A Condition Based Maintenance Implementation for an Automated People Mover Gearbox

Ali Ashasi-Sorkhabi[1], Stanley Fong[2], Guru Prakash[3], and Sriram Narasimhan[4,5]

[1, 2, 3, 4] *Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, ON, N2L3G1, CA*

[5]*Department of Mechanical & Mechatronics Engineering, University of Waterloo, Waterloo, ON, N2L3G1, CA*

aliashasi2010@gmail.com
stan.fong@uwaterloo.ca
gprakash66@gmail.com
snarasim@uwaterloo.ca

ABSTRACT

Data-driven condition-based maintenance (CBM) can be an effective predictive maintenance strategy for components within complex systems with unknown dynamics, non-stationary vibration signatures or a lack of historical failure data. CBM strategies allow operators to maintain components based on their condition in lieu of traditional alternatives such as preventive or corrective strategies. In this paper, the authors present an outline of the CBM program and a field pilot study being conducted on the gearbox, a critical component in an automated cable-driven people mover (APM) system at Toronto's Pearson airport. This CBM program utilizes a paired server-client "two-tier" configuration for fault detection and prognosis. At the first level, fault detection is performed in real-time using vibration data collected from accelerometers mounted on the APM gearbox. Time-domain condition indicators are extracted from the signals to establish the baseline condition of the system to detect faults in real-time. All tier one tasks are handled autonomously using a controller located on-site. In the second level pertaining to prognostics, these condition indicators are utilized for degradation modeling and subsequent remaining useful life (RUL) estimation using random coefficient and stochastic degradation models. Parameter estimation is undertaken using a hierarchical Bayesian approach. Degradation parameters and the RUL model are updated in a feedback loop using the collected degradation data. While the case study presented will primarily focus on a cable-driven APM gearbox, the underlying theory and the tools developed to undertake diagnostics and prognostics tasks are broadly applicable to a wide range of other civil and industrial applications.

1. INTRODUCTION AND OBJECTIVES

1.1. Background on the application

Condition-based maintenance (CBM) is a predictive maintenance strategy where the maintenance decisions are made based on the current health of the system. CBM has recently evolved as a viable alternative to traditional maintenance methods such as run-to-failure maintenance, which is performed in response to failures, and preventive maintenance, where maintenance actions are carried out periodically without a complete knowledge of the system's (or component) health. A properly implemented CBM framework addresses the main pitfalls of traditional maintenance methods by minimizing the unplanned system downtimes common in the run-to-failure maintenance method, while reducing the number of planned preventive maintenance actions required (Jardine et al., 2006).

CBM is comprised of two main components: diagnostics and prognostics. Diagnostics is a multi-level, sequential process consisting of fault detection, fault isolation, and fault diagnosis. The diagnostic capabilities of a CBM framework can be determined by the role the framework plays in the overall system maintenance. For systems that are designed to complement or aid maintenance personnel, such as the CBM framework presented herein, it is sufficient to only perform the first level of diagnostics (detection). Prognostics involve the prediction of the future performance of a monitored system. It includes the prediction of the remaining useful life (RUL) of a system, and the determination of, if and when a fault will occur, and the likelihood of that fault occurring. These measures of prognostics are obtained using degradation models generated from data collected from sensors mounted on the system. A CBM program can be designed to perform diagnostics or prognostics, or both, and is comprised of three major operational steps: collection of sensor data, post-processing the collected data, and maintenance decision-making (Jardine et al., 2006).

This paper concerns with the application of CBM to automated people movers (APMs), and presents a university-industry collaborative between the authors, Toronto's Pearson Airport, and Doppelmayr Cable Car Ltd., who commissioned this system and currently operate it at the airport. APMs are guided mass transit systems that utilize computer-controlled trains to transport passengers across a dedicated network. In many commercial airports, APMs are an integral piece of infrastructure to their operation, providing passengers with a quick and efficient means of transportation between key facilities within the airport and to offsite facilities.

The CBM application in this paper is for the gearbox of a live APM, which is a cable-driven system at Toronto's Pearson Airport, named the LINK train (LINK) (Figure 1). LINK provides the airport's 25 million annual passengers with continuous service between the airport's two terminals and parking facilities. Consequently, unexpected shutdowns due to faults in the system can lead to significant inconvenience to the passengers and indirect economic impact to the airport. Excessive unnecessary maintenance can also prove to be costly in the long term. CBM is an attractive solution to this paradigm, and this application provides a rich test-bed to pilot a CBM program and to demonstrate the interconnectivity between diagnosis and prognosis, and key implementation aspects in achieving the same.

The APM system studied here (Figure 1(a)) is of the cable-driven type that consists of a cable-propelled train mounted on a steel guideway. The trains themselves do not contain any drive assemblies. Rather, power is generated from a central station that houses all of the drive train machinery. Tractive forces are then transferred to the train through a cable that is fixed to the underside of the carriage (Doppelmayr Cable Car, 2017; Lee et al., 2016). The most critical component in this APM system is the gearbox (strictly speaking, a gearbox is a system consisting of a gears, shaft and bearings), and hence it is natural to start to the CBM pilot on this system. The overall health of the gearbox can be cost-effectively monitored using vibration measurements using accelerometers mounted directly on the gearbox housing, at critical locations, as shown in Figure 1(b). The locations shown on the gearbox housing in Figure 1(b) have been chosen in a way that the sensors are placed in the immediate proximity of the main shaft and bearings. Prior to describing the main contributions of this paper, a brief review of background literature is described next.

a)

b)

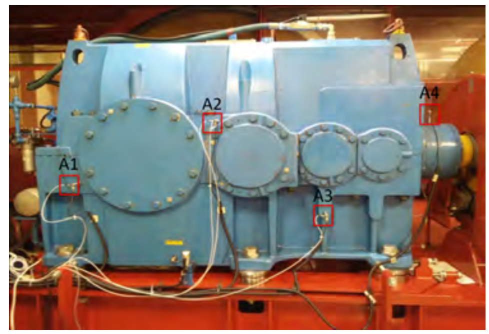

Figure 1. a) APM train (Doppelmayr Cable Car, 2017), b) gearbox housing with accelerometers A_1 to A_4

1.2. Literature Review

There exists a relatively large volume of literature on vibration-based machinery diagnostics and prognostics techniques, specifically related to systems with rotating components. The review paper by (Jardine et al., 2006) provides a comprehensive summary on vibration-based damage detection including time domain, frequency domain, and time-frequency domain methods, as well as pattern recognition analysis. The information provided can be effectively used for bearing and gear fault diagnostics. Another review paper (Robert B. Randall & Antoni, 2011), provides an excellent tutorial of the most effective contemporary techniques for rolling element diagnostics. Although the literature on the machinery diagnostics is rich, majority of the work is focused on the fault detection of systems with high speed rotating components, and there is relatively little study carried out towards the maintenance planning of machinery systems with low-speed components, such as critical components within cable-driven APMs. Recently, Bechhoefer et al., 2016 and Yin et al., 2014 studied a number of vibration based techniques for large, slow bearing fault diagnostics in wind turbines, which is one of the few studies that exist on this topic today.

Data-driven techniques have been shown to be efficient and reliable means to perform fault detection in rotating machinery systems (Bechhoefer et al., 2016; Timusk et al., 2008; Yin et al., 2014). This is due to the fact that in such methods fault classification (e.g., faulty or not) is obtained based on the statistical analysis of the vibration signatures even without a need for *a priori* knowledge regarding the physical characteristics of the component/system being studied. This reduces the need for extensive signal processing and more importantly reduces user intervention, which usually makes them more conducive to automation. The idea behind data-driven techniques is to employ suitable condition indicators (e.g., extracted from vibration signals) which are sensitive to fault signatures embedded within the measurements (Timusk et al., 2008). Root mean square value (RMS), crest factor (CF), kurtosis and skewness are some examples of statistical condition indicators previously used for condition monitoring of rotating components employing vibration measurements (Večeř et al., 2005). Over the recent decades, several statistical methods such as hidden Markov models (HMMs) (Rabiner, 1989; Bunks & Mccarthy, 2000; Boutros & Liang, 2011), semi-Markov models (Q. Liu et al., 2012), hybrid HMMs (Sadhu et al., 2016), support vector machines (Sharma, et al. 2014), Gaussian mixture models (GMMs) (Nelwamondo et al., 2006), and symbolic analysis have all been used successfully for rotating machinery diagnostics (Chin et al., 2005).

