

# A Methodology for Updating Prognostic Models via Kalman Filters

Venkatesh Rajagopalan<sup>1</sup> and Arun Subramaniyan<sup>2</sup>

<sup>1</sup> *Prognostics Laboratory, GE Global Research Center, Bangalore, Karnataka, 560078, India*  
*venkatesh.rajagopalan@ge.com*

<sup>2</sup> *Structures Laboratory, GE Global Research Center, Niskayuna, New York, 12309, USA*  
*arun.subramaniyan@ge.com*

## ABSTRACT

Prognostic models are built to predict the future evolution of the state or health of a system. Typical applications of these models include predictions of damage (like crack, wear) and estimation of remaining useful life of a component. Prognostic models may be data based, based on known physics of the system or can be hybrid, i.e., built through a combination of data and physics. To build such models, one needs either data from the field (i.e., real-world operations) or simulations/tests that qualitatively represent field observations. Often, field data is not easy to obtain and is limited in its availability. Thus, models are built with simulation or test data and then validated with field observations when they become available. This necessitates a procedure that allows for refinement of models to better represent real-world behavior without having to run expensive simulations or tests repeatedly. Further, a single prognostic model developed for an entire fleet may need to be updated with measurements obtained from individual units. In this paper, we describe a novel methodology, based on the Unscented Kalman Filter, that not only allows for updating such “fleet” models, but also guarantees improvement over the existing model.

## 1. INTRODUCTION

Prognostic models are built to make predictions on the future evolution of a system. Applications of these models include prediction of cracks, wear, scrap rate and estimation of remaining useful life in industrial components. These prognostic models can be completely data-driven, based on known physics of the system or can be hybrid, i.e., a combination of both data and physics.

Prognostic models can be built using field (real-world) data, if available, or with simulation/test data. Often, field data is not available or may be inadequate (i.e., not enough data points)

Venkatesh Rajagopalan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

for model building. In such scenarios, prognostic models are built with simulation data, like Design of Experiments (DOE) data, or test data. These models are then validated and refined with field data when it becomes available. Thus, there exists a requirement for a methodology that allows for model updating even with limited field data<sup>1</sup>. One of the techniques that can be used for model or transfer function updating is the Kalman Filter (Kalman, 1960) and its variants, namely the Extended Kalman Filter (EKF) (Swering, 1959) and the Unscented Kalman Filter (UKF) (Julier, Uhlmann, & Durrant-Whyte, 1995).

The Extended, Unscented and Ensemble Kalman Filters have been used for history matching and continuous model updating in diverse fields such as Petroleum Engineering (Ning & Oliver, 2005), Meteorology (Houtekamer & Mitchell, 2001) as well as for gas turbine performance diagnostics (Volponi, Ganguli, & Daguang, 2003). In all these examples, the model being updated represented a specific reservoir or an aircraft engine. The data for updating the model was obtained from a single source (either the reservoir or the engine) and it was ordered in time, i.e., a time series. Thus, the standard Kalman Filter framework is applicable as-is to these problems.

In this paper, we consider the problem of updating a model that represents, not a single entity, but all entities of a class. An example is a model built to predict cracks or scrap percentage in a fleet of gas turbines or aircraft engines. The model parameters are the same for all units of the fleet with the operational history of the units being the inputs of the model. The predictions of these models are usually verified through inspections during outages or shop visits. The challenge is to update the parameters of such “fleet” models based on field inspection measurements obtained from various units of the fleet. Given that there is no notion of (time) sequence in the measurements, i.e., there’s no ordering of data from different units of the fleet, and the fact that these measure-

<sup>1</sup>Re-building models with field data alone, when it becomes available, is generally not feasible since field data is usually very limited in quantity at a given time. Also, it is often desired to retain the model “kernel”, but modify the coefficients using field data.

ments become available in a batch, the conventional Kalman Filter framework does not apply. We present a methodology that modifies the Kalman Filter based model updating framework to allow for updating “fleet” models by utilizing measurements from different entities of the fleet. Further, the methodology is designed to ensure improvement in model accuracy after the update.

The rest of the paper is organized as follows: Section 2 provides a brief review of the Unscented Kalman Filter. Section 3 discusses the problem of updating “fleet” models and describes a novel methodology based on Kalman Filtering. In Section 4, the efficacy of the proposed methodology is illustrated with a simulated and a real-world example. Section 5 provides a summary of the paper.

Throughout the paper, scalars and vectors are respectively represented as small letters and small, bold letters. Furthermore,  $I_n$  denotes an  $n \times n$  identity matrix. Additionally, the symbol  $\mathcal{R}$  refers to the 1-dimensional space over the field of real numbers,  $\mathcal{R}^n$  refers to the  $n$ -dimensional space over the field of real numbers and  $\mathcal{R}^{n \times m}$  refers to the  $n \times m$  dimensional rectangular matrix real space (square if  $n = m$ ). The notation  $\mathcal{N}(\mu, \sigma)$  represents a normal/Gaussian distribution of mean  $\mu$  and standard deviation  $\sigma$ . The symbol  $\mathcal{D}$  refers to a compact region of the space of appropriate dimension. Thus  $\mathcal{D}_x \in \mathcal{R}$  represents a compact region of the variable  $x$  which is a subset of the state space of appropriate dimension. The operator  $E[x]$  denotes the expectation of the random variable  $x$ .

## 2. KALMAN FILTERS: A BRIEF REVIEW

The availability of all state variables for direct measurement is a rare. In physical systems, some components of the state are inaccessible internal variables, which either cannot be measured or the measurements require the use of very costly measurement devices. Hence in most practical scenarios, there is a true need to construct estimates of the unknown state variables via known measurements, albeit noisy. In the case of systems that are linear in the process and measurement, and which are corrupted by white process noise and measurement noise, the linear Kalman Filter (KF) offers a recursive solution in the sense of minimizing the trace of the error covariance of the system states (Kalman, 1960), (Kalman & Bucy, 1961), (Brown & Hwang, 1992). However, in the case of systems which are inherently nonlinear in either the process or the measurement or both, straightforward implementation of the KF is not guaranteed to yield optimum results in the sense of minimizing the root mean square of the estimation error. Thus the need to be able to effectively reconstruct the unknown states of a nonlinear system has promoted research in nonlinear filtering theory (Nijmeijer & Fossen, 1999).

Of the numerous attempts being made for the development of nonlinear filters, the Extended Kalman Filter, was the earliest

and the most prevalent approach. The design of the EKF is based on a first order local linearization of the system around the current state estimate at each time step (Eykhoff, 1974), (Jazwinski, 1970), (Daum, 2005). The first ever implementation of an EKF is credited to Peter Swerling (Brookner, 2001), and was called as the Swerling Filter for filtering problems (Swerling, 1959). This approach approximates the nonlinear equations by a Taylor series of up to the first order. The well known KF (Kalman, 1960) equations can then be applied to the linearized system to compute the Kalman gain and the covariance matrices. To address the limitations inherent in an EKF<sup>2</sup>, the Unscented Kalman Filter (UKF) (Julier et al., 1995), (Julier & Uhlmann, 1997) was developed, which neither relies on the linearization steps required by the EKF nor the computation of Jacobian matrices. Instead, the UKF uses a deterministic sampling approach to estimate the mean and covariance with a minimal set of sample points.

