

Model-based Prognostics with Concurrent Damage Progression Processes

Matthew J. Daigle, *Member, IEEE*, and Kai Goebel, *Member, IEEE*

Abstract—Model-based prognostics approaches rely on physics-based models that describe the behavior of systems and their components. These models must account for the several different damage processes occurring simultaneously within a component. Each of these damage and wear processes contribute to the overall component degradation. We develop a model-based prognostics methodology that consists of a joint state-parameter estimation problem, in which the state of a system along with parameters describing the damage progression are estimated, followed by a prediction problem, in which the joint state-parameter estimate is propagated forward in time to predict end of life and remaining useful life. The state-parameter estimate is computed using a particle filter, and is represented as a probability distribution, allowing the prediction of end of life and remaining useful life within a probabilistic framework that supports uncertainty management. We also develop a novel variance control algorithm that maintains an uncertainty bound around the unknown parameters to limit the amount of estimation uncertainty and, consequently, reduce prediction uncertainty. We construct a detailed physics-based model of a centrifugal pump that includes damage progression models, to which we apply our model-based prognostics algorithm. We illustrate the operation of the prognostic solution with a number of simulation-based experiments and demonstrate the performance of the approach when multiple damage mechanisms are active.

Index Terms—model-based prognostics, particle filters, variance control, centrifugal pumps

I. INTRODUCTION

Systems health management is integral to ensuring system safety while meeting system objectives. Traditionally, health management has consisted solely of fault detection and diagnosis. More recently, the extension to fault prognosis has become an important area of research. Prognostics is focused on predicting when a fault, damage, or wear of a component, subsystem, or system will progress to a point that is deemed unsafe, or in which the system does not function as specified. This time point is called end of useful life (EOL), and the time remaining until that point is called remaining useful life (RUL). With accurate predictions of EOL/RUL, future maintenance activities can be optimally planned, and component or system life can be extended by modifying its workload [1], [2].

Model-based prognostics approaches employ domain knowledge about a system, its components, and how they fail through the use of physics-based models that capture the

underlying physical phenomena [3]–[6]. Component damage progresses along several different dimensions, driven by different degradation phenomena. Further, due to manufacturing variances and differences in usage and environmental conditions, the damage progression rates for the different damage mechanisms vary among components of the same type. This poses considerable challenges to data-driven algorithms, because they try to generalize damage progression over a large set of cases [7]. Therefore, model-based approaches, which adapt online to the specific component being monitored, are able to obtain predictions with much higher accuracy, given correct models.

A model-based approach thus requires the development of models capturing the possible damage progression processes. During online monitoring, they must track both the state and unknown parameters whose values are unique to the specific component, obtaining a joint state-parameter estimate that may be used for prediction. Extending previous work in [3] and preliminary results in [8], we develop a model-based prognostics methodology that computes the joint state-parameter estimate using particle filters. The estimate is represented as a probability distribution, allowing the prediction of EOL and RUL within a probabilistic framework that supports uncertainty management. The use of particle filters for prognostics is not new (e.g., [4], [6]), and has been validated in real systems [9], however, previous approaches are developed for tracking only a single damage mode, whereas in this paper, we present a more general framework that can handle multiple damage modes simultaneously.

In particle filter-based parameter estimation, an artificial random walk evolution is assigned to the unknown parameters, which is necessary for convergence of the parameter estimates and proper tracking afterwards. But, the optimal variance of the random walk depends on the actual parameter value, which is unknown. As previously recognized [10]–[13], it is desirable to tune the value of this variance online in order to promote quick convergence of the parameter estimate and ensure a small tracking variance. To reduce the amount of this artificial uncertainty, we introduce a novel variance control algorithm, differing significantly from these previous techniques, that maintains an uncertainty bound around an unknown parameter being estimated, allowing the particle filter to tune itself online to improve performance.

We demonstrate our prognostics methodology on a centrifugal pump. Centrifugal pumps are used in a wide range of applications, from water supply to spacecraft fueling systems. Because pumps typically see high usage, they can particularly benefit from prognostics and health management solutions to

Manuscript received September 6, 2011; revised March 28, 2012. This work was supported by the NASA Fault Detection, Isolation, and Recovery (FDIR) and System-wide Safety and Assurance Technologies (SSAT) projects.

M. Daigle and K. Goebel are with NASA Ames Research Center, Moffett Field, CA 94035 USA (e-mail: matthew.j.daigle@nasa.gov, kai.goebel@nasa.gov).

ensure system performance, extended component lifetime, and limited downtime. We develop a detailed physics-based model of a centrifugal pump used for cryogenic spacecraft propellant loading that includes models of the most significant damage progression processes. With the centrifugal pump as a case study, we apply our approach to a number of simulation-based experiments under concurrent damage progression processes. We evaluate algorithm performance using established prognostics metrics [14].

The paper is organized as follows. Section II formally defines the prognostics problem and describes the prognostics architecture. Section III describes the modeling methodology and develops the centrifugal pump model for prognostics. Section IV describes the particle filter-based damage estimation method and develops the variance control scheme. Section V discusses the prediction methodology. Section VI provides results from a number of simulation-based experiments and evaluates the approach. Section VII describes related work, and Section VIII concludes the paper.

II. PROGNOSTICS APPROACH

Prediction is a general problem, but, in systems health management, we are interested in a particular kind of prediction, namely, EOL. The goal of fault prognostics is to predict when the EOL of a system of interest is reached. In this section, we first formally define the prognostics problem. We then describe a general model-based prognostics architecture.

A. Problem Formulation

We assume the system model may be generally defined as

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

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 is a general nonlinear model with no restrictions on the functional forms of \mathbf{f} or \mathbf{h} , or on how the noise terms are coupled with the states and parameters. The parameters $\boldsymbol{\theta}(t)$ evolve in an unknown way, but are typically considered to be constant in practice.

In prognostics, we are interested in when the performance of a system lies outside some desired region of acceptable behavior. Outside this region, we say that the system has failed. The desired performance is expressed through a set of c constraints, $\mathcal{C} = \{C_i\}_{i=1}^c$, where C_i is a function

$$C_i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{B}$$

that maps a given point in the joint state-parameter space, $(\mathbf{x}(t), \boldsymbol{\theta}(t))$, to the Boolean domain $\mathbb{B} \triangleq [0, 1]$, where $C_i(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$ if the constraint is satisfied. If $C_i(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 0$, then the constraint is not satisfied and the system has failed. For example, a constraint may require that a crack size is less than some critical value, or a valve opens within some specified time limit.

These individual constraints may be combined into a single threshold function T_{EOL} , where

$$T_{EOL} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \rightarrow \mathbb{B},$$

defined as

$$T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = \begin{cases} 1, & 0 \in \{C_i(\mathbf{x}(t), \boldsymbol{\theta}(t))\}_{i=1}^c \\ 0, & \text{otherwise.} \end{cases}$$

That is, T_{EOL} evaluates to 1, i.e., the system has failed, when any of the constraints are violated. This threshold defines an acceptable region of the joint state-parameter space, \mathcal{A} , that satisfies the performance constraints, i.e.,

$$\mathcal{A} = \{(\mathbf{x}(t), \boldsymbol{\theta}(t)) : T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 0\}.$$

At some point in time, t_P , the system is at $(\mathbf{x}(t_P), \boldsymbol{\theta}(t_P))$ and we are interested in predicting the time point t at which this state will evolve to $(\mathbf{x}(t), \boldsymbol{\theta}(t))$ such that $T_{EOL}(\mathbf{x}(t), \boldsymbol{\theta}(t)) = 1$, i.e., the time point at which the system exits region \mathcal{A} . Using T_{EOL} , we formally define EOL with

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

i.e., EOL is the earliest time point at which T_{EOL} is met. RUL is expressed using EOL as

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

Problem (Fault Prognostics). The fault prognostics problem is to, at prediction time t_P , compute $EOL(t_P)$ and/or $RUL(t_P)$.

Fig. 1 describes these concepts with a two-dimensional example where $\mathbf{x}(t) = [x_1(t) \ x_2(t)]^T$ and $\boldsymbol{\theta} = \emptyset$. Initially, at time t_0 , the system is at some state $\mathbf{x}(t_0)$. It then evolves to some point $\mathbf{x}(t_P)$ at the present time t_P . As time progresses, the system evolves along some trajectory within the joint state-parameter space. Eventually, the system will reach a point at time t , $\mathbf{x}(t)$, that does not belong to \mathcal{A} , and it is this time point that is EOL and must be predicted. In general, the system may follow a complex path in the multi-dimensional joint state-parameter space. In the end, a systems health management framework may attempt to alter the path the system takes within \mathcal{A} in order to extend the life to some alternate time $t' > t$, as shown in the figure, e.g., by reducing the system workload.

