Stochastic Prediction of Remaining Driving Time and Distance for a Planetary Rover

Matthew Daigle
NASA Ames Research Center
Moffett Field, CA 94035
matthew.j.daigle@nasa.gov

Shankar Sankararaman
SGT Inc., NASA Ames Research Center
Moffett Field, CA 94035
shankar.sankararaman@nasa.gov

Chetan S. Kulkarni
SGT Inc., NASA Ames Research Center
Moffett Field, CA 94035
chetan.s.kulkarni@nasa.gov

***Abstract*—The operations of a planetary rover depend critically upon the amount of power that can be delivered by its batteries. In order to plan the future operation of the rover, it is important to make reliable predictions regarding the end-of-discharge time, which, in turn, can be used to estimate the remaining driving time and distance of the rover. In addition, quantifying the uncertainty in these predictions is critical to making risk-informed decisions regarding the operations of the rover. This paper presents a computational methodology to stochastically predict end-of-discharge time, remaining driving time, and remaining driving distance for a planetary rover, based on monitoring the batteries that power the rover. We utilize a model-based prognostics framework that characterizes and incorporates the various sources of uncertainty into these predictions, thereby assisting operational decision-making. We consider two different types of driving scenarios, structured and unstructured driving, and characterize the uncertainty they create in the future usage of the rover. In structured driving, the rover navigates among a set of known waypoints, and in unstructured driving, the rover performs a sequence of unplanned maneuvers. Results from a set of field experiments illustrate these computational methods and demonstrate their applicability.**

TABLE OF CONTENTS

1 INTRODUCTION 1
2 MODEL-BASED PROGNOSTICS 2
3 ROVER MODELING FOR PROGNOSTICS 2
4 ESTIMATION 5
5 PREDICTION 5
6 RESULTS 6
7 CONCLUSIONS 7
ACKNOWLEDGMENTS 8
REFERENCES 8
BIOGRAPHY 9

1. INTRODUCTION

Autonomous planning integrates the two problems of search and prediction in order to find feasible sequences of actions and behaviors [1]. For planetary rovers exploring an environment, the plans depend on how much longer the batteries can discharge, and, consequently, how much remaining time and distance that the rover can drive [2–7]. Whereas in the robotics field significant effort has been expended on the search problem, relatively little has been expended on the prediction problem. In order to make accurate predictions, a high-fidelity battery health monitoring (BHM) system is needed, capable of estimating the state of charge and predicting end of discharge (EOD). BHM has been receiving increased attention in recent years, due to the increasingly widespread presence of electric cars and aircraft [8, 9]. However, in order to predict remaining driving time and distance for the rover, the BHM system must be placed in the context of the rover system as a whole [10].

In order to solve the prediction problem, the general framework of model-based prognostics may be used [11–13]. In this framework, we develop a model of the system (e.g., the rover), and use model-based algorithms to estimate the current system state and predict its future evolution. Specifically, the algorithms are focused on the prediction of some system event, which, in this context, is EOD. The advantage of a model-based approach is that, going from one system to another, the algorithms remain the same while only the model changes. Model-based prognostics approaches have been applied to a wide range of components and systems, from single components such as capacitors [14], valves [15, 16], and pumps [11], to larger systems such electric aircraft [17] and electric cars [7].

In the context of planetary rovers, in previous work we developed a BHM framework using model-based prognostics [18]. The main contribution of this work was the use of a high fidelity, computationally efficient electrochemistry-based battery model. Using this model, highly accurate EOD predictions, given known future power demands, were delivered, however, system-level predictions such as remaining driving time and remaining driving distance, which would be much more useful to a system operator or automated planner, were not predicted. Further, only a very simple model of future power demands was considered. In prognostics, it is just as important to accurately model the system itself as to quantify the uncertainty associated with predictions, and, in many cases, the most significant source of uncertainty is the future loading to the system [19–21].

In this work, we extend the rover BHM framework developed in [18] in two significant ways. First, we extend the model-based prognostics framework for prediction of both the time of the system event to be predicted (here, EOD) and the value of system variables at that time (here, remaining driving time and distance). Second, we employ recent advances in uncertainty characterization and quantification for prognostics in order to produce predictions that more accurately capture the prediction uncertainty [22]. We show how these new methods can be generally applied to two different driving

978-1-4799-5380-6/15/$31.00

scenarios: structured driving, in which the rover travels along a predetermined set of waypoints, and unstructured driving, in which the rover is driven freely by random sequences of maneuvers.

The paper is organized as follows. Section 2 describes the model-based prognostics framework. Section 3 summarizes the system modeling for prognostics. Sections 4 and 5 discuss the estimation and prediction approaches, respectively. Section 6 presents the experimental results, and Section 7 concludes the paper.

2. MODEL-BASED PROGNOSTICS

In this section, we formulate the prognostics problem, extending the framework presented in [18, 22]. We then provide a computational architecture for model-based prognostics that will be applied to the rover.

Problem Formulation

We assume the system model may be generally defined as

$$\mathbf{x}(k+1) = \mathbf{f}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{v}(k)), \quad (1)$$

$$\mathbf{y}(k) = \mathbf{h}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k), \mathbf{n}(k)), \quad (2)$$

where k is the discrete time variable, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ is the state vector, $\boldsymbol{\theta}(k) \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{v}(k) \in \mathbb{R}^{n_v}$ is the process noise vector, $\mathbf{f}$ is the state equation, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ is the output vector, $\mathbf{n}(k) \in \mathbb{R}^{n_n}$ is the measurement noise vector, and $\mathbf{h}$ is the output equation.[2] The unknown parameter vector $\boldsymbol{\theta}(k)$ is used to capture explicit model parameters whose values are unknown and time-varying stochastically.

