Course project: Residual life prediction by modelling device degradation signals

Jing Pan, Shan Li, Xuemiao Hao, Xiongwen Tang

December 4, 2006

Abstract

Condition monitoring is popular in industrial maintenance dicision-making. For example people record the degradation signals of a functioning device to check the health of the device and try to predict its residual life time. In our work, we investigate the degradation signals of 51 bearings. These signals can be used to represent the goodness of bearings. We apply Bayesian approach to model these signals under different kinds of assumptions. According to the posterior distribution of residual life time we can give an estimated residual life for a particular bearing.

1 Introduction

In [1], Gebraeel et al. use a Bayesian approach to develop a parameterized model of the degradation signal of a population devices. To do this, they combine the distribution of the parameters across the device population and sencored degradation information collected by condition monitoring for a particular device to obtain the updated information about the parameters. Using the updated parameters they can determine the residual-life distribution for this device. They investigate 2 two-parameter exponential models with parameters β and σ^2. In one model the error terms are suppose with independently identical $N(0, \sigma^2)$ with σ^2 fixed. While in the other model, the error terms are supposed as multiplicative Brownian motion with the volatility σ also fixed.

Our project is based on Gebraeel et al.'s work. We have a vibration-based degradation signal for each of 51 bearings. As the bearings degrade the vibration they exhibit tends to increase. When the vibration reaches a standard level the bearing is considered to have failed. It's common to consider these bearings have an identical degradation signal form. Moreover we suppose they share the same prior distribution of parameters.

We build both linear and exponential models for degradation signal. In each case we first consider error terms as *iid* $N(0, \sigma^2)$ and then multiplicative Brownian motion. So totally 4 models are investigated in our project. Actually as shown in Section 2.1 exponential model can be transformed into linear model. Further, since Brownian motion has independent normal distributed increments we can also transform Brownian motion error process into *iid* normal error terms.(see Section 2.2)

When we apply Bayesian method we use the degradation signal of 25 bearings to estimate the prior distribution of β, which is identical for each bearing. We suppose noninformative prior for σ^2. To get the posterior distribution of the residual-life for a bearing, we assume σ^2 as inverse gamma distributed and β as multivariate normal distributed conditioning on σ^2. From the $3rd$ step in the Bayesian analysis these are conjugate distributions which we show in Section 2.2.

In Section 2.3 we give the residual-life distribution. We want to point out that our results are a bit different from those in [1]. The reason is that we let the variance parameter σ^2 be random instead of fixed. This can be regarded as an improvement for [1].

We use R programming to help handle data. In Section 3.3 we give a point estimate for a paticular bearing knowing its sensored degradation information till time t. The estimator is the median of the most updated residual-life distribution. After that we compare the estimated residual life with the real residual life.

2 Component Degradation Signal Modeling

In this chapter, we will construct the degradation model first, and then study the posterior Residual Life Distribution by using Bayesian's theory.

2.1 The Degradation signal Model

Model Assumption 1: Exponential Degradation Signal Model

Let S_t denote the degradation signal at time t. It can be modeled as follows:

$$S_t = \phi + \theta \exp(\beta t + \epsilon(t) - \frac{\sigma^2 t}{2}) \tag{1}$$

where ϕ is a known constant, θ and β are two unknown parameters, and $\epsilon(t)$is an error term.

Some assumptions are applied here:

(1) The prior distribution of θ is

$$\pi(\theta) = Lognormal(\mu_0, \sigma_0^2) \tag{2}$$

The prior distribution of β follows

$$\pi(\beta) = N(\mu_1, \sigma_1^2) \tag{3}$$

(2) For the error term $\epsilon(t)$, which is independent of θ and β, we will consider two different assumptions respectively in the following discussion: IID Error and BM Error.