Prognostics literature on low-speed, live, non-stationary components that lack failure data, such as the case with the APM under consideration here, is scarce. Generally, there are two main approaches for RUL estimation or reliability predictions: model based and data-driven approaches. Model based approaches involve modeling the underlying degradation process based on the physics of the failure mechanism; while, data-driven approaches build a statistical model solely based on collected sensory data (Si et al., 2011). Model-based techniques are difficult to implement for components embedded in complex systems such as APMs, as the dynamics and degradation processes are seldom conducive to direct modeling. On the other hand, statistical models, such as random coefficient regression or power law based regression models, can operate directly on degradation surrogates obtained from measurements to model the degradation path, and to predict the RUL distribution given a predefined threshold. Such models have widely been used (Chen & Tsui, 2013; Kaiser & Gebraeel, 2009; X. Wang et al., 2014) for bearing prognosis. When these methods are integrated in a Bayesian framework, they can estimate and update the reliability predictions in real-time even when prior information is limited. In some cases, such as exponential models or linear models with Weibull distributed slopes, a closed form expression for time to failure can be derived, which makes Bayesian inference relatively straight-forward (Lu & Meeker, 1993). When such closed-form solutions are not possible, their reliability can be assessed using sampling methods such as Markov chain Monte Carlo (MCMC) based methods.

An issue often not addressed in the literature is the practical implementation on live systems. Automation of a CBM implementation with various interconnected tasks presents its own unique set of challenges. The mapping of real world observations to virtual instances, the separation of prior knowledge and the fluent and timely distribution of sensor data to various system components throughout different operating states are all non-trivial tasks to automate (Baum et al., 2017). Currently, at best, these routines have been developed to work semi-autonomously (Robert Bond Randall, 2012, Chapter 5), and most autonomous methods are limited to fault detection only and do not address the prognostics aspects.

1.3. Objectives and contributions

The first objective pursued in this paper is fault detection in the gearbox of the APM described earlier. The inherent stochastic nature of gearbox vibration signals, coupled with the addition of non-stationarity due to constant stop and start phases of the train and the multi-path convolved sources from the constituent components in the gearbox makes the signal isolation of a single component extremely challenging. In addition, the slow speed of the system may render traditional impact-based signal processing techniques for fault detection ineffective. Furthermore, such sophisticated signal processing techniques are not conducive to automation, which limits the application to near real-time fault-detection. All these challenges call for an alternate, simpler, data-driven approach to perform fault detection , which is conducive to automation.

The second objective is to develop a data-driven methodology for undertaking prognostics. This is accomplished using a degradation model employing surrogates extracted from vibration data. The problem of prognosis is compounded by the fact that prior failure information is unavailable and has to be updated over time. A Bayesian hierarchical approach is used to set and update failure thresholds in the absence of historical data, while the Gaussian mixture models reinforced with information criteria decomposes the system's behavior into its unique operating states to allow for state-based monitoring. The only sensors used in this application are accelerometers and the data processing platform consists of a commercially available programmable controller and a PC. This CBM platform is designed to complement, not replace, existing maintenance personnel. The CBM prognostics outputs, RUL and reliability, are invaluable metrics for asset management and long-term maintenance planning, respectively.

There are two main contributions of this paper. First, the presented approach is specifically targeted towards fault detection in a cable-driven APM gearbox based on operational data. To the authors' knowledge, this is a first of its kind in the literature. The proposed framework is designed to be conducive to automation. Secondly, the use of the hierarchical Bayesian approach for model updating allows the framework to periodically generate fault detection failure thresholds based on the actual condition of the system; and, provides the flexibility to model the degradation of systems with very little historical failure information. Many of the data-driven statistical tools used in this paper are available in the literature and a brief background on the key tools is given in this section.

2. BACKGROUND AND METHODOLOGY

In this study a statistical process control (SPC)-based methodology is employed for the fault detection and prognosis of the APM system described previously. SPC describes an approach in which statistical methods are used to monitor and control a process. This CBM framework is comprised of two unique phases: a training phase and a continuous monitoring phase, as depicted in Figure 2. In the training phase, which only occurs once at the onset of the project, a representative sample of vibration data, collected through a set of accelerometers, is used as an input to generate the initial GMM and a degradation model (described in detail in this section). From these models the initial failure thresholds and initial degradation path parameters are obtained. The duration of the training phase should be chosen such that all the possible operational conditions (e.g. loads on the system) affecting the performance of the system are captured. For instance, for the training phase of the case-study presented in this paper, a 16-hour long vibration data was used which was recorded at different times of the day and week to cover any fluctuations in load due to different passenger volumes. In the continuous monitoring phase, the initial failure thresholds and initial degradation path parameters appear as inputs, alongside continuous raw vibration data (i.e. raw acceleration signals) collected throughout continuous monitoring. The continuous feature data (i.e. condition indicators), which is extracted from raw vibration signals, is compared against the failure thresholds to check for exceedance as well to update the GMM and degradation models. In the case of an exceedance, an alarm is generated to notify maintenance personnel of a potential fault. Upon inspection, if a fault is observed, the appropriate maintenance actions can be scheduled. If no fault is observed, the incident is tagged as a false alarm, and this information is fed back into the process to update the GMM and degradation model. In the following section, various components and the techniques employed in this two-tier framework are described in detail.

2.1. Background on GMM

Vibration data obtained from the cable-driven APM depends on several characteristics such as speed, direction and load, broadly termed as "states". The proper identification and quantification of these state-induced vibration features is vital to the success of an SPC-based fault detection algorithm, since the premise of SPC is to detect statistical anomalies. GMMs can be used to help capture these anomalies in the observed data. A GMM is a probabilistic model that assumes that a given set of observed data can be modeled by a finite mixture of Gaussian distributions with unknown parameters (Nelwamondo et al., 2006). This is represented by Eq. (1) in one dimension, where K denotes the number of mixture components, and μ_i, σ_i, ϕ_i denote the mean, standard deviation and weight of mixture component i.

$$p(x) = \sum_{i=1}^{K} \phi_i \mathcal{N}(x \,|\mu_i, \sigma_i) \tag{1}$$

$$\mathcal{N}(x \,|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_i)^2}{2{\sigma_i}^2}\right) \tag{2}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{3}$$

Eq. (1) is expanded for a multi-variate case as represented in Eq. (4), where the variance σ_i is replaced by the covariance matrix Σ_i.

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i) \tag{4}$$

$$\mathcal{N}(x \,|\mu_i, \sigma_i) = \frac{1}{\sqrt{(2\pi)^K|\Sigma_i|}} \exp\left(-\frac{(\vec{x}-\vec{\mu}_i)^T}{\Sigma_i(\vec{x}-\vec{\mu}_i)}\right) \tag{5}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{6}$$

2.2. Bayes rule and posterior inference

For processes where additional knowledge becomes available over time, a means to integrate the newly acquired knowledge with existing prior knowledge is beneficial. Baye's rule is an effective method to combine such prior knowledge of the process parameters θ, (i.e. for the GMM, $\theta = (\mu, \sigma^2)$) with newly acquired knowledge. This approach is very well suited to applications where prior knowledge of the system is minimal at the beginning of the monitoring process. It allows for the constant refinement of model parameters as more information becomes available. Bayes rule is described by Eq. (7) below, where $p(\theta)$ and its likelihood $p(y\,|\theta)$ is obtained from the data, $y = \{y_1, y_2 \cdots y_t\}$, to obtain the posterior distribution for the parameters, $p(\theta|y)$:

$$p(\theta|\boldsymbol{y}) = \frac{p(\theta)p(\boldsymbol{y}|\theta)}{\int p(\theta)p(\boldsymbol{y}|\theta)\, d\theta} \tag{7}$$

where $y_1,\ y_2 \cdots y_t$ are some surrogate measures (condition indictors) of degradation at time $1, 2, \cdots\ t$, respectively, and

