. The approach utilizes mathematical models describing the underlying physics of valve degradation, and, employing the particle filtering algorithm for joint state-parameter estimation, determines the health state of the valve and the rate of damage progression, from which EOL and RUL predictions are made. We develop a prototype user interface for valve prognostics, and demonstrate the prognostics approach using historical pneumatic valve data from the Space Shuttle refueling system.

## Nomenclature

ISHM	Integrated Systems Health Management
EOL	end of life
RUL	remaining useful life
LH2	liquid hydrogen
$t$ or $k$	time
$t_P$ 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
RMAD	relative median absolute deviation
RA	relative accuracy

## I. Introduction

Within Integrated Systems Health Management (ISHM), technical advances in and increased acceptance of automated diagnosis technologies are leading to a shift in research focus to the area of automated prognos-

---

\*Associate Scientist, University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA 94035. Member, AIAA.

†Senior Scientist, Prognostics Center of Excellence, NASA Ames Research Center, Moffett Field, CA 94035. Member, AIAA.

tics. Prognostics algorithms focus on determining the health (or damage) state of a component or subsystem, and making *end of life* (EOL) and *remaining useful life* (RUL) predictions. EOL and RUL predictions enable system operators to make informed maintenance decisions and streamline operational and mission-level activities.

As a case study, we consider ground support equipment from the Space Shuttle cryogenic refueling system. This system transfers cryogenic propellants from a storage tank to the vehicle tank through a network of pipes and valves. Since the valves are used to control the flow of propellant, failures may have a significant impact on launch availability. Hence, valve prognostics can provide significant value to launch operations. We focus on one of the pneumatic transfer line valves in this system, and apply a model-based prognostics approach.

For the selected valve, we develop a complete prognostics solution, including models, algorithms, and a user interface. The prognostics algorithms are model-based, meaning that, unlike data-driven approaches,<sup>1</sup> we use mathematical models describing the underlying physical principles of valve operation and how valves fail.<sup>2-6</sup> We construct a detailed physics-based model of a pneumatic valve that includes models of different damage mechanisms and how they progress in time. Using this model, we employ the particle filtering algorithm for joint state-parameter estimation to determine the health state of the valve and the rates of damage progression, from which EOL and RUL predictions are made. Although we focus on a pneumatic valve, the approach is general in that it may be applied to any component, given its model. We also develop a prototype user interface to communicate the algorithm results, based on an analysis of what information the algorithms provide and how the information should be made available to the user. We demonstrate the prognostics approach as applied to a real-world case study using historical pneumatic valve data from the Space Shuttle refueling system.

The paper is organized as follows. Section II formulates the prognostics approach. Section III describes the valve model. Section IV describes the damage estimation algorithm, and Section V describes the prediction algorithm. Section VI develops the prognostics interface. Section VII demonstrates the approach on historical pneumatic valve data. Section VIII concludes the paper.

## II. Prognostics Approach

The goal 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.

### II.A. Problem Formulation

We assume the system may be described 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. This form represents 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, 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 expressed through 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

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

and RUL with

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

Due to process noise, measurement noise, and uncertainty in the future inputs of the system, we compute a probability distribution of the EOL or RUL, i.e., at time  $t_P$ , we compute  $p(EOL(t_P)|\mathbf{y}_{0:t_P})$  and/or  $p(RUL(t_P)|\mathbf{y}_{0:t_P})$ .

## II.B. Prognostics Architecture

We adopt a model-based approach that utilizes detailed physics-based models of components and systems that include descriptions of how faults evolve in time. These models are parameterized by 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 when desired.

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  $t = 0$ , 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 focus on the damage estimation and prediction modules, and begin prognostics at  $t = 0$  without the aid of an FDII solution.

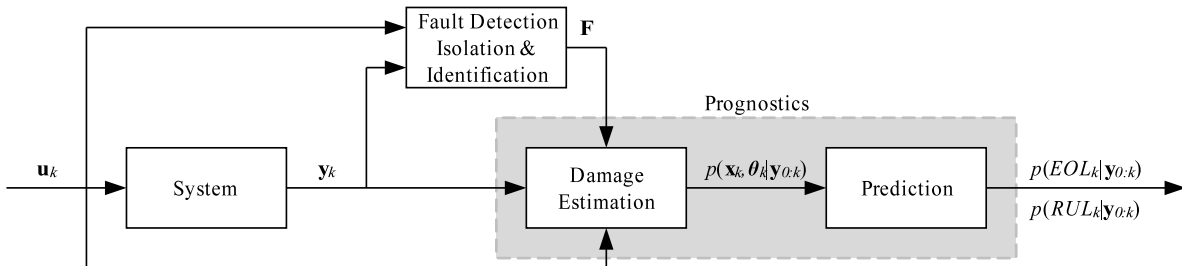


Figure 1. Prognostics architecture.

## III. 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 the valve closes when pressure is lost.

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 when the valve is fully closed is defined as  $x = 0$ , and when fully open is defined as  $x = L_s$ , where  $L_s$  is the stroke length of the valve.

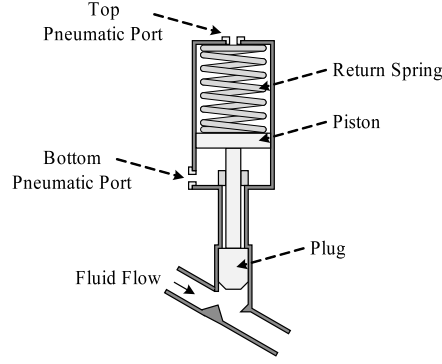


Figure 2. Pneumatic valve.

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, which 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.