B. Prognostics Architecture

In order to predict EOL/RUL, we require the current joint state-parameter description at prediction time t_P , $(\mathbf{x}(t_P), \boldsymbol{\theta}(t_P))$, and the system inputs $\mathbf{u}(t)$ for all $t \geq t_P$. However, there are several issues that make this problem difficult. First, $(\mathbf{x}(t_P), \boldsymbol{\theta}(t_P))$ is not known exactly because we measure only $\mathbf{y}(t_P)$ and these measurements are corrupted by the sensor noise $\mathbf{n}(t_P)$. So, we can only compute a probability distribution $p(\mathbf{x}_{t_P}, \boldsymbol{\theta}_{t_P} | \mathbf{y}_{0:t_P})$ that is estimated based on the history of measurements up to t_P , $\mathbf{y}_{0:t_P}$. Second, the process noise $\mathbf{v}(t)$ will corrupt the evolution of $(\mathbf{x}(t_P), \boldsymbol{\theta}(t_P))$ for $t > t_P$. Third, the future inputs of the system are usually uncertain. So, at best, we can obtain only a probability distribution of EOL or RUL, i.e., $p(EOL(t_P) | \mathbf{y}_{0:t_P})$ or $p(RUL(t_P) | \mathbf{y}_{0:t_P})$.

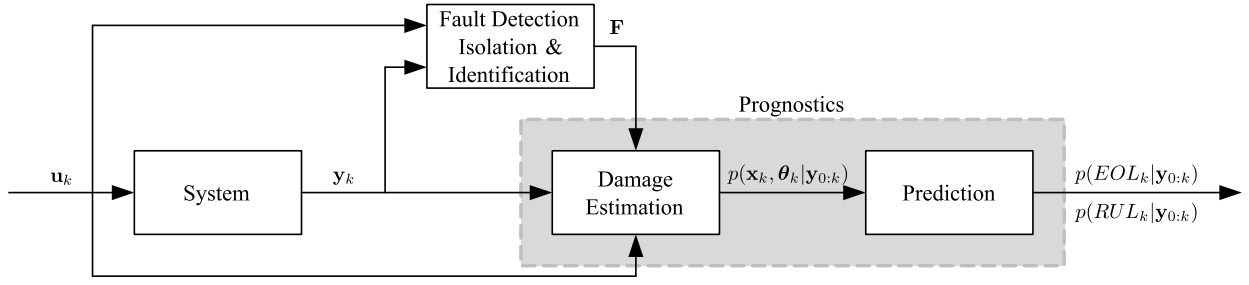


Fig. 2. Prognostics architecture.

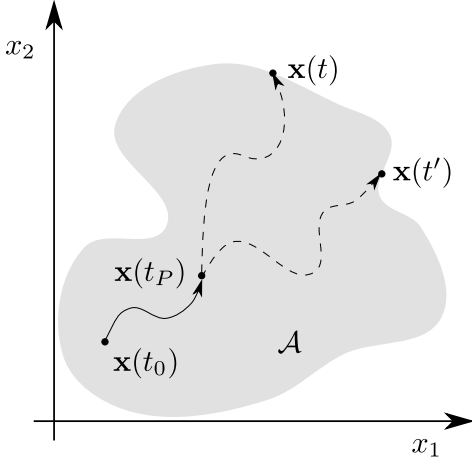


Fig. 1. Conceptual two-dimensional example of system trajectories to EOL.

These issues lead to the prognostics architecture shown in Fig. 2. In discrete time k , the system is provided with inputs \mathbf{u}_k and provides measured outputs \mathbf{y}_k . The damage estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k})$, represented as a probability distribution. The prediction module uses the joint state-parameter distribution and the system model, 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 many cases, a fault detection, isolation, and identification (FDII) module may be used in parallel 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 dimension of the joint state-parameter space that must be estimated. In this paper, we focus on the damage estimation and prediction modules, and assume that prognostics begins at $t = 0$ and that the FDII module does not inform the prognostics, i.e., all possible damage progression paths must be tracked starting from $t = 0$.

The scope of the prognostics application may be an entire system, a subsystem, or a single component. The problem formulation and architecture are general enough to consider any given scope. In this paper, we limit the scope to a single component.

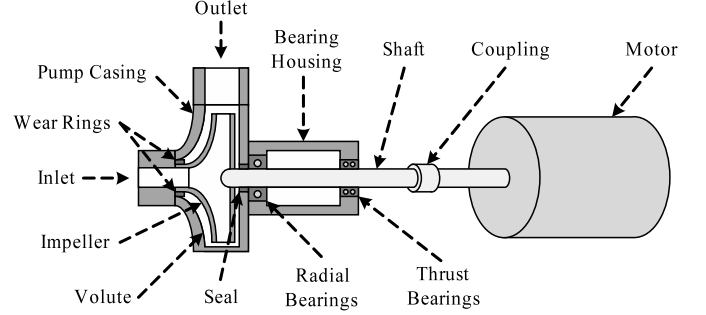


Fig. 3. Centrifugal pump.

III. PUMP MODELING

In order to apply the model-based prognostics architecture, we must develop a model of the system under consideration. This includes identifying the state vector $\mathbf{x}(t)$, the parameter vector $\boldsymbol{\theta}(t)$, the output vector $\mathbf{y}(t)$, the state equation \mathbf{f} , the output equation \mathbf{h} , and the set of performance constraints \mathcal{C} . In our modeling methodology, we first describe a nominal model of system behavior. We then extend the model by including *damage progression functions* within the state equation \mathbf{f} that describe how *damage variables* $\mathbf{d}(t) \subseteq \mathbf{x}(t)$ evolve over time. The damage progression functions are parameterized by unknown and possibly time-varying *wear parameters* $\mathbf{w}(t) \subseteq \boldsymbol{\theta}(t)$. We use a centrifugal pump as a case study. In this section, we first describe the nominal model of the pump, and then describe its damage progression models.

A. Nominal Model

Centrifugal pumps are used in a variety of domains for fluid delivery. We develop a model of a pump used for cryogenic spacecraft propellant (liquid oxygen) loading located at Kennedy Space Center. A schematic of a typical centrifugal pump is shown in Fig. 3. Fluid enters the inlet, and the rotation of the impeller forces fluid through the outlet. The impeller is driven by an electric motor, typically a three-phase alternating-current induction motor. The radial and thrust bearings help to minimize friction along the pump shaft. The bearing housing contains oil which lubricates the bearings. A seal prevents fluid flow into the bearing housing. Wear rings prevent internal pump leakage from the outlet to the inlet side of the impeller, but a small clearance is typically allowed to minimize friction (a small internal leakage is normal).

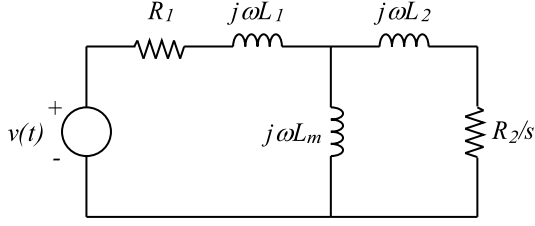


Fig. 4. Induction motor equivalent circuit.

The state of the pump is described by

$$\mathbf{x}(t) = [\omega(t) \quad T_t(t) \quad T_r(t) \quad T_o(t)]^T,$$

where $\omega(t)$ is the rotational velocity of the pump, $T_t(t)$ is the thrust bearing temperature, $T_r(t)$ is the radial bearing temperature, and $T_o(t)$ is the oil temperature.