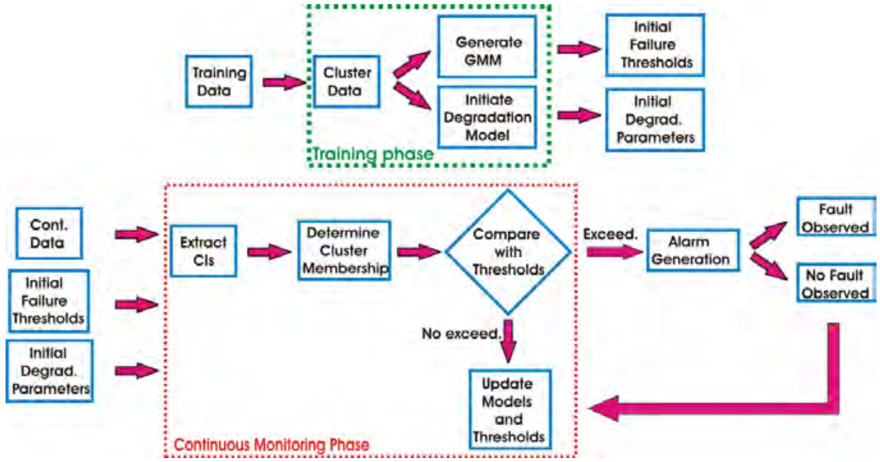

Figure 2. SPC for CBM Framework

can be assumed to be normally distributed with mean μ and variance σ^2. For the observed data y, the likelihood expression is given by:

$$f(\boldsymbol{y}|\mu,\sigma^2) = \prod_{i=1}^{n} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \ldots \atop \ldots \exp\left(-\frac{(y_i-\mu)^2}{2\,\sigma^2}\right) \right\} \tag{8}$$

2.3. Expectation-maximization

When modeling a system without sufficient prior knowledge of its characteristics, the estimation of the model's unknown parameters can be accomplished using the expectation maximization (EM) algorithm (Dempster et al., 1977). EM attempts to estimate the values of the unknown parameters $\hat{\theta}$ that maximize the log likelihood $\hat{L}$ of the observed data using an iterative two-step algorithm. First, in the expectation step, Baye's rule is applied to the dataset to determine the cluster affinity of each data point for a given estimated parameter set $\hat{\theta}^t$. A corresponding lower bound for the function $g(\hat{\theta}^t) = \hat{L}^t$ is obtained from the results of the posterior inference. Next, in the maximization step, the algorithm will attempt to find a new parameter set $\hat{\theta}^{t+1}$such that $g(\hat{\theta}^{t+1}) > g(\hat{\theta}^t)$. As the process is repeated, the log likelihood increases monotonically until it converges to a maximum.

2.4. Akaike and Bayesian information criteria

Complimentary to the EM algorithm, information criteria are used to evaluate the goodness of fit of a given statistical model relative to other models, by measuring the degree to which information is lost when that model is used to represent a set of observed data. In the context of the proposed CBM platform, the Akaike and Bayesian information criteria (AIC and BIC, respectively) are used to assess the goodness of fit for GMMs with different numbers of component mixtures. AIC and BIC are an aid in determining a model that minimizes information loss without over-fitting with excessive parameters. For a finite sample of size n, AIC is calculated using Eq. (9), where k is the number of estimated parameters, $\hat{L}$ is the maximized likelihood function of a model, $\hat{M}$, and $\hat{\theta}$ are the model parameters that maximize the likelihood function. Similarly, BIC for a finite sample n is calculated using Eq. (11) (Akaike, 1974).

$$AIC = \left(2k - 2\ln(\hat{L})\right) + \frac{2k(k+1)}{n-k-1} \tag{9}$$

$$\hat{L} = P(x\,|\hat{\theta},\hat{M}) \tag{10}$$

$$BIC = -2\ln(\hat{L}) + k\left(\ln(n) - \ln(2\pi)\right) \tag{11}$$

2.5. Failure thresholds using Bayesian updating

The accuracy of fault detection and RUL prediction algorithms depend upon the proper choice of the failure threshold(η_D). Generally, a failure threshold for a given machine is generated based on the available failure data available from other similar units. However, for long-lifetime machinery, failure data can be sparse or non-existent, making threshold setting solely relying on historical data is a challenging or an impossible task. For situations where there is a lack of historical data a Bayesian hierarchical approach can be an effective alternative for calculating failure thresholds. For this paper, a Bayesian hierarchical approach is used in conjunction with the 3-sigma rule for threshold setting and updating (Montgomery, 2009). The 3-sigma rule is common in SPC literature and has been widely applied in Gearbox monitoring (B. Liu & Makis, 2008), bearing fault detection (W. Wang & Zhang, 2008; Zhou et al., 2008) and structural health monitoring (Fugate et al., 2001).

Let the prior distribution for μ given σ^2 be normally distributed with mean μ_0 and variance σ_0^2 i.e.,$f(\mu|\sigma^2) = f_N(\mu;\mu_0,\sigma_0^2)$, then according to the Bayesian hierarchical principle, the posterior distribution of $f(\mu\,|\sigma^2, y)$ is given by:

$$f(\mu\,|\sigma^2,\boldsymbol{y}) \;\propto\; f(\boldsymbol{y}\,|\mu,\sigma^2)\, f(\mu\,|\sigma^2) \tag{12}$$

$$\propto\; f(\boldsymbol{y}|\mu,\sigma^2)\, f_N(\mu;\mu_0,\sigma_0^2)$$

$$\propto\; (2\,\pi\sigma^2)^{-\frac{n}{2}}\, exp\left(\frac{-1}{2\,\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2\right)\dots$$

$$\dots(2\,\pi\,\sigma_0^2)^{-\frac{1}{2}}\exp\left(\frac{1}{2\sigma^2}(\mu-\mu_0)^2\right)$$

$$\propto exp\left(\begin{array}{c} -\frac{1}{2}\left[\frac{n\,\sigma_0^2+\sigma^2}{\sigma^2\sigma_0^2}\right]\dots \\ \dots\left\{\mu^2-\;2\mu\left[\frac{n\,\bar{y}\sigma_0^2+\mu_0\sigma^2}{n\sigma_0^2+\sigma^2}\right]\right\}\end{array}\right)$$

The posterior described above is a normal distribution (which is conjugate to the prior) with mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$ given by:

$$E(\mu|\boldsymbol{y}) = \;\tilde{\mu}\; = \frac{(n\,\sigma_0^2\;+\sigma^2)}{(\sigma^2\sigma_0^2)}\;\; and \tag{13}$$

$$V(\mu|\boldsymbol{y}) \;=\; \tilde{\sigma}^2 = \frac{(n\bar{y}\sigma_0^2\;+\mu_0\sigma^2)}{(n\,\sigma_0^2\;+\sigma^2)}$$

The above mentioned posterior mean and variance can also be expressed as:

$$\tilde{\mu} = \mathrm{w}\bar{y} + \quad (1-w)\mu_0\;\; and\;\; \tilde{\sigma}^2 = \mathrm{w}\frac{\sigma^2}{n} \tag{14}$$

where $w = n\sigma_0^2/(n\sigma_0^2+\sigma^2\,)$. Note that the posterior mean is the weighted average of the prior mean $\mu_0$and sample mean $\bar{y}$. If the prior variance is low (and hence the prior information about μ is strong), then the posterior mean will be equal to prior mean, while if the prior variance is high then the posterior mean will be equal to sample mean. Once the $\tilde{\mu}$ and $\tilde{\sigma}^2$values are estimated, the failure threshold (η_D) can be set using statistical process control theory given by Eq. 15:

$$\eta_D = \;\tilde{\mu}\; +\; j\tilde{\sigma} \tag{15}$$

where j is a constant related to a given percentile of inverse normal distribution. For example, j is equal to 3 for 99.7 percentile. Figure 3 below illustrates an example of a failure threshold for a surrogate measure of degradation (Y) being refined over successive time intervals using Bayesian updating. Fault detection is simply performed by calculating the Euclidian distance of a data point and comparing it to the failure threshold η_D.

For a bivariate case, failure threshold setting is accomplished through Bayesian updating of μ and Σ for each cluster. Initial parameter estimation is done through sampling points from the joint posterior distribution using Markov Chain Monte Carlo sampling techniques. For subsequent updates, the posterior result from the previous cycle becomes the prior for the current update cycle. In contrast to the univariate case, the bi-variate case requires the calculation of the Mahalanobis distance – which is a measurement of the distance of a data point Y_t from the cluster mean as a function of the standard deviations and cross correlations in a multi-dimensional space. The Mahalanobis distance is given by:

$$D_M = \sqrt{(Y_t -\; \mu)^T\Sigma^{-1}(Y_t -\; \mu)} \tag{16}$$

and follows chi-square distribution. For the multivariate case the value of k in Eq.15 is set using a chi-square chart. Faults are then detected by comparing D_M to η_D and checking for exceedance.

