

# An Inference-based Prognostic Framework for Health Management of Automotive Systems

Chaitanya Sankavaram<sup>1</sup>, Anuradha Kodali<sup>2</sup>, Krishna Pattipati<sup>3</sup>, Satnam Singh<sup>4</sup>, Yilu Zhang<sup>5</sup> and Mutasim Salman<sup>6</sup>

<sup>1,2,3</sup> *Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA*

*chaitanya.sankavaram@uconn.edu*  
*krishna@enr.uconn.edu*

<sup>2</sup> *University of California Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035*

*anuradha.kodali@nasa.gov*

<sup>4</sup> *CA Technologies, Bangalore, Karnataka 560017, India*

<sup>4</sup> *Diagnosis and Prognosis Group, GM India Science Lab, Bangalore, Karnataka 560066, India*

<sup>1,5,6</sup> *Vehicle Systems Research Lab, General Motors Global R&D, Warren, MI 48090, USA*

*chaitanya.sankavaram@gm.com*  
*yilu.zhang@gm.com*  
*mutasim.a.salman@gm.com*

## ABSTRACT

This paper presents a unified data-driven prognostic framework that combines failure time data, static parameter data and dynamic time-series data. The framework employs proportional hazards model and a soft dynamic multiple fault diagnosis algorithm for inferring the degraded state trajectories of components and to estimate their remaining useful life times. The framework takes into account the cross-subsystem fault propagation, a case prevalent in any networked and embedded system. The key idea is to use Cox proportional hazards model to estimate the survival functions of error codes and symptoms (probabilistic test outcomes/prognostic indicators) from failure time data and static parameter data, and use them to infer the survival functions of components via soft dynamic multiple fault diagnosis algorithm. The average remaining useful life and its higher-order central moments (e.g., variance, skewness, kurtosis) can be estimated from these component survival functions. The framework is demonstrated on datasets derived from two automotive systems, namely hybrid electric

vehicle regenerative braking system, and an electronic throttle control subsystem simulator. Although the proposed framework is validated on automotive systems, it has the potential to be applicable to a wide variety of systems, ranging from aerospace systems to buildings to power grids.

## 1. INTRODUCTION

Conventional maintenance strategies, such as reactive and preventive maintenance, are inadequate in fulfilling the needs of high-availability in complex automotive systems. What is needed is a continuous monitoring and early warning capability that detects, isolates and estimates component degradations (fault diagnosis and prognosis) over time and thus minimizing downtime and operational costs via condition-based maintenance. Failure prognosis, an add-on capability to diagnosis, involves forecasting system degradation and time-to-failure (remaining useful life) based on “state awareness” gained from monitored data, for example, parameters collected from various sensors, such as vehicle speed, individual wheel speeds, yaw rate, master brake cylinder pressure, and so on.

The existing time-series based approaches to prognostic health management are component-centric and do not make use of widely available data in archived databases of vehicle equipment, such as historical usage patterns, error codes (i.e., codes that are recorded by onboard software units in case of a fault or malfunction of the component), observed failure modes, repair and inspection intervals, environmental factors, skill levels of personnel, and status parameters

Chaitanya Sankavaram 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.

A preliminary version of the paper was published as Sankavaram, C., Kodali, A., Pattipati, K., Wang, B., Azam, M., and Singh, S. A Prognostic Framework for Health Management of Coupled Systems. *IEEE International Conference on Prognostics and Health Management*, Denver, CO, June 2011.

collected periodically or at the onset of error codes. Examples of status parameters include operating parameter identifier data collected periodically or when an error code is recorded. Consequently, the time-series based prognostic health management approaches are both incomplete and inaccurate for coupled systems with cross-subsystem fault propagation. In coupled systems, the cross-subsystem fault propagation comes from the fact that when multiple components are interacting with each other, a fault in one component would propagate to other components, i.e., degradation in one component affects the other and new faults emerge as a result of interactions among multiple components. For example, interactions with a degraded battery would eventually damage the starter. Therefore, individual component-based prognostic algorithms are inadequate and a system-level prognostic algorithm taking into account the fault propagation effects due to coupling/interactions among the components is needed.

On the other hand, the classical survival theory-based approaches and reliability-based methods (Jardine & Tsang, 2005; Murthy, Xie, & Jiang, 2004) use only failure time data to estimate time to failure distribution. These methods rely on Weibull and other nonlinear regression models to infer time-to-failure, and these estimates are used to optimize the time-to-maintain or time-to-repair/replace. These techniques do not consider actual condition of a specific component and therefore result in large variability in the time-to-failure estimates. Evidently, the two disparate methodologies, viz., prognostic health management techniques based on dynamic time series data and survival theory-based techniques using archived data, need to be reconciled and unified under a common modeling framework that can work with *all* of the three types of data (see Fig. 1) listed below:

- (a) archived failure data (Type I data): age (or a surrogate function such as the mileage or operational time) of the vehicle at the time of failure, i.e., age when an error code or symptom is observed, or a component is replaced;
- (b) static environmental and status parameter data (Type II data); and
- (c) dynamic data (Type III data): time-series data and periodic status data.

In this paper, an integrated approach that seamlessly combines all three types of data to infer the component degradations and to estimate their remaining useful life (RUL) times is presented. The framework employs two key techniques: (i) Cox proportional hazards model (Cox PHM) (Klabfleisch & Prentice, 2002), and (ii) soft dynamic multiple fault diagnosis (soft DMFD) inference algorithm (Singh, Kodali, & Pattipati, 2009). The Cox PHM computes the survival functions of tests (or error codes), whereas the soft DMFD algorithm is used to infer failing components in coupled systems. The soft DMFD algorithm determines the most likely evolution of component states that best explains the observed soft test failure outcomes (i.e., complementary

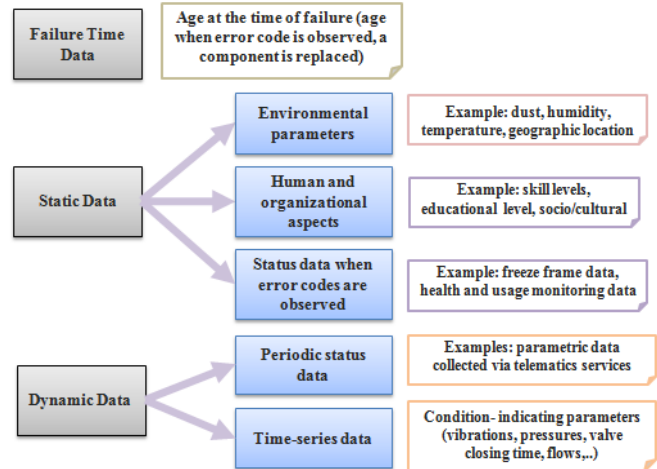


Figure 1. Categories of Data Available for Prognostics

test survival probabilities). Here, a soft Viterbi algorithm is employed to decode the most likely probabilistic evolution of the fault sequence. The prognostic framework is discussed in detail in the subsequent sections and the capabilities of the framework is illustrated on two datasets: (i) a dataset derived from an automotive electronic throttle control (ETC) system simulator with failure time data, static parameter data, and simulated test outcomes; and (ii) a dataset derived from an automotive regenerative braking system (RBS) with failure time data, and static as well as dynamic parameter data obtained from simulation-based fault injection experiments conducted in MATLAB®/Simulink®. The proposed framework is modular, leading to a flexible and evolvable software architecture for prognostic health management.

The paper is organized as follows. An overview of the existing prognostic methods is discussed in Section 2 and Section 3 presents the proposed prognostic framework to estimate the component degradations. A brief discussion on the soft DMFD inference algorithm is presented in Section 4 and the experimental results from two automotive applications are discussed in Section 5. Finally, section 6 concludes the paper with a summary.

## 2. PREVIOUS WORK

Prognosis is a salient component of condition-based maintenance (CBM) of systems. The Prognostic methods can be broadly classified into the following two approaches: model-based and data-driven (Chiang, Russel, & Braatz, 2001). These methods are discussed in detail in the following subsections.

### 2.1. Model-based methods

The *model-based methods* are applicable to systems where accurate mathematical models of the system and an adequate number of sensors to observe the state of the system are available. The model-based methods are also termed the

physics-based or physics-of-failure-based models in the literature. The model-based methods employ statistical estimation techniques to track residuals generated using observers (e.g., Kalman filters, reduced-order unknown input observers, interacting multiple models, particle filters) and parity relations (dynamic consistency checks among measured variables) in order to provide an estimate of the accumulated damage and assess the remaining life (Luo, Namburu, Pattipati, Qiao, Kawamoto, & Chigusa, 2003).

Several methods have been proposed based on damage accumulation models. Ray and Tangirala (1996) presented a nonlinear stochastic model of fatigue crack dynamics for damage rate computation in mechanical structures. Adams (2002) proposed to model damage accumulation in a structural dynamic system as first/second order nonlinear differential equations. Serrao, Onori, Rizzoni, and Guezennec (2009) proposed a damage accumulation model for automotive batteries by taking into account the deterioration of physical or functional parameters, for instance, battery state-of-charge, operating temperature, as well as load profile and its impact on the battery aging process. Chelidze, Cusumano, and Chatterjee (2002) modeled degradation as a “slow-time” process, which is coupled with a “fast-time” observable subsystem. Luo *et al.* (2003) developed a prognostic process based on data collected from model-based simulations under normal and degraded conditions; interacting multiple models were used to track the hidden damage. Daigle, Matthew, and Goebel (2011), Celaya, Saxena, and Goebel (2012), and Byington, Watson, Edwards, and Stoelting (2004) are some examples for degradation modeling and physics-based approaches.

The main advantage of a model-based approach is its ability to incorporate a physical understanding of the degradation process into the process monitoring scheme. However, it is difficult to apply the model-based approach to large-scale systems because it requires detailed analytical models in order to be effective.

## 2.2. Data-driven methods

The *data-driven approaches* to prognostics are derived directly from routinely monitored system operating data (e.g., vibration and acoustic signals, temperature, pressure, currents, voltages). The data-driven approaches are based on statistical and pattern classification techniques, ranging from multivariate statistical methods, linear and quadratic discriminants, partial least squares and canonical variate analysis, support vector machine and relevance vector machine regression, Gaussian processes, graphical models (Bayesian networks, hidden Markov models) to black-box methods based on neural networks (e.g., multi-layer perceptrons, probabilistic neural networks, learning vector quantization), self-organizing feature maps, signal analysis (filters, auto-regressive models etc), and fuzzy rule-based systems.