Prognostics is concerned with predicting the occurrence of some event E that is defined with respect to the states, parameters, and inputs of the system. We define the event as the earliest instant that some event threshold $T_E : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_u} \to \mathbb{B}$, where $\mathbb{B} \triangleq \{0, 1\}$, changes from the value 0 to 1. That is, the time of the event k_E at some time of prediction k_P is defined as

$$k_E(k_P) \triangleq \inf\{k \in \mathbb{N} \colon k \geq k_P \wedge T_E(\mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)) = 1\}. \quad (3)$$

The time remaining until that event, Δk_E, is defined as

$$\Delta k_E(k_P) \triangleq k_E(k_P) - k_P. \quad (4)$$

We may also be interested in predicting the values of system variables, $\mathbf{z}$, at time k_E. The variables $\mathbf{z}$ are expressed as a function of the system states, parameters, and inputs:

$$\mathbf{z}(k) = \mathbf{g}(k, \mathbf{x}(k), \boldsymbol{\theta}(k), \mathbf{u}(k)), \quad (5)$$

where $\mathbf{g}$ is a function mapping the states, parameters, and inputs to these variables. We may also need to predict $\Delta\mathbf{z}(k_E)$, defined as

$$\Delta\mathbf{z}(k_E) = \mathbf{z}(k_E) - \mathbf{z}(k_P). \quad (6)$$

[2] Bold typeface denotes vectors, and n_a denotes the length of a vector $\mathbf{a}$.

Uncertainty Representation

The system evolves stochastically due to (*i*) the process noise $\mathbf{v}(k)$ and (*ii*) the future inputs to the system, $\mathbf{u}(k)$ for $k > k_P$, which are, in general, uncertain. Thus, k_E and its derived variables are random variables, and we must compute the probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$, where $\mathbf{y}(k_0{:}k_P)$ denotes the system outputs from time k_0 to k_P [20, 23]. Further, additional uncertainty arises due to lack of knowledge of the system, namely, (*i*) the initial state at time k_P, $\mathbf{x}(k_p)$; and (*ii*) the unknown parameters $\boldsymbol{\theta}(k)$.

So, in order to compute k_E and its derived variables, we need estimates of (*i*) the initial state at time k_P, $\mathbf{x}(k_p)$; (*ii*) the parameter values $\boldsymbol{\theta}(k)$ for all $k \geq k_P$, denoted as $\boldsymbol{\Theta}_{k_P}$ (the subscript k_P indicates the start time of the trajectory); (*iii*) the inputs $\mathbf{u}(k)$ for all $k \geq k_P$, denoted as $\mathbf{U}_{k_P}$; and (*iv*) the process noise $\mathbf{v}(k)$ for all $k \geq k_P$, denoted as $\mathbf{V}_{k_P}$ [22].

In order to make a prediction that accounts for this uncertainty, we require the probability distributions $p(\mathbf{x})$, $p(\boldsymbol{\Theta}_{k_P})$, $p(\mathbf{U}_{k_P})$, and $p(\mathbf{V}_{k_P})$ to be defined, and this is part of the modeling problem. Because it is often difficult to describe the probability distribution of a trajectory directly, we use the method of surrogate variables to define them indirectly, as introduced in [22]. For describing the probability distribution of a generic trajectory $\mathbf{A}_k$, we introduce a set of *surrogate* random variables $\boldsymbol{\lambda}_a = [\lambda_a^1\ \lambda_a^2\ \ldots]$. We describe a trajectory using $\boldsymbol{\lambda}_a$ and instead define $p(\boldsymbol{\lambda}_a)$, which in turn defines $p(\mathbf{A}_k)$. These surrogate variables can be used to describe trajectories in many ways. For the parameter, input, and process noise trajectories we have the surrogate variables $\boldsymbol{\lambda}_\theta$, $\boldsymbol{\lambda}_u$, and $\boldsymbol{\lambda}_v$. Additional discussion on the use of surrogate variables can be found in [22].

Prognostics Architecture

We adopt a model-based prognostics architecture [11, 22], in which there are two sequential problems, (*i*) the *estimation* problem, which requires determining a joint state-parameter estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k_P))$ based on the history of observations up to time k, $\mathbf{y}(k_0{:}k_P)$, and (*ii*) the *prediction* problem, which determines at k_P, using $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k_P))$, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$, a probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$, as well as those for Δk_E, $\mathbf{z}(k_E)$ and $\Delta\mathbf{z}(k_E)$.

The prognostics architecture is shown in Fig. 1. In discrete time k, the system is provided with inputs $\mathbf{u}_k$ and provides measured outputs $\mathbf{y}_k$. The estimation module uses this information, along with the system model, to compute an estimate $p(\mathbf{x}(k), \boldsymbol{\theta}(k)|\mathbf{y}(k_0{:}k))$. The prediction module uses the joint state-parameter distribution and the system model, along with the distributions for the surrogate variables, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$, to compute the probability distribution $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$, $p(\Delta k_E(k_P)|\mathbf{y}(k_0{:}k_P))$, $p(\mathbf{z}(k_E)|\mathbf{y}(k_0{:}k_P))$, and $p(\Delta\mathbf{z}(k_E)|\mathbf{y}(k_0{:}k_P))$ at given prediction times k_P. We describe an approach to solve the estimation problem in Section 4, and an approach for the prediction problem in Section 5.

3. ROVER MODELING FOR PROGNOSTICS

The system model for prognostics must include the system dynamics, as defined by the states, inputs, outputs, process noise, sensor noise, state equation, and output equation. For prediction purposes, it must also include a definition of the system event to be predicted, E, and the corresponding

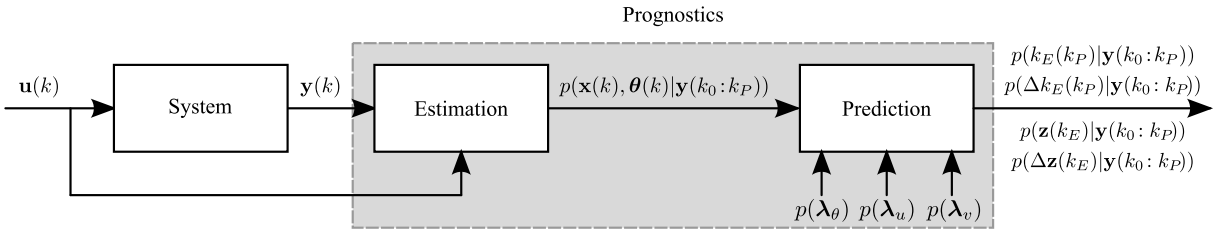

Figure 1. Prognostics architecture.

threshold function, T_E, along with descriptions of the future process noise, uknown parameters, and input trajectory distributions.