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

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

$$p_b(t) = \frac{m_b(t)R_gT}{V_{b_0} + A_px(t)}$$

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.<sup>7</sup> Non-choked flow for  $p_1 \geq p_2$  is given by  $f_{g,nc}(p_1, p_2) =$

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

where  $\gamma$  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  $p_1 \geq p_2$ , which occurs when  $p_1/p_2$  exceeds  $\left(\frac{\gamma+1}{2}\right)^{\gamma/(\gamma-1)}$ , is given by

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

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

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

where  $f_v$  is the fluid flow through the valve:

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

$C_v$  is the (dimensionless) flow coefficient of the valve,  $\rho$  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 which output 1 if the valve is in the fully opened or fully closed state:

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

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

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 opens to the supply pressure. When the force on the underside of the piston is large enough to overcome the opposing forces, the valve begins to move upward as the pneumatic gas continues to flow into and out of the valve actuator. In a little over 1 s, the valve is completely open. The valve is commanded to close at 10 s. The bottom pneumatic port opens to atmosphere and the top opens to the supply pressure. When the force balance becomes negative, the valve starts to move downward, and completely closes within 1 s. The valve closes faster than it opens due to the return spring.

### III.A. Damage Modeling

From valve documentation and historical maintenance records, we have identified friction damage, spring damage, internal valve leaks, and external valve leaks as faults that lead to EOL. The variables that describe the magnitude of damage augment the state vector  $\mathbf{x}(t)$ , and the parameters defining their progression form the unknown parameter vector  $\boldsymbol{\theta}(t)$ .

One damage mechanism present in valves is sliding wear, which results in an increase in friction.<sup>8</sup> We characterize friction damage as change in the friction coefficient, and model the damage progression in a form similar to the sliding wear equation:<sup>2</sup>

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

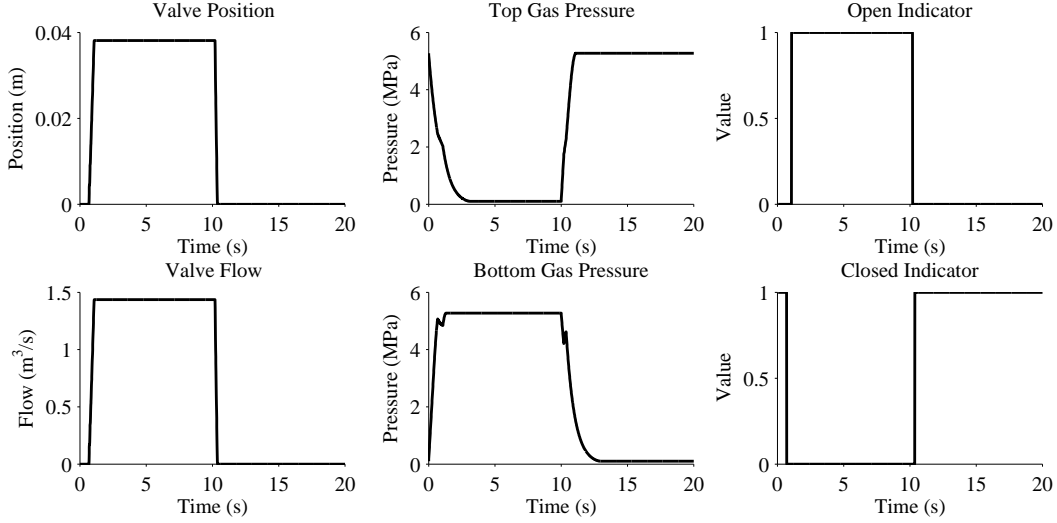


Figure 3. Nominal valve operation.

where  $w_r$  is the wear coefficient, and  $F_f(t)$  is the friction force. Friction damage causes the valve to both open and close more slowly. The valve is considered to be at EOL when it cannot open or close within required timing limits. We define  $r^+$  as the maximum value of the friction parameter at which the valve still behaves within these limits. So,  $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$  if  $r(t) > r^+$ .

We assume a similar equation form for spring damage:<sup>2</sup>

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

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. Spring damage causes the valve to open faster and close slower. The spring must also remain strong enough to close the valve against system pressure when the actuating pressure is lost, and the loss of this function also defines EOL. We define  $k^-$  as the smallest value of  $k$  at which the valve will both close within the timing limit and 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:<sup>2</sup>

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

where  $w_i$  is the wear coefficient. 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  $A_i(t) > A_i^+$ .

External leaks may also form at the actuator connections to the pneumatic gas supply, due to corrosion and other environmental factors. We assume the growth of the area of the leak holes,  $A_{e,t}(t)$  and  $A_{e,b}(t)$ , is linear:

$$\begin{aligned} \dot{A}_{e,t}(t) &= w_{e,t} \\ \dot{A}_{e,b}(t) &= w_{e,b}, \end{aligned}$$

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. A leak at the top pneumatic port causes the valve to open more quickly and close more slowly, whereas a leak at the bottom pneumatic port causes the valve to open more slowly and close more quickly. 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  $A_{e,b}(t) > A_{e,b}^*$ .

---

**Algorithm 1** SIR Filter

---

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

---

## IV. Damage Estimation

In model-based prognostics, damage estimation is fundamentally a joint state-parameter estimation problem, i.e., computation of  $p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k})$ .<sup>2,3</sup> Particle filters offer a general solution to this problem for nonlinear systems with non-Gaussian noise terms,<sup>9,10</sup> and have become a favored approach within prognostics.<sup>2,3,6,11,12</sup> As described in Section III, for the pneumatic valves, both the state equation  $\mathbf{f}$  and the output equation  $\mathbf{h}$  are highly nonlinear, and, further,  $\mathbf{h}$  contains the discrete measurements *open* and *closed*, necessitating the use of particle filters.