2.6. Degradation modeling and RUL estimation

While real-time fault detection can be an effective means at detecting faults early on in their development phase, it does not provide a long term forecast of the machine's health. Degradation modeling is a useful prognostic tool for making long-term predictions regarding the health of the machine. The degradation process can either be modeled through the physics of the system (model-based) or through derived parameters obtained through sensor measurements (data-driven). Furthermore, data-driven degradation models can be implemented alongside data-driven fault detection techniques since they could potentially utilize the same underlying dataset.

In general, a degradation path of a monitored unit can be convex, concave, or linear in shape as shown in Figure 4. The selection of the appropriate shape for a given application is guided by field data, engineering judgment and some understanding of the mechanical laws that describe a system's performance. The most suitable model for the degradation path depends on the application, and ranges from simple regression type models (e.g., linear, exponential, power law, logistic and Gompertz) to more complex stochastic models (e.g., Gamma, Wiener and Markov process) (Lu & Meeker, 1993; Meeker & Escobar, 2014; van Noortwijk, 2009; Whitmore & Schenkelberg, 1997). The regression model with power law is most appropriate for gearbox degradation where the shape of degradation path is not known *a priori*. This is because, a wide variety of degradation paths (including convex, concave and linear) can be generated by varying the exponent of power law, as also illustrated in Figure 4.This functional form has been applied widely; for e.g., degradation modeling of concrete due to corrosion of reinforcement (Ellingwood & Yasuhiro, 1993), reliability analysis of hydraulic systems (Kumar & Klefsjo, 1992) and bearing health (Ali et al., 2014).

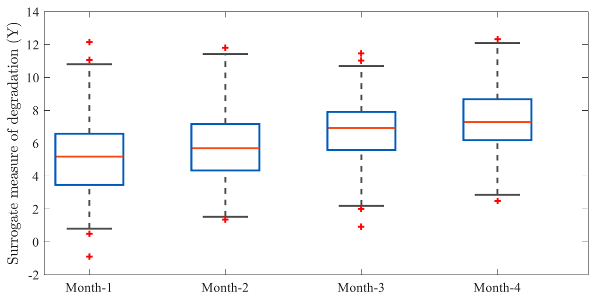

Figure 3. Bayesian threshold updating of surrogate measure of degradation (Y)

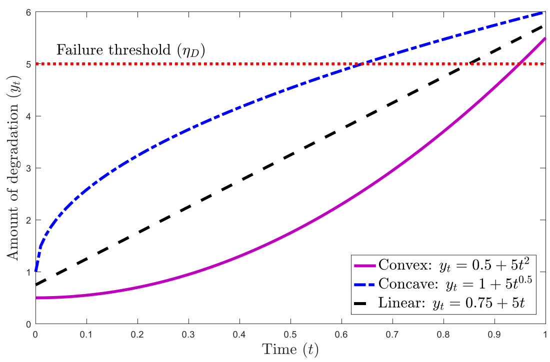

Figure 4. Various Shapes for Degradation Paths

Let the Y_{ij} be the observed surrogate measure of degradation for the i^{th} bearing at time t_{ij}. Then, the general degradation model is given as:

$$Y_{ij} = h(t_{ij}, \theta_i, \phi) + \epsilon_{ij} \tag{17}$$

where $h(\cdot)$ is some function of t, θ_i and ϕ are the vector of random and fixed effect respectively. The measurement errors ϵ_{ij} are assumed to be additive, conditionally independent and distributed as $N(0, \sigma_\epsilon^2)$. With the following parameters $\theta = (\theta_1, \theta_2)$, $\phi = \phi_0$ and the power functional form (Kumar & Klefsjo, 1992; Sánchez-Silva et al., 2016; van Noortwijk, 2009) for $h(\cdot)$ in Eq. 17, the degradation model can be written as:

$$Y_j = \phi_0 + \theta_1 \theta_2 t_j^{\theta_2} + \epsilon_j \tag{18}$$

Furthermore, it is assumed that $\theta = (\theta_1, \theta_2)$ follow a multivariate normal $(MVN(\mu_{\theta,}\Sigma_\theta))$ distribution. A Bayesian approach is employed to estimate and update the model parameters ($\Theta = \mu_{\theta,}\Sigma_\theta, \sigma_\epsilon^2$) as more vibration data becomes available. For each GMM cluster, the Mahalanobis distance of each feature pair is calculated and the marginal posterior density of parameters is estimated by taking samples from a joint posterior distribution through MCMC. The failure thresholds are also updated for each cluster and the RUL is predicted at each update interval. Total system failure occurs when the surrogate degradation measure reaches the predefined threshold η_D and the RUL is the measure of the time between the current time and the predicted failure time.

3. IMPLEMENTATION OF CBM FOR THE APM GEARBOX

The CBM test setup, termed "two-tier framework", in this study has been designed as an auto-updating platform that is suitable for performing CBM on the APM gearbox. It is comprised of two distinct workstations: a CBM client and CBM server. The CBM client is installed on-site in the immediate proximity of the gearbox (Figure 1(b) and Figure 6) and the CBM server could be placed anywhere with a network access (in our case, this was placed on site). Figure 5 shows the overview of the two-tier framework components and how various components are interconnected for data sharing.

3.1. CBM client and CBM server

An embedded real-time controller with an onboard 667 MHz dual-core processor and a user-reconfigurable field programmable gate array (FPGA) both residing on a Compact RIO (or cRIO) system are the key components of the CBM client workstation (see Figure 6 (a)).

Compact RIO is a real-time embedded automation and data acquisition platform made by National Instruments. This system combines the reliability of FPGA technology with the high-speed computational capabilities of an embedded real-time processor and is suitable for applications that require high performance and reliability. The FPGA is a reconfigurable hardware chip that contains logic blocks, programmable interconnections and input/output (or I/O) blocks (see Figure 6 (b)). If programmed properly, FPGA can execute predefined tasks at very fast and deterministic rates without a need for CPU resources.

In this application, the FPGA is the communication hub of the CBM client with the sensing hardware and the alarm system. It uses NI9234 and NI9263 C-series modules to acquire the vibration data from the sensors and issues command signals to the alarms, respectively. NI 9234, the accelerometer module, is a 4-channel, 24 bit analog input module that reads the accelerometer output voltage at up to 50 kS/s (kilo sample per second) sampling rate per channel and then performs the analog to digital conversion internally ("NI 9234 User Guide and Specifications," 2014). NI 9263, is a high performance 4-channel +/- 10 VDC analog I/O module that is used to handle the alarm system within the CBM implementation structure ("NI 9263 User Guide and Specifications," 2014).In addition, a network of PCB piezoelectric accelerometers were utilized (described in the results section) to acquire vibration data for monitoring. The accelerometer arrangement and specifications can vary depending on the application.

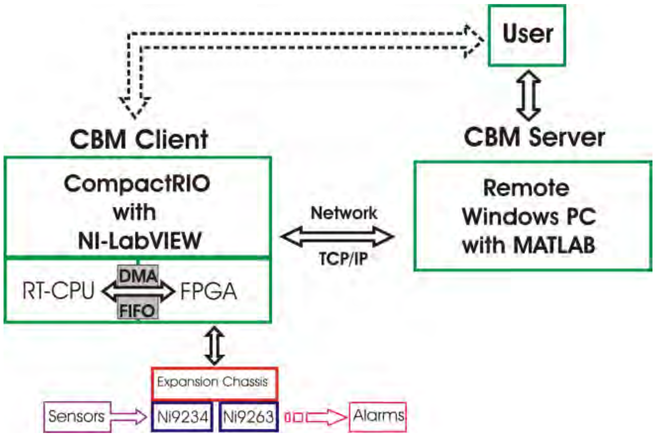

Figure 5. Overview of the CBM test Setup

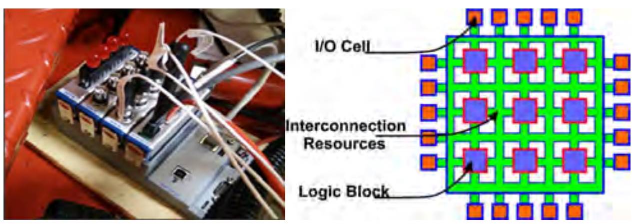

Figure 6. a) cRIO (left), b) FPGA chip schematic (right)

The CBM server is a windows based computer with a generic programming toolbox such as MATLAB and Python. It serves as the off-site computational workstation of the setup and is designed to handle the computationally intensive CBM tasks, such as GMM and degradation model updating that could not be loaded on the CBM client. The CBM server is interconnected with the CBM client via network using TCP/IP protocol.