For the rover, we want to be able to predict when its batteries reach EOD; this defines the event E. Typically, EOD is described using a voltage threshold V_{EOD}, such that EOD has occurred once the battery voltage V has reached V_{EOD}, i.e., T_E evaluates to 1 when $V \leq V_{EOD}$.

At the time of event E, we want to know the remaining driving time, Δt_d, defined as the amount of remaining time until EOD that the rover can be driven, and the remaining driving distance Δd_d, defined as the remaining distance until EOD that can be traveled by the rover. These particular quantities are useful to an operator or an automated planner, because they describe how much longer the rover can be operated and how much distance it is able to cover in that time.

In order to predict these quantities, we need to develop a model of the rover as a system that includes its kinematics (to compute t_d, d_d, and the power demanded from the batteries) and the dynamics of the batteries (to compute voltage and predict EOD). We describe this model in the following subsections, followed by representations of the sources of uncertainty.

Rover Kinematics

We consider a four-wheel skid-steered rover, in which the speeds of the two left-side wheels are always commanded to the same speed, and the speeds of the two right-side wheels are always commanded to the same speed. The rover moves straight when these speeds are the same and turns when these speeds are different. Therefore, the inputs to the rover are u_L, the left-side speed, and u_R, the right-side speed.

We assume that the dynamics of the control system are sufficiently fast such that the actual wheel speeds are approximately equal to the commanded speeds, i.e., $v_L = u_L$ and $v_R = u_R$. Given the velocities v_L and v_R, driving distance, $d_d(k)$, is computed using:

$$d_d(k+1) = d_d(k) + \frac{v_L + v_R}{2}\Delta t, \tag{7}$$

where Δt is the sampling time. Driving time, $t_d(k)$, is simply described using:

$$t_d(k+1) = t_d(k) + \left(\frac{v_L + v_R}{2} > 0\right)\Delta t. \tag{8}$$

Given wheel speeds v_L and v_R, the power demanded from the batteries, P_B, is approximated by the linear relationship

$$P_B(k) = \frac{v_L + v_R}{2} * g_b + b_b, \tag{9}$$

where g_b and b_b are empirical parameters determined from rover field data.

Battery Modeling

The rover electrical power system consists of 24 batteries, with two parallel branches of 12 batteries in series [5, 24]. Each lithium-ion battery produces around 4 V, leading to a total output of 48 V supplied to the rover motors. We use the electrochemistry battery model originally developed in [25], which describes how the internal charge moves within the battery, given current as an input, and how voltage is computed based on the internal charge. We summarize the main details here, and refer the reader to [25] for additional details and model parameter values. The battery model has been validated with data from the rover operating in the field (see [18]).

The battery is divided into two electrodes, the positive (subscript p) and the negative (subscript n). Each electrode is split into two control volumes, a bulk volume (subscript b) and a surface layer (subscript s). As the battery discharges, Li ions move from the surface layer at the negative electrode, through the bulk, and to the surface layer at the positive electrode, in order to match the flow of electrons. So, we have four states representing charge (q), described by

$$\dot{q}_{s,p} = i_{app} + \dot{q}_{bs,p}, \tag{10}$$
$$\dot{q}_{b,p} = -\dot{q}_{bs,p} + i_{app} - i_{app}, \tag{11}$$
$$\dot{q}_{b,n} = -\dot{q}_{bs,n} + i_{app} - i_{app}, \tag{12}$$
$$\dot{q}_{s,n} = -i_{app} + \dot{q}_{bs,n}, \tag{13}$$

where i_{app} is the applied electric current. The flow of charge between the surface and bulk volumes is driven by diffusion based on Li ion concentration. The concentrations are computed as

$$c_{b,i} = \frac{q_{b,i}}{v_{b,i}}, \tag{14}$$
$$c_{s,i} = \frac{q_{s,i}}{v_{s,i}}, \tag{15}$$

where, for control volume C in electrode i, $c_{C,i}$ is the concentration and $v_{C,i}$ is the volume. The diffusion rate from the bulk to the surface in electrode i is expressed as

$$\dot{q}_{bs,i} = \frac{1}{D}(c_{b,i} - c_{s,i}), \tag{16}$$

where D is the diffusion constant.

The battery voltage, which is comprised of several electrochemical potentials, is computed as a function of the charge variables. These potentials include the equilibrium potential, concentration overpotential, surface overpotential, and ohmic overpotential [26]. The equilibrium potential is captured using the Nernst equation:

$$V_{U,i} = U_{0,i} + \frac{RT}{nF} \ln\left(\frac{1-x_i}{x_i}\right) + V_{\text{INT},i}, \tag{17}$$

where for electrode i, $U_{0,i}$ is a reference potential, R is the universal gas constant, T is the electrode temperature (in K), n is the number of electrons transferred in the reaction ($n = 1$ for Li-ion), F is Faraday's constant, x is the mole fraction of lithium ions in the lithium-intercalated host material [27]. Mole fraction x_i is related to charge using

$$x_i = \frac{q_i}{q^{\max}}, \tag{18}$$

where $q^{\max} = q_p + q_n$ refers to the total amount of available Li ions. It follows that $x_p + x_n = 1$. In the Nernst equation, $V_{\text{INT},i}$ is the activity correction term for which we use the Redlich-Kister expansion [27]:

$$V_{\text{INT},i} = \frac{1}{nF}\left(\sum_{k=0}^{N_i} A_{i,k}\left((2x_i-1)^{k+1} - \frac{2x_i k(1-x_i)}{(2x_i-1)^{1-k}}\right)\right). \tag{19}$$

The concentration overpotential is the difference in voltage between the surface and bulk control volumes due to the difference in concentration and is captured by using $x_{s,i}$ in the expression for equilibrium potential.

The ohmic resistance is described using the constant R_o, producing the ohmic overpotential:

$$V_o = i_{app} R_o. \tag{20}$$

The surface overpotentials are described by the Butler-Volmer equation, which, for Li ion batteries, reduces to

$$V_{\eta,i} = \frac{RT}{F\alpha}\text{arcsinh}\left(\frac{J_i}{2J_{i0}}\right), \tag{21}$$

where J_i is the current density, J_{i0} is the exchange current density, and α is the symmetry factor (0.5 for Li ion). The current densities are defined as

$$J_i = \frac{i}{S_i}, \tag{22}$$

$$J_{i0} = k_i(1-x_{s,i})^{\alpha}(x_{s,i})^{1-\alpha}, \tag{23}$$

where k_i is a lumped parameter of several constants including a rate coefficient, electrolyte concentration, and maximum ion concentration.