Error Assumption 1: Independent and Indentically Distribution (IID)

Firstly, we assume $\epsilon(t)$'s are independent with respect to t. That is $\epsilon(t_i)$ is independent of $\epsilon(t_j)$ where t_i and t_j are arbitrarily two different time points. $\epsilon(t)$ follows a normal distribution as

$$\epsilon(t) \sim N(0, \sigma^2) \tag{4}$$

Error Assumption 2: Brownian Motion (BM)

Secondly, we consider $\epsilon(t) = \sigma W(t)$ is proportional to a centered Brownian motion such that the mean of $\epsilon(t)$ is zero and variance of $\epsilon(t)$ is $\sigma^2 t$.

(3) σ^2 is also considered as a variable with a prior distribution

$$\pi(\sigma) = IG(a, b) \tag{5}$$

Let $L_t \equiv \ln(S_t - \phi)$, The above $Exponential Model$ can be transformed to a linear model as

$$L_t = \phi' + \beta' t + \epsilon(t) \tag{6}$$

where $\phi' \equiv \ln \phi$ and $\beta' \equiv \beta - \frac{\sigma^2 t}{2}$.

To integrate and simplify the model, in this exponential degradation model case, we let $y_t = L_t$, for all t.

Model Assumption 2: Linear Degradation Signal Model

Besides the Exponential Model or Transformed-Exponential Model, we also consider a possible linear relationship between degradation signal and time t. The linear model can be written as

$$S_t = \phi_1 + \beta_1 t + \epsilon(t) \tag{7}$$

where ϕ_1 and β_1 are two independent variables with the prior distribution $\pi(\phi_1) = N(\mu_0, \sigma^2)$, $\pi(\beta_1) = N(\mu_1, \sigma^2)$. $\epsilon(t)$ follows the same assumptions that we have stated in the Exponential Model.

So in our project, we will study four models: transformed exponential model and direct linear model under IID error and Brownian error situations. Let $y_t \equiv L_t$ for the transformed exponential model, and $y_t \equiv S_t$ in the linear model, the four models can be unified as a single form:

$$y_t = \phi_0 + \beta_0 t + \epsilon(t) = X_t\beta^* + \epsilon(t) \tag{8}$$

in which $X_t \equiv \begin{pmatrix} 1 & t \end{pmatrix}$, and $\beta^* \equiv \begin{pmatrix} \phi_0 & \beta_0 \end{pmatrix}^T$. We also assume the prior distribution of β^* given σ^2 follow a multivariate normal distribution with mean $\mu_\beta = \begin{pmatrix} \mu_0 & \mu_1 \end{pmatrix}^T$ and covariance matrix $\sigma^2\Sigma$.

Based on the above assumptions, we will derive 4 models in total with different combinations of the model assumptions and error assumptions. In the following sections, we will use y_t in the derivative. Note the y_t has different meanings under different model assumptions.

2.2 Bayesian Posterior Calculation

In this section, we will derive the posterior distribution of the model parameter β and σ^2 for IID case. However, we will not derive the posterior distribution for BM case seperately, but will show that the BM error could be transformed into IID case.

2.2.1 IID Case

Suppose the degradation signals y_{t_i}'s are measured at time $t_1, \ldots, t_n$,under IID error situation, the likelihood function can be written as

$$p(y_{t_1}, \ldots, y_{t_n}|\phi_0, \beta_0, \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n exp\left\{-\frac{1}{2\sigma^2}(Y - X\beta^*)^2\right\} \tag{9}$$

where $Y = (y_{t_1}, \ldots, y_{t_n})^T$ and the $n \times 2$ matrix $X = (x_{t_1} \ldots, x_{t_n})^T$, .