The rotational velocity of the pump is described using a torque balance,

$$\dot{\omega} = \frac{1}{J} (\tau_e(t) - r\omega(t) - \tau_L(t)),$$

where J is the lumped motor/pump inertia, τ_e is the electromagnetic torque provided by the motor, r is the lumped friction parameter, and τ_L is the load torque. In an induction motor, a voltage is applied to the stationary part, the stator, which creates a current through the stator coils. With a polyphase supply, this creates a rotating magnetic field that induces a current in the rotating part, the rotor, causing it to turn. A torque is produced on the rotor only when there is a difference between the synchronous speed of the supply voltage, ω_s and the mechanical rotation, ω . This difference, called *slip*, is defined as

$$s = \frac{\omega_s - \omega}{\omega_s}.$$

The expression for the torque τ_e is derived from an equivalent circuit representation for the three-phase induction motor, shown in Fig. 4, based on rotor and stator resistances and inductances, and the slip s [15]:

$$\tau_e = \frac{npR_2}{s\omega_s} \frac{V_{rms}^2}{(R_1 + R_2/s)^2 + (\omega_s L_1 + \omega_s L_2)^2},$$

where R_1 is the stator resistance, L_1 is the stator inductance, R_2 is the rotor resistance, L_2 is the rotor inductance, n is the number of phases (typically 3), and p is the number of magnetic pole pairs. For a 3600 rpm motor, $p = 1$. The dependence of torque on slip creates a feedback loop that causes the rotor to follow the rotation of the magnetic field. The rotor speed may be controlled by changing the input frequency ω_s , e.g., through the use of a variable-frequency drive.

The load torque τ_L is a polynomial function of the flow rate through the pump and the impeller rotational velocity [16], [17]:

$$\tau_L = a_0\omega^2 + a_1\omega Q - a_2Q^2,$$

where Q is the flow, and a_0 , a_1 , and a_2 are coefficients derived from the pump geometry [17].

The rotation of the impeller creates a pressure difference from the inlet to the outlet of the pump, which drives the pump flow, Q . The pump pressure is computed as

$$p_p = b_0\omega^2 + b_1\omega Q - b_2Q^2,$$

where b_0 , b_1 , and b_2 are coefficients derived from the pump geometry. The parameter b_0 is proportional to impeller area A [18]. Flow through the impeller, Q_i , is computed using the pressure differences:

$$Q_i = c\sqrt{|p_s + p_p - p_d| \text{sign}(p_s + p_p - p_d)},$$

where c is a flow coefficient, p_s is the suction pressure, and p_d is the discharge pressure. The small (normal) leakage flow from the discharge end to the suction end due to the clearance between the wear rings and the impeller is described by

$$Q_l = c_l\sqrt{|p_d - p_s| \text{sign}(p_d - p_s)},$$

where c_l is a flow coefficient. The discharge flow, Q , is then

$$Q = Q_i - Q_l.$$

For the particular pump under consideration, pump temperatures are monitored as indicators of pump condition. The oil heats up due to the radial and thrust bearings and cools to the environment:

$$\dot{T}_o = \frac{1}{J_o} (H_{o,1}(T_t - T_o) + H_{o,2}(T_r - T_o) - H_{o,3}(T_o - T_a)),$$

where J_o is the thermal inertia of the oil, the $H_{o,i}$ terms are heat transfer coefficients, and T_a is the ambient temperature. The thrust bearings heat up due to the friction between the pump shaft and the bearings, and cool to the oil and the environment:

$$\dot{T}_t = \frac{1}{J_t} (r_t\omega^2 - H_{t,1}(T_t - T_o) - H_{t,2}(T_t - T_a)),$$

where J_t is the thermal inertia of the thrust bearings, r_t is the friction coefficient for the thrust bearings, and the $H_{t,i}$ terms are heat transfer coefficients. The radial bearings behave similarly:

$$\dot{T}_r = \frac{1}{J_r} (r_r\omega^2 - H_{r,1}(T_r - T_o) - H_{r,2}(T_r - T_a))$$

where J_r is the thermal inertia of the radial bearings, r_r is the friction coefficient for the radial bearings, and the $H_{r,i}$ terms are heat transfer coefficients. Note that r_t and r_r contribute to the overall friction coefficient r .

The overall input vector \mathbf{u} is given by

$$\mathbf{u}(t) = [p_s(t) \quad p_d(t) \quad T_a(t) \quad V(t) \quad \omega_s(t)]^T.$$

The measurement vector \mathbf{y} is given by

$$\mathbf{y}(t) = [\omega(t) \quad Q(t) \quad T_t(t) \quad T_r(t) \quad T_o(t)]^T.$$

Fig. 5 shows nominal pump operation, with the parameters (estimated from pump data) given in Table I. The input voltage (and frequency) are varied to control the pump speed. The electromagnetic torque is produced initially as slip is 1. This causes a rotation of the motor to match the rotation of the magnetic field, with a small amount of slip remaining, depending on how large the load torque is. As the pump rotates, fluid flow is created. The bearings heat up as the pump rotates and cool when the pump rotation slows.

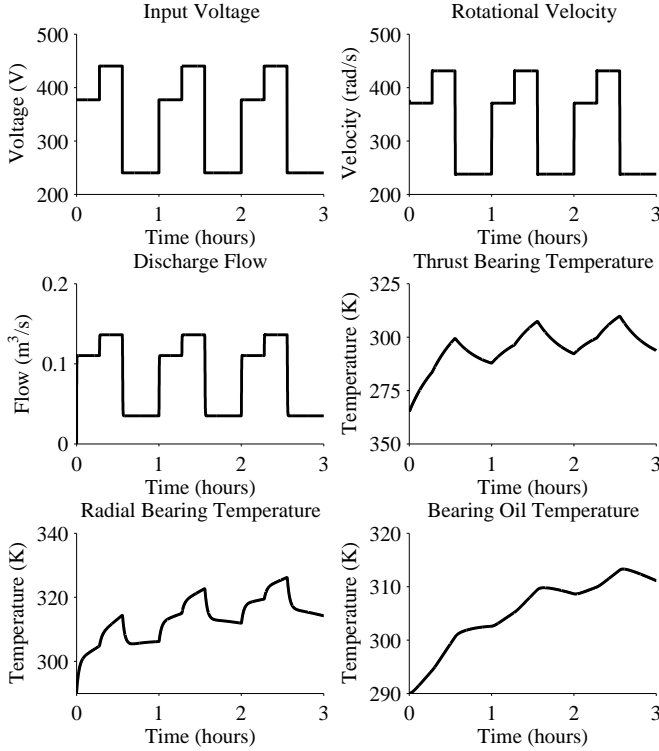


Fig. 5. Nominal pump operation.

TABLE I
NOMINAL PUMP PARAMETERS

Parameter	Value
J	50 kg m^2
r	$8.0 \times 10^{-3} \text{ N m s}$
n	3 phases
p	1 pole pair
R_1	$3.6 \times 10^{-1} \Omega$
R_2	$7.6 \times 10^{-2} \Omega$
$L_1 + L_2$	$6.3 \times 10^{-4} \text{ H}$
a_0	$1.5 \times 10^{-3} \text{ kg m}^2$
a_1	5.8 kg/m
a_2	$9.2 \times 10^3 \text{ kg/m}^4$
b_0	12.7 kg/m
b_1	$1.8 \times 10^4 \text{ kg/m}^4$
b_2	0 kg/m^7
c	$8.2 \times 10^{-5} \text{ m}^{7/2}/\text{kg}^{1/2}$
c_l	$1.0 \times 10^{-10} \text{ m}^{7/2}/\text{kg}^{1/2}$
J_o	$8.0 \times 10^3 \text{ K/J/s}$
$H_{o,1}$	1.0 W/K
$H_{o,2}$	3.0 W/K
$H_{o,3}$	1.5 W/K
J_t	7.3 K/J/s
r_t	$1.4 \times 10^{-6} \text{ N m s}$
$H_{t,1}$	$3.4 \times 10^{-3} \text{ W/K}$
$H_{t,2}$	$2.6 \times 10^{-2} \text{ W/K}$
J_r	2.4 K/J/s
r_r	$1.8 \times 10^{-6} \text{ N m s}$
$H_{r,1}$	$1.8 \times 10^{-3} \text{ W/K}$
$H_{r,2}$	$2.0 \times 10^{-2} \text{ W/K}$

B. Damage Modeling

The performance constraints of the pump are specified by efficiency and temperature limits. The first constraint, C_1 ,

is that the efficiency $\eta > 0.75\eta_0$, where η_0 is the nominal efficiency. Efficiency is defined as the input electrical power divided by the output hydraulic power, i.e., $\eta = (VI)/((p_d - p_s)Q)$. The remaining constraints are limits on the temperatures:

$$\begin{aligned} C_2 : T_o(t) &< T_o^+ \\ C_3 : T_t(t) &< T_t^+ \\ C_4 : T_r(t) &< T_r^+, \end{aligned}$$

where the $+$ superscript denotes the maximum allowable temperature. When the maximum temperatures are reached, irreversible damage occurs. Here, for the pump under consideration, $T_o^+ = 333 \text{ K}$ and $T_t^+ = T_r^+ = 370 \text{ K}$.