### 2.1. Discrete Nonlinear Time Invariant System

Consider the following observable<sup>3</sup>, discrete, nonlinear dynamical system:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k, & k \geq 1 \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \end{aligned} \quad (1)$$

where,  $\mathbf{x}_k \in \mathcal{D}_x \subset \mathcal{R}^n$  denotes the  $n$ -dimensional state vector of the system,  $\mathbf{u}_k \in \mathcal{D}_u \subset \mathcal{R}^m$  denotes the  $m$ -dimensional known input vector,  $\mathbf{f}(\cdot) : (\mathcal{D}_x \subset \mathcal{R}^n) \rightarrow \mathcal{R}^n$  is a finite nonlinear mapping of the system states and system input,  $\mathbf{z}_k \in \mathcal{D}_z \subset \mathcal{R}^p$  denotes the  $p$ -dimensional system measurement,  $\mathbf{h}(\cdot, \cdot) : (\mathcal{D}_x \subset \mathcal{R}^n) \times (\mathcal{D}_u \subset \mathcal{R}^m) \rightarrow \mathcal{R}^p$  is a nonlinear mapping of the system states to the output,  $\mathbf{w}_k \in \mathcal{D}_w \subset \mathcal{R}^w$  denotes the  $w$ -dimensional random process noise vector and  $\mathbf{v}_k \in \mathcal{D}_v \subset \mathcal{R}^v$  denotes the  $v$ -dimensional random measurement noise vector. The process and measurement noise are assumed to be zero mean, band-limited, uncorrelated, additive, white Gaussian noise processes such that:

$$E[\mathbf{w}_k \mathbf{w}_j^T] = Q_k \delta_{kj} = \begin{cases} Q_k, & k = j \\ 0, & k \neq j \end{cases}$$

$$E[\mathbf{v}_k \mathbf{v}_j^T] = R_k \delta_{kj} = \begin{cases} R_k, & k = j \\ 0, & k \neq j \end{cases}$$

$$E[\mathbf{w}_k \mathbf{v}_j^T] = E[\mathbf{w}_k] E[\mathbf{v}_j^T] = 0,$$

where,  $Q_k$  is the process noise covariance,  $R_k$  is the measurement noise covariance and  $\delta_{kj}$  is the Kronecker delta function (discrete equivalent of the Dirac delta function). The Gaussian random variables  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are commonly denoted as  $\mathbf{w}_k \sim \mathcal{N}(0, Q_k)$  and  $\mathbf{v}_k \sim \mathcal{N}(0, R_k)$  respectively, where

<sup>2</sup>A major limitation of the EKF is that it approximates the expected value of a nonlinear function  $f(x)$  as a function of the expected value, i.e.,  $E[f(x)] \approx f(E[x])$ .

<sup>3</sup>For the definition on observability of nonlinear systems, refer (Isidori, 1995).

$w_k$  and  $v_k$  are 0 mean distributions and of variance  $Q_k$  and  $R_k$  respectively. The initial state of the system in Eq.(1) is assumed to be a Gaussian random vector with mean  $\bar{x}_0$  and covariance  $\bar{P}_0$  and can be denoted as  $x_0 \sim \mathcal{N}(\bar{x}_0, \bar{P}_0)$ .

### 2.1.1. The Unscented Kalman Filter

The Unscented Transform (UT) is a mathematical function used to estimate the result of applying a given nonlinear transformation to a probability distribution that is characterized only in terms of a finite set of statistics. In other words, the UT approximates a Gaussian distribution with a set of deterministically chosen sample points. These points completely capture the mean and covariance of the Gaussian distribution such that when propagated through a nonlinear transformation map, the transformed points accurately capture the mean and covariance of the new density (due to the transformation) up to the third order. The Unscented Kalman Filter (UKF), which is based on propagating the mean and covariance through the UT (Julier et al., 1995), is a method for calculating the statistics of a random variable that undergoes a nonlinear transformation.

Consider the observable, discrete, nonlinear dynamical system in Eq.(1). Initialize the state and state error covariance estimates to  $\hat{x}_0$  and  $P_0$ , respectively. Let  $n$  denote the state dimension. For each time index  $k \geq 1$ , the typical UKF predict-update steps are:

1. Generate sigma points  $\xi_{k-1}$ , based on  $\hat{x}_{k-1}$  and  $P_{k-1}$ , as:

$$\begin{aligned}\xi_{k-1}^{(0)} &= \hat{x}_{k-1} \\ \xi_{k-1}^{(i)} &= \hat{x}_{k-1} + \gamma \left( \sqrt{P_{k-1}} \right)_i \\ \xi_{k-1}^{(i+n)} &= \hat{x}_{k-1} - \gamma \left( \sqrt{P_{k-1}} \right)_i\end{aligned}$$

where,  $i = 1, \dots, n$ ,  $\gamma = \sqrt{n + \lambda}$  and  $\lambda = \alpha^2 (n + \kappa) - n$  such that  $10^{-4} \leq \alpha \leq 1$ , where  $\alpha$  determines the spread of the sigma points around the mean,  $\kappa$  is a secondary scaling parameter and  $\left( \sqrt{P_{k-1}} \right)_i$  denotes the  $i^{th}$  column (or  $i^{th}$  row transpose) of the matrix  $\sqrt{P_{k-1}}$ . If  $P_{k-1}$  is positive definite, the Cholesky decomposition can be used to obtain the square root. If  $P_{k-1}$  is positive semi-definite, perform an eigen decomposition to identify the eigen values that are 0, reset those eigen values alone to a very small positive number (E.g.  $10^{-8}$ ), regroup the eigen value and eigen vector matrices to obtain  $P_{k-1}$  and then use Cholesky decomposition.

2. At time index  $k$ , PREDICT:  
Sigma points as:

$$\xi_k^{-(j)} = f \left( \xi_{k-1}^{(j)} \right), \quad j = 0, \dots, 2n \quad (2)$$

State estimate  $\hat{x}_k^-$  as:

$$\hat{x}_k^- = \sum_{j=0}^{2n} W_j^{(m)} \xi_k^{-(j)}, \quad (3)$$

with  $W_0^{(m)} = \frac{\lambda}{n + \lambda}$ ,  $W_j^{(m)} = \frac{1}{2(n + \lambda)}$

State error covariance matrix as:

$$P_k^- = \sum_{j=0}^{2n} W_j^{(c)} \left( \xi_k^{-(j)} - \hat{x}_k^- \right) \left( \xi_k^{-(j)} - \hat{x}_k^- \right)^T + Q_k \quad (4)$$

where,  $W_0^{(c)} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta)$  and  $W_j^{(c)} = \frac{1}{2(n + \lambda)}$  and  $\beta$  is used to incorporate prior knowledge of the distribution of  $x$  ( $\beta = 2$  for Gaussian is optimal).