Depending on the type of information used, the prognostic techniques may also be categorized into three types: (i) time-to-failure data-based, (ii) stressor-based, and (iii) degradation-based. *Time-to-failure data-based* methods use failure time data to estimate the lifetime of a component (e.g., Weibull analysis). *Stressor-based* methods consider the operating conditions, such as temperature, humidity, vibrations, load, input current and voltage. *Degradation-based* methods estimate and track the degradation parameters and predict when the total degradation (damage) exceeds a predefined threshold of functional failure. These degradation parameters are directly measured from the system or via a fusion of multiple parameters (Coble, 2010).

A survey of data-driven prognostics is provided by Schwabacher (2005). Si, Wang, Hu, and Zhou (2011) presented a detailed review of the statistical data-driven approaches. Gorjian, Ma, Mittinty, Yarlagadda, and Sun (2009a; 2009b) discussed the existing state-of-the-art literature based on covariate-based models, and the commonly used degradation models in reliability analysis. Wang and Vachtsevanos (1999) employed a dynamic wavelet neural network trained on vibration signals of defective bearings with varying crack depth and width to predict the crack evolution and to estimate their remaining useful life times. Gebraeel and Lawley (2008) discussed a feedforward neural network based method for predicting the remaining useful life of ball bearings. Swanson (2001) proposed to use a Kalman filter to track the dynamics of mode frequency of vibration signals in a tensioned steel band. Garga, McClintic, Campbell, Yang, and Lebold (2000) presented a signal analysis approach for prognostics of an industrial gearbox. The main features used include the root mean square value, kurtosis and wavelet magnitude of vibration data. Cox and Oakes (1984) developed proportional hazard models that merge both failure time data and stress data (vibration signals) to estimate the remaining useful life. Kumar, Torres, Chan, and Pecht (2008) described a hybrid prognostic framework utilizing both data-driven and physics-of-failure models to estimate the remaining useful life in electronic systems. The monitored parameters included fan speed, temperature and percentage of CPU utilization.

Most of the prognostic approaches in the literature consider one or two categories of data delineated in Fig. 1. For instance, the reliability-based methods use historical failure times (Type I data) to generate life time models. These methods can only provide an average estimate for component degradations based on historical failure time data. It is evident that a prognostic algorithm built solely on the historical data cannot provide an accurate estimate for component degradations because typically, components operating in a harsh environment will fail more quickly than those operating in a mild environment. Hence, a prognostic algorithm should consider operating parameter data and

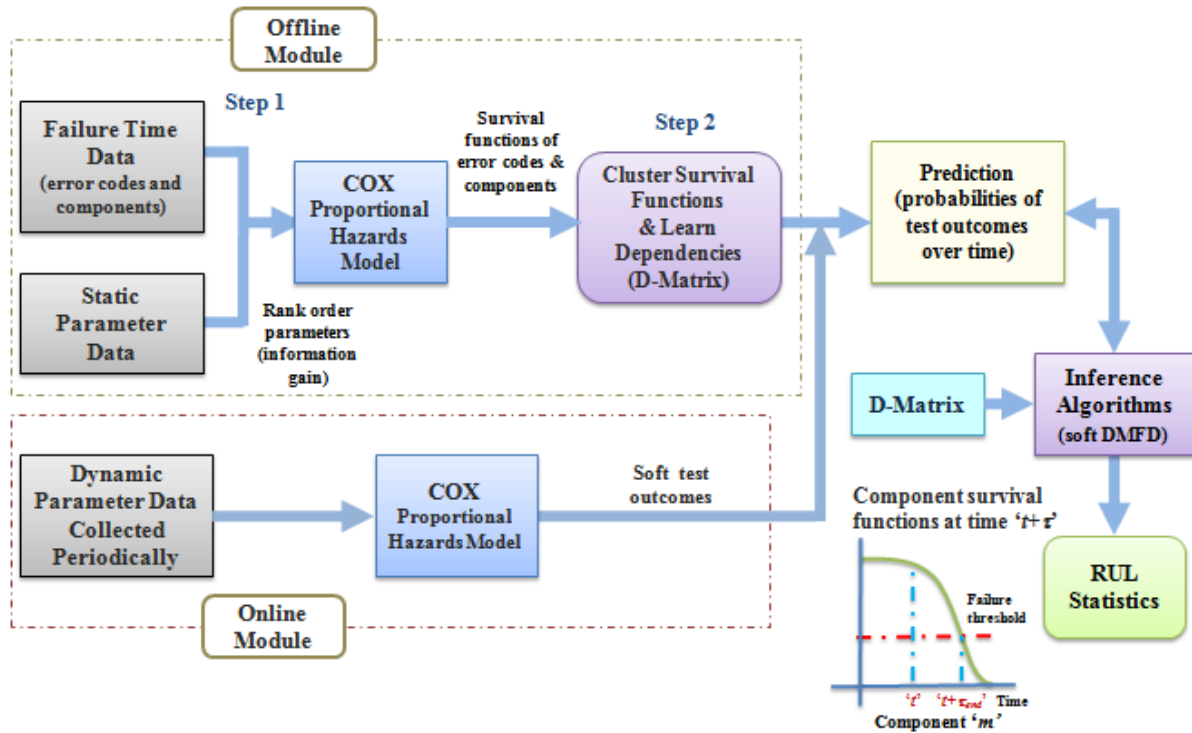


Figure 2. Cox PHM-based Approach to Prognosis of Coupled Systems

condition monitoring data (periodic or time-series data) along with the historical data to provide an accurate estimate for component degradations so that an appropriate predictive maintenance action can be taken on an as-needed basis. Coble (2010) compared the performance of prognostic models based on reliability data and condition monitoring data and demonstrated that reliability data alone does not result in satisfactory results for prognostics. Pecht, Das, and Ramakrishnan (2002), Lall, Pecht, and Harkim (1997), and Vichare, Rodgers, Evely and Pecht (2004) are few other examples where it is emphasized that methods based on Type I data are insufficient for prognostics – these methods tend to either underestimate or overestimate the remaining useful life. Therefore, it is vital to take into consideration the current operating status of the components (i.e., Type III data) as well as the historical data (Type I and II data) to estimate component degradations with good degree of accuracy.

In the literature, there are many methods developed exclusively for prognostics of a single component. The methods usually employ feature data collected over a period of time until the component fails to develop a prognostic model. More significantly, they are component-centric and they primarily focus on predicting the remaining useful life of one particular component in isolation. The approach presented in this paper overcomes these two fundamental limitations by developing a prognostic framework that is capable of tracking degradations of multiple interacting components in a system. Also, the framework has the ability to incorporate data generated via a model-based approach, or a sensor monitoring approach or both.

The novel contributions of this paper are: (i) a unified framework combining failure time data, static parameter data and dynamic parameter data, (ii) Cox proportional hazards model to estimate the survival functions of error codes (tests) using failure time data and static parameter data, (iii) estimating multiple component degradations that are coupled via observations (test outcomes) using a novel inference algorithm, and (iv) simulation results on degradation estimation of multiple components in electronic throttle control (ETC) subsystem simulator, and regenerative braking system in hybrid electric vehicles. Since coupled systems are common in aircraft, automobiles, power systems, nuclear energy systems, and indeed any networked embedded cyber-physical system, the proposed prognostic framework is readily applicable to these systems.

### 3. PROGNOSTIC FRAMEWORK

As shown in Fig. 2, the proposed prognostic framework consists of two phases: an *offline* training and validation (“model learning”) phase, and an *online* testing (“deployment”) phase.

#### 3.1. Training and Validation Phases (offline module)

The training phase consists of two steps (see Fig. 2). In *Step 1*, Type I and Type II data are used to compute static data-modulated survival functions for components, error codes, symptoms and any observable test outcomes via Cox proportional hazards model. Here, mutual information algorithm (Duda, Hart, & Stork (2000); Bishop (1997)) is

employed to select minimum number of Type II static parameter data for the Cox proportional hazards model.

The Cox PHM assumes a hazard function of the form (Jardine & Tsang, 2005; Murthy et al., 2004):

$$\phi_i(t, \underline{z}) = \phi_0(t) e^{\underline{a}^T \underline{z}} \quad (1)$$

where  $i$  denotes a component, diagnostic error code, symptom or any failure event of interest,  $\underline{z}$  is a vector of covariates (Type II static data such as freeze frame data),  $\underline{a}$  is a vector of regression parameters, and  $\phi_0(t)$  is the failure rate without any covariates (i.e.,  $\underline{z} = 0$ ), i.e., it is the baseline hazard function. The baseline hazard function can be from any of the standard failure time distributions (e.g., exponential, Weibull, normal, log normal, Gamma, etc.) or it can be nonparametric. The baseline hazard function and the regression parameters are estimated via the maximum likelihood method (Klein, and Moeschberger, 2003).

The survival functions of components,  $R_i(t, \underline{z})$  and the associated failure time density functions,  $f_i(t, \underline{z})$  can be computed from the hazard function  $\phi_i(t, \underline{z})$  as follows,

$$\begin{aligned} R_i(t, \underline{z}) &= \exp\left(-\int_0^t \phi_i(\tau, \underline{z}) d\tau\right) \\ f_i(t, \underline{z}) &= \phi_i(t, \underline{z}) R_i(t, \underline{z}) \\ &= \phi_i(t, \underline{z}) \exp\left(-\int_0^t \phi_i(\tau, \underline{z}) d\tau\right) \\ &= -\frac{dR_i(t, \underline{z})}{dt} \end{aligned} \quad (2)$$

In *Step 2*, the survival functions corresponding to an event of interest  $i$  are grouped via clustering techniques, such as k-means, learning vector quantization (LVQ), Gaussian mixture models (GMM), or hierarchical clustering (Duda et al, 2000; Bishop, 1997). These clusters represent different usage profiles of the components, depending on the usage conditions, environmental factors, etc.