The most significant damage mechanism for pumps is impeller wear. It is represented as a decrease in impeller area A [18], [19]. Since the impeller area is proportional to b_0 , a decrease causes a decrease in the pump pressure, and, hence, the pump efficiency. We use the erosive wear equation [20] to describe how the impeller area changes over time. The erosive wear rate is proportional to fluid velocity times friction force. Fluid velocity is proportional to volumetric flow rate, and friction force is proportional to fluid velocity. We lump the proportionality constants into the wear coefficient w_A to obtain

$$\dot{A}(t) = -w_A Q_i(t)^2.$$

Because A is proportional to b_0 , then $\dot{b}_0(t) = k\dot{A}(t) = -kw_A Q_i(t)^2$, so we estimate $b_0(t)$ and $w_{b_0} = kw_A$.

Another significant damage mechanism for pumps is bearing wear, which is captured as an increase in the friction coefficient. Sliding and rolling friction generate wear of material which increases the coefficient of friction [3], [20]:

$$\begin{aligned} \dot{r}_t(t) &= w_t r_t \omega^2 \\ \dot{r}_r(t) &= w_r r_r \omega^2, \end{aligned}$$

where w_t and w_r are the wear coefficients. The slip compensation provided by the electromagnetic torque generation masks small changes in friction, so it is only with very large increases that a change in ω will be observed. Changes in friction manifest more strongly in the bearing temperatures, eventually driving them to the temperature limits.

So, the damage variables are given by

$$\mathbf{d}(t) = [b_0(t) \quad r_t(t) \quad r_r(t)]^T,$$

and the full state vector becomes

$$\mathbf{x}(t) = [\omega(t) \quad T_t(t) \quad T_r(t) \quad T_o(t) \quad b_0(t) \quad r_t(t) \quad r_r(t)]^T.$$

The initial conditions for the damage variables are given in Table I. The wear parameters form the unknown parameter vector, i.e.,

$$\mathbf{w}(t) = \boldsymbol{\theta}(t) = [w_{b_0} \quad w_t \quad w_r]^T.$$

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 $p(\mathbf{x}_k, \boldsymbol{\theta}_k | \mathbf{y}_{0:k})$. A general solution to this problem is the *particle filter*, which may be directly applied to nonlinear systems with non-Gaussian noise terms [21]. In particle filters, the state distribution is approximated by a set of discrete weighted samples, called *particles*, i.e.,

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

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

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

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

To apply a filtering approach, some estimate of the process and sensor noise vectors, $\mathbf{v}(t)$ and $\mathbf{n}(t)$, must be determined. The distribution describing sensor noise can be estimated from the system measurements, and in practice is often assumed to be Gaussian. The process noise can be estimated by comparing measured system behavior with that predicted in the absence of process noise. Within a filtering framework, it is typically not critical to obtain very accurate estimates of the noise, because the amount of noise assumed by the filter is almost always tuned to improve performance.

We use the sampling importance resampling (SIR) particle filter, using systematic resampling [22]. The pseudocode for a single step of the SIR filter is shown as Algorithm 1. Each particle is propagated forward to time k by first sampling new parameter values, and then sampling new states using the model. The particle weight is assigned using \mathbf{y}_k . The weights are then normalized, followed by the resampling step [21].

Here, the parameters $\boldsymbol{\theta}_k$ evolve by some unknown process that is independent of the state \mathbf{x}_k . However, we need to assign some type of evolution to the parameters in order for the particle filter to estimate them. The typical solution is to use a random walk, i.e., $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \boldsymbol{\xi}_{k-1}$, where $\boldsymbol{\xi}_{k-1}$ is sampled from some distribution (e.g., zero-mean Gaussian). With this

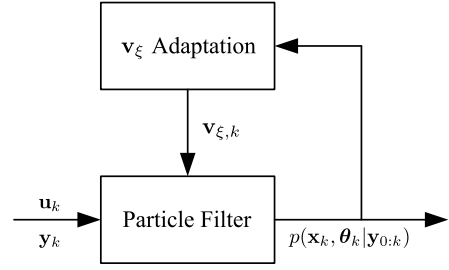


Fig. 6. \mathbf{v}_ξ adaptation scheme.

Algorithm 2 \mathbf{v}_ξ Adaptation

Inputs: $\{(\mathbf{x}_k^i, \boldsymbol{\theta}_k^i), w_k^i\}_{i=1}^N$
State: $\mathbf{v}_{\xi,k-1}, \mathbf{l} \leftarrow \mathbf{1}$
Outputs: $\mathbf{v}_{\xi,k}$
for all $j \in \{1, 2, \dots, n_\theta\}$ **do**
 $v_j \leftarrow \text{RMAD}(\{\theta_k^i(j)\}_{i=1}^N)$
if $v_j < \mathbf{t}_j(\mathbf{s}(j))$ **then**
 $\mathbf{s}(j) \leftarrow \mathbf{s}(j) + \mathbf{1}$
end if
 $\mathbf{v}_{\xi,k}(j) \leftarrow \mathbf{v}_{\xi,k-1}(j) \left(1 - \mathbf{P}_j(\mathbf{s}(j)) \frac{v_j - \mathbf{v}_j^*(\mathbf{s}(j))}{\mathbf{v}_j^*(\mathbf{s}(j))}\right)$
end for
 $\mathbf{v}_{\xi,k-1} \leftarrow \mathbf{v}_{\xi,k}$

type of evolution, the particles generated with parameter values closest to the true values should be assigned higher weight, thus allowing the particle filter to converge to the true values.

The selected variance of the random walk noise determines both the rate of this convergence and the estimation performance once convergence is achieved. Therefore, it is very desirable to tune this parameter to obtain the best possible performance. A large random walk variance will yield quick convergence but tracking with too wide a variance, whereas too small a random walk variance will yield a very slow convergence, if at all, but, once achieved, tracking will proceed with a very small variance. We develop an adaptation method for the variances of $\boldsymbol{\xi}$, denoted as \mathbf{v}_ξ , with the following features. First, we consider a multi-dimensional damage space, therefore, we must simultaneously adapt the random walk noise for multiple parameter values. Second, we cannot use prediction error to drive the adaptation, because we cannot, in general, map errors in prediction to specific wear parameters, since each output is dependent on multiple damage mechanisms. Instead, we try to control the variance of the hidden wear parameter estimate to a user-specified range by modifying the random walk noise variance. Since the random walk noise is artificial, we should reduce it as much as possible, because this uncertainty propagates into the EOL predictions. So, controlling this uncertainty helps to control the uncertainty of the EOL prediction. Reducing the variance of the wear parameter can reduce the variance of the EOL prediction by several factors, and the improvement is substantial over long time horizons.

The algorithm for the adaptation of the \mathbf{v}_ξ vector is given as Algorithm 2, and Fig. 6 shows how it interacts with the particle filter. We assume that the distributions that the elements of $\boldsymbol{\xi}$ are drawn from can be specified using a variance value,

and that the variance values are tuned initially based on the maximum expected wear rates, e.g., if the pump is expected to fail no earlier than 100 hours, then this corresponds to particular maximum wear rate values. The initial wear rate estimate values may start at 0. We use the relative median absolute deviation (RMAD) as the measure of variance:

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

where X is a data set and X_i is an element of that set. We use RMAD because it is statistically robust, and, since it is a *relative* measure of spread, it can be treated equally for any wear parameter value.

The adaptation proceeds in multiple stages, maintained with an s_j variable for each parameter (with j referring to the parameter index), with the number of stages specified by S_j . The s_j values are initialized to 1. Each stage is specified using three variables, (i) a lower threshold that, once crossed, signals that the next stage should be entered, (ii) the desired RMAD value for the stage, and (iii) a proportional gain term used to control the degree of adaptation during that stage. For each parameter, the threshold vector \mathbf{t}_j , the desired RMAD vector \mathbf{v}_j^* , and the proportional gain vector \mathbf{P}_j are of size S_j .