In particle filters, the state distribution is approximated by a set of discrete weighted samples, called *particles*. The particle approximation to the state distribution is given by

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

with  $N$  denoting the number of particles. For particle  $i$ ,  $\mathbf{x}_k^i$  denotes the state estimates,  $\boldsymbol{\theta}_k^i$  denotes the parameter estimates, and  $w_k^i$  denotes the weight. The posterior density is approximated by

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

with  $\delta_{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)}(d\mathbf{x}_k d\boldsymbol{\theta}_k)$  denoting the Dirac delta function located at  $(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i)$ .

We utilize the sampling importance resampling (SIR) particle filter, given as Algorithm 1. Each particle is propagated forward to time  $k$  by sampling new parameter and state values. The particle weight is assigned using  $\mathbf{y}_k$ . The weights are then normalized, and then the particles are resampled, e.g., using the systematic resampling algorithm.<sup>13</sup>

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, we assign a random walk evolution, i.e., for parameter  $\theta$ ,  $\theta_k = \theta_{k-1} + \xi_{k-1}$ , where  $\xi_{k-1}$  is a noise term. During the sampling step, particles are generated with parameter values that will be different from the current values of the 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 with similar values. The particle filter converges to the true values as the process is repeated over each step of the algorithm.

The selected variance of the random walk noise determines both the rate of this convergence and the estimation performance once convergence is achieved. Therefore, this parameter should be tuned to obtain the best possible performance, but the optimal value is dependent on the value of the hidden wear parameter, which is unknown. We use the variance control method presented in Ref. 6, shown as Algorithm 2. In this approach, the variance of the hidden wear parameter estimate is controlled to a user-specified range by modifying the random walk noise variance. We assume that the  $\boldsymbol{\xi}$  values are tuned initially based on the maximum expected wear rates. The algorithm uses relative median absolute deviation (RMAD) as the measure of spread, defined as:

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

---

**Algorithm 2**  $\xi$  Adaptation

---

**Inputs:**  $\{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i), w_k^i\}_{i=1}^N, \boldsymbol{\xi}_{k-1}$   
**State:**  $\mathbf{a}$   
**Outputs:**  $\boldsymbol{\xi}_k$   
**if**  $k = 0$  **then**  
     $\mathbf{a} \leftarrow \mathbf{0}$   
**end if**  
**for all**  $j \in \{1, 2, \dots, n_\theta\}$  **do**  
     $v_j \leftarrow \text{RMAD}(\{\theta_k^i(j)\}_{i=1}^N)$   
    **if**  $\mathbf{a}(j) = 0$  **and**  $v_j < T$  **then**  
         $\mathbf{a}(j) \leftarrow 1$   
    **end if**  
    **if**  $\mathbf{a}(j) = 0$  **then**  
         $v_j^* \leftarrow v_{j0}^*$   
    **else**  
         $v_j^* \leftarrow v_{j\infty}^*$   
    **end if**  
     $\boldsymbol{\xi}_k(j) \leftarrow \boldsymbol{\xi}_{k-1}(j) \left(1 + P \frac{v_j - v_j^*}{v_j^*}\right)$   
**end for**

---

where  $X$  is a data set and  $X_i$  is an element of that set. The adaptation scheme resembles a proportional control law, where the error between the actual RMAD of a parameter  $\boldsymbol{\theta}(j)$ , denoted as  $v_j$  in the algorithm, and the desired RMAD value (e.g., 10%), denoted as  $v_j^*$  in the algorithm, is normalized by  $v_j$ . The error is then multiplied by a factor  $P$  (e.g.,  $1 \times 10^{-3}$ ), and the corresponding variance  $\boldsymbol{\xi}(j)$  is increased or decreased by that percentage. There are two different setpoints. First, we allow for a convergence period, with setpoint  $v_{j0}^*$  (e.g., 50%). Once  $v_j$  reaches  $T$  (e.g.,  $1.2v_{j0}^*$ ), we mark it using the  $\mathbf{a}(j)$  flag, and begin to control it to a new setpoint  $v_{j\infty}^*$  (e.g., 10%).

## V. 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<sup>14</sup>

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

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.

The pseudocode for the prediction procedure is given as Algorithm 3.<sup>2</sup> 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. Prediction requires hypothesizing future inputs of the system,  $\hat{\mathbf{u}}_k$ . For the valve, this problem is simplified, because each valve cycle corresponds to the same set of inputs, because the fluid pressures  $p_L$  and  $p_R$  can safely be assumed to be constant in our application domain, and, further, 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.