Recalculate the sigma points based on  $\hat{x}_k^-$  and  $P_k^-$  to incorporate the effect of process noise as:

$$\begin{aligned}\chi_k^{-(0)} &= \hat{x}_k^- \\ \chi_k^{-(i)} &= \hat{x}_k^- + \gamma \left( \sqrt{P_k^-} \right)_i \\ \chi_k^{-(i+n)} &= \hat{x}_k^- - \gamma \left( \sqrt{P_k^-} \right)_i\end{aligned}$$

Measurement as:

$$\hat{z}_k^- = \sum_{j=0}^{2n} W_j^{(m)} \mathcal{Z}_k^{-(j)}, \quad (5)$$

with,  $\mathcal{Z}_k^{-(j)} = h \left( \chi_k^{-(j)}, \mathbf{u}_k \right)$ ,  $j = 0, \dots, 2n$

3. At time index  $k$ , UPDATE: Innovation covariance as:

$$P_{\hat{z}_k \hat{z}_k} = \sum_{j=0}^{2n} W_j^{(c)} \left( \mathcal{Z}_k^{-(j)} - \hat{z}_k^- \right) \left( \mathcal{Z}_k^{-(j)} - \hat{z}_k^- \right)^T + R_k \quad (6)$$

Cross covariance between  $\hat{x}_k^-$  and  $\hat{z}_k^-$  as:

$$P_{\hat{x}_k^- \hat{z}_k} = \sum_{j=0}^{2n} W_j^{(c)} \left( \chi_k^{-(j)} - \hat{x}_k^- \right) \left( \mathcal{Z}_k^{-(j)} - \hat{z}_k^- \right)^T \quad (7)$$

State estimate as:

$$\hat{x}_k = \hat{x}_k^- + K_k \left( z_k - \hat{z}_k^- \right) \quad (8)$$

with,  $K_k = P_{\hat{x}_k^- \hat{z}_k} P_{\hat{z}_k \hat{z}_k}^{-1}$

State error covariance matrix as:

$$P_k = P_k^- - K_k P_{\hat{z}_k \hat{z}_k} K_k^T \quad (9)$$

**Remark 2.1** Although, theoretically, the matrix  $P_k$  is symmetric, for real data this condition might be violated. Therefore, after every update step it is necessary to force a symmet-

ric constraint as  $P_k = \frac{P_k + P_k^T}{2}$ .

**Remark 2.2** For details on the iterative UKF, refer (Zhan & Wan, 2007).

### 3. UPDATING MODELS WITH KALMAN FILTERS

In this section, we describe the methodology for updating models with the Unscented Kalman Filters. The model of the measured output is given in Eq.(10)

$$\mathbf{z}_k = \mathbf{h}(\Theta_k, \mathbf{u}_k) + \mathbf{v}_k \quad (10)$$

The function  $\mathbf{h}$  could be as simple as a single equation, can be a system of equations or may represent a series of computational steps resulting in the output  $\mathbf{z}$ . The parameter vector  $\Theta$ , given by

$$\Theta = [\Theta^1, \Theta^2, \dots, \Theta^n] \quad (11)$$

represents the coefficients associated with various terms in the model and  $\mathbf{u}$  represents the inputs to the model. The inputs, for example, could be temperature, stress, pressure or any other measured input to the system. The purpose of the update methodology is to modify the values of the coefficients, utilizing the actual measurements of the output and the model predictions, in a Kalman Filter framework. To achieve this, we postulate the parameter vector  $\Theta$  to represent the “states” of the filter. The states are modeled as random walk process and hence we obtain the process and measurement equations as:

$$\begin{aligned} \Theta_k &= \Theta_{k-1} + \mathbf{w}_k, \quad k \geq 1 \\ \mathbf{z}_k &= \mathbf{h}(\Theta_k, \mathbf{u}_k) + \mathbf{v}_k \end{aligned} \quad (12)$$

While such a formulation is standard for model updating with Kalman Filters, it should be noted that the procedure described in literature cannot be applied as-is to updating “fleet” models. This is due to the fact that there is no ordering when measurements are obtained from different units of the fleet. Therefore, one can easily end up with a completely different updated model if the order of measurements used for updating is changed.

Also, with the standard procedure, the model gets updated to minimize the error between the model prediction and the measurement at that point (in the “fleet” case, measurement obtained from a particular unit). This is acceptable, as long as the latest measurement represents the most current information obtained about the system. Since this is not the case with measurements used for updating “fleet” models, standard procedure may result in an updated model that is worse than the original model in terms of overall performance against all measurements. Therefore, there is a need for a methodology that ensures an improvement in prediction accuracy for the updated model when compared to the original model.

In the proposed methodology, the available data is split into

training and testing data set. Then, the training data set is permuted  $M$  times to create different training data sets. The  $M$  training data sets contain the same measurement points, but the ordering in each set is different. The original model is updated with each of the permuted training data sets to generate  $M$  updated models. To guarantee, that an updated model is always better (or at least no worse) than the original/previous model, the original/previous model is updated with a measurement only if the updated model performance (for example, in terms of overall average error computed across all training data), is better than that of the previous model.

A limited comparison of this approach can be made with stepwise regression. In stepwise regression, an existing model is modified at each step, by adding a term in case of forward stepwise or by removing a term with backward stepwise, only if such an addition or deletion improves the existing model. Similarly, in our “stepwise” model updating approach, an existing model is a modified at a step (in this case, a measurement in the training set) only if the modification results in a more accurate model. Otherwise, the modification is discarded and the existing model is retained.

Such a selective updating process can be effective in mitigating the effect of excessively noisy measurements and outliers in the data. This process is depicted in Figure 1. Each of the  $M$  updated models is validated with the test data set and the model that provides the least test error is chosen as the updated model. Figure 2 provides an illustration of the overall procedure.

The salient aspects of the model update procedure can be summarized as:

- Separation of data for training (updating) and testing
- Permutations of training data to generate multiple updated models
- Monotonic reduction in error during the update
- Validation with test data to select the best updated model
- Guarantee that the updated model is no worse than the original model (across all data).

### 4. RESULTS AND DISCUSSION

In this section, the analysis and results of applying the update methodology to a simulated and a real-world example are presented. The simulated example is a non-linear model with two inputs and an output. The real-world example is a prognostic model for scrap rate prediction in an industrial gas turbine. It is noted that measurements from several individual gas turbines are used for updating this model.