The algorithm works as follows. For each parameter, indexed by j , the current RMAD is computed as v_j . If this value is below the threshold value for the current stage, $\mathbf{t}_j(s(j))$, then the stage number is increased. Then the new random walk variance $\mathbf{v}_{\xi,k}(j)$ is computed. The error between the actual and the desired RMAD value for the current stage, $v_j - \mathbf{v}_j^*(s(j))$, is normalized by $\mathbf{v}_j^*(s(j))$. This normalized error is then multiplied by the proportional gain term for the current stage, $\mathbf{P}_j(s(j))$, and the corresponding variance $\mathbf{v}_{\xi,k-1}(j)$ is increased or decreased by that percentage to compute the new variance value $\mathbf{v}_{\xi,k}(j)$.

Because there is some inertia to the process of v_j changing in response to a new value of $\mathbf{v}_{\xi,k}(j)$, the gains \mathbf{P}_j cannot be too large, otherwise v_j will not converge to the desired value, instead, it will continually shrink and expand. This is illustrated in Fig. 7, where the value of \mathbf{P}_j is varied for estimation of w_{b_0} for the pump. For $\mathbf{P}_j(s) = 1 \times 10^{-2}$ for all s , this oscillatory behavior occurs because \mathbf{P}_j is too large. In contrast, if \mathbf{P}_j is too small, such as when $\mathbf{P}_j(s) = 1 \times 10^{-5}$, v_j will converge to the final value of \mathbf{v}_j^* much more slowly.

In our experiments, for all parameters, setting $S_j = 2$ with $\mathbf{v}_j^* = [50, 10]$, $\mathbf{T}_j = [60, 0]$, and $\mathbf{P}_j = [1 \times 10^{-4}, 1 \times 10^{-4}]$, worked well over the entire range of values considered for each wear parameter. Ideally, the wear parameter variance would be zero, but the particle filter needs some amount of noise to accurately track the parameter. So, $\mathbf{v}_j^*(S_j)$ cannot be too small, and we have found that controlling to an RMAD of 10% at the final stage introduces an acceptable amount of uncertainty while allowing for accurate tracking.

V. PREDICTION

Prediction is initiated at a given time k_P . Using the current joint state-parameter estimate, $p(\mathbf{x}_{k_P}, \boldsymbol{\theta}_{k_P} | \mathbf{y}_{0:k_P})$, which represents the most up-to-date knowledge of the system at time k_P , the goal is to compute $p(\text{EOL}_{k_P} | \mathbf{y}_{0:k_P})$ and

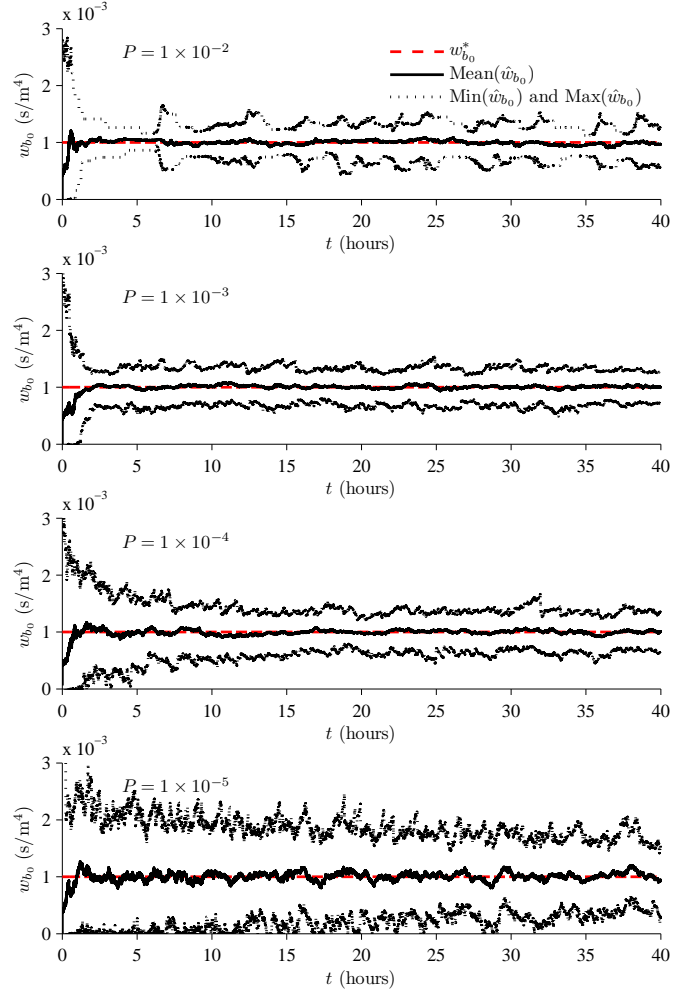


Fig. 7. Estimation of w_{b_0} with different values of P .

$p(\text{RUL}_{k_P} | \mathbf{y}_{0:k_P})$. As discussed in Section IV, 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 [23]

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

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

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

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: $\{(\mathbf{x}_{k_P}^i, \boldsymbol{\theta}_{k_P}^i), w_{k_P}^i\}_{i=1}^N$
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

If an analytic solution exists for the prediction, this may be directly used to obtain the prediction from the state-parameter distribution. An analytical solution is rarely available, so the general approach to solving the prediction problem is through simulation. Each particle is simulated forward to EOL to obtain the complete EOL distribution. The pseudocode for the prediction procedure is given as Algorithm 3 [3]. 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.

Note that prediction requires hypothesizing future inputs of the system, $\hat{\mathbf{u}}_k$, because damage progression is dependent on the operational conditions. For example, in the pump, an increased rotation speed will cause bearing friction to increase at a faster rate, and will cause an increased pump flow, which, in turn, will cause impeller wear to increase at a faster rate. The choice of expected future inputs depends on the knowledge about operational settings and the type of information the user is interested in, e.g., for a worst-case scenario, one would consider the pump running at its maximum rotation.

Fig. 8 shows results from the simultaneous prediction of thrust bearing and radial bearing wear for $N = 100$ (not all trajectories are shown in the lower plot). Initially, the particles have a very tight distribution of friction coefficient values, but the distribution of the wear parameters, w_t and w_r , is relatively large. As a result, the individual trajectories are easily distinguishable as EOL is approached. The EOL threshold is multi-dimensional, and in this example we restrict to the performance constraints related to T_t and T_r , with the maximum permissible values denoted in the figure (370 K for each). We show also the projections of the trajectories onto the temperature-time planes (gray lines). The projection onto the T_t - t plane (right) shows the progression of T_t towards the T_t^+ threshold as a function of time. The projections stop when EOL is reached, and the dotted lines connecting the projections to the time axis indicate individual EOL predictions. Similarly, the projection onto the T_r - t plane (bottom) shows the progression of T_r towards the T_r^+ threshold as a function of time. The dotted lines connecting to the trajectory endpoints are used as a visual aid to place the endpoints in the three-dimensional space. For some particles, T_t^+ is reached first, while for others, T_r^+ is reached first. The different EOL

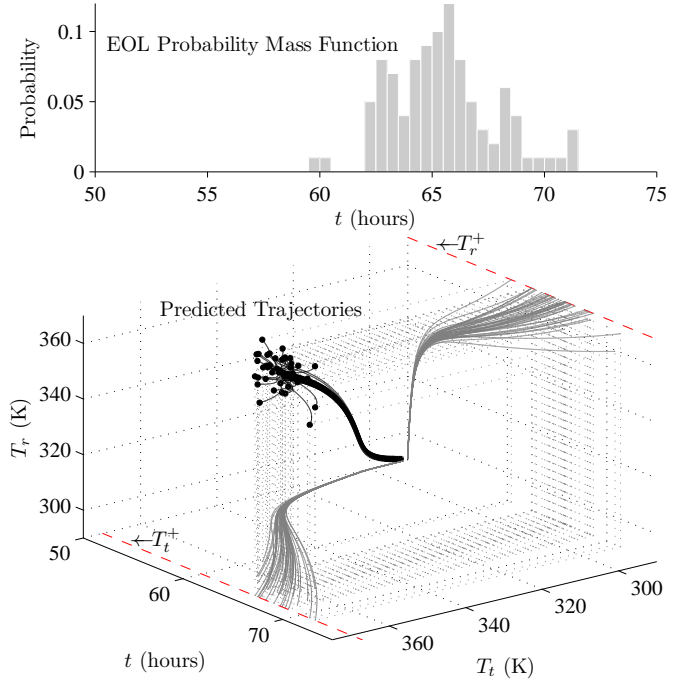


Fig. 8. Simultaneous prediction of thrust bearing and radial bearing wear in the pump. The damage trajectories are coming out of the page, increasing in T_t , increasing in T_r , and increasing in t .

values along with particle weights form an EOL distribution approximated by the probability mass function shown in the upper plot.