Battery voltage can now be expressed as follows:

$$V = V_{U,p} - V_{U,n} - V_o' - V_{\eta,p}' - V_{\eta,n}', \tag{24}$$

where

$$\dot{V}_o' = (V_o - V_o')/\tau_o, \tag{25}$$

$$\dot{V}_{\eta,p}' = (V_{\eta,p} - V_{\eta,p}')/\tau_{\eta,p}, \tag{26}$$

$$\dot{V}_{\eta,n}' = (V_{\eta,n} - V_{\eta,n}')/\tau_{\eta,n}, \tag{27}$$

and the τ parameters are empirical time constants (used since the voltages do not change instantaneously).

Recall that the batteries are arranged in two parallel branches of 12 batteries in series. So the total voltage provided by the battery system is

$$V_B = V_1 + V_2 + \ldots + V_{12}, \tag{28}$$

$$= V_{13} + V_{14} + \ldots + V_{24}. \tag{29}$$

For given wheel speeds, we have the power P_B demanded from the set of batteries, so the battery system as a whole sees $i_B = P_B/V_B$. Throughout discharge, the battery voltages remain balanced, so the current is split evenly between the two parallel branches, with each seeing $i_B/2$.

Dynamic Model Summary

To summarize, the model contains as states $\mathbf{x}$, four charge variables and three voltages per battery, along with t_d and d_d. Its inputs $\mathbf{u}$ include u_L and u_R. The voltages of all batteries, along with the total battery current are measured, so its outputs $\mathbf{y}$ include V_i for all batteries i, and i_B. The variables $\mathbf{z}$ consist of t_d and d_d.

Uncertainty Characterization

For the rover, we assume all parameters are known and process noise is negligible relative to future input uncertainty, so we need to define surrogate variables for the future input trajectories only. For prediction, the inputs to the system are the commanded wheel speeds u_L and u_R. However, since the demanded power is split evenly among the batteries, we can simplify the prediction and include the power for a battery P_b, as an input, and predict E for only one battery.

These inputs, u_L, u_R, and P_b, depend on how the rover is to be driven in the future. We consider two different driving scenarios, *unstructured* and *structured* driving. The associated future input uncertainty is different in these cases.

Unstructured Driving—In unstructured driving, the rover is driven around by the operator freely without any known goals. Therefore the power demanded on the batteries varies significantly. Thus, we have the least information about the usage of the rover, and the most future input uncertainty. The commanded wheel speeds, and hence required power, are all random. By making some simplifying assumptions, we can reduce the complexity of the problem. First, we assume that the rover travels only in a straight line, i.e., $u_L = u_R$, thus, removing one random variable. We can then use the known relationship between commanded speed and power so that only power remains as the random variable. So, we need only describe $P_b(k)$ for $k \geq k_P$ (from which we can compute corresponding values for u_L and u_R). Second, instead of the rover going at variable speeds, we assume that it moves at a constant speed, which is chosen randomly. We thus transform the distribution of variable-power trajectories into a distribution of constant-power trajectories, requiring only a single random variable to describe.

With these simplifications, we can express power as $P_b(k) = p$ for $k \geq k_P$, where we require only one surrogate variable to capture the statistics of p. In order to characterize the statistics of this variable, we can look at past unstructured driving scenarios, and compute the average power in each. Then, we can find the mean and variance of the average power over the set of scenarios. We use these statistical parameters to define the constant-power trajectory distribution.

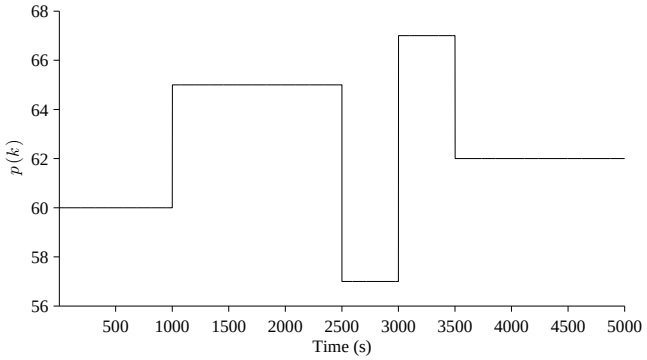

Figure 2. Sampled power trajectory for structured driving.

Structured Driving—In structured driving, the rover is driven to known waypoints at a fixed average speed. In this case, uncertainty is reduced because distances between waypoints and rover speed are known. Thus, we have more information about the usage of the rover, and this should reduce the future input uncertainty. In this driving scenario, the set of ordered waypoints the rover will travel to are known, and the rover travels a fixed overall speed. As such, power is the only uncertain input (even with known speeds, power becomes random due to variations in terrain). As with unstructured driving, we approximate the trajectory going between any two consecutive waypoints using a constant-power trajectory. So, between each pair of consecutive waypoints, we sample the power. In order to determine how long the rover will be operating at this power, we need to know the distance between the waypoints. Due to small variations in the paths taken between waypoints, a straight line is not followed, and so the distance is also random. As a random variable we consider the bias above the straight-line distance that the rover actually travels going between two consecutive waypoints, expressed as a percent increase. After the final waypoint is achieved, we assume the rover goes straight at constant speed.

So, for n remaining waypoints, we require $2n + 1$ surrogate variables. Fig. 2 shows a sample power trajectory for structured driving. The power has a step change at each waypoint. The amount of power drawn for each segment is selected randomly, as is the distance traveled between the waypoints. Each step change indicates that a waypoint has been met and a new segment begun.

In order to compute the statistics of these random variables, we look at past structured driving scenarios. We compute the average power between consective waypoints, and the bias to the straight-line distance traveled. Considering all waypoint segments, we can compute the mean and variance for average power and the distance bias. The surrogate variables will be sampled from these distributions.