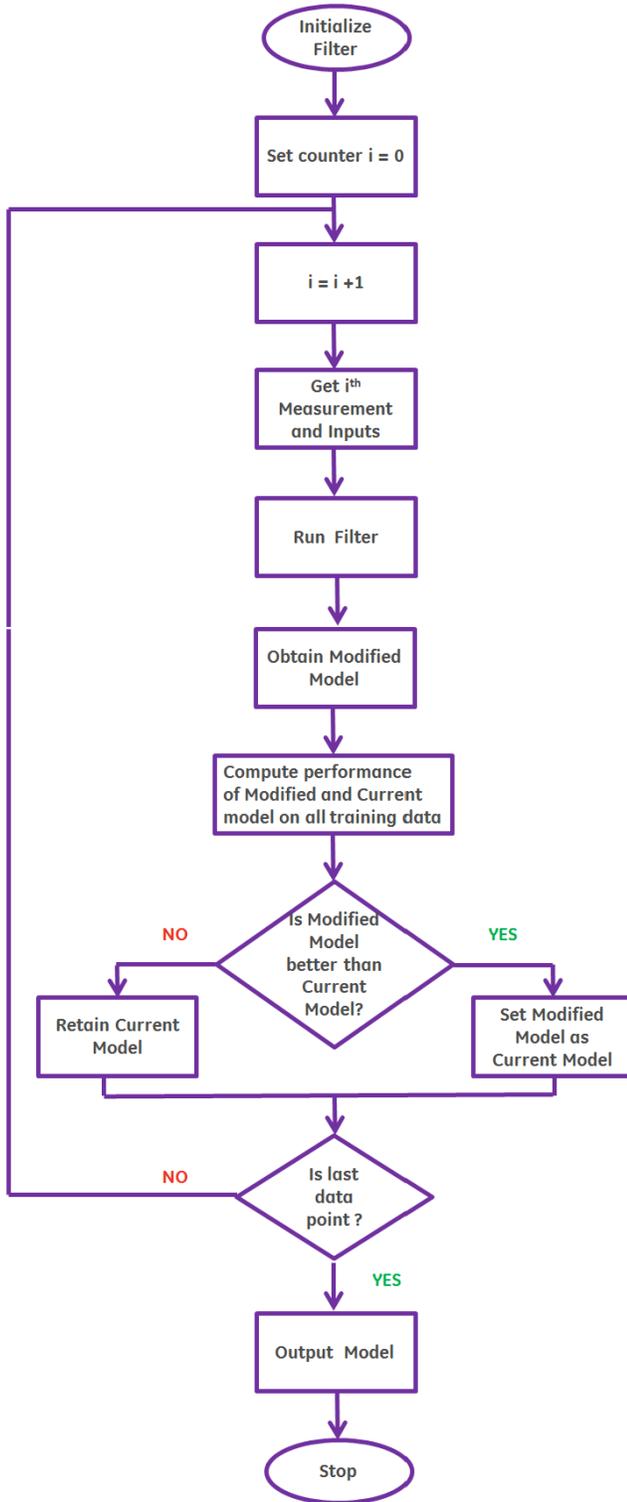


Figure 1. Model update algorithm

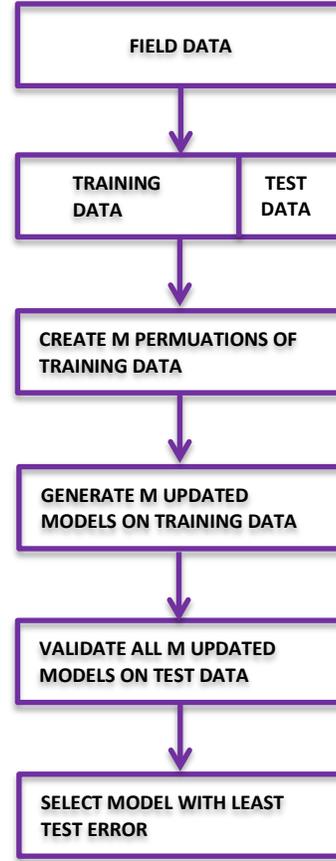


Figure 2. Methodology for model update

#### 4.1. Updating a Nonlinear Model

A model of the form given in Eq.(13) is considered for updating with the Unscented Kalman Filter.

$$z = \theta^1 + \theta^2 x_1 + \theta^3 x_2 + \theta^4 x_1 x_2 + \theta^2 \theta^4 x_1^2 \quad (13)$$

where  $x_1$  and  $x_2$  are the inputs and  $z$  is the output.

The model coefficients  $\{\theta^i\}$ ,  $i = \{1, 2, 3, 4\}$  are given by

$$\Theta = [\theta^1, \theta^2, \theta^3, \theta^4] \quad (14)$$

It is noted that the model is non-linear both in the coefficients as well as the inputs. The initial model is thus represented as  $\Theta_0$ . The coefficient values of the initial model are provided in Table 1.

##### 4.1.1. Generating Data for Update

This section describes the process by which the data for updating the initial model was generated. The model parameters used for the simulation are given in Table 2.

The inputs,  $x_1$  and  $x_2$ , were sampled randomly from a Uni-

Table 1. Initial model:  $\Theta_0$ 

Coefficient	Value
$\Theta^1$	0.20
$\Theta^2$	0.20
$\Theta^3$	0.10
$\Theta^4$	0.40

Table 2. Model parameters for generating update data

Coefficient	Value
$\Theta^1$	0.35
$\Theta^2$	0.25
$\Theta^3$	0.15
$\Theta^4$	0.40

form distribution of  $[0, 0.5]$ . A total of 35 data points were generated for  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . The model output is then obtained with generated inputs as per Eq.(13). Additive White Gaussian Noise (AWGN) is then added to the model output to simulate the effect of measurement noise. This noisy output is treated as “actual” measurement  $\tilde{\mathbf{z}}$ . 25 out of the 35 data points were corrupted with zero-mean AWGN with a standard deviation of 0.01. The remaining 10 points were corrupted with zero-mean AWGN with a standard deviation of 0.1. This is done to explore the robustness of the proposed update methodology vis-a-vis the standard Kalman Filter approach with respect to excessively noisy measurements and outliers. The  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\tilde{\mathbf{z}}$  values corresponding to the data set are provided in Table 3. Out of the 35 data points, 80% of the data (28 points) were randomly chosen as the training data. The remaining 7 points constitute the test data. The training data and test data remain the same for the all experimental results presented subsequently in this paper.

#### 4.1.2. Updating with Standard UKF

The plot of predictions of the initial model  $\Theta^0$ , for the update data set, against the “actual” values (obtained through simulation that was described earlier) is shown in Figure 3.

From the figure, it is obvious that the initial model predictions are not very accurate. The prediction bounds are shown merely to visualize the data in an easy to interpret manner. The average absolute error, for the training data, was found to be 0.1649 and that for the test data was found to be 0.1989. Also, 34/35 predictions lie beyond the prediction bounds. Therefore, this model accuracy can be improved with an update. Further, it is also desirable if the updated values of the model coefficients are close to the “true” values given in Table 2.

To begin with, the initial is updated with a standard UKF, i.e., no multiple runs with shuffled training data and no checking on whether an update with a measurement improves overall

Table 3. Data for model update

Sample	$\mathbf{x}_1$	$\mathbf{x}_2$	$\tilde{\mathbf{z}}$
1	0.2565	0.4669	0.3277
2	0.2302	0.1134	0.4710
3	0.1752	0.3930	0.4045
4	0.0475	0.2054	0.3959
5	0.2168	0.0597	0.4092
6	0.3546	0.3172	0.5365
7	0.0580	0.4312	0.4584
8	0.0390	0.0791	0.3436
9	0.1846	0.3006	0.4767
10	0.0168	0.0588	0.3515
11	0.0961	0.3130	0.3836
12	0.2357	0.4176	0.6718
13	0.0725	0.4702	0.4441
14	0.3589	0.2078	0.5145
15	0.3309	0.1360	0.4845
16	0.2159	0.4640	0.5224
17	0.2230	0.4607	0.5274
18	0.2542	0.2710	0.4866
19	0.2640	0.4064	0.5350
20	0.2864	0.0832	0.4559
21	0.1804	0.1602	0.4210
22	0.1682	0.3289	0.4760
23	0.0866	0.0009	0.3791
24	0.0431	0.3145	0.4456
25	0.1967	0.3926	0.4736
26	0.4022	0.1473	0.6683
27	0.0055	0.3948	0.4164
28	0.1166	0.1079	0.4086
29	0.2016	0.0714	0.4170
30	0.4012	0.2887	0.5565
31	0.4311	0.1280	0.5139
32	0.0719	0.0232	0.3752
33	0.1725	0.1355	0.4339
34	0.2062	0.3389	0.4903
35	0.3551	0.1097	0.3610