3.2. Task scheduling

The CBM client accomplishes the following tasks continuously in near real-time: vibration data collection from accelerometers, feature calculations (condition indicator extraction), fault detection by comparing the extracted features to the thresholds received from the CBM server, alarm generation in the event of threshold exceedance, and curating of a text file containing information regarding a fault if one is detected by the system (i.e. time, date and location). In addition, the CBM client streams collected data and feature sets to the CBM server through the network connection.

The tasks handled by the CBM server include feature selection, feature clustering, degradation modeling and RUL estimation. In general, the CBM server routine is conducted in two distinct phases: an initial training phase that occurs only at the onset of the project, and a continuous monitoring phase. During the training phase, the CBM server routine executes a training algorithm where it calls the client to perform frequent sampling and feature calculation for an extended period of time. This training phase is used to develop a representative dataset that encompasses all of the system's operating states. For each feature or pair of features within the initial dataset, the optimal clustering configuration for a mixture of Gaussians is calculated using the Akaike or Bayesian Information Criterions. For each feature or feature pair, initial Bayesian thresholds are calculated, packaged and sent back to the client. The optimal feature set can be determined objectively using a method such as Hart Decision trees, or subjectively by selecting a feature clustering scheme that corresponds to some physical phenomena in the system. The CBM platform in this study is capable of monitoring and setting thresholds for a number of feature sets simultaneously. In the continuous monitoring phase, the CBM server uses the vibration and feature data streamed from the CBM client to regularly update the GMM, Bayesian thresholds and degradation models for RUL estimation. Once the model updating is completed the new failure thresholds are sent back to the CBM client. Figure 7 summarizes the task scheduling in the test setup.

3.3. Software development

Unlike turnkey systems, the computational/data acquisition platform that is used in this application must be programmed by the user to run all the designated tasks. The CBM client software is developed using NI LabVIEW. It consists of an FPGA VI, a Host VI (also called as real-time VI), and several sub VIs (equivalent to sub functions in MATLAB), all coordinated by a LabVIEW project.

The Host VI (hereinafter referred to as "real-time VI") is planned and programmed based on state machine design. The state machine is one of the most effective tools available in LabVIEW to handle applications with distinguishable states. State machines enable complicated decisions, summarized in a state flow diagram to be implemented. In a majority of applications with the state machine architecture, the process starts with an initialization state and ends with a stop or shut down state that clears all the actions undertaken by the system in the previous states. While the system is running, based on user inputs or in-state calculations, the state machine determines which state to go next.

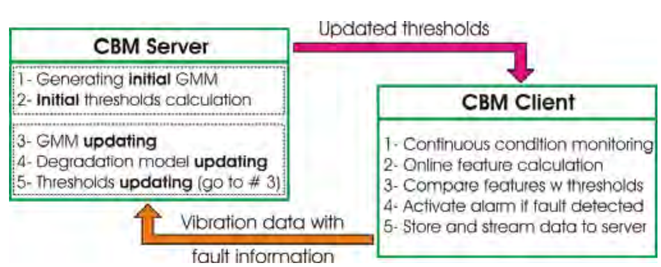

Figure 7. Summary of the tasks executed the CBM client and server

Figure 8 shows the state flow diagram of the host VI that was designed for the CBM client. The current version of the CBM client includes five operational states with distinguishable tasks. The CBM client begins with the "initialization" state that starts/ resets the FPGA target and writes all the user defined information such as sensor calibration data and data sampling rate to the associated constants in the software. This state is followed by the "start-up" state where the initial GMM input file that has been sent by the CBM server is read and the containing thresholds are written to the cRIO memory to be used throughout continuous monitoring. In addition, the DMA FIFO protocol is initiated for transferring the raw vibration data between the FPGA target and the Host VI. Next, the program moves into the "monitoring" state. In this state the sensor data is continuously acquired, calibrated and monitored. To minimize computational effort, the transition to the writing state is only triggered when a prescribed threshold for vibration is exceeded. The following three tasks are carried out concurrently in the "writing state":

1. The raw vibration data from sensors that is acquired through FPGA is passed to the host VI using FIFO protocol and then stored in the daily raw output files.
2. The statistical feature data is computed based on the raw data and stored in the daily feature data.
3. Posterior inference using Bayes rule is employed to determine the cluster membership of the incoming data. Once the correct data cluster is determined, the data is checked against the associated failure thresholds through calculation of Euclidian distance (1D) or Mahalanobis distance (2D). If an exceedance is detected, the user is notified via a LED indicator attached to the CBM client hardware and the corresponding time stamp, sensor name, and feature values are sent to the CBM server. If the user performs maintenance after a potential fault is detected and determines that there is no fault present, the CBM server will update the GMM model and thresholds accordingly.

In the current version of the CBM platform the duration of the "writing" state is set to 9 sec. Due to the storage limitations it is not feasible to have vibration data continuously written onto the output file. After completion of the "write " state, the state machine enters the "wait" state where the system goes on hold for one hour. Having the "wait state" enables the user to acquire/write vibration data at different times of day and night, hence capturing the system performance under different load conditions. It should be pointed out that during the "wait state" the condition indicators are still computed and monitored but nothing is written to the output files. Once the waiting time is over the state machine goes back to the "monitoring" state and the process repeats from there.

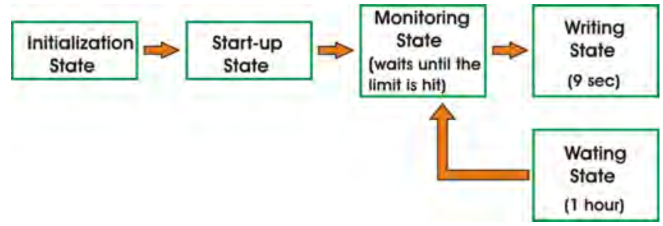

Figure 8. State flow diagram of the Host VI for CBM client implementation

The FPGA VI is programmed to handle the interfacing tasks with the hardware (e.g. vibration sensors and limit switches) and FIFO DMA protocol for transferring raw data from FPGA to the host VI. The CBM client is configured as an unsupervised computational/DAQ platform. Once the developed software is fully deployed onto the cRIO and the system is energized, it can perform all the required tasks automatically and unattended. Hence, the current implementation can be accurately described as a semi-autonomous one, rather than a fully autonomous implementation.

4. CASE STUDY AND RESULTS

4.1. Case study description

A field pilot was conducted on the APM gearbox described earlier to validate the developed CBM strategy, during late 2016 and early 2017. For this particular case study, several decisions in the CBM server-side routine involved expert intervention. It is the authors' goal in the future to develop and refine the criteria necessary to automate these decisions as well. This APM system consists of a cable-driven, computer-controlled train that travels along a 1.5 km track over variable grade and connects three passenger stations (see Figure 9). The train travels in both directions along the same track, with a capacity of 2,500 pphpd (people per hour per direction).

The APM gearbox was instrumented with four uniaxial PCB accelerometers (model #: 352C68) with sensitivity of 100 mV/g, linearity between 0.5 Hz to 10 kHz and an output range of +/-50g, mounted radially on the gearbox housing with respect to the bearings, as depicted in Figure 1(b)). Data acquisition and on-site computation is handled using the CBM client platform. During the initial training period, training data for the APM was collected over eight discrete 2-hour long sampling periods taken throughout the course of one month. The sampling periods were scheduled at different times of the day and week to encapsulate any fluctuations in load due to different passenger volumes. During each sampling period, the train acceleration was sampled at 1000 Hz (the authors recognize that a much higher sampling rate may be needed for bearing fault isolation, but this is not addresses in this paper) and the precise location of the train was recorded. Onboard feature calculation occurred at one-second intervals.