---

**Algorithm 3** EOL Prediction

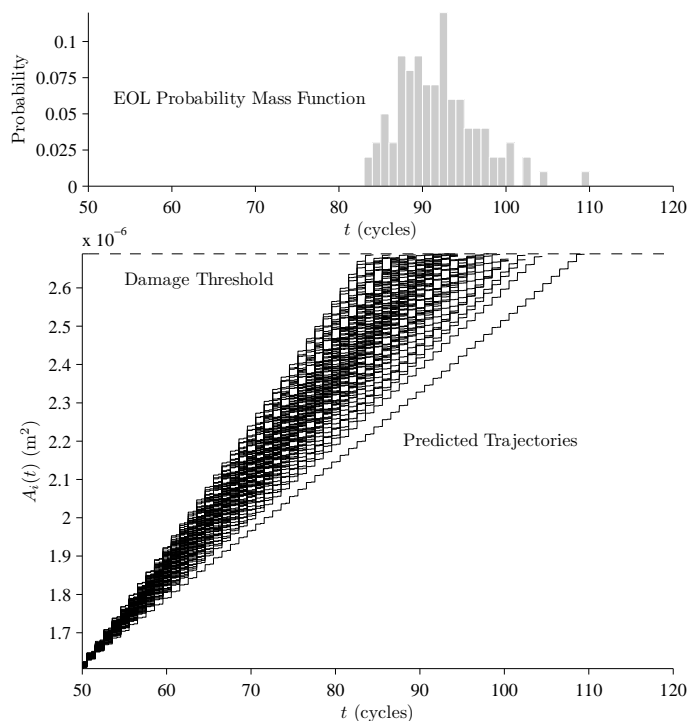
---

**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  $\hat{\mathbf{u}}_k$   
         $\boldsymbol{\theta}_{k+1}^i \sim p(\boldsymbol{\theta}_{k+1}^i | \boldsymbol{\theta}_k^i)$   
         $\mathbf{x}_{k+1}^i \sim p(\mathbf{x}_{k+1}^i | \mathbf{x}_k^i, \boldsymbol{\theta}_k^i, \hat{\mathbf{u}}_k)$   
         $k \leftarrow k + 1$   
         $\mathbf{x}_k^i \leftarrow \mathbf{x}_{k+1}^i$   
         $\boldsymbol{\theta}_k^i \leftarrow \boldsymbol{\theta}_{k+1}^i$   
    **end while**  
     $EOL_{k_P}^i \leftarrow k$   
**end for**

---

So, the future inputs for each valve cycle are deterministic. We can simply provide repeated valve cycles as input, and the prediction algorithm will determine EOL in the number of valve cycles.

Fig. 4 shows results from the prediction of internal leak damage for  $N = 100$ . Initially, the particles have a very tight distribution of internal leak area values,  $A_i(t)$ , but the distribution of the wear parameter,  $w_i$ , is relatively large. As a result, the individual 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. The internal leak damage accrues step-wise as the valve moves, but over large time scales the damage progression appears linear, as shown in the figure.



**Figure 4.** Prediction of internal leak damage.

## VI. Prognostics Interface Design

Since prognostics is still a nascent field, there are no real examples of successful prognostics interfaces. Therefore, in this section, we try to establish some fundamental guidelines and principles for the design of prognostics interfaces, based on our experience developing a prototype interface for our case study. We first describe the information available from prognostics algorithms that may be presented to a user, then provide some design guidelines, followed by a description of the interface we designed.

### VI.A. Information Provided by Prognostics

A prognostics interface must present the results of prognostics algorithms to a user and allow the user to interact with the algorithm to obtain additional results, view information in a variety of ways, and access long-term historical information. In contrast to most systems interfaces, e.g. diagnostics interfaces, in which information is viewed only over short time scales, such as the duration of a mission, prognostics interfaces cope with information over very large time scales, such as sequences of missions. Therefore it is important to distinguish these two time scales clearly. Further, it is more useful to an operator for results to be presented in an operational or missions context, e.g., how many missions a component will survive as opposed to how much time or how many cycles it will last.

Prognostics algorithms are capable of presenting a wealth of information to the user. Foremost, algorithms generate predictions of EOL and RUL, i.e.,  $p(EOL(k_p)|\mathbf{y}_{0:k_p})$  and/or  $p(RUL(k_p)|\mathbf{y}_{0:k_p})$  at given prediction times  $k_p$ . Predictions may be computed automatically by the algorithm, but also on request from a user. As described in Section IV, model-based prognostics algorithms also compute supportive information, such as state and parameter estimates,  $p(\mathbf{x}_k, \boldsymbol{\theta}_k|\mathbf{y}_{0:k})$ , which includes estimates of the amount of damage and the rates of damage progression.

As shown in Algorithm 3, the state estimates are used directly to compute EOL. Using the same basic procedure, other predictions can be made as well, which may be useful to the user. Future state, parameter, and output trajectories can be computed and displayed to the user, which can show how the component is expected to perform in the future, and how damage is expected to progress. Note also that a number of health or damage indices may be computed from the state estimates, e.g., that a valve is at 50% health or its spring is at 75% health. Therefore, predictions of these health indices may also be obtained from the predictions of the state trajectories. This type of information is useful to the user because it helps in understanding the EOL and RUL predictions made by the algorithm.

Once a component has failed, and ground truth is hence known, the outputs of the algorithm over the life of the component can be used to evaluate the performance of the algorithm. Various offline performance metrics may be used for this purpose, such as relative accuracy, convergence, prognostics horizon, and  $\alpha$ - $\lambda$  performance.<sup>15</sup> This higher-level information is also beneficial to a system operator.

### VI.B. Design Guidelines for a Prognostics Interface

With the wealth of information available from prognostics algorithms, a prognostics interface must avoid overwhelming the user with too much technical information. In the early life of a component, prognostics is not a concern, so the interface should indicate when damage has reached a threshold where EOL predictions become relevant so that the user knows when to start viewing the algorithm results. The most critical information, which should be displayed most prominently, includes overall component health and the life predictions. The predicted values should be shown over a large time window, so that users can see how the RUL and EOL predictions have evolved in time. Because it takes time for prognostics algorithms to converge, the interface should distinguish between predictions obtained before and after algorithm convergence, since the results before convergence are unreliable.

State, parameter, and output estimates provide additional supportive information that helps the user to understand the current life predictions made by the algorithm. In particular, the evolution of damage states (e.g., spring rate or leakage area) over time demonstrate how the component is degrading, especially when shown against damage limits. It is equally important to show damage as it manifests in the functional performance of the component. For example, with the valves, EOL is largely defined by valve opening and closing times, so showing the progression of these values over time against the functional limits is also useful. Since damage often manifests as nonlinear changes in the system outputs over time (e.g., exponential

degradation), showing the predicted future output values helps the user understand the connection between functional performance and the life predictions with additional clarity.

Since most prognostics algorithms provide results as probability distributions, there are multiple ways to view the results. Histograms or probability density functions can be shown, as well as box plots, or the life predictions at specified confidence levels. The choice of how to display the distribution depends on the type of reasoning that the user is performing. It is then also useful to show summary statistics for the various distributions, including measures of central tendency and statistical dispersion, as well as higher-order moments such as skew and kurtosis.