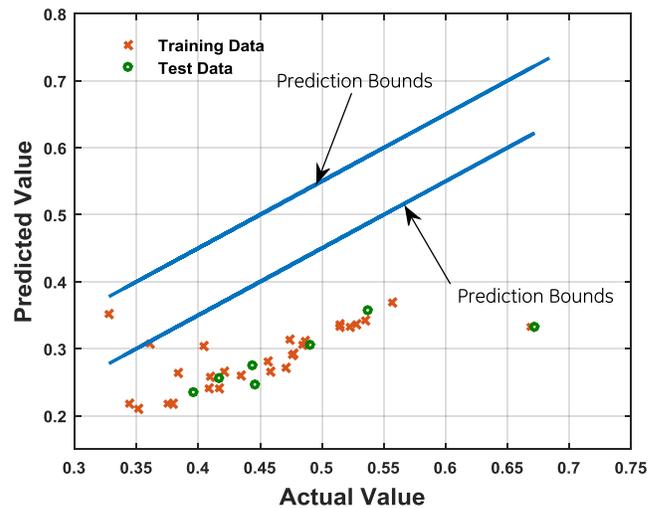


Figure 3. Initial model: actual Vs prediction

accuracy. The parameters to be updated (“the states”) are modeled as a random walk process as in Eq.(12). The process noise covariance  $Q$  was assumed to be diagonal matrix with a value of  $1e-4$  for the elements in the diagonal. The measurement noise variance  $R$  is assumed to be  $1e-4$ . The UKF parameters are chosen as  $\alpha = 1$ ,  $\beta = 2$  and  $\kappa = 0$ . The number of iterations of the filter is chosen to be 1.

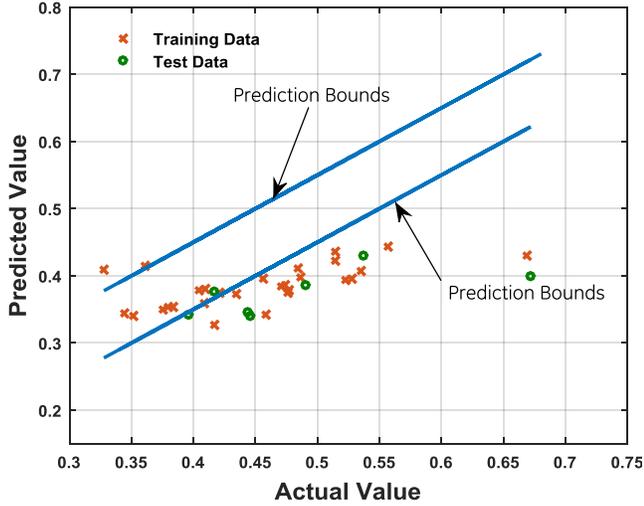


Figure 4. Standard UKF update: actual Vs prediction

The updated model predictions are shown in Figure 4. It can be observed that the update did improve the accuracy of model predictions. But, more than 65% of the training data (19/28) predictions and over 85% of the test data predictions (6/7) are still beyond the prediction bounds. The training error was found to be 0.0767 and the test error was found to be 0.1113. Significant difference between training and test error suggests that the updated model obtained through this process may not be optimal.

The plot of the average error, at each update step, is depicted in Figure 5. The average error, at an update step, is calculated by predicting on all data with the updated model obtained at that step. Though the final error of 0.0767 is lower than the initial model error of 0.1649, it can be observed that the error does not decrease monotonically as more data is utilized for the update. This is due to the fact that, at each update step, the model is being modified in a way to reduce the error between prediction and actual value at that particular data point. Thus, an updated model need not necessarily be more accurate than the previous model across all data.

Further, the significant increases in error, at update step 11 and between update steps 24 and 27, suggest that the standard UKF based model update is highly sensitive to noisy data. A few outliers in the data can have a significant impact on the filter estimates. In fact, it is not inconceivable that the final updated model may actually be worse than the initial model,

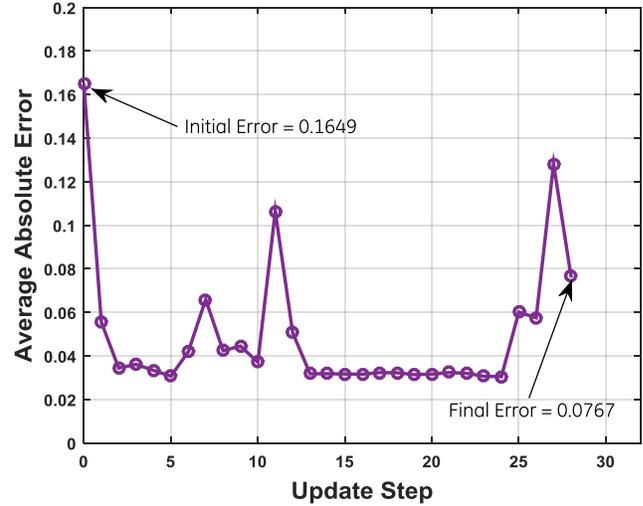


Figure 5. Error at each update step - standard UKF

in terms of accuracy, due to a few very extremely measurements in the data used for the model update.

Table 4. Updated coefficients - standard UKF

Coefficient	True Value	Updt. Value	Abs. Error
$\Theta^1$	0.35	0.3382	0.0118
$\Theta^2$	0.25	0.1699	0.0801
$\Theta^3$	0.15	-0.0332	0.1832
$\Theta^4$	0.40	0.3211	0.0789

Finally, the coefficients of the updated model along with the “true” values are presented in Table 4. It can be observed that while the update did improve the prediction accuracy, the updated model coefficients are quite different from “true” values. The average absolute error between the “true” values and the updated model coefficients was found to be 0.0885.

#### 4.1.3. Updating with Standard UKF - Multiple Runs

It is possible to improve the results obtained with a single run of standard UKF by shuffling the training data and generating multiple models and then selecting the model that best performs on the test data. To illustrate this,  $M = 5$  shuffles were performed on the training data and an updated model was generated with all the  $M$  permuted training data sets. Figure 6 shows the average errors obtained at each update step for all the runs.

The test errors for each of the runs is given in Table 5. The lowest test error of 0.0326, obtained from run 2, is smaller than the value of 0.1113 obtained earlier. The training error for this run was 0.0329.