Also, the probabilistic dependencies between the error codes (and other observables i.e., symptoms), and the component failure modes are evaluated using maximum likelihood estimation procedure. This dependency matrix captures the relationships between failure modes and error codes (or symptoms) and thus can be either binary (hard (0 or 1)) or probabilistic (soft). The probabilistic dependencies are in the form of a matrix of likelihoods of observing an error code or other observables given a failure mode and is given by,

$$\hat{p}_{ij} = \frac{n_{ij}}{n_i} \Rightarrow \hat{P}(o_j | FM_i) = \frac{n(o_j, FM_i)}{n(FM_i)} \quad (3)$$

where  $o_j$  is the error code  $j$ ,  $n_{ij}$  is the number of times  $o_j$  is associated with failure mode  $i$  ( $FM_i$ ) and  $n_i$  is the total number

of observed cases with  $FM_i$ . In order to avoid the problem with ML estimates, viz., the possibility of having a zero probability because of an unseen combination of  $(o_j, FM_i)$  in the training data (the so-called “black swan” problem), Laplacian smoothing (Metzler, Lavrenko, & Croft, 2004) is used as follows:

$$\hat{P}(o_j | FM_i) = \frac{n(o_j, FM_i) + 1}{n(FM_i) + |FM|} \quad (4)$$

where  $|FM|$  is the number of system failure modes.

Another way to generate the fault-test dependency matrix is via a model-based approach. The model-based methods employ residuals as features, where the residuals are the deviations of actual sensor measurements from the expected ones. The residuals can be generated, for instance, based on parameter estimation, observers, parity relations, or via simulation-based fault injection experiments (Ciccarella, Dalla Mora, & Germani, 1993; Bar-Shalom, Li, & Kirubarajan, 2001; Gertler, 1997). The residuals thus generated are used to devise tests and to extract fault-test relationships. Statistical hypothesis testing techniques (e.g., change detection techniques (Basseville & Beneveniste 1986), such as the generalized likelihood ratio test, cumulative sum test, sequential probability ratio test, etc.) can be used to define thresholds to detect the presence of faults. The tests that can detect a fault are represented as “1” in the corresponding row of the  $D$ -matrix; thus, each row corresponds to a “signature” for the fault associated with the row. Refer to (Singh, Holland, & Bandyopadhyay, 2010), and (Kodali, Ponizovskaya-Devine, Robinson, Luchinsky, Bajwa, Khasin, Perotti, & Brown, 2015) for various methods of generating dependency matrices.

### 3.2. Testing (Deployment) Phase (online module)

In the testing phase, when new feature data  $\underline{y}(t)$  (Type III dynamic data) is obtained via online data acquisition systems, the survival probabilities of error codes (see Eq. (5)) are estimated using the Cox PHM model as well as the baseline hazard functions obtained from the offline module (from Type I and Type II data).

$$\begin{aligned} \phi_j(t, \underline{y}(t)) &= \phi_0(t) e^{\underline{a}^T \underline{y}(t)} \\ \Rightarrow R_j(t, \underline{y}(t)) &= \exp\left(-\int_0^t \phi_j(\tau, \underline{y}(\tau)) d\tau\right) \end{aligned} \quad (5)$$

Here  $j$  refers to error codes (tests, symptoms). These soft error code outcomes are used to infer the failing components via the soft DMFD based on the  $D$ -matrix. The soft DMFD determines the evolution of fault states (complementary survival functions) given the soft error code outcomes at the observed time  $t$ . A brief explanation of the soft DMFD algorithm is presented in Section 4.

Once the component survival functions are obtained, the first and second order moments of time-to-failure  $T$  of each component can be computed from the survival functions  $R_i(t, \underline{z})$  via Eqs. (6) and (7), respectively (Ma & Krings, 2008).

$$E_i(T) = \int_0^{\infty} R_i(\tau, \underline{z}) d\tau \quad (6)$$

$$\text{Var}_i(T) = 2 \left( \int_0^{\infty} \tau R_i(\tau, \underline{z}) d\tau \right) - \left[ \int_0^{\infty} R_i(\tau, \underline{z}) d\tau \right]^2 \quad (7)$$

Alternately, the remaining useful life (RUL) of a component at any time  $t$  can be computed from the survival function by defining a threshold on the survival probability. Mathematically, it is written as,

$$\text{RUL}_i(t) = \arg \min_{\tau} (R_i(t + \tau, \underline{z}) \leq \varepsilon_0) \quad (8)$$

where  $\varepsilon_0$  denotes the threshold for a functional failure.

#### 4. SOFT DYNAMIC MULTIPLE FAULT DIAGNOSIS

The soft dynamic multiple fault diagnosis (DMFD) is a factorial HMM-based inference algorithm to determine the evolution of component fault states based on the observed soft evidence on the test states. Formally, the soft DMFD problem is represented as  $\text{SDMFD} = \{S, \kappa, \Gamma, O, D, P, A\}$  where  $S = \{s_1, \dots, s_m\}$  is a finite set of components (failure sources) associated with the system. The state of component  $s_i$  is denoted by  $x_i$ , where  $x_i(k) = 1$  if  $s_i$  is faulty; and  $x_i(k) = 0$  otherwise. At epoch  $k$ , the status of all components at epoch  $k$  is denoted by  $\underline{x}(k) = [x_1(k), x_2(k), \dots, x_m(k)]$ . Similarly,  $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_n\}$  represent a finite set of available tests where  $\gamma_j(k) = 1$  implies that the test  $\gamma_j$  is in a failed state at time epoch  $k$ ; and  $\gamma_j(k) = 0$  otherwise. Here,  $\kappa = \{0, 1, \dots, k, \dots, K\}$  is the set of discretized observation epochs. The observations  $O = \{o_1, o_2, \dots, o_n\}$  constitute the soft evidence on the test states. For each component state, e.g., for component  $s_i$  at epoch  $k$ ,  $A = (Pa_i(k), Pv_i(k))$  denotes the set of fault appearance probability and fault disappearance probability defined as  $\Pr(x_i(k) = 1 | x_i(k-1) = 0)$  and  $\Pr(x_i(k) = 0 | x_i(k-1) = 1)$ , respectively. Also,  $P = (Pd_{ij}, Pf_{ij})$  represent probability of detection and probability of false alarm associated with test outcome  $j$  and fault class  $i$ .

The inference problem can be formulated as one of finding the maximum *a posteriori* (MAP) configuration:

$$\hat{X}^K = \arg \max_{X^K} \Pr(X^K | O^K, \underline{x}(0)) \quad (9)$$

where  $X^K = [\underline{x}(k)]_{k=1}^K$ ,  $O^K = [o(k)]_{k=1}^K$ ,  $K$ =total number of epochs and  $o(k) = [o_1(k), o_2(k), \dots, o_n(k)]$ .

The solution is a primal-dual optimization framework that employs Lagrangian relaxation method for decomposing the original soft DMFD problem into parallel decoupled

subproblems, one for each fault. Each subproblem corresponds to finding the optimal fault-state sequence, which is solved using a soft Viterbi decoding algorithm. The subproblems are coordinated by updating the Lagrange multipliers using a subgradient method.

The inputs to the algorithm are test outcomes  $o(k)$  at each sampling time  $k$ , their reliabilities, and a fault diagnostic matrix (D-matrix) that captures the relationships between failure sources and tests (see Eq. 4). The test outcomes could be statistical test decisions derived from sensor data. These test outcomes, together with the fault-dependency matrix, are processed through a primal-dual optimization method that exploits temporal correlations of test outcomes over time for inference. A detailed description of soft DMFD algorithms can be found in our previous work (Sankavaram, Kodali, Pattipati, Wang, Azam, & Singh (2011)).

#### 5. EXPERIMENTAL RESULTS

The prognostic framework is applied to datasets derived from two automotive systems, namely, electronic throttle control (ETC) subsystem simulator, and regenerative braking system (RBS). The experimental results are briefly discussed in the following subsections.

##### 5.1. Application to Electronic Throttle Control System

In the first application, the prognostic approach is applied to a dataset derived from an automotive ETC subsystem simulator. The function of an ETC subsystem is to determine the necessary throttle opening using sensors (such as the accelerator pedal position, the engine RPM, and the vehicle speed) and drive the actuator to obtain the required throttle position via a closed-loop control algorithm in the engine control module (ECM). The ECM also monitors the health of the engine subsystem by processing parameter identifier data (PIDs) collected from various sensors and generates diagnostic trouble codes (DTCs or error codes) when a failure occurs in any component. Refer to Appendix for a detailed description of DTCs and repair codes (RCs) in an ETC subsystem.

The dataset derived from the ETC simulator consisted of 11 error codes (DTCs), 479 status parameters (PIDs) collected at the time of DTC firing, accelerated age of the vehicle and the repair/replacement actions (i.e., repair codes (RCs)) performed on the system. A total of five different repair codes (replaceable components) are present in our training data. Here, the age of the vehicle at the time of repair/replacement action (i.e., the failure time of the replaced component) is the Type I data and the set of 479 PIDs collected at the time of DTC occurrence forms the Type II data.

On the available Type II data, information gain (mutual information) algorithm is employed to rank order the PIDs and optimal number of PIDs are then selected for the analysis (Duda et al., 2000; Bishop, 1997). The idea of IG algorithm is to evaluate the amount of information contributed by each



feature to a particular class; and the subset of features with high information content is used for analysis. In the current application, IG algorithm is employed on 479 status parameters and the top 16 PIDs are selected based on the failure probability estimation accuracy on the test data. Fig. 3 shows the value of mutual information for each of the status parameters and the top features are circled in red. The total data available for Type I and Type II data is of the dimension  $\beta \times 1$  and  $\beta \times 16$  respectively where  $\beta$  is the total number of observed failure cases.

The fault-test dependency matrix (D-Matrix) between DTCs and RCs are learned via maximum likelihood estimation of probabilities (see (3) and (4)), which is later used in the inference algorithm for inferring the failing components. Table 1 shows the diagnostic matrix with 5 RCs and 11 DTCs. The diagnostic matrix represents fault signatures for component faults, these components are coupled via observations. For instance, when any of the components RC2, RC4 and RC5 are faulty, these faults will generate a set of observations that will fail the test P0121 (i.e., set to 1 – failed test outcome). It is evident that RC1 and RC3 are ambiguous, and hence are grouped into a single repair code throughout our analysis (RC1/RC3). Also, the fault signatures of components RC1/RC3, RC2, and RC5 are hidden within the fault signature of RC4 and hence are termed as hidden faults.