Besides choosing how information should be displayed, the user may also interact directly with the algorithm by setting assumptions about the future inputs of the component. For example, a user may be interested only in worst-case performance, so will assume maximum loading on the component. Instead, a user may be interested in average performance over a distribution of possible future loading profiles. Each of these cases should be made available to the user.

### VI.C. Valve Prognostics Interface

A prototype prognostics interface for the pneumatic valve case study was developed for demonstration purposes, and provides an implementation of the design concepts discussed in the previous sections. The interface is organized in a hierarchical structure that presents prognostics information at various levels, allowing the user to drill down into more detailed views when desired.

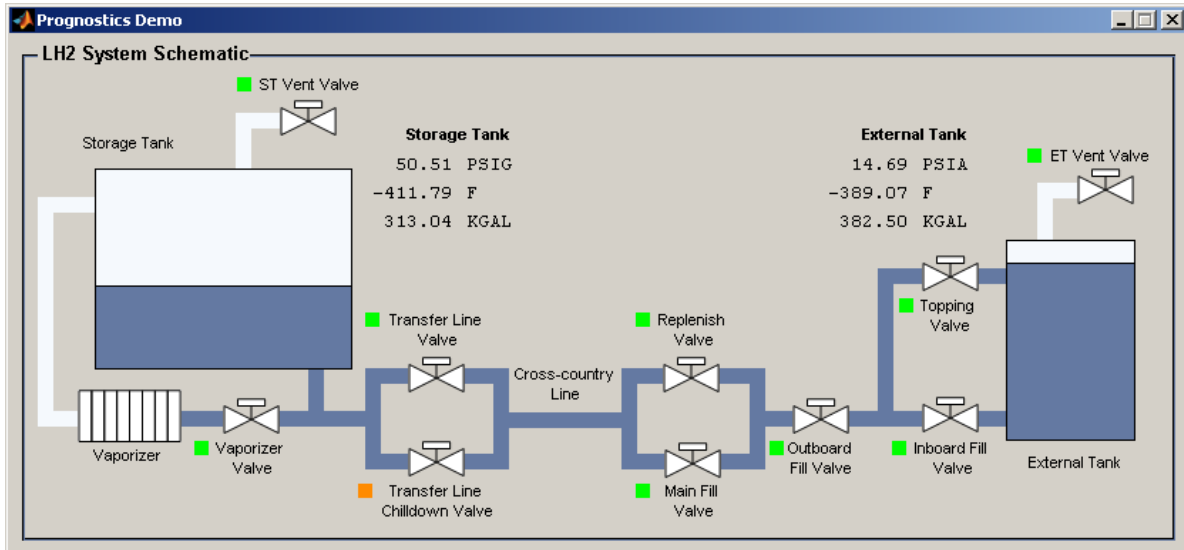


Figure 5. Top-level systems interface with prognostic indicators.

At the top level of the interface, shown in Fig. 5, is a system schematic integrated with sensor values. The only prognostics information communicated to the user at this level is the component health, shown using a simple color indicator with green indicating healthy, yellow indicating degraded health, and red indicating near-failure. This indicator changes continuously from green to yellow to red, in order to indicate the accumulation of component damage. When the indicator becomes yellow, this serves as a notification to the operator to investigate more closely. The next level of the interface, which is a component-centric view of prognostics results, allows this functionality.

The lower-level prognostics interface, shown in Figs. 6 and 7, shows a summary of component health, the current life predictions, and supportive information. The number of fuelings the component has undergone, along with the overall health and RUL, are shown in text. The RUL is plotted on the left side, with the mean with confidence intervals shown in the figure. Here, multiple views are supported, including plots for the RUL at different confidence levels, the EOL mean with confidence intervals, the RUL probability distribution, the EOL probability distribution, and an  $\alpha$ - $\lambda$  plot<sup>15</sup> that is available once the component has failed. Here, color is used to indicate the severity of the predictions, as the colors of the points are selected based on operational RUL thresholds. For example, the points are yellow when less than 15 remaining fuelings are predicted, as

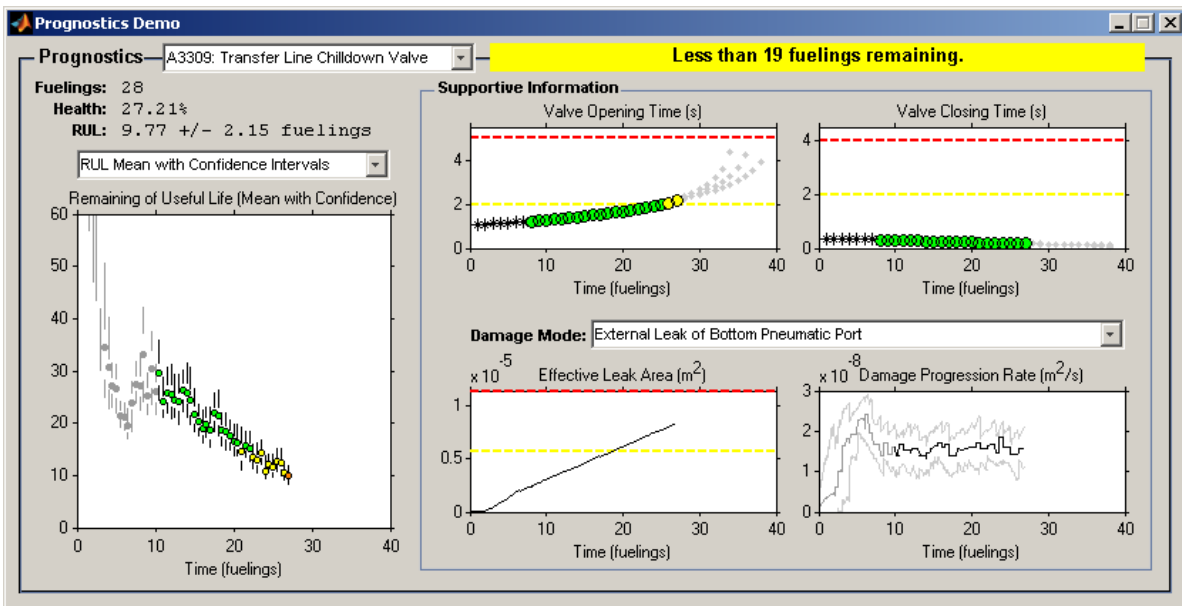


Figure 6. Lower-level prognostics interface.