Based on our assumption under IID error scenario, the joint posterior distribution of β^* and σ^2 given Y is

$$\begin{aligned} p(\beta^*, \sigma^2|Y) &\propto p(Y|\beta^*, \sigma^2)\pi(\beta^*|\sigma^2)\pi(\sigma^2) \\ &= \frac{1}{(\sqrt{2\pi\sigma^2})^n} \exp\left\{-\frac{1}{2\sigma^2}(Y - X\beta^*)^T(Y - X\beta^*)\right\} \\ &\quad \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(\beta^* - \mu_\beta)^T\Sigma^{-1}(\beta^* - \mu_\beta)}{2\sigma^2}\right\} \frac{b^a}{\Gamma(a)} \left(\sigma^2\right)^{-(a+1)} \exp\left(-\frac{b}{\sigma^2}\right) \end{aligned} \tag{10}$$

Let $\hat{\beta^*}$ denote the least squared estimator of β^* in model $Y = X\beta^* + \epsilon$, the above joint posterior distribution can be further written as

$$\begin{aligned} p(\beta^*, \sigma^2|Y) &\propto \left(\sigma^2\right)^{-(a+n/2+1)-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma^2}(Y - X\hat{\beta^*})^T(Y - X\hat{\beta^*})\right. \\ &\quad \left.-\frac{1}{2\sigma^2}(\beta^* - \hat{\beta^*})^T X^T X(\beta^* - \hat{\beta^*}) - \frac{1}{2\sigma^2}(\beta^* - \mu_\beta)^T\Sigma^{-1}(\beta^* - \mu_\beta)\right\} \exp\left(-\frac{b}{\sigma^2}\right) \\ &\propto \left(\sigma^2\right)^{-(a+n/2+1)} \exp\left\{-\frac{b}{\sigma^2} - \frac{1}{2\sigma^2}\left(Y - X\hat{\beta^*}\right)^T\left(Y - X\hat{\beta^*}\right)\right. \\ &\quad \left.-\frac{1}{2\sigma^2}\left[\hat{\beta^*}^T X^T X\hat{\beta^*} + {\mu_\beta}^T\Sigma^{-1}\mu_\beta - \beta^{\circ T}\left(X^T X + \Sigma^{-1}\right)\beta^\circ\right]\right\} \\ &\quad \cdot \frac{1}{\sqrt{\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}\left(\beta^* - \beta^\circ\right)^T\left(X^T X + \Sigma^{-1}\right)\left(\beta^* - \beta^\circ\right)\right\} \end{aligned} \tag{11}$$

where, $\beta^\circ \equiv (\hat{\beta}^T X^T X + {\mu_\beta}^T \Sigma^{-1})(X^T X + \Sigma^{-1})^{-1}$

It can be showed that the posterior distribution of β^* given Y and σ^2 follows a multivariate normal distribution

$$\beta^*|Y, \sigma^2 \sim N(\beta^\circ, \sigma^2 \Sigma^\circ) \tag{12}$$

where, $\Sigma^\circ = (X^T X + \Sigma^{-1})^{-1}$

And the posterior distribution of σ^2 given Y is

$$\sigma^2|Y \sim IG(a + \frac{n}{2}, b^\circ) \tag{13}$$

where, $b^\circ = b + \frac{1}{2} Y^T (I - X(X^T X)^{-1} X^T) Y + \frac{1}{2} \hat{\beta^*}^T X^T X \hat{\beta^*} + {\mu_\beta}^T \Sigma^{-1} \mu_\beta - \frac{1}{2} {\beta^\circ}^T (X^T X + \Sigma^{-1}) \beta^\circ$.

Therefore,with the degradation signals y_{t_i}'s $(i = 1, \ldots, n)$ observed, the posterior distribution of β^* can be obtained. Based on this posterior distribution, we use mean of the posterior distribution β° as the point estimator of β^*, and predict a new degradation signal at time $t_n + t$ as $\hat{y}_{t_n+t} = X_{t_n+t} \beta^\circ$