4. ESTIMATION

The first step of prognostics is estimation; to predict the future behavior of a system we first require an estimate of its state at the time of prediction. Since our system model is nonlinear, we use the unscented Kalman filter (UKF) [28, 29]. The UKF is based on the unscented transform (UT), which takes a random variable $\mathbf{x} \in \mathbb{R}^{\mathbf{n_x}}$, with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P_{xx}}$, which is related to a second random variable $\mathbf{y}$ by some nonlinear function $\mathbf{y} = \mathbf{g}(\mathbf{x})$, and computes the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P_{yy}}$ using a set of *deterministically* selected weighted samples, called *sigma points* [28]. $\boldsymbol{\mathcal{X}}^i$ denotes the ith sigma point from $\mathbf{x}$ and w^i denotes its weight. The sigma points are always chosen such that the mean and covariance match those of the original distribution, $\bar{\mathbf{x}}$ and $\mathbf{P_{xx}}$. Each sigma point is passed through $\mathbf{g}$ to obtain new sigma points $\boldsymbol{\mathcal{Y}}$, i.e.,

$$\boldsymbol{\mathcal{Y}}^i = \mathbf{g}(\boldsymbol{\mathcal{X}}^\mathbf{i}) \tag{30}$$

with mean and covariance

$$\bar{\mathbf{y}} = \sum_i w^i \boldsymbol{\mathcal{Y}}^i, \tag{31}$$

$$\mathbf{P}_{yy} = \sum_i w^i (\boldsymbol{\mathcal{Y}}^i - \bar{\mathbf{y}})(\boldsymbol{\mathcal{Y}}^i - \bar{\mathbf{y}})^T. \tag{32}$$

The symmetric unscented transform selects $2n_x + 1$ sigma points symmetrically about the mean [29]:

$$w^i = \begin{cases} \dfrac{\kappa}{(n_x + \kappa)}, & i = 0 \\ \dfrac{1}{2(n_x + \kappa)}, & i = 1, \ldots, 2n_x \end{cases}, \tag{33}$$

$$\boldsymbol{\mathcal{X}}^i = \begin{cases} \bar{\mathbf{x}}, & i = 0 \\ \bar{\mathbf{x}} + \left(\sqrt{(n_x+\kappa)\mathbf{P}_{xx}}\right)^i, & i = 1, \ldots, n_x \\ \bar{\mathbf{x}} - \left(\sqrt{(n_x+\kappa)\mathbf{P}_{xx}}\right)^i, & i = n_x+1, \ldots, 2n_x \end{cases}, \tag{34}$$

where $\left(\sqrt{(n_x + \kappa)\mathbf{P}_{xx}}\right)^i$ refers to the ith column of the matrix square root of $(n_x+\kappa)\mathbf{P}_{xx}$. Here, κ is a free parameter that can be used to tune the higher order moments of the distribution. A value of $\kappa = 3 - n_x$ is recommended if $\mathbf{x}$ is assumed to be Gaussian [28].

As with other filters, the UKF operates in two steps, prediction and correction [30]. In the prediction step, sigma points for the states are generated based on the current estimated mean and covariance. The sigma points are passed through the state equation from which new estimates of mean and covariance can be derived from the transformed sigma points. Output sigma points are then computed by passing the transformed state sigma points through the output function. In the update step, the Kalman gain is computed based on the predicted outputs and the measured outputs. The state estimates are then modified based on the Kalman gain. Mathematical details of the filter can be found in [18, 28, 29].

5. PREDICTION

Prediction is initiated at a given time k_P using the current joint state-parameter estimate, $p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P)|\mathbf{y}(k_0{:}k_P))$. The goal is to compute $p(k_E(k_P)|\mathbf{y}(k_0{:}k_P))$ using the state-parameter estimates and assumptions about uncertainty regarding the future parameter, input, and process noise values.

As described in Section 3, we assume all parameters are known, so $\boldsymbol{\theta}(k)$ is empty. Further, we assume that, because the model is so accurate, process noise is negligible. Therefore, only uncertainty in the future inputs is considered. In the following, we describe the approach for the general case as originally developed in [22].

Algorithm 1 $(k_E(k_P), \Delta k_E(k_P), \mathbf{z}(k_E), \Delta\mathbf{z}(k_E)) \leftarrow \mathtt{P}(\mathbf{x}(k_P), \mathbf{\Theta}_{k_P}, \mathbf{U}_{k_P}, \mathbf{V}_{k_P})$

$k \leftarrow k_P$
$\mathbf{x}(k) \leftarrow \mathbf{x}(k_P)$
$\mathbf{z}(k_P) \leftarrow \mathbf{g}(k, \mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k))$
while $T_E(\mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k)) = 0$ **do**
  $\mathbf{x}(k+1) \leftarrow \mathbf{f}(k, \mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k), \mathbf{V}_{k_P}(k))$
  $k \leftarrow k + 1$
  $\mathbf{x}(k) \leftarrow \mathbf{x}(k+1)$
end while
$k_E(k_P) \leftarrow k$
$\Delta k_E(k_P) \leftarrow k - k_P$
$\mathbf{z}(k_E) \leftarrow \mathbf{g}(k, \mathbf{x}(k), \mathbf{\Theta}_{k_P}(k), \mathbf{U}_{k_P}(k))$
$\Delta\mathbf{z}(k_E) \leftarrow \mathbf{z}(k_P)$

For one realization of each of the uncertain quantities at prediction time k_P, i.e., the state $\mathbf{x}(k_P)$, the parameter trajectory $\mathbf{\Theta}_{k_P}$, the input trajectory $\mathbf{U}_{k_P}$, and the process noise trajectory $\mathbf{V}_{k_P}$, we compute the corresponding realization of k_E using the system model with the function $\mathtt{P}$, shown as Algorithm 1. The function $\mathtt{P}$ simulates the system until the event threshold T_E evaluates to 1.

To use $\mathtt{P}$, we need realizations of the state-parameter distribution, the parameter trajectory, the input trajectory, and the process noise trajectory. The distribution for the state comes from the UKF, and the distributions for the parameter, input, and process noise trajectories are defined indirectly by the set of surrogate variables, as described in Section 2. So, we are interested in computing the distribution for k_E and its derived variables from the distributions for $p(\mathbf{x}(k_P), \boldsymbol{\theta}(k_P))$, $p(\boldsymbol{\lambda}_\theta)$, $p(\boldsymbol{\lambda}_u)$, and $p(\boldsymbol{\lambda}_v)$ (see Fig. 1).