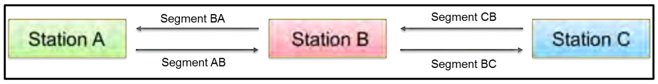

Figure 9. APM train and stations configuration

4.2. Data analysis

The following time-domain condition indicators were calculated from the vibration data obtained from each accelerometer: root-mean square, variance, kurtosis, crest-factor and hyper-kurtosis (6th normalized moment). A preliminary study was conducted to verify a normality assumption and the probability paper plot revealed that the dataset was left-skewed due to significant zero entries as a result of periodic stops of the train. Hence, prior to clustering, the zeros were manually removed, resulting in a uniformly distributed dataset. The removal of zeros from the dataset is justified due to the fact that the periods of interest to this CBM implementation is where the system is under load and in motion. Furthermore, the data obtained during continuous monitoring will only contain non-zero entries due to the triggered sampling algorithm implemented.

The discussion presented hereafter focuses on the data from a single accelerometer (i.e. A_1 as shown on Figure 1(b)) and utilizes the feature pair subset of hyper-kurtosis and crest factor, as the clustering for this feature pair was observed to be consistent with the position/direction of the train. Figure 10 shows the GMM for the feature data for three clusters, where the number of clusters was obtained by using the minimum AIC or BIC value.

A study was conducted to determine whether the clustering was related to any physical system characteristics. The APM feature data was sorted by train location and plotted in Figure 11 and Figure 12. Comparison of Figure 10 and Figure 11 shows that clusters 1 and 2 correspond to different directions of travel for the train, while cluster 3 corresponds to scatter, containing points from all track segments. The study illustrates a major advantage of using a state-based CBM approach: separating the behavior of a system into its different operational states not only improves the accuracy of SPC-based fault detection, but can also grant the user additional insight into how and where faults begin to develop within a system.

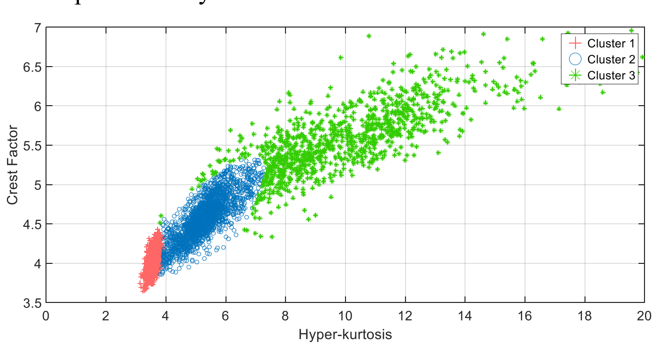

Figure 10. GMM clusters (n=3), hyper kurtosis vs. CF

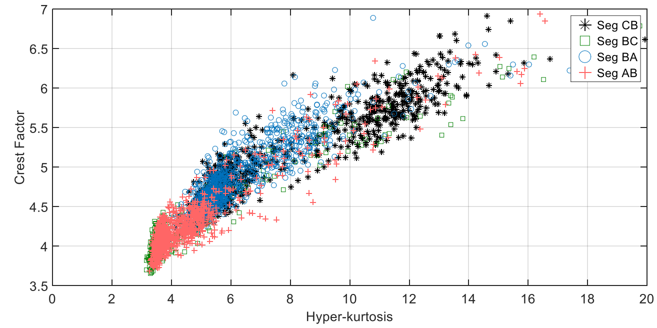

Figure 11. Feature points based on train location, hyper kurtosis vs. crest factor

Two-dimensional Bayesian thresholds were calculated for each of the clusters found in the hyper-kurtosis and crest factor feature pair at three time instances. First, the initial thresholds were set using the training dataset collected throughout the first month, designated by time t_1. Next, the thresholds were updated bi-weekly at times t_2 and t_3. The Bayesian thresholds are shown in Table 1.

Table 1: APM Posterior Parameters at Various Times

t1 = End of Month 1 (end of training period)							
	Mean	sd	2.5%	25%	50%	75%	97.50%
μ_1	4.64	0.06	4.54	4.60	4.64	4.68	4.75
μ_2	4.38	0.02	4.35	4.37	4.38	4.40	4.42
σ_{11}	0.90	0.08	0.77	0.85	0.90	0.95	1.06
σ_{12}	0.30	0.03	0.25	0.28	0.30	0.31	0.35
σ_{22}	0.12	0.01	0.10	0.11	0.12	0.13	0.14
t2 = Middle of Month 2							
μ_1	4.59	0.05	4.50	4.56	4.59	4.62	4.68
μ_2	4.36	0.02	4.32	4.35	4.36	4.37	4.39
σ_{11}	0.89	0.06	0.78	0.85	0.89	0.93	1.02
σ_{12}	0.29	0.02	0.25	0.28	0.29	0.31	0.34
σ_{22}	0.12	0.01	0.10	0.11	0.12	0.12	0.14
t3 = End of Month 2							
μ_1	4.62	0.04	4.54	4.59	4.62	4.64	4.69
μ_2	4.36	0.01	4.33	4.35	4.36	4.37	4.39
σ_{11}	0.91	0.05	0.81	0.87	0.91	0.94	1.02
σ_{12}	0.30	0.02	0.26	0.28	0.29	0.31	0.33
σ_{22}	0.12	0.01	0.11	0.11	0.12	0.12	0.13

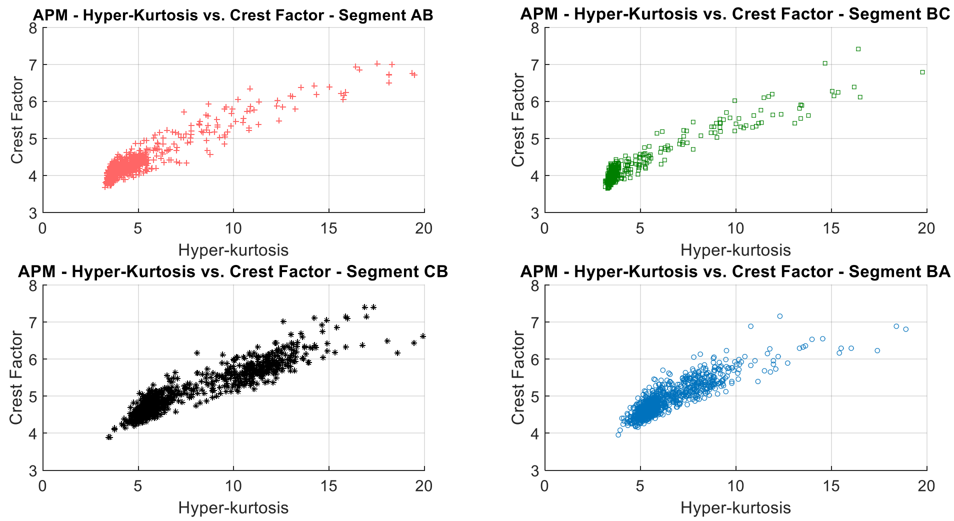

Figure 12. 2-dimensional feature points separated based on train location and direction, hyper kurtosis vs. crest factor

Table 1 shows very little fluctuation in the **μ** and **σ** values throughout the month-long period. The lack of change in the threshold parameters can be attributed to a number of factors: during the initial period, there is a bias towards the training data set since the size of the training sample is much larger than those from subsequent updating intervals (nt1 >> nt2, nt3). Secondly, the length of time considered for the study is small in comparison to the lifespan of the system. It is anticipated that a much longer monitoring period would be required to detect any changes in the threshold parameters due to degradation, which is planned for through a long-term monitoring program currently in place at this facility.

4.3. Degradation modeling and reliability predictions

As discussed in the methodology section, a regression model with the power law (see Eq. 5) is suitable for modelling the degradation path of the gearbox. The key steps in building the degradation model are to calculate the Mahalanobis distance for each feature pair (i.e., hyper-kurtosis, crest-factor) in a given GMM cluster from the corresponding threshold (i.e.,μ, Σ assigned in the previous step), and to estimate the degradation model parameters.