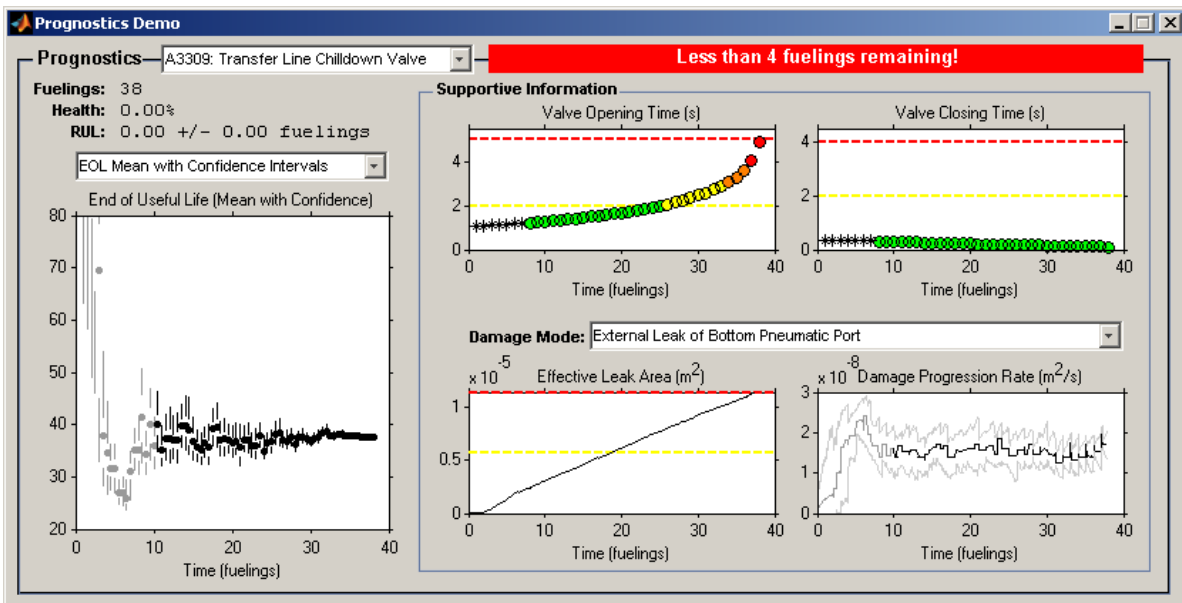


Figure 7. Lower-level prognostics interface at component failure.

at this point an operator might start to plan for future maintenance. The predictions up to the 10th fueling are shown in gray, as the algorithm has not yet converged until then in this example.

The interface provides supportive information to the right of the RUL plot. At the top of this portion is shown the measured valve timing, which in this case is the only directly observable information for the selected component. The opening and closing time requirements are shown with red dashed lines. The expected future valve times are shown with the trajectories consisting of gray dots, with the mean, maximum, and minimum timing values shown. These timing predictions help to explain why the algorithm predicts EOL at around 38 fuelings, because extrapolating the times linearly would yield a much larger EOL. Below the timing plots, the operator may select a specific damage mode, and view the estimated amount of damage and the estimated progression rate. Damage thresholds corresponding to EOL are shown with red dashed lines.

Prognostics Report (Demo)						
Component	Health (%)	RUL (95%)	Open (s)	Open Req. (s)	Close (s)	Close Req. (s)
Transfer Line Valve	95.18	137	13.12	26	9.90	20
Transfer Line Chillover Valve	0.00	0	4.89	5	0.10	4
Replenish Valve	98.23	89	4.99	10	3.59	8
Main Fill Valve	85.78	54	5.00	10	3.81	8
Outboard Fill Valve	79.00	101	4.26	10	4.14	8
Inboard Fill Valve	88.12	89	5.13	10	4.24	8
Topping Valve	98.78	74	5.14	10	4.10	8
Vaporizer Valve	95.44	36	5.29	10	3.60	7
ST Vent Valve	97.70	88	6.29	14	6.79	14
ET Vent Valve	93.33	71	5.21	10	3.68	8
Component	Friction Dmg. (%)	Spring Dmg. (%)	Int. Leak Dmg. (%)	Top Ext. Leak Dmg. (%)	Bottom Ext. Leak Dmg. (%)	
Transfer Line Valve	1.00	9.00	0.34	5.78	9.00	
Transfer Line Chillover Valve	2.79	0.80	1.58	0.12	100.00	
Replenish Valve	0.34	5.78	9.00	7.70	5.78	
Main Fill Valve	5.78	6.18	8.12	3.33	6.18	
Outboard Fill Valve	9.00	7.70	5.78	1.00	7.70	
Inboard Fill Valve	8.12	3.33	6.18	2.05	3.33	
Topping Valve	5.78	1.00	7.70	0.34	1.00	
Vaporizer Valve	6.18	2.05	3.33	5.78	2.05	
ST Vent Valve	7.70	0.34	1.00	9.00	0.34	
ET Vent Valve	3.33	5.78	2.05	8.12	5.78	

Figure 8. System report of prognostics information.