To compute the realizations of the surrogate variables for the future input trajectories, we use the unscented transform sampling method [23], in which we sample from the surrogate variable distributions using the unscented transform. The advantages of this method are that (*i*) it is deterministic, since the unscented transform does not sample stochastically; and (*ii*) it is computionally efficient, as the number of samples needed is linear in number of the surrogate variable dimensions. In contrast, Monte Carlo sampling is stochastic, and stochastic algorithms cause difficulties for certification [23], and typically a very large number of samples are required to accurately represent the statistics of the distribution. By using the unscented transform method, we sample a small number of sufficient samples, run them through $\mathtt{P}$, and then reconstruct the statistics of k_E (i.e., mean and variance) from the sigma points transformed via $\mathtt{P}$.

6. RESULTS

We use a rover testbed developed at NASA Ames Research Center [5]. With the rover, both unstructured and structured driving experiments have been performed in the field. In this section, we provide results for both driving scenarios using real data from these experiments in order to demonstrate the prediction methods.

Unstructured Driving

We consider first an unstructured driving scenario, in which it so happens that the rover is simply driven in circles until EOD is reached. From a set of unstructured driving scenarios, we determined that power averaged to 127 W with a standard deviation of about 8.9 W, thus defining the statistics of P_B.

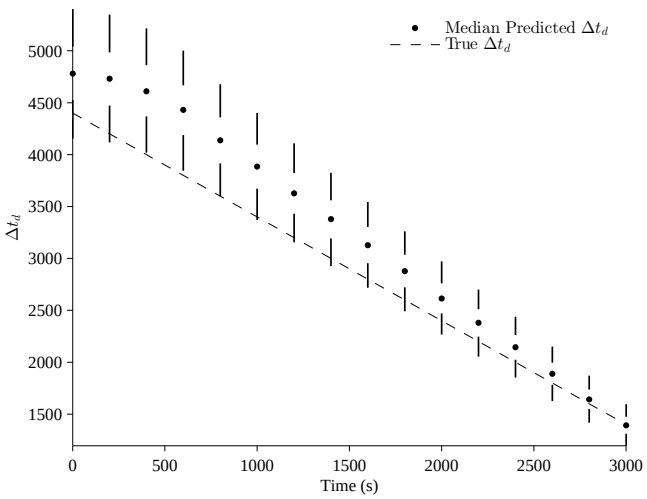

Figure 3. Remaining driving time predictions for unstructured driving.

This power is split between the 24 cells, so the surrogate variable p, introduced in Section 3, is defined with a normal distribution with a mean of $127/24 = 5.3$ W and a standard deviation of $8.9/24 = 0.37$ W.

The predictions for remaining driving time and remaining driving distance are shown in Figs. 3 and 4. Predictions are made every 20 s. Since the rover did not stop, remaining time until discharge and remaining driving time predictions are the same, and so the former are not shown. In each plot, the medians are shown along with the 5–25% and 75–95% percentiles indicated by the black lines.

When we assume that the future power draw is higher than it will be, remaining driving time and distance will be underpredicted, i.e., the predictions will be conservative. On the other hand, when we assume that the future power draw is lower than it will be, remaining driving time and distance will be overpredicted, i.e., the predictions will be overly optimistic. In this scenario, the assumed future power used, based on previous scenarios, is lower than the actual average power, so the median predictions for remaining driving time are biased above the true values, however, the true values are still captured within the uncertainty of the predictions. We can see that the uncertainty is quite large, with the difference between the 5th and 95th percentiles for the first prediction being about 1000 s.

Structured Driving

We next consider a structured driving scenario. The desired waypoints and the path traversed by the rover are shown in Fig. 5. Note that the path taken by the rover between waypoints is not direct; on average, the rover travels 11% farther than the straight-line distance between two consecutive waypoints. On average, the rover requires 140 W of power to travel, with a standard deviation of 9.6 W. The variation is due to differences in terrain and rover handling (including turns, which require more power than going straight). These statistics define those used by the surrogate variables for the constant power draw between waypoints and the relative distance bias.

The predictions for remaining driving time and remaining driving distance are shown in Figs. 6 and 7. The spread is not-

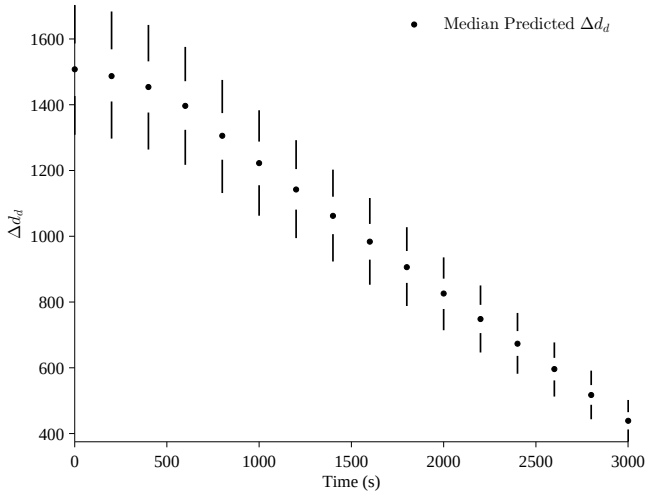

Figure 4. Remaining driving distance predictions for unstructured driving.

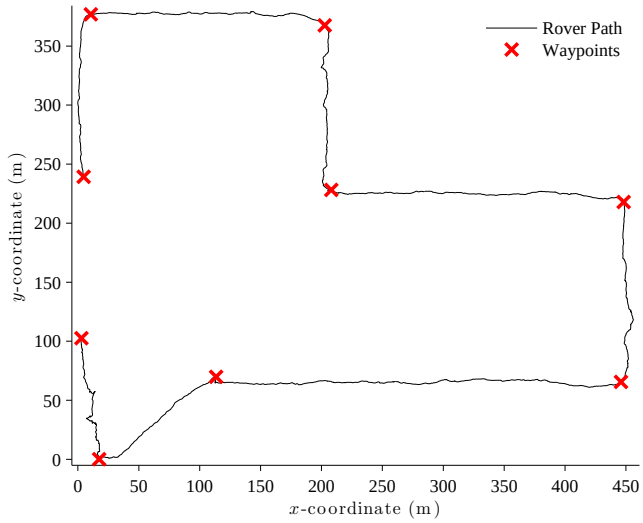

Figure 5. Rover path and waypoints.