As mentioned in Section 3, the survival functions for components and tests are initially learned using the Cox PHM model. Then, a  $k$ -means clustering technique is employed to group the survival functions of RCs as well as DTCs. Here, there appear to be 3 clusters of survival functions. Fig 4 shows the original DTC survival functions. The different colors represent different clusters. The averaged survival function clusters for DTC P1682 and RC2 are shown in Figs. 5 and 6, respectively. When the number of clusters were increased from 3 to 5 and 6, there was no significant improvement in terms of failure probability estimation. Hence, for this work, the number of clusters for the DTC and the RC was selected to be 3.

In order to validate the prognostic framework, the detection and false alarm probabilities of tests are initially learned from the averaged DTC survival function and the averaged RC survival function by minimizing the objective function derived from a noisy OR model (Singh et al., 2009b) given in Eq. (10).

$$\min_{\{Pd_{ij}, Pf_{ij}\}_{i=1, j=1}^{m, n}} J = \sum_{k=1}^K \sum_{j=1}^n \{N_j(k) + \sum_{i=1}^m \ln[R_i(k)(1 - Pf_{ij}) + (1 - R_i(k))(1 - Pd_{ij})]\}^2$$

$$\Rightarrow \min_{\{Pd_{ij}, Pf_{ij}\}_{i=1, j=1}^{m, n}} J = \sum_{k=1}^K \sum_{j=1}^n \{N_j(k) + \sum_{i=1}^m \ln[R_i(k)(1 - Pf_{ij}) + (1 - R_i(k))(1 - Pd_{ij})]\}^2$$

$$\begin{aligned} \text{where } N_j(k) &= -\ln y_j(k) = -\ln(1 - v_j(k)) \\ &= -\ln \Pr(\gamma_j(k) = 0 | \underline{x}(k)) \end{aligned} \quad (10)$$

$$v_j(k) = \text{complementary survival function of } DTC_j$$

$$R_i(k) = \text{survival function of } RC_i$$

$$\Pr(\gamma_j(k) = 1 | \underline{x}(k)) = v_j(k) = 1 -$$

$$\prod_{i=1}^m \{R_i(k)(1 - Pf_{ij}) + (1 - R_i(k))(1 - Pd_{ij})\} \quad (11)$$

A nonlinear least squares estimation technique is implemented to solve the problem in Eq. (10) using the MATLAB's *fmincon* function from the optimization toolbox to determine the optimal parameters  $\{pd_{ij}, pf_{ij}\}$ . Once the parameters are learned, continuous test outcomes are generated from each of the RC survival function clusters using Eq. (11). These soft test outcomes (equivalent to the test outcomes obtained from the Type III data in the Online phase) are used as input to the soft dynamic multiple fault

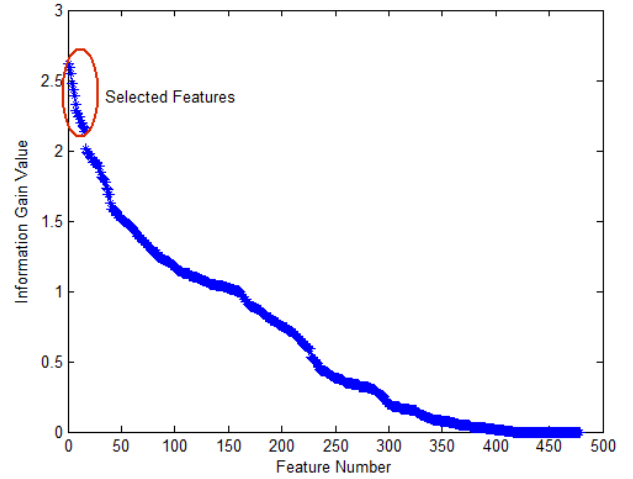


Figure 3. Plot of Mutual Information for 479 Status Parameters

	P01 01	P01 02	P01 21	P06 01	P15 16	P16 82	P21 01	P21 22	P21 27	P21 38	P21 76
RC1	0	0	0	0	0	0	0	1	1	1	0
RC2	0	0	1	0	1	0	1	0	0	0	1
RC3	0	0	0	0	0	0	0	1	1	1	0
RC4	1	1	1	1	1	1	1	1	1	1	1
RC5	0	0	1	0	0	0	0	0	0	0	0

Table 1. Diagnostic (D-) Matrix for ETC System

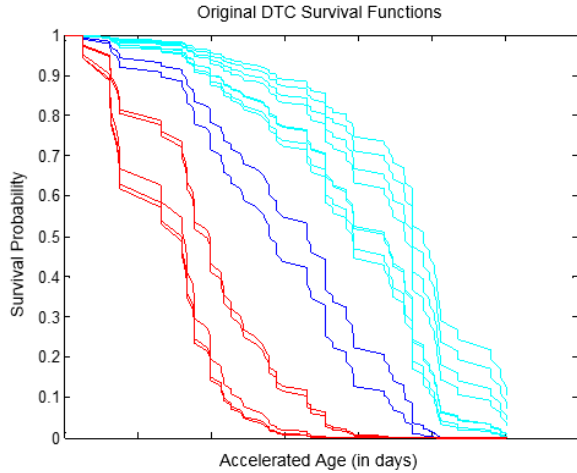


Figure 4. Original Survival Functions for “DTC P1682- Ignition 1 Switch Circuit 2”

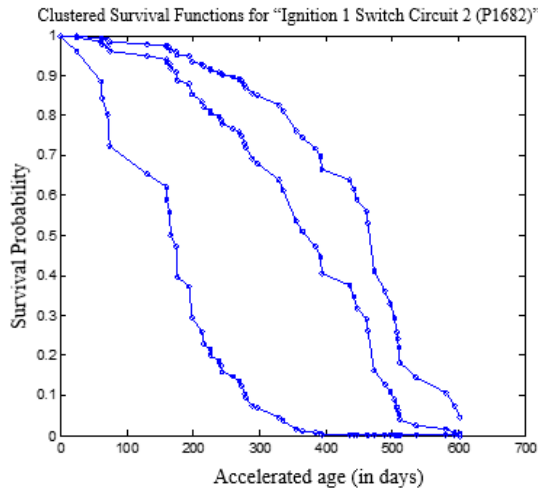


Figure 5. Survival Function Clusters for “DTC P1682- Ignition 1 Switch Circuit 2”

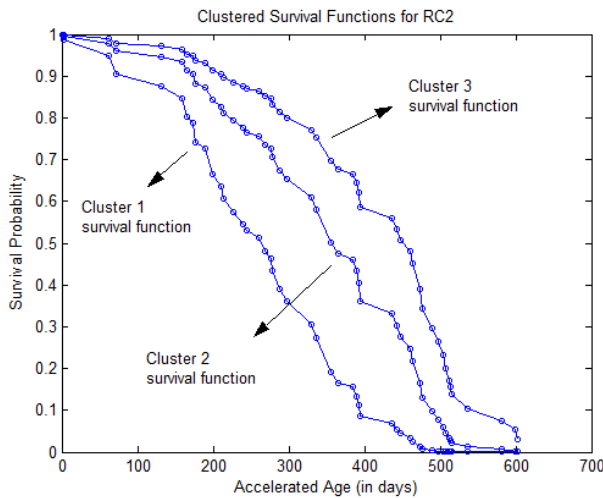


Figure 6. Survival Function Clusters for Repair Code “Throttle Body Assembly Replacement” (RC2)

diagnosis inference algorithm to infer the component degradations at any time epoch  $k$ . Under a single fault assumption, Figures 7, 8 and 9 show the estimated component failure probability for repair codes RC2 (cluster 2), RC5 (Cluster 2), and RC4 (Cluster 1), respectively. In the first two cases i.e., Figs. 7 and 8, the R- square fit is 94%, where as in Fig. 9 the R-square fit is only 78%. Although the R-square fit is low (in Fig. 9), at around 400 in the (accelerated) age axis, the estimated failure probability is higher than the actual component fault probability. This suggests that the algorithm can estimate/predict the failing component before the actual component reaches the failing threshold – which is expected of an effective prognostic algorithm. In addition to RC4 in Fig 9, the failure probability of RC2 is also significant compared to other repair codes, this is because RC2 is a hidden fault of RC4 and the algorithm inferred RC2 as failing component with failure probability of  $<0.5$ . Fig. 10 shows the mean square error (MSE) in the estimation of failure probabilities.

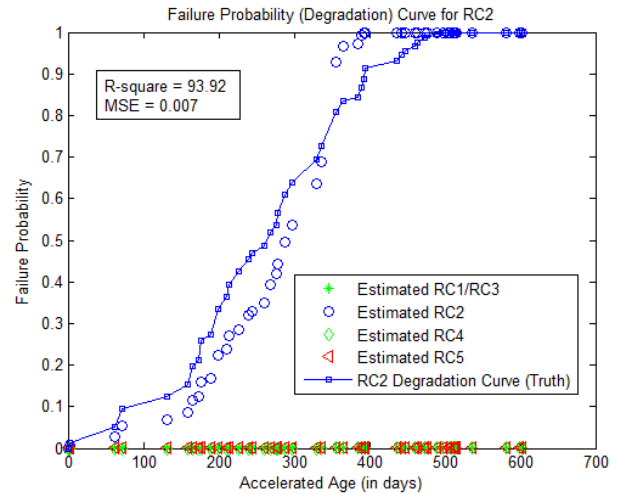


Figure 7. Estimated Degradation Curve for “Throttle Body Assembly Replacement” (RC2 Cluster 2)

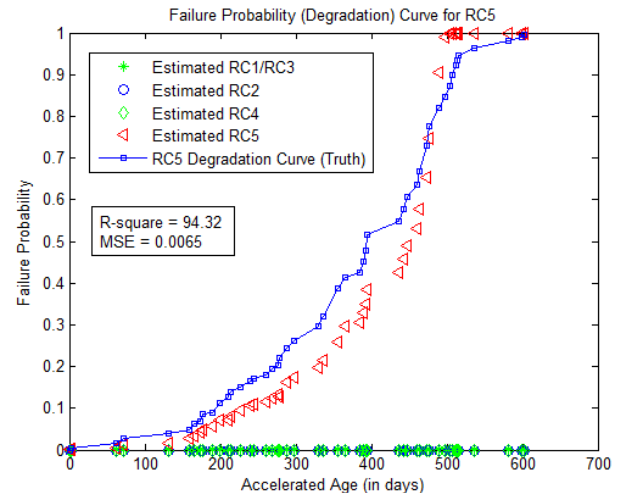


Figure 8. Estimated Degradation Curve for “Throttle Position Sensor Replacement” (RC5 Cluster 2)