A summary of prognostics information for multiple components, shown in Fig. 8, is also available. For each component, the overall health is shown, the RUL at the 95th percentile, and current opening and closing times, as well as the timing requirements. For each valve, the amount of damage for each damage mode is also shown. Damage modes with high damage percentages indicate where repairs will have to be performed. Such a report may be used to track the health of various components after each fueling operation.

## VII. Demonstration

We apply the prognostics methodology to our case study, the transfer line chilldown valve of the Space Shuttle LH2 refueling system, using historical valve degradation 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  $w_{e,b} = 3.0 \times 10^{-7} \text{ m}^2/\text{cycle}$ , up to EOL at 38 cycles, after which the valve fails to open within the 5 second limit. Fig. 9 shows the valve timing data used for the demonstration. Note that only the *open* and *closed* sensors are available here for prognosis, and we have established in previous work that prognostics may still be performed under this constraint.<sup>2</sup>

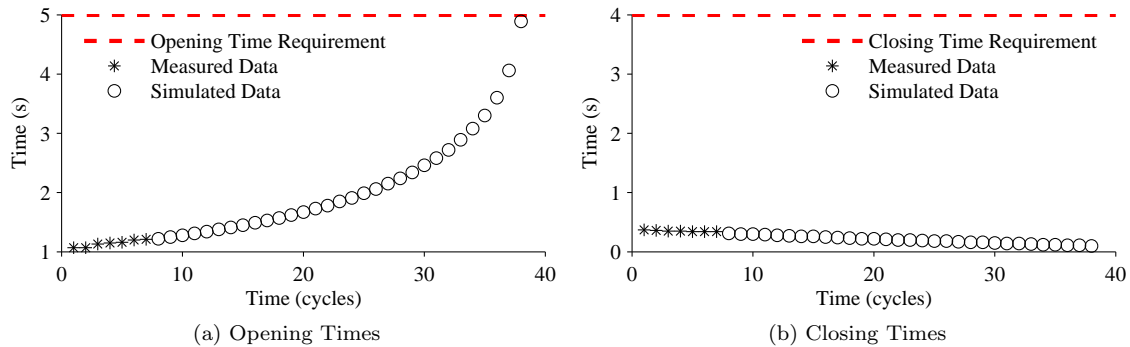


Figure 9. Real and simulated valve timing.

We assumed that no FDII algorithm was present to limit the number of damage modes to estimate, and therefore damage estimation was started immediately, jointly estimating all the damage modes of the valve. Because of the large state space, we set  $N = 2000$ . 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. For the variance control algorithm, we used  $T = 60\%$ ,  $v_0^* = 50\%$ ,

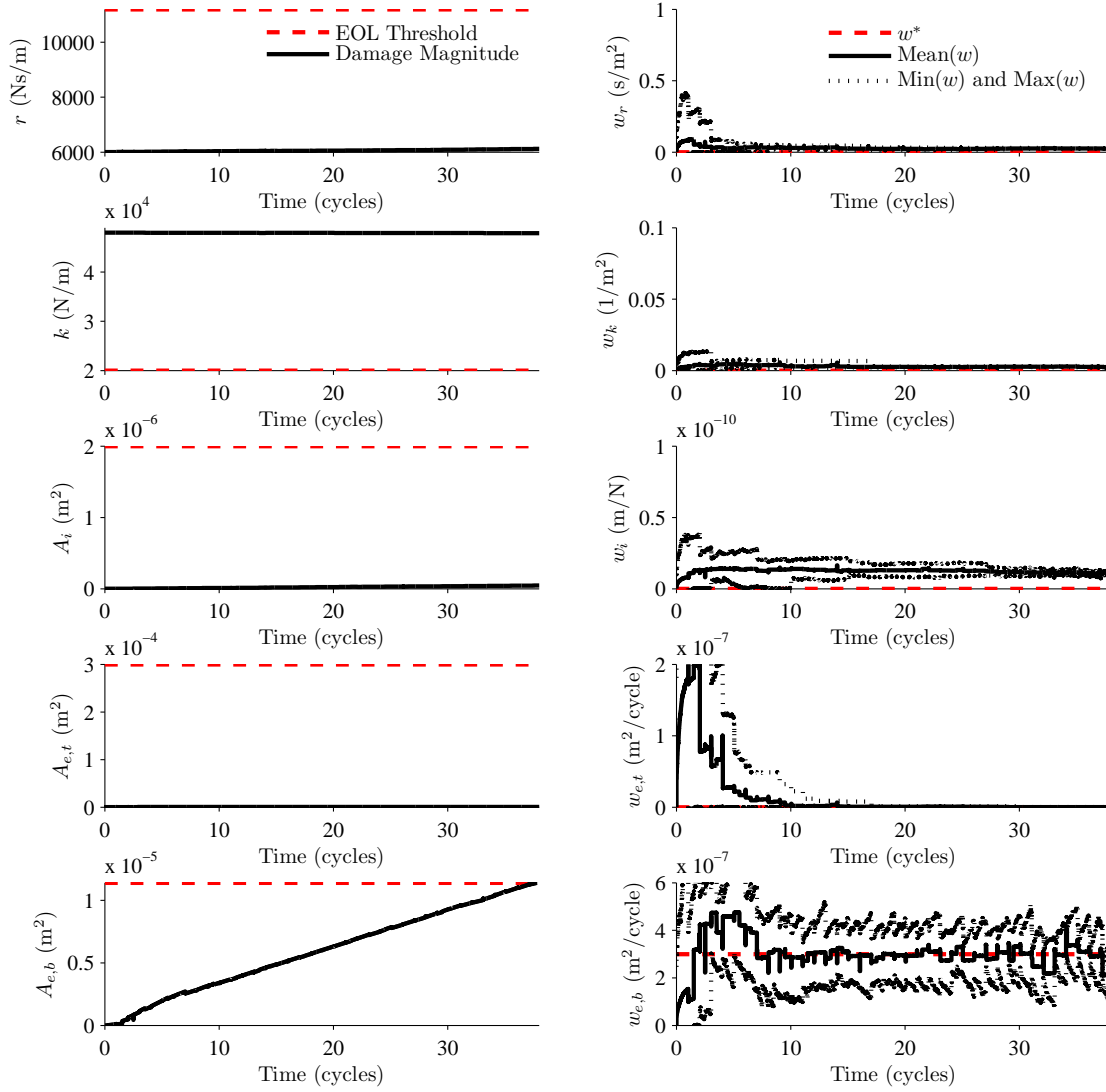


Figure 10. Estimation of pneumatic valve damage modes.