VI. RESULTS

In this section, we present simulation-based experiments to analyze the performance of the prognostics algorithm under concurrent damage progression processes. We first provide detailed results for a single experiment to demonstrate the approach, followed by results summarizing a large number of experiments.

A. Demonstration of Approach

We first provide an example scenario to illustrate the approach. We use $N = 500$ and set $S_j = 2$ with $\mathbf{v}_j^* = [50, 10]$, $\mathbf{T}_j = [60, 0]$, and $\mathbf{P}_j = [1 \times 10^{-3}, 1 \times 10^{-3}]$ for all wear parameter indices j within the adaptation algorithm. Fig. 9 shows the estimation results for the hidden parameters, with the true values given by $w_{b_0}^* = 2.0 \times 10^{-3}$, $w_t^* = 2.0 \times 10^{-11}$, and $w_r^* = 4.5 \times 10^{-11}$. The estimated wear parameter distributions begin very wide, but quickly converge first to 50% RMAD and then to 10% RMAD due to the variance control algorithm. After convergence, the wear parameters are tracked with percent root mean square errors (PRMSEs) of $\text{PRMSE}_{w_{b_0}} = 4.32$, $\text{PRMSE}_{w_t} = 5.03$, and $\text{PRMSE}_{w_r} = 2.82$, and with average RMAD of $\text{RMAD}_{w_{b_0}} = 8.57$, $\text{RMAD}_{w_t} = 8.54$, and $\text{RMAD}_{w_r} = 8.22$.

Prediction performance is shown by the α - λ plot of Fig. 10. The α - λ metric requires that for a given prediction time λ , at least β of the RUL probability mass lies within α of

TABLE II
ESTIMATION AND PREDICTION PERFORMANCE

n	PRMSE $_{w_A}$	PRMSE $_{w_t}$	PRMSE $_{w_r}$	RMAD $_{w_A}$	RMAD $_{w_t}$	RMAD $_{w_r}$	RA	RMAD $_{RUL}$
1	3.70	3.58	2.54	11.58	11.27	10.03	97.28	11.61
10	4.15	2.81	2.74	12.25	11.48	10.63	96.58	12.34
100	6.30	3.46	3.23	13.46	12.38	11.59	94.69	14.09
1000	12.93	6.25	5.29	13.92	12.99	12.64	79.37	15.32

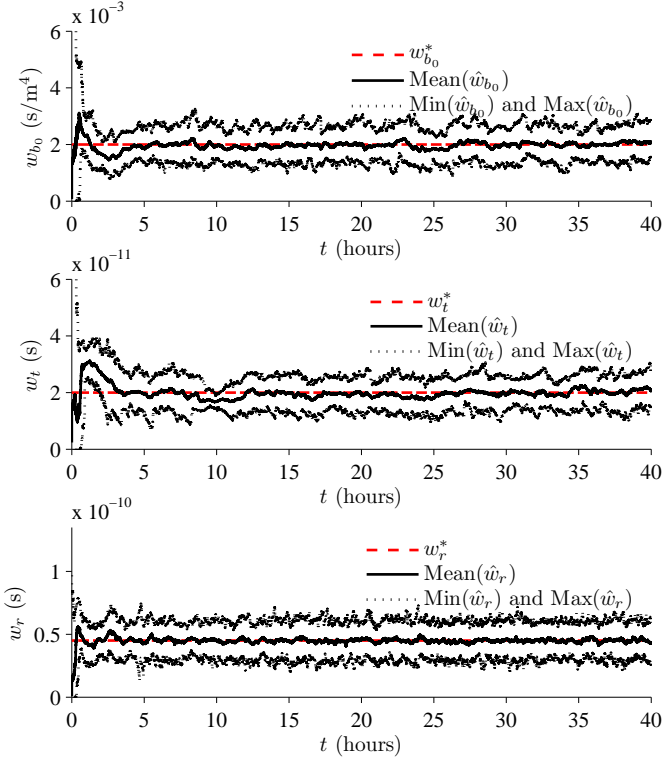


Fig. 9. Simultaneous estimation of pump wear parameters for $N = 500$ and $S_j = 2$ with $\mathbf{v}_j^* = [50, 10]$, $\mathbf{T}_j = [60, 0]$, and $\mathbf{P}_j = [1 \times 10^{-3}, 1 \times 10^{-3}]$ for all j .

the true value [14]. Here, we use $\alpha = 0.1$ and $\beta = 0.5$. In this example, thrust bearing wear dominates the EOL prediction. The accurate and precise wear parameter estimates yield correspondingly accurate and precise RUL predictions, and the α - λ metric is satisfied at all points. Prediction accuracy is evaluated using the relative accuracy (RA) metric [14], where for a prediction time k_P ,

$$RA_{k_P} = 100 \left(1 - \frac{|RUL_{k_P}^* - \widetilde{RUL}_{k_P}|}{RUL_{k_P}^*} \right),$$

where $RUL_{k_P}^*$ denotes the true RUL at time k_P , and \widetilde{RUL}_{k_P} denotes the median RUL prediction at k_P . Here, we use the median as the point of central tendency because the prediction distributions are skewed, due to the nonlinear damage progressions, and so it is a better description of central tendency than the mean. In this example, RA averaged over all prediction points is $\overline{RA} = 96.70\%$. RMAD of the RUL distribution averaged over all prediction points is $\overline{RMAD}_{RUL} = 8.56\%$. Maintaining the variance of the wear parameter estimates helps

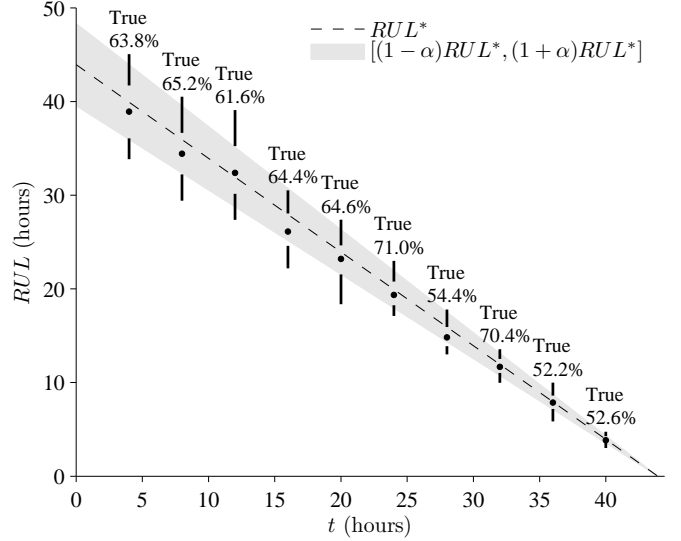


Fig. 10. α - λ performance with $\alpha = 0.1$ and $\beta = 0.5$ for $N = 500$ and $S_j = 2$ with $\mathbf{v}_j^* = [50, 10]$, $\mathbf{T}_j = [60, 0]$, and $\mathbf{P}_j = [1 \times 10^{-3}, 1 \times 10^{-3}]$ for all j .

to maintain also the RMAD of the RUL (though not to the same setpoint).

B. Simulation Results

We performed a number of simulation experiments in which combinations of wear parameter values were selected randomly within a range. We selected values in $[0.5 \times 10^{-3}, 4 \times 10^{-3}]$ at increments of 0.5×10^{-3} for w_{b_0} , in $[0.5 \times 10^{-11}, 7 \times 10^{-11}]$ at increments of 0.5×10^{-11} for w_t , and in $[0.5 \times 10^{-11}, 7 \times 10^{-11}]$ at increments of 0.5×10^{-11} for w_r , such that the maximum wear rates corresponded to a minimum EOL of 20 hours. Note that these wear rates are higher than one would observe in practice, and are selected only to reduce the experiment time to a practical level. In order to confirm that the wear parameter variance could still be maintained with additional sensor noise, we varied the sensor noise variance by factors of 1, 10, 100, and 1000, and performed 30 experiments for each case. In all experiments, we set $N = 500$ and $S_j = 2$ with $\mathbf{v}_j^* = [50, 10]$, $\mathbf{T}_j = [60, 0]$, and $\mathbf{P}_j = [1 \times 10^{-4}, 1 \times 10^{-4}]$ for all j , and the particle filter assumed the sensor noise variance was 100 times its true value. We considered the case where the future input of the pump is known, and it is always operated at a constant RPM. Hence, the only uncertainty present is that involved in the noise terms and that introduced by the particle filtering and variance control algorithms.