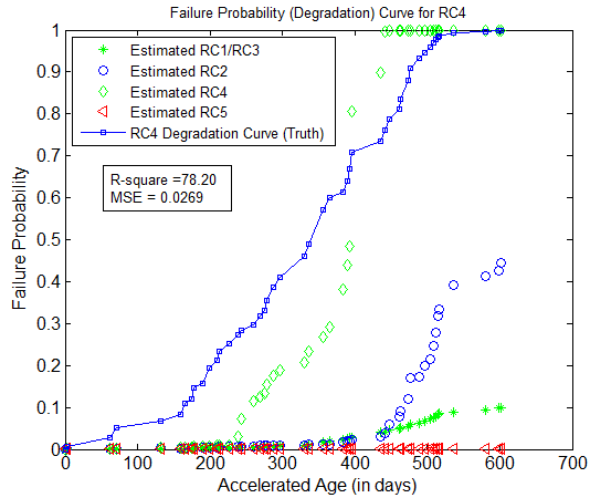


Figure 9. Estimated Degradation Curve for “Powertrain Control Module Replacement” (RC4 Cluster 1)

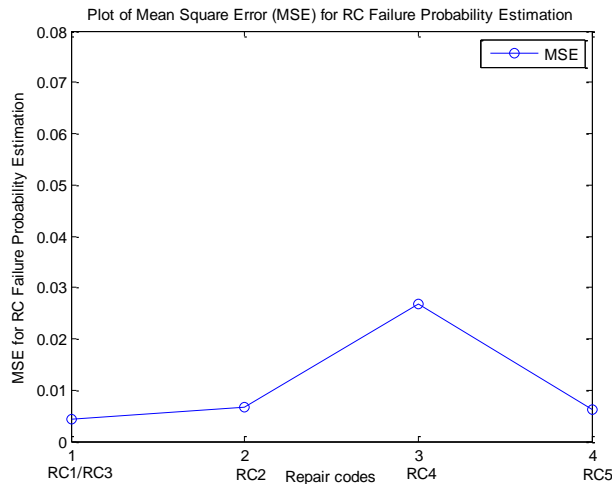


Figure 10. Plot of MSE Error for Repair Code Failure Probability Estimation

To demonstrate the validity of the proposed prognostic framework, the component survival functions prior to clustering are used for further validation. Fig. 11 shows the component degradation curves prior to clustering. Using the learned parameters  $\{pd_{ij}, pf_{ij}\}$ , the continuous test outcomes are generated for each of the RC survival functions using Eq. (11). These soft test outcomes are used as input to the inference algorithm to infer the component degradations. The component degradations are estimated with an R-square fit of about 90% except for RC4 (see Fig 12). The performance of the soft-DMFD algorithm is also compared with some of the state of the art data-driven prognostic techniques, including, support vector machine regression (SVMR), relevance vector machine (RVM), and Gaussian process (GP) regression (Goebel, Saha, & Saxena, 2008). To train these regression techniques, 5x2 cross validation is employed, i.e., 50% of the data for each of the repair codes is randomly chosen for

training the prognostic algorithm and is validated using the remaining 50% of the data and vice-versa. The process is repeated five times and the average performance in terms of R-Square and MSE are computed. Figures 12 and 13 shows the performance statistics of all the prognostic techniques in comparison. The Soft-DMFD inference technique performed better than other inference techniques. The average R-square with RVM was 85% whereas the other techniques, SVMR and GP had R-square fit of about 78% and 75%, respectively. The better performance of the soft-DMFD algorithm over others could be attributed to the Bayesian inference framework maximizing the MAP objective function and the soft-Viterbi algorithm for tracking the evolution of coupled component fault states that best explains the observed test outcomes over time. Similar performance results were observed in terms of MSE performance (Fig 13).

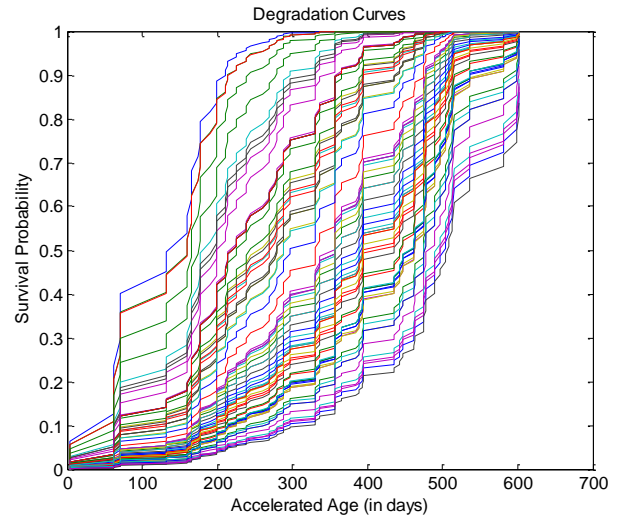


Figure 11. Component Degradation Curves

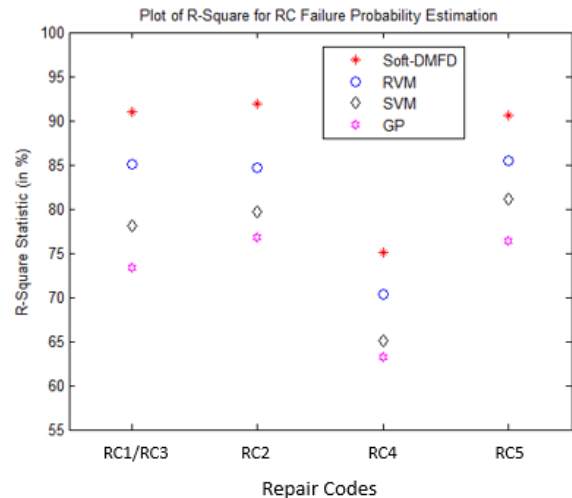


Figure 12. Plot of R-Square Statistics for Repair Code Failure Probability Estimation

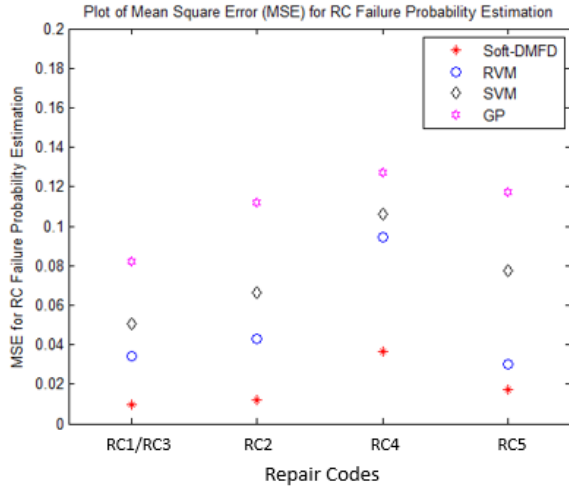


Figure 13. Plot of MSE Error for Repair Code Failure Probability Estimation

## 5.2. Application to Regenerative Braking System

In this application, the prognostic framework described in section 3 is applied to estimate the sensor and parameter degradations in a regenerative braking system (RBS). The RBS model with a series-parallel drivetrain configuration (Ehsani, Gao, & Emadi, 2009) is built using the PSAT software. It consists of a driver model, a powertrain controller, component controllers, and the powertrain model (see Fig 14).

The *driver model* simulates the drive cycles (e.g., Urban Dynamometer Driving Schedule (UDDS) [EPA]), by setting accelerator and brake pedal positions to achieve the desired vehicle speed. The output from this block is the driver's torque demand at the wheels; this acts as the input to the powertrain controller (PTC). The *powertrain controller* is

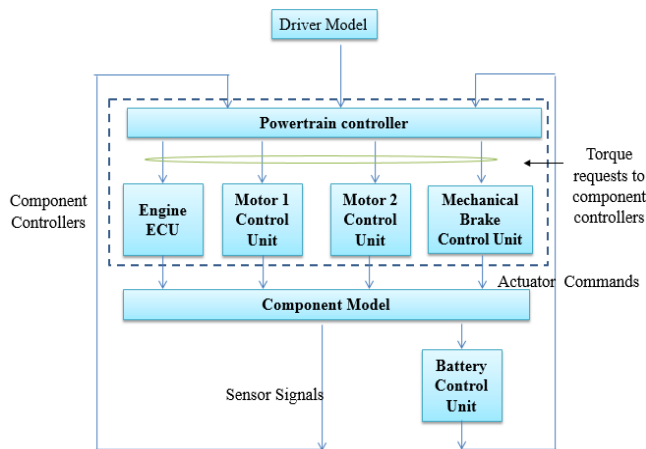


Figure 14. Functional Flow Diagram of Regenerative Braking System

supervisory controller making the high-level decisions that affect the general state of the powertrain (e.g., engine on/off), the operating mode of the vehicle (e.g., propelling, regenerative braking etc.), and accordingly deliver the torque requests to the component controller. Subsequently, the *component controller* converts these torque requests into component commands. These commands are, in turn, treated as the actuator commands by the individual components in the *powertrain model* to achieve the requested torque and, consequently, report the system status (e.g., engine speed, battery state of charge) to the supervisory controller. The powertrain model comprises of all the components that mimic the behavior of hardware components, such as the engine, the battery, and the motor. Fig. 15 shows the individual component blocks in the powertrain model (with a series-parallel drivetrain configuration). The detailed model description and mathematical details are presented in (Sankavaram, Pattipati, Pattipati, Zhang, & Howell, 2014).

There are 25 signals that are being monitored in the RBS system including (a) *sensor signals*, such as temperature, speed, and current measurements from the hardware components in the powertrain model; (b) *motor, wheel, and engine torque demands* sent from the powertrain controller to the component controllers; and (c) *component commands* sent from the individual ECU's to the hardware components in the powertrain model. The list of monitored signals is provided in Table 2.