Two months of data from initial data collection period were used for this purpose and the aforementioned parameters were estimated to be $\mu_{\theta 1} = 5.7, \mu_{\theta 2} = 3.5, \Sigma_{11} = 1.7, \Sigma_{12} = 0.15, \Sigma_{22} = 1.35$. Note that these are the degradation model parameters and not the mean and covariance matrix estimated for GMM clusters presented previously. Next, since run-to-failure data is not available for this system, the initial parameters are used to predict the future Mahalanobis distances and degradation signal. Figure 13 (left panel) shows three simulated degradation paths using the initial parameters, at the end of 3, 5 and 6 years, respectively. Bayesian inference was then performed with the following diffuse prior distributions

$$\mu_\theta = \begin{bmatrix} \mu_{\theta 1} \\ \mu_{\theta 2} \end{bmatrix}$$

$$\sim \text{Multivriate Normal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}\right)$$

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} \sim \text{Inverse Wishart}\left(\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, 2\right)$$

where $\sigma_\epsilon^2 \sim \text{Inverse gamma}\,(5, 0.0001)$ and the parameters were estimated and updated by taking random samples from the joint posterior distribution of $(\mu_\theta, \Sigma_\theta, \sigma_\epsilon^2)$ using MCMC sampling. The right half of Figure 13 shows the updated posterior distribution of θ at the end of 3, 5 and 6 year respectively. Note that, over time (commensurate with higher degradation), the posterior distribution becomes narrower and the parameters converge to their true values.

Finally, the reliability at the end of 3, 5 and 6 year is predicted using Monte Carlo simulation, as presented in Figure 14. Upon closer inspection, Figure 14 reveals that the reliability remains approximately equal to one up until the sixth year and then decreases abruptly beyond that for all the three paths. Moreover, one can see the improvement in reliability predictions when more data is utilized in the analysis i.e., at the later stages of the degradation. For example, when the reliability is assessed by utilizing data up to 3 years, the model predicts a failure by the end of 7.6 years with 80 percent probability. However, when the similar analysis was performed with 6 years of available data, it predicts failure at the age of 6.9 years. In other words, the 80 percent RUL CDF estimated using 3, 5 and 6 years of data are 4.6, 2.4 and 0.9 years, respectively. Evidently, as more degradation data becomes available, the RUL prediction becomes more accurate and increasingly valuable for maintenance planning. Finally, it should be noted this model is only for illustration, and the simulated degradation paths will be updated over time as live data from the gearbox becomes available.

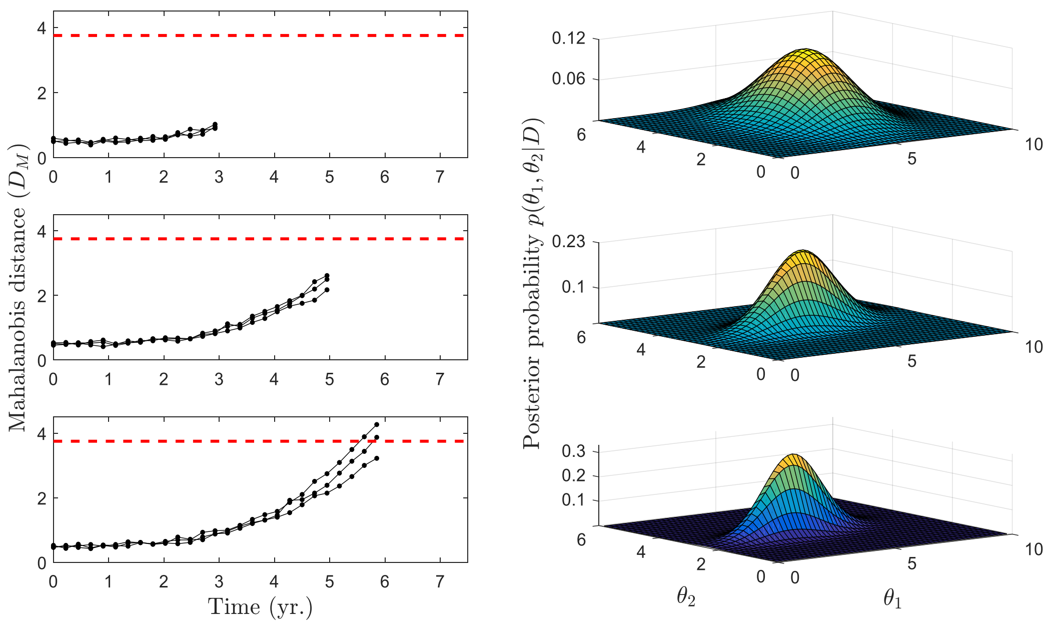

Figure 13. a) Degradation signal until 3, 5 and 6 yr., b) updated posterior distribution

5. Conclusion

A data-driven automated two-tier CBM frame work for the gearbox in a cable-driven APM was presented in this paper. APM gearboxes typically present these main challenges for CBM: they are low-speed, highly non-stationary and complex systems with multi-path convolved vibration sources and often accompanied with scarce availability of historical failure data. The CBM framework presented is able to address many of these challenges through the use of GMMs and degradation models, which are updated using a hierarchical Bayesian approach. The generated GMM clusters were able to decompose the behavior of the system based on the direction of travel of the train, which adds to information contained in each of these clusters. Bayesian thresholds were derived for each cluster to perform SPC-based fault detection, which makes this approach more robust than using a single set of thresholds. A regression model with power law was used to model the degradation and was shown to be effective when prior information regarding the nature of degradation is not available. Bayesian hierarchical updating is used to refine the model parameters periodically as new data becomes available, which fits within the long-term monitoring goals of this project. Results from this pilot show that the CBM framework is able to address many of the aforementioned challenges, which present themselves in low-speed, long-life components.

The framework can be further improved upon by introducing more sophisticated signal processing techniques to isolate and monitor specific components within the gearbox; work on this is ongoing, but is considered outside the scope of the current paper. Future work on the CBM platform is largely focused developing protocols to monitor the health of individual critical components within the APM gearbox. Signal separation and filtering techniques will be used to de-convolve the gearbox signals into its constituent components. Feature sets from frequency domain and time-frequency domain will be investigated. The feasibility of contemporary fault detection and diagnostic techniques will be investigated in the context of low-speed rotating components to address the gaps in literature. Alternative feature space reduction tools such as linear discriminant analysis and principle component analysis will also be considered.

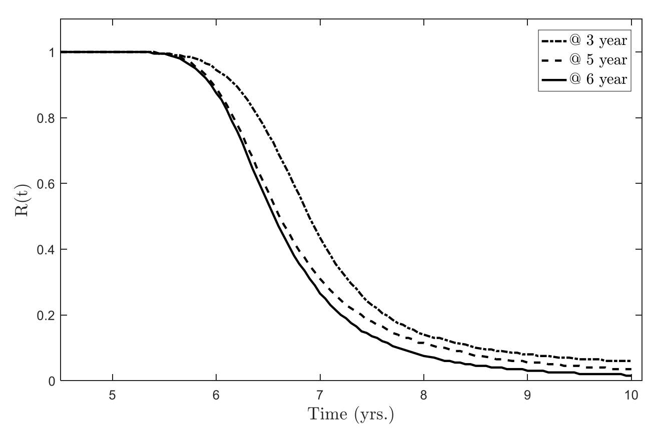

Figure 14. Reliability predictions for the gearbox at the end of 3, 5 and 6 years

ACKNOWLEDGMENTS

Funding provided by The Natural Sciences Engineering Research Council of Canada, Greater Toronto Airports Authority and Doppelmayr Cable Car Ltd (DCC). is gratefully acknowledged. We would also like to thank Sean Best, Dr. Ignacio Sesma and Johannes Winter from DCC Ltd. for their technical assistance during the data collection process and during the preparation of this manuscript.