The coefficients of the best model, obtained from run 2, is presented in Table 6. It can be observed that the estimates are

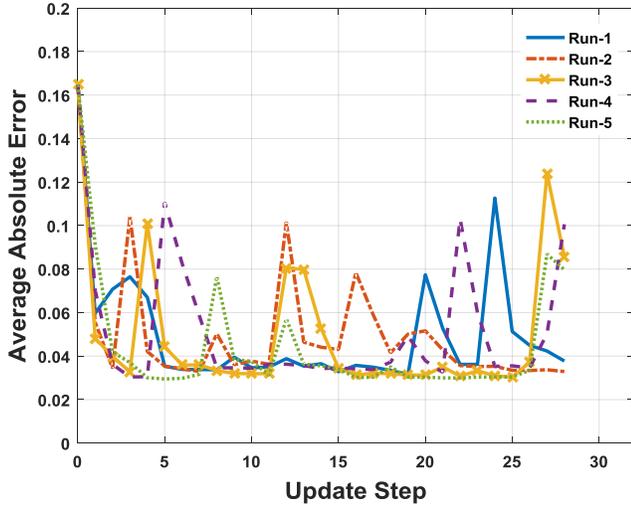


Figure 6. Error at each step - std UKF, multiple runs

Table 5. Test error - std UKF, multiple runs

Run	Test Error
1	0.0423
2	0.0326
3	0.1109
4	0.1265
5	0.1291

better than those obtained in section 4.1.2. The average error is 0.0282 as compared to 0.0885 obtained with a single run of standard UKF.

Table 6. Updated coefficients - standard UKF

Coefficient	True Value	Updt. Value	Abs. Error
$\Theta^1$	0.35	0.3661	0.0161
$\Theta^2$	0.25	0.2006	0.0494
$\Theta^3$	0.15	0.1719	0.0219
$\Theta^4$	0.40	0.3748	0.0252

#### 4.1.4. Updating with Proposed Methodology

In this section, we present the results of updating the non-linear model with the methodology described in Section 3. The Unscented Kalman Filter parameters, namely  $\alpha$ ,  $\beta$  and  $\kappa$ , the number of iterations, the process covariance  $Q$  and the measurement variance  $R$  are the same as those used in Section 4.1.2. The training and test data used are also the same as that used in the case of the standard UKF.

To illustrate the efficacy of the proposed methodology, only a single run (i.e.,  $M = 1$ ) is performed on the training data. The predictions of the updated model are shown in Figure 7.

It can be observed from Figure 7 that the updated model pre-

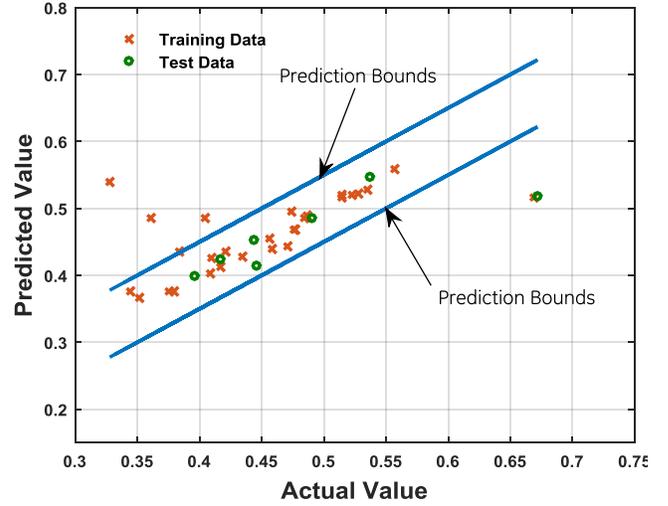


Figure 7. Updated model: actual Vs prediction

dictions are significantly more accurate than that of the initial model. Only 5/28 training data predictions and 1/7 test data prediction are outside the prediction bounds. The training error was found to be 0.0297 and the test error was found to be 0.0309, which are less compared to their respective values reported in the previous sections.

The plot of the average error, at each update step, is depicted in Figure 8. It can be observed that the error non-increasing as more data is utilized for the update. This is due to the fact that in the proposed approach, at each update step, the model modified if and only if the update leads to an overall decrease in error across all data. Thus, the updated model, at any given update step, is guaranteed to be no worse than the initial model when compared with all data.

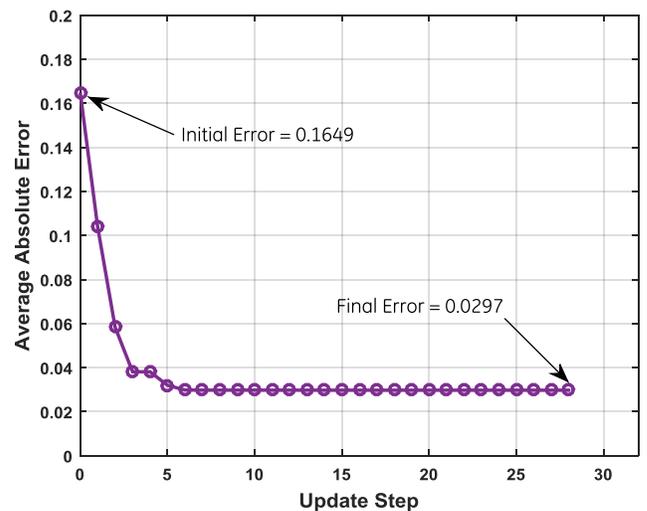


Figure 8. Error at each update step - proposed methodology

Table 7. Updated coefficients - proposed methodology

Coefficient	True Value	Updt. Value	Abs. Error
$\Theta^1$	0.35	0.3525	0.0025
$\Theta^2$	0.25	0.2497	0.0003
$\Theta^3$	0.15	0.1443	0.0057
$\Theta^4$	0.40	0.4141	0.0141

Finally, the coefficients of the updated model along with the "true" values are presented in Table 7. It can be observed that the updated model coefficients are quite similar to the "true" values. The average absolute error between the "true" values and the updated model coefficients was found to be 0.0056, much less than the error of 0.0885 obtained in the case of a standard UKF and 0.0282 obtained with multiple runs of standard UKF. It can be observed that while the performance of the standard UKF with multiple runs and proposed approach (single run) are similar in terms of prediction error on test data, the proposed methodology is superior in estimating the coefficients of the underlying model. While, in this example, superior results were obtained with a single run of the proposed model update methodology, it is suggested to use a  $M$  value greater than 1 in practical applications.

Even though, the data for update came from a single source (i.e., simulation with a set of model parameter values), this example illustrated the robustness of the proposed approach with respect to excessively noisy measurements and outliers. Thus, the proposed update methodology can be more effective than the standard approach for updating models. Metrics for the all the three approaches discussed so far are summarized in Table 8.

Table 8. Summary of metrics

Method	Train. Error	Test Error	Coeff. Error
Std UKF	0.0767	0.1113	0.0885
Multi Runs	0.0329	0.0326	0.0282
Proposed	0.0297	0.0309	0.0056

#### 4.2. Updating a Damage Model

A model that tracks the damage in a part was built to predict the fraction of the overall fielded components expected to be declared as scrap at the time of inspection. A component is deemed "scrap" if there exists a defect of size greater than a certain limit.

Blades, nozzles and shrouds in high pressure turbine sections of a gas turbine are prime examples of parts that require accurate damage models. This is because, these parts experience the harshest conditions in the gas turbine in terms of high temperatures and pressures. Damage of these parts are thus a function of operating conditions of the gas turbine as well as the variations in design parameters such as material proper-

ties and manufacturing variability.