To demonstrate the framework, two faults are considered – motor speed sensor fault and wheel inertia fault. These faults are injected into the model as additive biases on the measured signals. For instance, a 10% deviation in the motor speed is used to model the motor speed sensor fault. Similarly, the parametric fault i.e., wheel inertia fault is simulated as 10% deviation from its nominal value. Mathematically, the fault scenarios are simulated using the following equation,

$$X_{faulty} = X_{nominal} (1 + \Delta). \quad (12)$$

In Eq. (12),  $X_{faulty}$  is the parameter value under faulty condition,  $X_{nominal}$  is the nominal parameter value and  $\Delta$  is the

Monitored Signals: Signal Description	
<b>S1:</b> Battery State of Charge;	<b>S2:</b> Motor2 Torque Demand;
<b>S3:</b> Wheel Torque Demand;	<b>S4:</b> Motor1 Torque Demand;
<b>S5:</b> Engine Torque Demand;	<b>S6:</b> Battery Temperature;
<b>S7:</b> Battery Current;	<b>S8:</b> Driver Torque Demand;
<b>S9:</b> Motor1 Command;	<b>S10:</b> Gearbox Speed;
<b>S11:</b> Wheel Input Speed;	<b>S12:</b> Wheel Output Speed;
<b>S13:</b> Wheel Torque;	<b>S14:</b> Vehicle Linear Speed;
<b>S15:</b> Motor1 Speed;	<b>S16:</b> Motor1 Current;
<b>S17:</b> Clutch Input Speed;	<b>S18:</b> Engine Command;
<b>S19:</b> Motor2 Command;	<b>S20:</b> Motor2 Speed;
<b>S21:</b> Motor2 Current;	<b>S22:</b> Engine Speed;
<b>S23:</b> Clutch Output Speed;	<b>S24:</b> Mechanical Accessory Torque;
<b>S25:</b> Wheel Command	

Table 2. List of monitored signals

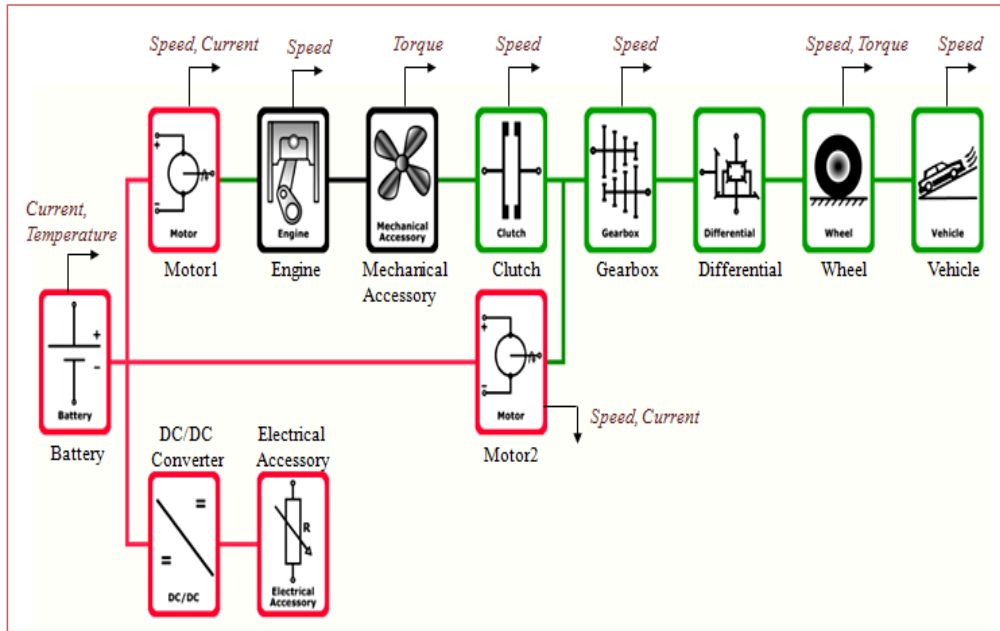


Figure 15. Vehicle Powertrain Model with Series-Parallel Configuration

fractional change in the parameter value (fault severity). The simulation data (monitored signals) thus obtained is used in the prognostic framework for estimating the component degradations.

The component degradations (or) fault evolution is assumed to be of sigmoid (S-) shape as shown in Fig. 16 with degradation progressing gradually from low severity level (2%) to high severity level (10%) and eventually leading to a complete failure. Fig. 16 illustrates the fault progression from nominal to failure. Here, 10% severity level is treated as the component failure, i.e., the component has failed and is not considered to be operational at this point.

As a first step, the simulated failure time data (Type I data) for sensor and parametric faults is considered as shown in Table 3. Since the component failure is assumed to occur at 10% severity level, the corresponding feature data i.e., monitored variables is treated as the static parameter data (features at the time of failure) – Type II data. This feature data corresponding to 10% fault severity level is obtained via simulation-based fault injection experiments. During the offline training phase, the failure time data and static parameter data are used to learn the component survival functions using Cox PHM model (see (1)). Figures 17 and 18 show the baseline survival function and the complementary survival functions (component degradation curves) for motor speed sensor (F2) and wheel inertia (F3) faults.

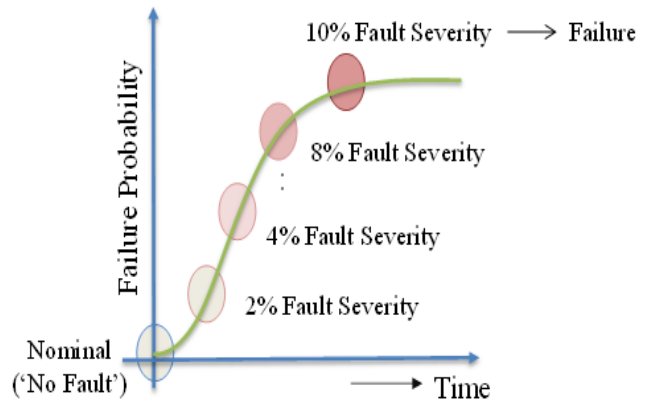


Figure 16. An Illustration of Failure Progression from Nominal to Faulty

Failure Times (Motor Speed Sensor)	Failure Times (Wheel Inertia)
200	250
400	450
600	650
800	850
1000	1050

Table 3. Simulated Failure Time Data for Components

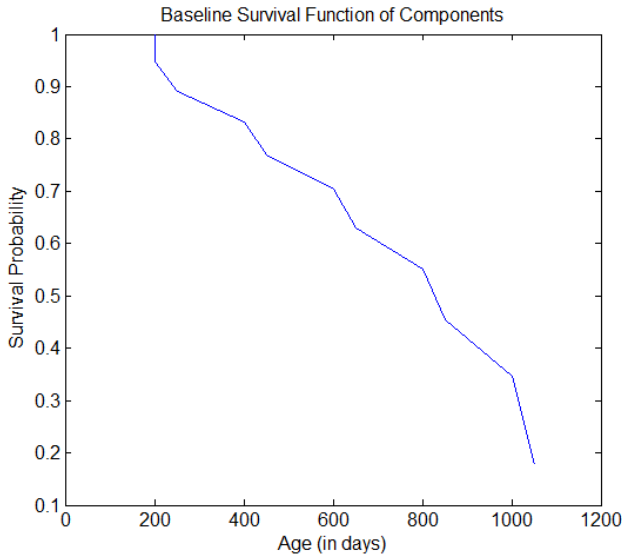


Figure 17. Baseline Survival Function of Components

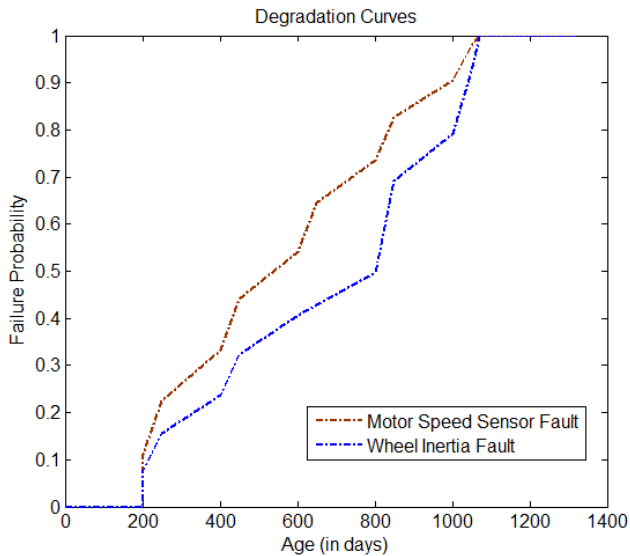


Figure 18. Degradation Curves (Complementary Survival Functions) of Motor Speed Sensor and Wheel Inertia Faults

Next, the fault-test dependency matrix (or) a diagnostic matrix (D-Matrix) is prepared for the faults in the training phase. Table 4 shows a partial view of the D-Matrix. To generate this matrix of relationships between faults and tests, a simple mean threshold test is employed on the residual signals as a detection technique. A model is first used to generate nominal and faulty data, and the residual signals are computed as the deviations of faulty measurements from the nominal ones. When there is no fault, the mean of the residual signals is expected to be close to zero, whereas in the

	T1	T2	T3	T4	T5	T6	T7	...	T21	T22	T23	T24	T25
F1	0	0	0	0	0	0	0	...	0	0	0	0	0
:								...					
F5	0	0	0	0	1	1	1	...	1	1	1	0	0
:								...					
F7	0	1	0	0	0	1	0	...	0	1	0	0	0
:								...					

Table 4. Partial View of Diagnostic Matrix  
(F1: Nominal, F5: Motor Speed Sensor Fault,  
F7: Wheel Inertia Fault, T1-T25: Tests for 25 monitored signals)

Failure Times (Test '1')	Failure Times (Test 'k')
180	260
380	420
580	620
780	820
980	1030

Table 5. Sample Failure Time Data for Tests

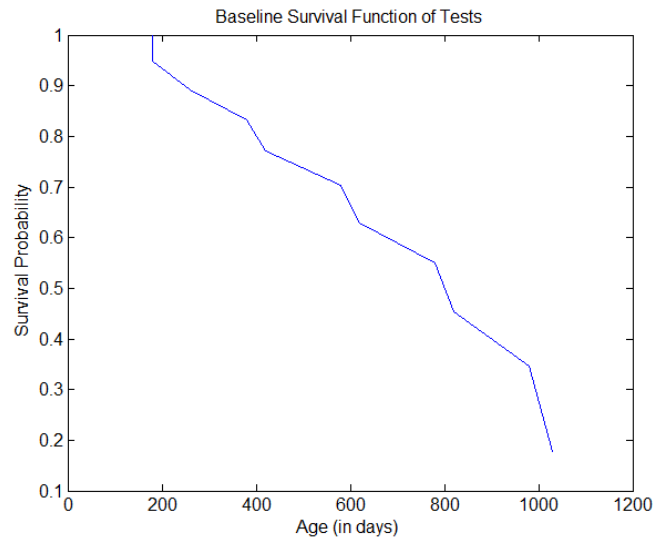


Figure 19. Baseline Survival Function of Tests

presence of a fault, the residual signal does not have a zero-mean. Hence, a threshold test on the mean of the residual signal indicates whether the system is deviating from its nominal state. These tests are assumed to fail at least 10 days prior to the component failure as shown in Table 5. Similar to the component survival functions, the baseline survival function for tests is also obtained via Cox Model as in Fig 19 and a sample of test survival functions is provided in Fig 20.