NOMENCLATURE

CBM	condition based monitoring
APM	automated people mover
RUL	remaining useful life
MCMC	Markov chain Monte Carlo
SPC	statistical process control
GMM	Gaussian mixture model
AIC	Akaike information criterion
BIC	Bayesian information criterion
EM	expectation maximization
cRIO	compact RIO
FPGA	field programmable gate array
DAQ	data acquisition
p	probability density function
K	number of mixture components
μ_i	mean
σ_i	standard deviation
ϕ_i	weight
θ	process parameters
y_i	condition indicators
$\hat{\theta}$	unknown process/model parameter
$\hat{M}$	model with unknown process parameters θ
$\hat{L}$	log likelihood function
k	number of estimated parameters
j	constant for percentile of inverse normal distribution
η_D	failure threshold
μ_0, σ_0^2	prior mean and variance
$\tilde{\mu}, \tilde{\sigma}^2$	posterior mean and variance
$\bar{y}$	sample mean
D_M	Mahalanobis distance
Y	observed surrogate measure of degradation
t_{ij}	time
ϵ_{ij}	measurement errors

REFERENCES

Akaike, H. (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. http://doi.org/10.1109/TAC.1974.1100705

Ali, J. Ben, Chebel-morello, B., Saidi, L., & Malinowski, S. (2014). Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network. *Mechanical Systems and Signal Processing*, 1–23. http://doi.org/10.1016/j.ymssp.2014.10.014

Baum, M., Gheţa, I., Belkin, A., Beyerer, J., & Hanebeck, U. D. (2017). Data association in a world model for autonomous systems. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (pp. 187–192). Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-78649315410&partnerID=40&md5=b64018bbc8d668cea4afa18356d89a75

Bechhoefer, E., Schlanbusch, R., & Waag, T. I. (2016). Techniques for Large, Slow Bearing Fault Detection. *International Journal of Prognostics and Health Management*, 2153–2648.

Boutros, T., & Liang, M. (2011). Detection and diagnosis of bearing and cutting tool faults using hidden Markov models. *Mechanical Systems and Signal Processing*, *25*(6), 2102–2124. http://doi.org/10.1016/j.ymssp.2011.01.013

Bunks, C., & Mccarthy, D. (2000). Condition-Based Maintenance of Machines Using Hidden Markov Models. *Mechanical Systems and Signal Processing*, *14*(4), 597–612. http://doi.org/10.1006

Chen, N., & Tsui, K. L. (2013). Condition monitoring and remaining useful life prediction using degradation signals: revisited. *IIE Transactions*, *45*(9), 939–952. http://doi.org/10.1080/0740817X.2012.706376

Chin, S. C., Ray, A., & Rajagopalan, V. (2005). Symbolic time series analysis for anomaly detection: A comparative evaluation. *Signal Processing*, *85*(9), 1859–1868. http://doi.org/10.1016/j.sigpro.2005.03.014

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). *Maximum likelihood from incomplete data via the EM algorithm*. *J R Stat Soc Series B Stat Methodol* (Vol. 39).

Doppelmayr Cable Car. (2017). Cable-propelled automated people mover systems. Retrieved from https://www.dcc.at/solutions/system-features/

Ellingwood, B., & Yasuhiro, M. (1993). Probabilistic methods for condition assessment and life prediction of concrete structures in nuclear power plants. *Nuclear Engineering and Design*, *142*, 155–166. http://doi.org/10.1016/0029-5493(93)90199-J

Fugate, Michael L and Sohn, Hoon and Farrar, C. R. (2001). Vibration-based damage detection using statistical process control. *Mechanical Systems and Signal Processing*, *15*, 707–721.

Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. http://doi.org/10.1016/j.ymssp.2005.09.012

Kaiser, K. A., & Gebraeel, N. Z. (2009). Predictive maintenance management using sensor-based degradation models. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, *39*(4), 840–849. http://doi.org/10.1109/TSMCA.2009.2016429

Kumar, U., & Klefsjo, B. (1992). Reliability analysis of hydraulic systems of LHD machines using the power law process model. *Reliability Engineering & System Safety*, *35*, 217–224. http://doi.org/10.1016/0951-8320(92)90080-5

Lee, M. D., Houssein, M. A. E., & Shahidul, M. I. (2016). Production Machinery Maintenance Cost Optimization : A Review. *International Journal of Advanced Engineering Research and Application*, *2*(3), 131–146.

Liu, B., & Makis, V. (2008). Gearbox failure diagnosis based on vector autoregressive modelling of vibration data and dynamic principal component analysis. *IMA Journal of Management Mathematics*, *19*(1), 39–50. http://doi.org/10.1093/imaman/dpm002

Liu, Q., Dong, M., & Peng, Y. (2012). A novel method for online health prognosis of equipment based on hidden semi-Markov model using sequential Monte Carlo methods. *Mechanical Systems and Signal Processing*, *32*, 331–348. http://doi.org/10.1016/j.ymssp.2012.05.004

Lu, C. J., & Meeker, W. Q. (1993). Using Degradation Measures to Estimate a Time-to-Failure Distribution. *Technometrics*, *35*(2), 161. http://doi.org/10.2307/1269661

Meeker, W. Q., & Escobar, A. L. (2014). *Statistical Methods for Reliability Data*. John Wiley & Sons, Inc.

Montgomery, D. (2009). *Introduction to statistical quality control*. *John Wiley & Sons Inc*. http://doi.org/10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C

Nelwamondo, F., Marwala, T., & Mahola, U. (2006). Early Classifications of Bearing Faults Using Hidden. *Information and Control*, *2*(6), 1281–1299.

NI 9234 User Guide and Specifications. (2014). Retrieved from http://www.ni.com/datasheet/pdf/en/ds-316

NI 9263 User Guide and Specifications. (2014), (866), 1–10. Retrieved from http://www.ni.com/datasheet/pdf/en/ds-59

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. http://doi.org/10.1109/5.18626

Randall, R. B. (2012). *Vibration-based Condition Monitoring*. John Wiley & Sons, Inc.

Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics-A tutorial. *Mechanical Systems and Signal Processing*, *25*(2), 485–520. http://doi.org/10.1016/j.ymssp.2010.07.017

Sadhu, A., Prakash, G., & Narasimhan, S. (2016). A hybrid hidden Markov model towards fault detection of rotating components. *Journal of Vibration and Control*, (August 2015). http://doi.org/10.1177/1077546315627934

Sánchez-Silva, Mauricio and Klutke, G.-A. (2016). Degradation: Data Analysis and Analytical Modeling. In *Reliability and Life-Cycle Analysis of Deteriorating Systems* (pp. 79--116). Springer.

Sharma, A., Amarnath, M., & Kankar, P. (2014). Feature extraction and fault severity classification in ball bearings. *Journal of Vibration and Control*, (February), 1–17. http://doi.org/10.1177/1077546314528021

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation - A review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14. http://doi.org/10.1016/j.ejor.2010.11.018

Timusk, M., Lipsett, M., & Mechefske, C. K. (2008). Fault detection using transient machine signals. *Mechanical Systems and Signal Processing*, *22*(7), 1724–1749. http://doi.org/10.1016/j.ymssp.2008.01.013

van Noortwijk, J. M. (2009). A survey of the application of gamma processes in maintenance. *Reliability Engineering and System Safety*, *94*(1), 2–21. http://doi.org/10.1016/j.ress.2007.03.019

Večeř, P., Kreidl, M., & Šmíd, R. (2005). Condition Indicators for Gearbox Condition Monitoring Systems. *Acta Polytechnica*, *45*(6), 35–43. http://doi.org/10.14311/782

Wang, W., & Zhang, W. (2008). Early defect identification: application of statistical process control methods. *Journal of Quality in Maintenance Engineering*, *14*(3), 225–236. http://doi.org/10.1108/13552510810899445

Wang, X., Jiang, P., Guo, B., & Cheng, Z. (2014). Real-time reliability evaluation with a general Wiener process-based degradation model. *Quality and Reliability Engineering International*, *30*(2), 205–220. http://doi.org/10.1002/qre.1489

Whitmore, G. a, & Schenkelberg, F. (1997). Modelling accelerated degradation data using Wiener diffusion with a time scale transformation. *Lifetime Data Analysis*, *3*(1), 27–45. http://doi.org/10.1023/A:1009664101413

Yin, S., Wang, G., & Karimi, H. R. (2014). Data-driven design of robust fault detection system for wind turbines. *Mechatronics*, *24*(4), 298–306. http://doi.org/10.1016/j.mechatronics.2013.11.009

Zhou, W., Habetler, T. G., & Harley, R. G. (2008). Bearing fault detection via stator current noise cancellation and statistical control. *IEEE Transactions on Industrial Electronics*, *55*(12), 4260–4269. http://doi.org/10.1109/TIE.2008.2005018