The damage model is a hierarchical model that takes the operational inputs such as airflow pressures and temperatures that are measured and compute the damage as a function of cycles and stresses in specific locations. The stresses are in turn computed as a function of thermal and mechanical loads on the part. The thermal and mechanical loads are in turn functions of the operational inputs. Thus, the hierarchy of a damage model can get very complex depending on the part and the operational characteristics. Since details of all the constituent models and parts are proprietary, we have represented the model mathematically as a collection of coefficients ( $\Theta$ ) and operational parameters. Although the proprietary restrictions prevent us from sharing the underlying models and data sets, we hope that details presented below throw light on the applicability of the techniques developed here for real-world non-linear applications.

The scrap rate  $\mathbf{z}$  is modeled as:

$$\mathbf{z} = \mathbf{h}(T_h, T_c, H, N, \Gamma, d_0, d_1, d_2, d_3, d_4, d_5) \quad (15)$$

where  $T_h$  and  $T_c$  respectively, are the hot and cold temperatures experienced by the component,  $H$  is the hours of operation,  $N$  is the number of cycles and  $\Gamma$  represents the material properties. A Bayesian estimation was performed to compute the initial coefficients of the model. The model predicts the distribution of crack lengths as a function of operating cycles. The model predictions are distributions because the coefficients are computed through Bayesian estimation accounting for variability and uncertainty in the system. This predicted damage distribution was then used with a damage threshold to compute the predicted scrap rate. All results presented in this section are provided in terms of a scaled version of scrap rate. The median values of the posterior distribution are used as the initial value of the model coefficients in the UKF.

The parameter vector  $\Theta$ , represents the coefficients  $d_0, d_1, \dots, d_5$ .

$$\Theta = [d_0, d_1, d_2, d_3, d_4, d_5] \quad (16)$$

An initial model ( $\Theta_0$ ) was developed with data obtained from 14 industrial gas turbines. The model coefficients are given in Table 9.

Table 9. Initial model:  $\Theta_0$ 

Coefficient	Value
$d_0$	7.6370E-01
$d_1$	9.4537E-01
$d_2$	1.0709E+00
$d_3$	4.6795E+00
$d_4$	3.2849E-04
$d_5$	5.5086E-01

The model predictions versus the actual scrap, for the data set used for model building, are depicted in Figure 9. The prediction bounds are derived from acceptable error limits for scrap prediction. These limits are proprietary. The normalized scrap rate and the model predictions are always greater than zero. The parallel prediction bounds are shown for the purpose of easier interpretation. It can be observed that over 85% of the predictions are within the prediction bounds. Only 2 out of 14 points have an error beyond the prediction limits. The average absolute prediction error, for the scaled scrap rate, was found to be 0.0080.

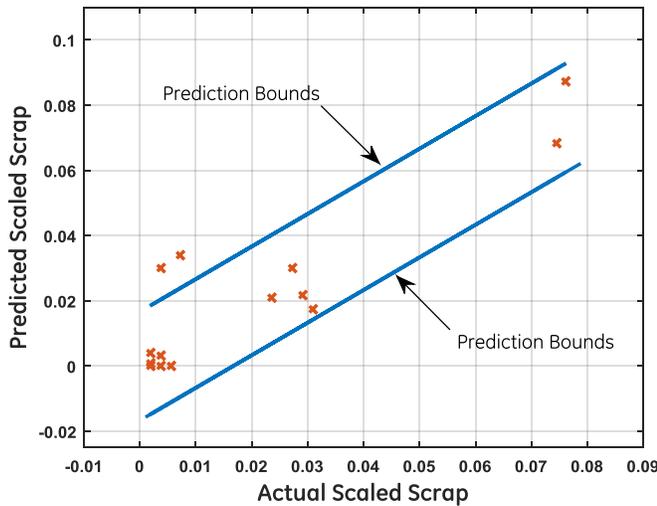


Figure 9. Initial model: actual Vs prediction, training data

Predictions were then obtained for a new data set comprising 13 industrial gas turbines with this model. Figure 10 shows the plot of actual scaled scrap Vs model predictions. It can be observed that almost 50% (6/13) of the predictions have error beyond the prediction limits. The average absolute error was found to be 0.0229.

The initial model is then updated with the new data to improve its prediction accuracy. The model parameters are modeled as a random walk process as in Eq.(12). To illustrate the efficacy of the procedure described in Section 3, the model is first updated with a standard Unscented Kalman Filter. The updated model predictions are provided in Figure 11.

It can be seen from Figure 11 that the performance, in terms of average error, of the updated model is worse than the original model shown in Figure 10. This is due to the fact that in a standard UKF based update of “fleet” models, there is no guarantee that an update with a measurement leads to an overall improvement in performance. The overall average absolute error, calculated at each update step, is plotted in Figure 12. It can be observed that an update can lead to an increase in error, across all points, even if it reduces the error for that particular measurement. Also, there’s no guarantee that the

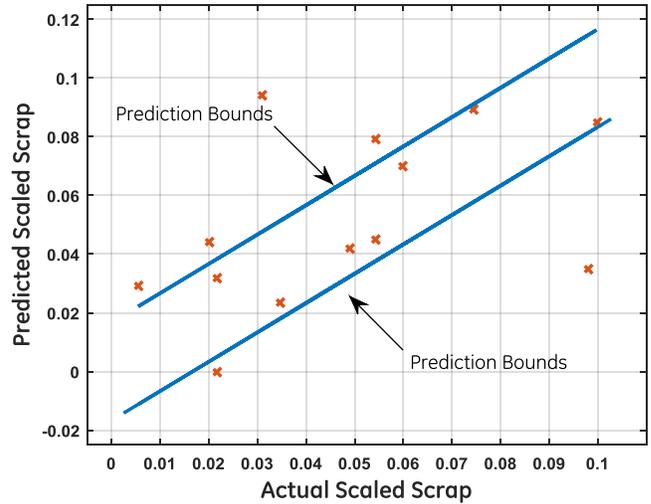


Figure 10. Initial model: actual Vs prediction, new Data

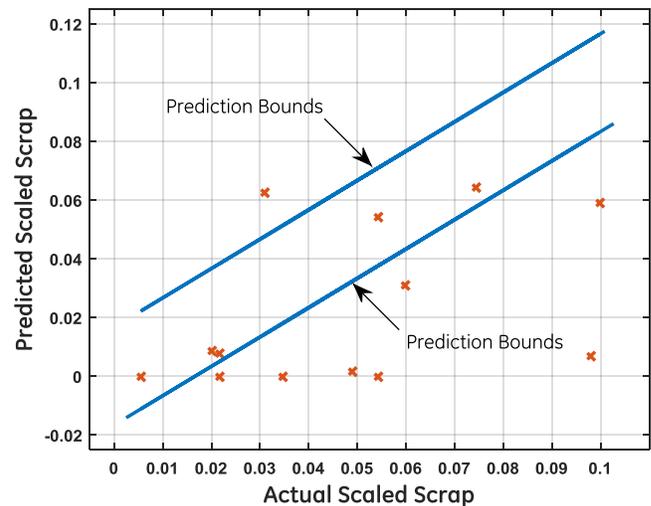


Figure 11. Std UKF update: actual Vs prediction, new data

model obtained at the end of the update process is better in performance than the original model.