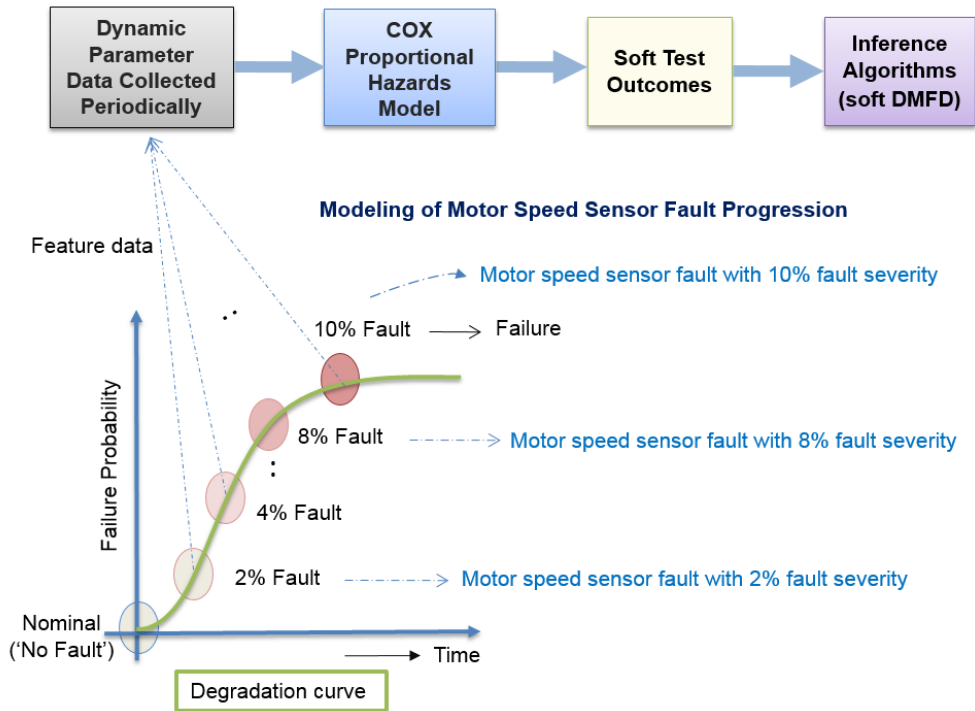


Figure 21. Steps involved in Testing Phase (Online Phase)

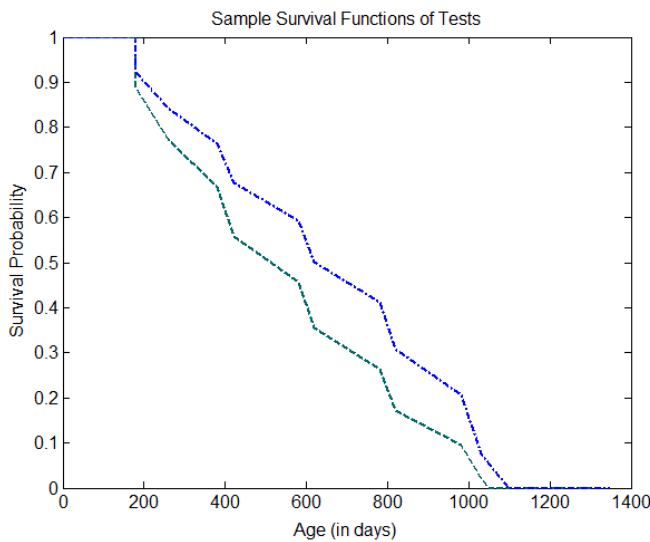


Figure 20. Sample Survival Function of Tests

During the testing phase, when the dynamic feature data (25 monitored signals) is collected periodically, the learned baseline survival functions and the feature data is used to compute test probabilities (see Eq. 2). These test probabilities are fed to soft-DMFD to infer the component degradations (complementary survival functions or failure probabilities). Refer to Fig. 21 for the steps involved in the testing phase.

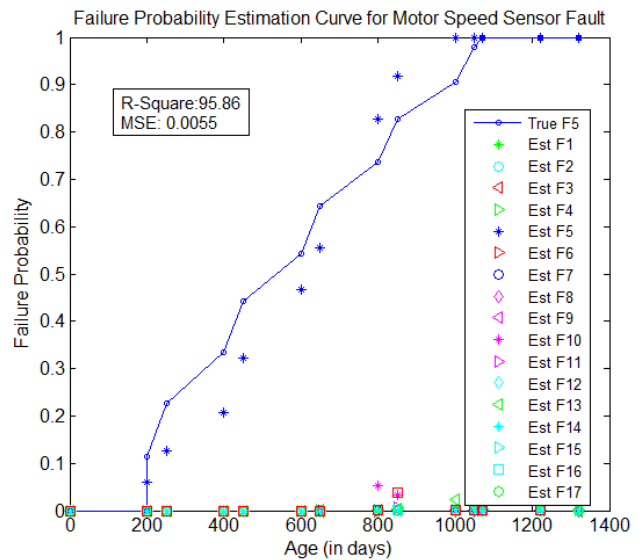


Figure 22. Degradation Curve Estimation for Motor Speed Sensor Fault

Figures 22 and 23 show the estimated component failure probability for motor speed sensor fault and wheel inertia faults, respectively. The estimated component degradations are in good agreement with the truth with an R-square fit of about 96%. An estimate of RUL at any time  $t$  can be obtained by defining a threshold on the failure probability.

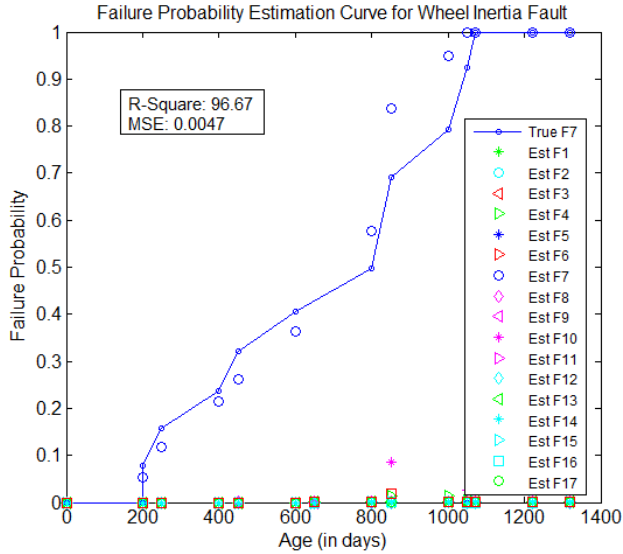


Figure 23. Degradation Curve Estimation for Wheel Inertia Fault

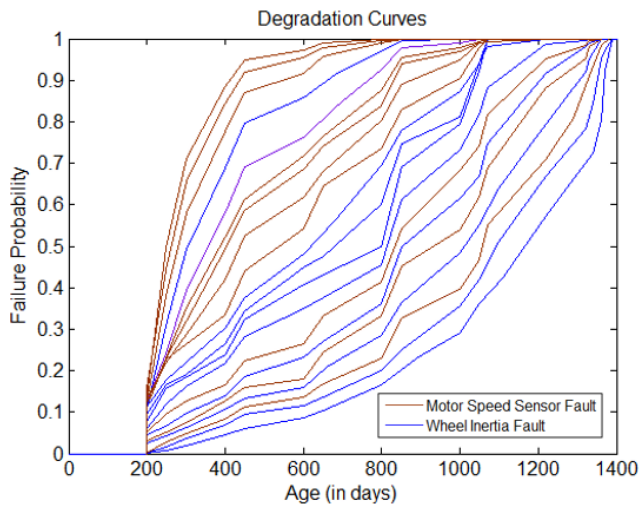


Figure 24. Degradation Curves of Motor Speed Sensor and Wheel Inertia Faults

To establish the validity of the proposed approach, a Gaussian measurement noise with random seed is added to the monitored signals to generate additional patterns for the two faults considered in this application. The variance of the added noise is 0.6% of the squares of the magnitude of the signals; this corresponds to a signal-to-noise (SNR) ratio of 22.2 dB. A total of 10 patterns are generated for each fault, Fig. 24 shows the evolution of component degradations for motor speed sensor and wheel inertia faults. Here, the evolution of the faults is assumed to be of sigmoid shape (monotonically increasing), other forms of fault evolution will be studied as part of our future research. The dynamic feature data, i.e., feature data corresponding to 2%, 4%, ..., 10% fault severity levels are used along with the learned baseline survival functions to generate test probabilities (see

Eq. 2). When these test probabilities are fed to the soft-DMFD algorithm, the component degradations are estimated with an R-square fit of about 94%. Table 6 shows the average R-square and MSE in estimating the failure probabilities of motor speed sensor and wheel inertia faults. Table 6 also shows the comparison of soft-DMFD performance with other data-driven prognostic techniques. The soft-DMFD inference technique performed better than other inference techniques with 94% R-square fit. RVM was the next best technique with an average R-square value of 83% whereas the other techniques SVMR and GP had an R-square fit of about 75% and 68%, respectively. Similar performance results were observed in terms of MSE performance.

Fault Type	Inference Technique	R-square	MSE
Motor Speed Sensor Fault	Soft DMFD	93.64	0.0071
	SVMR	78.84	0.0589
	RVM	84.18	0.0462
	GP	72.27	0.1109
Wheel Inertia Fault	Soft DMFD	94.42	0.0059
	SVMR	74.66	0.0633
	RVM	83.33	0.0587
	GP	68.49	0.1812

Table 6. Performance Results

## 6. CONCLUSIONS

The paper presented a novel approach for fault prognosis problem in coupled systems by combining three types of data, i.e., failure time data, static environmental and status parameter data, and dynamic data. The framework employed the Cox PHM to infer the survival functions of components and subsequently estimated the component degradations via the soft dynamic multiple fault diagnosis algorithm. The framework is applied to two different automotive applications to infer the component degradations (complementary survival functions) and the inference algorithm estimated the component failure probabilities with a good R-square fit.