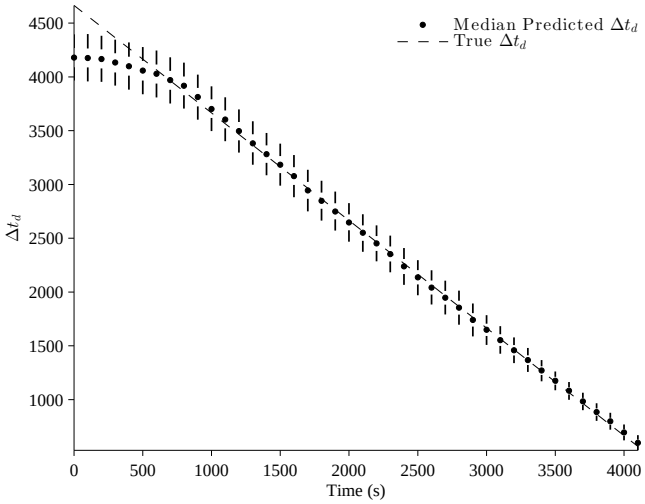

Figure 6. Remaining driving time predictions for structured driving.

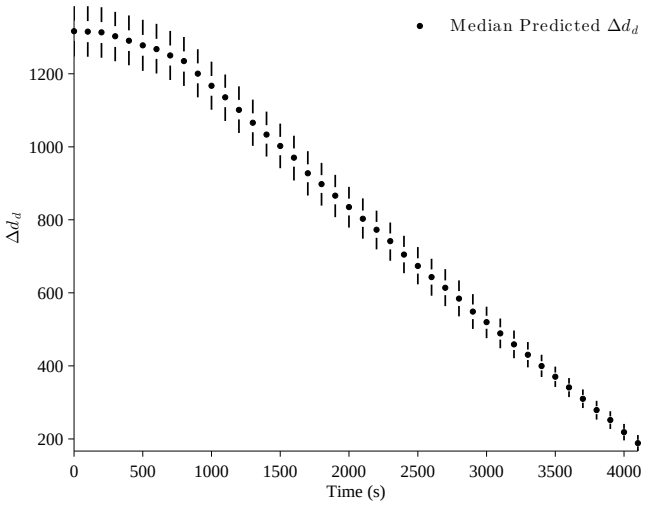

Figure 7. Remaining driving distance predictions for structured driving.

icably smaller than for unstructured driving. As EOD is approached, which occurs around 4600 s, the prediction spread reduces since E is closer and hence there is less time for the uncertainty to spread out. On average, relative accuracy of the predictions is 98%. Initially, remaining driving time and distance are underpredicted because the power drawn in the first 500 s is only about half the average, but predictions are based on the expected average. Therefore the predictions are based on a faster discharge rate and hence remaining driving time and distance are underpredicted. Since the state is being tracked, once the power approaches the expected average, predictions are accurate.

For structured driving, the uncertainty is much less than with unstructured driving. The difference between the 5th and 95th percentiles for the first prediction are about 500 s for structured driving, half that as for unstructured. Although the variance in power drawn for a single segment is larger than that for unstructured driving, uncertainty overall is reduced because it is less likely to be in the extreme for several consecutive segments than for a single segment (which is what is considered for unstructured driving).

7. Conclusions

In this paper, we developed an approach to predict remaining driving time and distance for a rover, extending the framework originally presented in [18]. We considered both unstructured and structured driving scenarios, in which we described how to characterize the uncertainty associated with these scenarios and methods to predict remaining driving time and distance given this information. Experimental results from rover field tests demonstrated the utility of the approach.

In this work, we also assumed that all batteries were at the same age, thus having the same capacity, which was a reasonable assumption in this case, since all the batteries were fairly new. In reality, batteries age and thus each battery has a different discharge rate, and this has an effect on the system-level EOD. In that case, the battery models must include aging dynamics [31], and in future work, aging pa-

rameters should be estimated online. Further, if each battery behaves differently, then a distributed prognostics approach is required [32]. Diagnostics must be integrated as well, because faults will change the system state and the power demands on the rover, thus modifying the predictions [24].

ACKNOWLEDGMENTS

This work was funded in part by the NASA System-wide Satefy Assurance Technologies (SSAT) project under the Aviation Safety (AvSafe) Program of the Aeronautics Research Mission Directorate (ARMD). The authors also acknowledge Adam Sweet, NASA Ames Research Center, and Brian Bole, Indranil Roychoudhury, and George Gorospe, SGT, Inc., for obtaining field test data.

REFERENCES

[1] F. Ingrand and M. Ghallab, "Robotics and artificial intelligence: A perspective on deliberation functions," *AI Communications*, vol. 27, no. 1, pp. 63–80, 2014.

[2] E. Balaban, S. Narasimhan, M. Daigle, J. Celaya, I. Roychoudhury, B. Saha, S. Saha, and K. Goebel, "A mobile robot testbed for prognostics-enabled autonomous decision making," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2011*, September 2011, pp. 15–30.

[3] L. Tang, E. Hettler, B. Zhang, and J. Decastro, "A testbed for real-time autonomous vehicle PHM and contingency management applications," in *Annual Conference of the Prognostics and Health Management Society 2011*, Oct. 2011, pp. 56–66.

[4] S. Narasimhan, E. Balaban, M. Daigle, I. Roychoudhury, A. Sweet, J. Celaya, and K. Goebel, "Autonomous decision making for planetary rovers using diagnostic and prognostic information," in *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Aug. 2012, pp. 289–294.

[5] E. Balaban, S. Narasimhan, M. Daigle, I. Roychoudhury, A. Sweet, C. Bond, and G. Gorospe, "Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms," *International Journal of Prognostics and Health Management*, vol. 4, no. 1, May 2013.

[6] A. Sweet, G. Gorospe, M. Daigle, J. Celaya, E. Balaban, I. Roychoudhury, and S. Narasimhan, "Demonstration of prognostics-enabled decision making algorithms on a hardware mobile robot test platform," in *Annual Conference of the Prognostics and Health Management Society 2014*, September 2014.