The averaged estimation and prediction performance results are shown in Table II. The sensor noise variance multiplier is given in the column labeled with n . Overall, the estimation results are very good, with PRMSE kept under 5% in most cases. As sensor noise increases, tracking becomes more difficult, and with the highest level of sensor noise, estimation performance was poor in some cases. However, the estimation spread is maintained close to the desired level of 10%. As sensor noise increases, this becomes more difficult and the average spread increases.

When estimation performance is good, this translates to accurate and precise predictions, since future inputs were assumed to be known. As sensor noise is increased, the accuracy and precision of the RUL predictions decrease. Even though sensor noise increases significantly, prediction spread does not, since the estimation spread is being controlled. With the highest level of sensor noise, around 17% of the cases had poor estimation performance, and this resulted in a significant drop in average RA to a little less than 80%. Omitting these cases, the RA averages around 90%.

Additional analysis of the performance at the highest noise level showed that increasing the value of P makes the variance control algorithm too aggressive, and does not give the filter enough time to converge, resulting sometimes in a loss of convergence. As sensor noise decreases, a higher value of P may be used without tracking problems.

Fig. 11 shows the RMAD of the wear parameters as a function of wear parameter value. Here, it is shown that the RMAD of a wear parameter is successfully controlled largely independently of its specific value. Therefore, one may tune only the initial random walk variances, based on anticipated minimum EOL values, and the algorithm self-tunes to optimize performance for the actual wear parameter value. Here, it is clear that as sensor noise is increased, RMAD is generally higher, but still close to the desired final setpoint, denoted with v_j^* in the figure. In some cases, an increase in RMAD is observed as the wear parameter value decreases. This is due somewhat to the slower convergence in those cases.

The computational complexity of damage estimation using the particle filter is a function of the number of particles. Here, damage estimation using 500 particles, implemented in Matlab running on a Windows system with a dual-core 2.49 GHz processor with 3 GB RAM, took on average 1.25 hours to run through 40 hours of data, so the approach is capable of running 32 times faster than real time. It is expected that the run time would reduce by an order of magnitude implemented in a compiled language, such as C. For the prediction step, the computational complexity depends on both the number of particles but also on the wear rates of the particles, since particles with smaller wear rates will take longer to simulate to EOL. In our experiments, in the worst case it took about 6 minutes to obtain a prediction 40 hours ahead. As EOL is approached, this time reduces since the amount of time to simulate to reach EOL is reduced. Prediction times can be improved by only simulating forward a reduced set of the particles chosen to preserve the statistical properties of the distribution and its prediction, as described in [24].

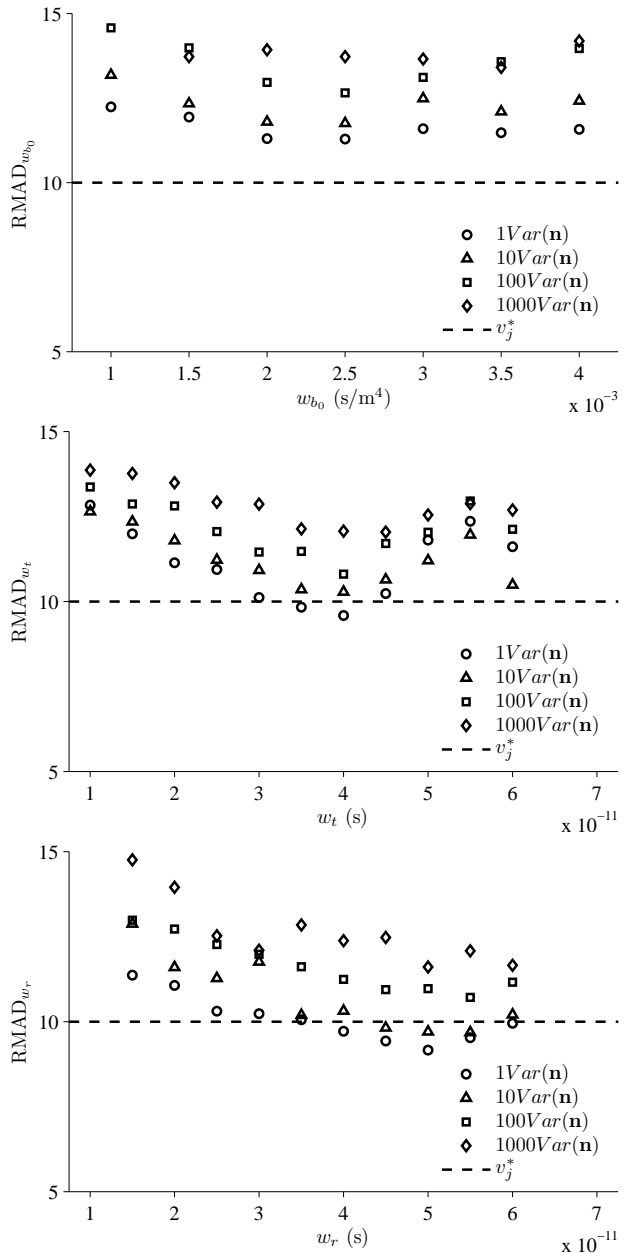


Fig. 11. RMAD of the wear parameter as a function of wear parameter value.

VII. RELATED WORK

Model-based diagnosis has been investigated previously with application to centrifugal pumps [16]–[18]. In contrast, pump prognostics approaches have mostly been data-driven, usually based on pump vibration signals. A principal component analysis method is applied for condition monitoring of a pump using vibration and acceleration signals in [25]. However, only a subset of possible damage modes manifest in the vibration sensors. Further, when using such methods it is difficult to map changes in vibration back to changes in the thrust bearings, radial bearings, or both, while also quantifying the amount of damage. A model-based approach, on the other hand, does this easily, with an appropriate model. A model-based prognostics approach for pumps is presented in [19],

however, it considers only a single degradation mode.

Model-based prognostics approaches have been developed previously and applied to other components and fault modes, such as batteries [4], [26], fatigue cracks [6], [27], and automotive suspension systems [5]. Although particle filtering approaches have been most common, other filtering methodologies may be used depending on the complexity of the underlying model. In [28], damage is tracked using a Kalman filter. In a related approach, a model-based prognosis methodology is developed in [5] using an interacting multiple model filter for state-parameter estimation and prediction. We use particle filters because they may be generally applied, and the pump model is nonlinear. Particle filters have also been used in [4], [6], [27], [29], among others.

All of these approaches either assume only a single damage variable, or a very restricted form of the EOL threshold. For example, in [27], EOL is directly linked to a state variable exceeding some static threshold. However, this approach is not general. The concept of a *hazard zone* has also been used [6], which generalizes for a given damage mode from a single threshold value to a bounded distribution, estimated from historical failures. This assumes a fundamentally different view, a damage-centric view, in which failure is directly tied to some amount of damage. In our approach, we link failure directly to violations of functional or performance specifications, taking a performance-centric view. These performance constraints are precise, and representing them through distributions does not apply. Violations of the performance constraints will invariably be caused by some amount of damage accumulation, but the same amount of damage may be tolerated in different operating regimes, so the performance-centric view is more general. For the pumps, EOL is defined as a combination of limits on efficiency and three different temperatures, and the different damage variables each contribute to the complete system state moving towards those thresholds. Even if information defining a hazard zone is available, it is more practical to use only the lower bound; a system operator would never drive the system past that point since there is a chance of failure at that point, effectively making the lower bound a precise performance constraint.

Methods to adjust the random walk variance in the particle filter have also been previously investigated. One approach is to use kernel shrinkage, in which the random walk noise is diminished over time [10]. This approach assumes that the parameter is constant, but in reality, this may not be the case, so some amount of noise should still be included to account for unmodeled deviations in the parameter value over time. In [6], [11], [12], this noise (viewed as a hyper-parameter) is tuned using outer correction loops based on prediction error. In this case, the underlying prognostic model is assumed to contain only a single fault dimension, therefore it cannot be applied in our case. It is also fundamentally different from our approach because it is prediction error that drives the adaptation. Our method is based on the observation that the particle filter, if tuned appropriately, will naturally converge to the true values with some uncertainty, so we drive the adaptation based on the error between that uncertainty and the desired level of uncertainty. Since it does not rely on

performing a prediction in order to derive the error, it is also computationally more efficient. A similar method is presented in [13], again assuming a single fault dimension, where the adaptation is driven by estimation error of the fault based on a sensitivity analysis. This assumes that the fault variable can be directly measured. In our case study, this is not the case.