The future work should include the application of this approach to continuous parameter identifier data and account for the uncertainty in RUL estimation. Another extension of the Cox-PHM framework for prognosis is by modeling coupled survival dynamics as monotone positive linear systems or monotone Markov processes (Zaslavski, 1984; Zaslavski, 1987). In monotone positive linear systems, the state matrix is a Metzler matrix (i.e., has nonnegative off-diagonal elements) to ensure that the state variables (in our case, survival functions) are nonnegative. In monotone Markov processes, the state generator matrix is upper triangular. Investigation of these concepts in the context of prognosis to estimate survival functions is a novel extension of the proposed prognosis framework.



## APPENDIX

APPENDIX A  
Diagnostic Trouble Code Description

Diagnostic Trouble Code	Diagnostic Trouble Code Description
P0101	Mass Air Flow Sensor Performance
P0102	Mass Air Flow Sensor Circuit Low Frequency
P0121	Throttle Position Sensor Performance
P0601	Control Module Read Only Memory
P1516	Throttle Actuator Control Module Actuator Position Performance
P1682	Ignition 1 Switch Circuit 2
P2101	Control Module Throttle Actuator Position Performance
P2122	Accelerator Pedal Position Sensor 1 Circuit Low Voltage
P2127	Accelerator Pedal Position Sensor 2 Circuit Low Voltage
P2138	Accelerator Pedal Position Sensor 1-2 Correlation
P2176	Minimum Throttle Position Not Learned

APPENDIX B  
Repair Code Description

Repair Code	Repair Code Description
RC1	Accelerator Pedal Replacement
RC2	Throttle Body Assembly Replacement
RC3	Accelerator Pedal Position Sensor Replacement
RC4	Powertrain Control Module Replacement
RC5	Throttle Position Sensor Replacement

## ACKNOWLEDGMENTS

We thank GM ISL and NSF (ECCS-0931956 (NSF CPS), ECCS-1001445 (NSF GOALI), CCF-1331850 (Cyber-SEES)) for their support of this work. Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsors.

## REFERENCES

Adams, D. E. (2002). Nonlinear damage models for diagnosis and prognosis in structural dynamic systems. In *SPIE Conference Proceedings*, vol. 4733, pp. 1-12.

Bishop, C. M. (1997). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Byington, C. S., Watson, M., Edwards, D. and Stoelting, P. (2004). A model-based approach to prognostics and health management for flight control actuators. *Proceedings of IEEE Aerospace Conference*, Vol. 6, pp. 3551.

Celaya, J. R., Saxena, A., and Goebel, K. (2012). *Uncertainty representation and interpretation in model-based prognostics algorithms based on kalman filter estimation*. NASA Ames Research Center, Moffett Field, CA.

Chelidze, D., Cusumano, J. P., & Chatterjee, A. (2002). Dynamical systems approach to damage evolution tracking, part I: The experimental method. *Journal of Vibration and Acoustics*, vol. 124, pp. 250-257.

Chiang, L. H., Russel, E., & Braatz, R. (2001). *Fault detection and diagnosis in industrial systems*. London: Springer-Verlag.

Coble, J. B. (2010). *Merging data sources to predict remaining useful life – an automated method to identify prognostic parameters*, PhD dissertation, University of Tennessee.

Cox, D. R., & Oakes, D. (1984). *Analysis of survival data*. Chapman and Hall.

Daigle, M. J., and Goebel, K (2011). A model-based prognostics approach applied to pneumatic valves. *International Journal of Prognostics and Health Management*, vol. 2: 84.

Duda, R. O., Hart, P. E., & Stork, D. (2000). *Pattern classification*. John Wiley & Sons, New York.

Ehsani, M., Gao, Y., and Emadi, A. (2009). *Modern electric, hybrid electric, and fuel cell vehicles: fundamentals, theory, and design*. CRC press.

Environmental Protection Agency (EPA). *Dynamometer Drive Schedules*. [Online]. Available: <http://www.epa.gov/nvfel/testing/dynamometer.htm>. [Accessed: 02-Jul-2015].

Garga, A. K., McClintic, K. T., Campbell, R. L., Yang, C. C., & Lebold, M. S. (2001). Hybrid reasoning for prognostic learning in cbm systems. *Proceedings of IEEE Aerospace Conference*, pp. 2957-2969.

Gabraeel, N. and Lawley, M. A. (2008). A neural network degradation model for computing and updating residual life distributions. *IEEE Trans. Automation Science and Engineering*, vol. 5, no. 1, pp. 154–163.

Goebel, K., Saha, B., and Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. 62<sup>nd</sup> meeting of the society for machinery failure prevention technology (MFPT), pp. 119-131.

Gorjian, N., Ma, L., Mittinty, M., Yarlagaadda, P., Sun Y. (2009a). A review on reliability models with covariates. *Proceedings of the 4th World Congress on Engineering Asset Management*, Athens, Greece, September.

Gorjian, N., Ma, L., Mittinty, M., Yarlagaadda, P., Sun Y. (2009b). A review on degradation models in reliability analysis. In: *Proceedings of the 4th World Congress on*

- Engineering Asset Management, Athens, Greece, September
- Jardine, A. K. S., & Tsang, A. H. C. (2005). *Maintenance, Replacement, and Reliability: Theory and Applications*, New York: CRC Press.
- Klabfleisch, J. D., & Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data 2<sup>nd</sup> edition*, New York, Wiley.
- Klein, J. P., & Moeschberger, M. L. (2003). *Survival Analysis Techniques for Censored and Truncated Data*. Springer-Verlag, New York.
- Kodali, A., Ponizovskaya-Devine, E., Robinson, P., Luchinsky, D., Bajwa, A., Khasin, M., Perotti, J, and Brown, B. (2015). D-matrix based fault modeling for Cryogenic loading systems. *Annual Conference of the Prognostics and Health Management Society*.
- Kumar, S., Torres, M., Chan, Y. C., & Pecht, M. (2008) A hybrid prognostics methodology for electronic products. *IEEE International Joint Conference on Neural Networks*.
- Lall, P., Pecht, M., and Harkim, E. (1997). *Influence of Temperature on Microelectronics and System Reliability*, New York: CRC Press.
- Luo, J., Namburu, M., Pattipati, K., Qiao, L., Kawamoto, M., & Chigusa, S. (2003). Model-based prognostic techniques. *IEEE Autotestcon Conference*, pp. 330-340.
- Ma, Z., & Krings, A. W. (2008). Survival analysis approach to reliability, survivability and Prognostics and Health Management (PHM). *IEEE Aerospace Conference*, May.
- Metzler, D., Lavrenko, V., & Croft, W. B. (2004). Formal Multiple Bernoulli Models for Language Modeling. *Proceedings of 27th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR' 04)*, Sheffield, UK, pp. 540-541.
- Murthy, D. N. P., Xie, M., & Jiang, R. (2004). *Weibull Models*, New York: Wiley.
- Pattipati, K., & Alexandridis, M. G. (1988). Application of heuristic search and information theory to sequential fault diagnosis. *IEEE International Symposium on Intelligent Control*, pp. 291-296, August.
- Pearl, J. (1986). Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, Vol. 29, No. 3, pp. 241-288.
- Pecht, M., Das, D., and Ramakrishnan, A. (2002). The IEEE standards on Reliability Program and Reliability Prediction methods for Electronic Equipment," *Microelectronic Reliability*, pp. 1259-1266.
- Ray, A. and Tangirala, S. (1996). Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, Vol.4, pp. 443-451.
- Sankavaram, C., Kodali, A., Pattipati, K., Wang, B., Azam, M., and Singh, S. (2011). A Prognostic Framework for Health Management of Coupled Systems. *IEEE International Conference on Prognostics and Health Management*, Denver, CO, June.
- Sankavaram, C., Pattipati, B., Pattipati, K., Zhang, Y., and Howell, M. (2014) "Fault Diagnosis in Hybrid Electric Vehicle Regenerative Braking System," *IEEE Access*, vol. 2, pp. 1225-1239, October.
- Schwabacher, M. A. (2005). A survey of data-driven prognostics. Infotech@Aerospace, American Institute of Aeronautics and Astronautics 2005-7002, September, Arlington, Virginia.
- Serrao, L., Onori, S., Rizzoni, G., and Guezennec, Y. (2009). A novel modelbased algorithm for battery prognosis. 7th *IFAC SAFEPROCESS*.
- Si, X., Wang, W., Hu, C., and Zhou, D. (2011) Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, Vol. 213, No. 1, pp 1-14.
- Singh, S., Kodali, A., & Pattipati, K. (2009a). A factorial hidden Markov model-based reasoner for diagnosing multiple intermittent faults. *IEEE CASE*, Bangalore, August.
- Singh, S., Kodali, A., Choi, K., Pattipati, K., Namburu, S. M., Chigusa, S., Prokhorov, D. V., & Qiao, L. (2009b). Dynamic Multiple Fault Diagnosis Problem Formulations and Solution Techniques. *IEEE Trans. on Systems, Man and Cybernetics: Part A*, vol. 39, no. 1, pp. 160-176, January.
- Singh, S., Holland, S. W., & Bandyopadhyay, P. (2010). Trends in the development of system-level fault dependency models. *IEEE Aerospace Conference*, March.
- Swanson, D. (2001). A general prognostic tracking algorithm for predictive maintenance. *Proceedings of IEEE International Conference on Aerospace*, vol.6, pp. 2971-2977.
- Vichare, N., Rodgers, P., Evely, V., and Pecht, M (2004). In situ temperature measurement of a notebook computer – A case study of health and usage monitoring of electronics. *IEEE trans. On Device and Materials Reliability*, pp. 658 – 663, December.
- Vlok, P. J., Coetzee, J. L., Banjevic, D., Jardine, A. K. S., & Makis, V. (2002). Optimal Component Replacement Decisions Using Vibration Monitoring and the PHM. *Journal of the Operational Research Society*, Vol. 53, 193-202.
- Wang, P., & Vachtsevanos, G. (1999). Fault prognosis using dynamic wavelet neural networks. in *Maintenance and Reliability Conference (MARCON 99)*, May.