[7] J. Oliva, C. Weihrauch, and T. Bertram, "A model-based approach for predicting the remaining driving range in electric vehicles," in *Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 438–448.

[8] B. Saha, C. C. Quach, and K. Goebel, "Optimizing battery life for electric UAVs using a Bayesian framework," in *Proceedings of the 2012 IEEE Aerospace Conference*, Mar. 2012.

[9] B. Saha, K. Goebel, S. Poll, and J. Christophersen, "An integrated approach to battery health monitoring using Bayesian regression and state estimation," in *2007 IEEE Autotestcon*, 2007, pp. 646–653.

[10] M. Daigle, A. Bregon, and I. Roychoudhury, "A distributed approach to system-level prognostics," in *Annual Conference of the Prognostics and Health Management Society*, Sep. 2012, pp. 71–82.

[11] M. Daigle and K. Goebel, "Model-based prognostics with concurrent damage progression processes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, pp. 535–546, May 2013.

[12] 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, Jun. 2009.

[13] 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*, Sep. 2009.

[14] C. S. Kulkarni, J. R. Celaya, G. Biswas, and K. Goebel, "Towards a model-based prognostics methodology for electrolytic capacitors: A case study based on electrical overstress accelerated aging," *International Journal of Prognostics and Health Management*, vol. 5, no. 1, p. 16, 2012.

[15] M. Daigle and K. Goebel, "A model-based prognostics approach applied to pneumatic valves," *International Journal of Prognostics and Health Management*, vol. 2, no. 2, August 2011.

[16] C. Kulkarni, M. Daigle, G. Gorospe, and K. Goebel, "Validation of model-based prognostics for pneumatic valves in a demonstration testbed," in *Annual Conference of the Prognostics and Health Management Society 2014*, September 2014.

[17] C. Quach, B. Bole, E. Hogge, S. Vazquez, M. Daigle, J. Celaya, A. Weber, and K. Goebel, "Battery charge depletion prediction on an electric aircraft," in *Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 503–512.

[18] M. Daigle and C. Kulkarni, "A battery health monitoring framework for planetary rovers," in *2014 IEEE Aerospace Conference*, March 2014.

[19] S. Sankararaman, "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction," *Mechanical Systems and Signal Processing*, 2014.

[20] S. Sankararaman, M. Daigle, A. Saxena, and K. Goebel, "Analytical algorithms to quantify the uncertainty in remaining useful life prediction," in *Proceedings of the 2013 IEEE Aerospace Conference*, Mar. 2013.

[21] S. Sankararaman, M. Daigle, and K. Goebel, "Uncertainty quantification in remaining useful life prediction using first-order reliability methods," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 603–619, June 2014.

[22] M. Daigle and S. Sankararaman, "Advanced methods for determining prediction uncertainty in model-based prognostics with application to planetary rovers," in *Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 262–274.

[23] M. Daigle, A. Saxena, and K. Goebel, "An efficient deterministic approach to model-based prediction uncertainty estimation," in *Annual Conference of the Prognostics and Health Management Society*, Sep. 2012, pp. 326–335.

[24] M. Daigle, I. Roychoudhury, and A. Bregon, "Integrated

diagnostics and prognostics for the electrical power system of a planetary rover," in *Annual Conference of the Prognostics and Health Management Society 2014*, September 2014.

[25] M. Daigle and C. Kulkarni, "Electrochemistry-based battery modeling for prognostics," in *Annual Conference of the Prognostics and Health Management Society 2013*, Oct. 2013, pp. 249–261.

[26] C. D. Rahn and C.-Y. Wang, *Battery Systems Engineering*. Wiley, 2013.

[27] D. K. Karthikeyan, G. Sikha, and R. E. White, "Thermodynamic model development for lithium intercalation electrodes," *Journal of Power Sources*, vol. 185, no. 2, pp. 1398–1407, 2008.

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

[29] ——, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[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*, March 2012.

[31] B. Bole, C. Kulkarni, and M. Daigle, "Adaptation of an electrochemistry-based li-ion battery model to account for deterioration observed under randomized use," in *Annual Conference of the Prognostics and Health Management Society 2014*, September 2014.

[32] M. Daigle, A. Bregon, and I. Roychoudhury, "Distributed prognostics based on structural model decomposition," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 495–510, June 2014.

BIOGRAPHY

Matthew Daigle *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. 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 and the IEEE.*

Shankar Sankararaman *received his B.S. degree in Civil Engineering from the Indian Institute of Technology, Madras in India in 2007 and later, obtained his Ph.D. in Civil Engineering from Vanderbilt University, Nashville, Tennessee, U.S.A. in 2012. His research focuses on the various aspects of uncertainty quantification, integration, and management in different types of aerospace, mechanical, and civil engineering systems. His research interests include probabilistic methods, risk and reliability analysis, Bayesian networks, system health monitoring, diagnosis and prognosis, decision-making under uncertainty, treatment of epistemic uncertainty, and multidisciplinary analysis. He is a member of the American Institute of Aeronautics (AIAA), the American Society of Civil Engineers (ASCE), the Institute for Electrical and Electronics Engineers (IEEE), and the Prognostics and Health Management (PHM) Society. Currently, Shankar is a researcher at NASA Ames Research Center, where he develops algorithms for uncertainty assessment and management in the context of system health monitoring, prognostics, and decision-making.*

Chetan S. Kulkarni *received the B.E. (Bachelor of Engineering) degree in Electronics and Electrical Engineering from University of Pune, India in 2002 and the M.S. and Ph.D. degrees in Electrical Engineering from Vanderbilt University, Nashville, TN, in 2009 and 2013, respectively. In 2002 he joined Honeywell Automation India Limited (HAIL) as a Project Engineer. From May 2006 to August 2007 he was a Research Fellow at the Indian Institute of Technology (IIT) Bombay with the Department of Electrical Engineering. From Aug 2007 to Dec 2012, 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. Since January 2013 he has been a Research Engineer II with SGT Inc. at the Prognostics Center of Excellence, NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis focused towards electrical and electronic devices and systems. Dr. Kulkarni is a member of the Prognostics and Health Management (PHM) Society, AIAA and the IEEE.*