$v_{\infty}^* = 10\%$ , and  $P = 2 \times 10^{-4}$ . The algorithm keeps the RMAD of the wear parameters controlled to within 10%. The estimation algorithm correctly identifies the bottom external leak as the dominant damage mode, as shown by the convergence of its wear rate to the true value. The estimates for all other wear rates are driven to zero, and so the corresponding damage modes do not grow significantly. During the first 7 cycles (where measured data is available), the algorithm is still converging, and by the time this portion of the data has ended, the algorithm estimates the wear parameter for the bottom external leak within 30% of the true value, and the mean RUL prediction is within 25% of the true RUL. The predicted value of the RUL at that time point is less than the true value, resulting in a conservative estimate. By 8 cycles, the mean RUL prediction has improved to within 5% of the true RUL. At this point the algorithm has converged and the user may then begin to use the results in the decision-making process.

Fig. 11 shows the RUL 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. After algorithm convergence, the predictions are fairly smooth. The relative accuracy of the RUL, using the mean values, averages 91% over all predictions, and the RMAD of the RUL averages 9.88%.

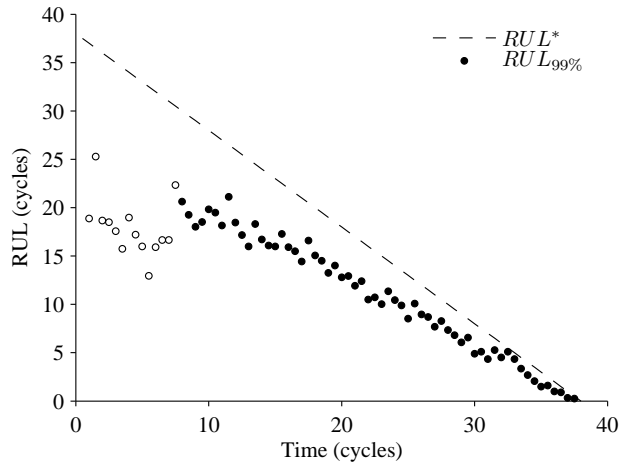


Figure 11. RUL predictions at the 99% confidence level. Predictions before algorithm convergence are represented by unfilled circles, and predictions after convergence by filled circles.

## VIII. 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 RUL distributions. We adopted a pneumatic valve from the Space Shuttle refueling system as a case study. We developed a detailed physics-based model that includes damage progression, and applied the algorithmic approach to historical valve degradation data augmented with simulation data to EOL. A prototype user interface was also developed to communicate the prognostics results to a system operator. The results here demonstrated the effectiveness of a model-based approach for valve 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

- <sup>1</sup>Schwabacher, M., "A survey of data-driven prognostics," *Proceedings of the AIAA Infotech@ Aerospace Conference*, 2005.
- <sup>2</sup>Daigle, M. and Goebel, K., "Model-based prognostics under limited sensing," *Proceedings of the 2010 IEEE Aerospace Conference*, March 2010.
- <sup>3</sup>Saha, B. and Goebel, K., "Modeling Li-ion battery capacity depletion in a particle filtering framework," *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009*, Sept. 2009.
- <sup>4</sup>Chelidze, D., "Multimode damage tracking and failure prognosis in electromechanical system," *Proceedings of the SPIE Conference*, Vol. 4733, 2002, pp. 1–12.
- <sup>5</sup>Luo, J., Pattipati, K. R., Qiao, L., and Chigusa, S., "Model-Based Prognostic Techniques Applied to a Suspension System," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 38, No. 5, Sept. 2008, pp. 1156–1168.
- <sup>6</sup>Daigle, M. and Goebel, K., "Multiple damage progression paths in model-based prognostics," *Proceedings of the 2011 IEEE Aerospace Conference*, March 2011.
- <sup>7</sup>Perry, R. and Green, D., *Perry's chemical engineers' handbook*, McGraw-Hill Professional, 2007.
- <sup>8</sup>Hutchings, I. M., *Tribology: friction and wear of engineering materials*, CRC Press, 1992.
- <sup>9</sup>Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 174–188.
- <sup>10</sup>Cappe, O., Godsill, S. J., and Moulines, E., "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, Vol. 95, No. 5, 2007, pp. 899.
- <sup>11</sup>Orchard, M., *A Particle Filtering-based Framework for On-line Fault Diagnosis and Failure Prognosis*, Ph.D. thesis, Georgia Institute of Technology, 2007.

<sup>12</sup>Daigle, M. and Goebel, K., “Model-based prognostics with fixed-lag particle filters,” *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2009*, Sept. 2009.

<sup>13</sup>Kitagawa, G., “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, Vol. 5, No. 1, 1996, pp. 1–25.

<sup>14</sup>Doucet, A., Godsill, S., and Andrieu, C., “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, Vol. 10, 2000, pp. 197–208.

<sup>15</sup>Saxena, A., Celaya, J., Saha, B., Saha, S., and Goebel, K., “Metrics for Offline Evaluation of Prognostic Performance,” *International Journal of Prognostics and Health Management (IJPHM)*, Vol. 1, 2010.