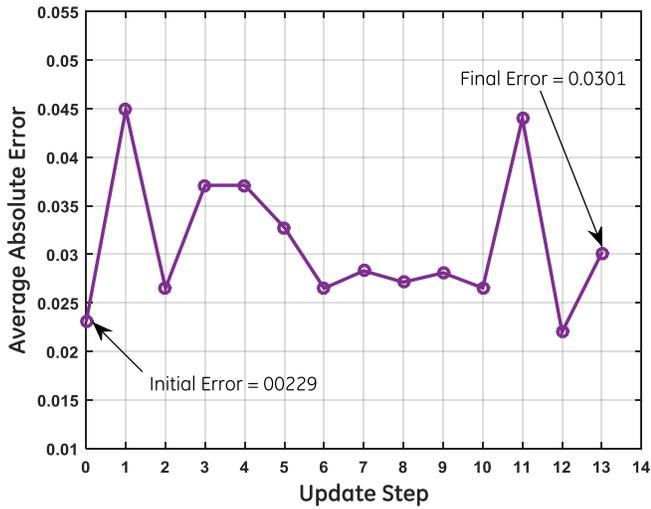


Figure 12. Error at each update step - standard UKF

The initial model is then updated as per the procedure described in Section 3. Of the 13 data points, 11 points were chosen randomly to form the training data set and 2 data points were used for testing. The results are depicted in Figure 13.

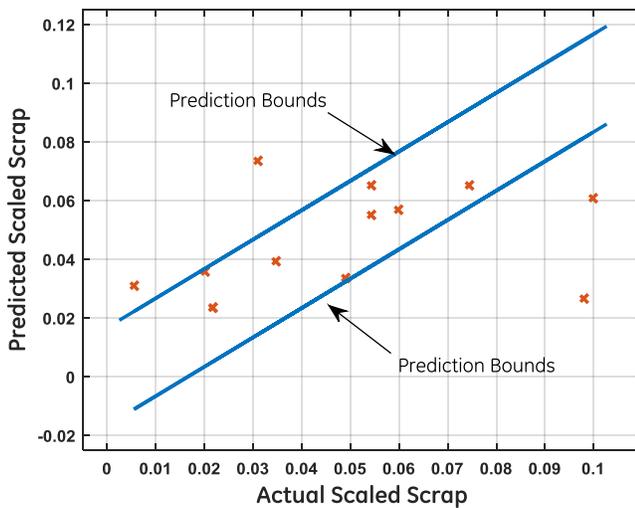


Figure 13. Updated model- actual Vs prediction, new data

It can be observed that the updated model is more accurate than the original model in predicting scrap rate. The number of predictions with error beyond the prediction limits has been reduced from 6 to 4 and the average absolute error has been reduced from 0.0229 to 0.0185. The updated model coefficients are provided in Table 10.

Table 10. Updated model coefficients.

Coefficient	Value
$d_0$	8.0416E-01
$d_1$	9.6754E-01
$d_2$	1.2450E+00
$d_3$	4.8198E+00
$d_4$	4.7651E-04
$d_5$	1.8026E+00

## 5. CONCLUSION

This paper presented a Kalman Filter based methodology to update prognostic models. Unlike prior approaches, where unit-specific models (applicable to a single reservoir or an aircraft engine) were updated with sensor measurements, the problem of updating “fleet” models (applicable to all entities of a fleet) was considered in this paper. The various challenges involved in updating such “fleet” models were discussed and a methodology for mitigating the challenges has been proposed. In this approach, the standard Kalman Filter based model updating framework is adapted in a manner where a portion of the data is randomly chosen for training the filter while the remainder is used for testing purposes. Further, the methodology guarantees non-decreasing accuracy by choosing to update the model coefficients with new data only if the updated model outperforms the current model in terms of reducing the error across all measurements. Also, in an effort to ensure robustness, the training data is permuted several times and an updated model is obtained with each permuted set. Finally, the best among these updated models is chosen through validation with the test data. The efficacy of the proposed update methodology was demonstrated through its application to a simulated non-linear model and to a real-world problem of scrap rate prediction in an industrial gas turbine. While the train-test approach was employed for selecting the best updated model in this paper, a k-fold cross-validation approach could potentially yield better results.

## ACKNOWLEDGMENT

The authors would like to thank KMK Genghis Khan, Senior Principal Engineer, GE Research and Ajay Kumar Behera, Principal Engineer, GE Power for the guidance and inputs that they provided during the development of the model update methodology presented in this paper.

## REFERENCES

- Brookner, E. (2001). The multifaceted peter swerling. *IEEE AESS Systems Magazine*.
- Brown, R., & Hwang, P. (1992). *Introduction to random signals and applied kalman filtering*. John Wiley and Sons, Inc.
- Daum, F. (2005). Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8), 57-69.
- Eykhoff, P. (1974). *System identification*. Wiley.
- Houtekamer, P. L., & Mitchell, H. (2001). A sequential ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1), 123-137.
- Isidori, A. (1995). *Nonlinear control systems*. Springer.
- Jazwinski, A. (1970). *Stochastic processes and filtering theory*. Academic Press.
- Julier, S., & Uhlmann, J. (1997). A new extension of the kalman filter to nonlinear systems. In *The 11<sup>th</sup> intl. symp. on aerospace/defense sensing, simulation and controls, multi sensor fusion, tracking and resource management*.
- Julier, S., Uhlmann, J., & Durrant-Whyte, H. (1995). A new approach for filtering nonlinear systems. In *Proceedings of american control conference*.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series-D), 35-45.
- Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Transactions of the ASME - Journal of Basic Engineering*, 83(Series-D), 95-108.
- Nijmeijer, H., & Fossen, T. (1999). *New directions in nonlinear observer design*. Lecture Notes in Control and Information Sciences, Springer Verlag.
- Ning, L., & Oliver, D. (2005). Ensemble kalman filter for automatic history matching of geologic facies. *Journal of Petroleum Science and Engineering*, 47(34), 147 - 161.
- Swerling, P. (1959). First order error propagation in a stage-wise smoothing procedure for satellite observations. *Journal of the Astronomical Sciences*, 6(3).
- Volponi, A., Ganguli, R., & Daguang, C. (2003). The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparative study. *ASME Journal of Engineering for Gas Turbines and Power*, 125(4), 917-924.
- Zhan, R., & Wan, J. (2007). Iterated unscented kalman filter for passive target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(3), 1155-1162.

## BIOGRAPHIES

**Venkatesh Rajagopalan** is a researcher in the Prognostics Lab at GE Global Research, Bangalore, India. He obtained his Ph.D in Electrical Engineering from the Pennsylvania State University, University Park, PA in the year 2007. His doctoral research focused on development of an anomaly detection methodology based on Symbolic Time Series Analysis. He has, to his credit, 16 international conference and journal publications and 11 granted patents. His research interests include Signal Processing, Estimation Theory, Statistical Learning and Prognostics. His research work at GE involves development and application of prognostic algorithms for health management of various industrial assets.

**Arun Subramaniyan** leads the Structures Lab at GE Global Research in Niskayuna, NY. He joined GE GRC in January, 2009 after completing doctoral studies in Aerospace Engineering from Purdue University. He is a prolific researcher with over 40 international publications that have been cited more than 300 times in the areas of Probabilistics, Data Mining, Uncertainty Quantification, Structural Mechanics, Composite Materials and Nanomechanics. He is actively involved in the development and application of probabilistic mechanics tools for a number of GE businesses.