VIII. CONCLUSIONS

We developed a model-based prognostics framework that handles concurrent damage progression processes. Damage progression processes are characterized by functions, parameterized by a set of wear parameters, describing how a fault or damage variable evolves in time. Particle filters perform joint state-parameter estimation in order to estimate the health state of the component. The state-parameter distribution is then extrapolated to the EOL threshold to compute EOL and RUL predictions in the presence of multiple damage progressions. A novel variance control mechanism maintains the random walk variances of the particle filter, in order to maintain the uncertainty of the unknown wear parameters at a desired level, and, consequently, reduce prediction uncertainty. The framework was applied to a centrifugal pump, and the results demonstrated good performance over a range of wear parameter values and sensor noise levels. Current work involves validating the approach as applied to a real pump system.

Although quite robust and generally applicable, the particle filter suffers from a high computational complexity. Using 500 particles was sufficient for this particular case study, but, in general, as the dimension of the joint state-parameter space increases, the number of particles needed for successful estimation increases with it. Therefore, for large systems, the approach presented in this paper may not achieve the desired efficiency. When appropriate, other filtering methods may be applied to improve computational efficiency [30]. In recent work, methods for improving the efficiency based on model decomposition have been explored, allowing the amount of computation to be decreased without a loss of estimation performance [31]. With such an approach, the extension of the prognostics framework presented here to system-level prognostics becomes feasible, and will be investigated in future work.

REFERENCES

- [1] J. W. Sheppard, M. A. Kaufman, and T. J. Wilmering, "Ieee standards for prognostics and health management," in *2008 IEEE AUTOTESTCON*, 2008, pp. 97–103.
- [2] M. Roemer, C. Byington, G. Kacprzynski, and G. Vachtsevanos, "An overview of selected prognostic technologies with reference to an integrated PHM architecture," in *Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace*, 2005.
- [3] M. Daigle and K. Goebel, "Model-based prognostics under limited sensing," in *2010 IEEE Aerospace Conference*, Mar. 2010.
- [4] B. Saha and K. Goebel, "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*, Sept. 2009.
- [5] J. Luo, K. R. Pattipati, L. Qiao, and S. Chigusa, "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, pp. 1156–1168, Sept. 2008.

- [6] M. Orchard and G. Vachtsevanos, "A particle filtering approach for on-line fault diagnosis and failure prognosis," *Transactions of the Institute of Measurement and Control*, no. 3-4, pp. 221-246, June 2009.
- [7] M. Schwabacher, "A survey of data-driven prognostics," in *Proceedings of the AIAA Infotech@Aerospace Conference*, 2005.
- [8] M. Daigle and K. Goebel, "Multiple damage progression paths in model-based prognostics," in *Proceedings of the 2011 IEEE Aerospace Conference*, Mar. 2011.
- [9] B. Saha, E. Koshimoto, C. C. Quach, E. F. Hogge, T. H. Strom, B. L. Hill, S. L. Vazquez, and K. Goebel, "Battery health management system for electric UAVs," in *2011 IEEE Aerospace Conference*, Mar. 2011.
- [10] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," *Sequential Monte Carlo Methods in Practice*, pp. 197-223, 2001.
- [11] M. Orchard, G. Kacprzyński, K. Goebel, B. Saha, and G. Vachtsevanos, "Advances in uncertainty representation and management for particle filtering applied to prognostics," in *Proceedings of International Conference on Prognostics and Health Management*, Oct. 2008.
- [12] M. Orchard, F. Tobar, and G. Vachtsevanos, "Outer feedback correction loops in particle filtering-based prognostic algorithms: Statistical performance comparison," *Studies in Informatics and Control*, no. 4, pp. 295-304, Dec. 2009.
- [13] B. Saha and K. Goebel, "Model adaptation for prognostics in a particle filtering framework," *International Journal of Prognostics and Health Management*, vol. 2, no. 1, 2011.
- [14] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for offline evaluation of prognostic performance," *International Journal of Prognostics and Health Management*, vol. 1, no. 1, 2010.
- [15] S. E. Lyshevski, *Electromechanical Systems, Electric Machines, and Applied Mechatronics*. CRC, 1999.
- [16] A. Wolfram, D. Fussel, T. Brune, and R. Isermann, "Component-based multi-model approach for fault detection and diagnosis of a centrifugal pump," in *Proceedings of the 2001 American Control Conference*, vol. 6, 2001, pp. 4443-4448.
- [17] C. Kallesøe, "Fault detection and isolation in centrifugal pumps," Ph.D. dissertation, Aalborg University, 2005.
- [18] G. Biswas and S. Mahadevan, "A hierarchical model-based approach to systems health management," in *2007 IEEE Aerospace Conference*, Mar. 2007.
- [19] F. Tu, S. Ghoshal, J. Luo, G. Biswas, S. Mahadevan, L. Jaw, and K. Navarra, "PHM integration with maintenance and inventory management systems," in *Proc. of the 2007 IEEE Aerospace Conference*, Mar. 2007.
- [20] I. M. Hutchings, *Tribology: friction and wear of engineering materials*. CRC Press, 1992.
- [21] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.
- [22] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1-25, 1996.
- [23] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197-208, 2000.
- [24] M. Daigle and K. Goebel, "Improving computational efficiency of prediction in model-based prognostics using the unscented transform," in *Annual Conference of the Prognostics and Health Management Society 2010*, Oct. 2010.
- [25] S. Zhang, M. Hodkiewicz, L. Ma, and J. Mathew, "Machinery condition prognosis using multivariate analysis," *Eng. Asset Management*, pp. 847-854, 2006.
- [26] M. Abbas, A. A. Ferri, M. E. Orchard, and G. J. Vachtsevanos, "An intelligent diagnostic/prognostic framework for automotive electrical systems," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 352-357.
- [27] E. Zio and G. Peloni, "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliability Engineering & System Safety*, vol. 96, no. 3, pp. 403-409, 2011.
- [28] D. Chelidze, "Multimode damage tracking and failure prognosis in electromechanical system," in *Proceedings of the SPIE Conference*, vol. 4733, 2002, pp. 1-12.
- [29] N. Bolander, H. Qiu, N. Eklund, E. Hindle, and T. Rosenfeld, "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*, Oct. 2010.
- [30] M. Daigle, B. Saha, and K. Goebel, "A comparison of filter-based approaches for model-based prognostics," in *Proceedings of the 2012 IEEE Aerospace Conference*, Mar. 2012.
- [31] M. Daigle, A. Bregon, and I. Roychoudhury, "Distributed damage estimation for prognostics based on structural model decomposition," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011*, Sept. 2011, pp. 198-208.



Matthew J. Daigle (S'07-M'08) received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively.

From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. From June 2008 to December 2011, he was an Associate Scientist with the University of California, Santa Cruz, at NASA Ames Research Center. Since January 2012, he has been with NASA Ames Research Center as a Research Computer Scientist. His current research interests include physics-based modeling, model-based diagnosis and prognosis, simulation, and hybrid systems.

Dr. Daigle is a member of the Prognostics and Health Management Society. He is a recipient of a University Graduate Fellowship from Vanderbilt University, a best paper award in the Annual Conference of the Prognostics and Health Management Society 2011, a NASA Ames Group Achievement Award in 2011, and an Ames Contractor Council Excellence Award in 2011. He has published over 40 peer-reviewed papers in the area of Systems Health Management.



Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively.

Dr. Goebel is currently a Deputy Branch Chief of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence and is the Technical Lead for Prognostics and Decision Making in NASA's System-wide Safety and Assurance Technologies Project. Prior to joining NASA in 2006, he was a Senior Research Scientist at General Electric Corporate Research and Development Center since 1997. He was also an Adjunct Professor of the Computer Science Department at Rensselaer Polytechnic Institute, Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems.

Dr. Goebel holds 15 patents and has co-authored more than 200 technical papers in the field of Systems Health Management. He is currently member of the board of directors of the Prognostics and Health Management Society and Associate Editor of the International Journal of Prognostics and Health Management.