The mean and variance of the predicted value $\hat{y}_{t_n+t}$ are

$$\tilde{\mu}(t + t_n|Y) = E\left[E\left[\hat{y}_{t_n+t}|Y\right]\right] = E\left[E\left[{x_{t_n+t}}^T \beta^\circ|\sigma^2, Y\right]\right] = {x_{t_n+t}}^T \mu_\beta \tag{14}$$

$$\begin{aligned} \tilde{\sigma}^2(t + t_n) &= Var\left[E(\hat{y}_{t_n+t}|\sigma^2, Y)\right] + E\left[Var(\hat{y}_{t_n+t}|\sigma^2, Y)\right] \\ &= E\left[{x_{t_n+t}}^T (\sigma^2 \Sigma^\circ) x_{t_n+t}|Y\right] + E\left[\sigma^2|Y\right] = \frac{b}{a-1}\left({x_{t_n+t}}^T \Sigma^\circ x_{t_n+t} + 1\right) \\ &= {x_{t_n+t}}^T \left(\frac{b}{a-1}\Sigma^\circ\right) x_{t_n+t} + \frac{b}{a-1} \end{aligned} \tag{15}$$

2.2.2 BM Case

If we assume that the error term is followed by a Brownian Motion, that means the increments of the degradation signal in different intervals with same length and without overlap should be IID. Therefore we can use a method to transform the BM case into the IID case we have already done above.

As the definition of Brownian Motion, the increments between two adjunct degradation signals, i.e. $Y_{i+1} - Y_i$, for all $i = 2, ..., n_k - 1$, are followed by indenpendent normal distributions with defferent variances which depend on the length of the interval $|t_{i+1} - t_i|$, for all $i = 2, ..., n_k - 1$, i.e. $(Y_{i+1} - Y_i) \sim N((t_{i+1} - t_i)\beta, (t_{i+1} - t_i)\sigma^2)$. Notice that our goal is to get identical distribution, then the next step is to standardlize it. A multipler $1/\sqrt{t_{i+1} - t_i}$ should be applied to each increment $Y_{i+1} - Y_i$. And one can check that $(Y_{i+1} - Y_i)/\sqrt{t_{i+1} - t_i} \sim N(\sqrt{t_{i+1} - t_i}\beta, \sigma^2)$, for all $i = 2, ..., n_k - 1$. For the first signal, we use the transformation $(Y_1/\sqrt{t_1}) \sim N(\phi/\sqrt{t_1} + \sqrt{t_1}\beta, \sigma^2)$. The matrix form of the transformation is given below:

$$\Delta y_i \doteq (y_i - y_{i-1})/\sqrt{t_i - t_{i-1}} \tag{16}$$

$$\Delta x_i \doteq [0 \quad \sqrt{t_i - t_{i-1}}] \tag{17}$$

for all $i = 2, \cdots, n_k - 1$, where n_k is the observations of the kth bearing B_k. And for $i = 1$ case, we have

$$\Delta y_1 \doteq y_1/\sqrt{t_1} \tag{18}$$

$$\Delta x_1 \doteq [\ 1/\sqrt{t_1} \quad \sqrt{t_1}\ ] \tag{19}$$

After the transformation (16) and (18), we could prove that the new varible Δy_i could be writen as:

$$\Delta y_i = \Delta x_i \beta + \Delta \epsilon_i \tag{20}$$

where the β is the same as the one defined in IID case, and $\Delta\epsilon_i$ is independent identically distributed with a normal distribution with mean 0 and a common variance not depends on i, say $\Delta\epsilon_i \sim N(0, \sigma^2)$. Notice the σ^2 here is different from the definition in IID case, since it is standardarized in the transformation.

Similarly we can define $\Delta Y_t = (\Delta y_1, \Delta y_2, \cdots, \Delta y_{n_k-1})'$, and $\Delta X_t = (\Delta x_1, \Delta x_2, \cdots, \Delta x_{n_k-1})'$, and perform the same analysis as we derived in IID case.

The predicted degradation signal at

2.3 Residual-life distribution

Given the posterior distribution of β^*, we would like to determine the distribution of the time until failure for the component. From the perspective of degradation signal, we consider a component is failed when the degradation signal reaches and exceeds a threshold D. We will discuss the residual life distribution under IID and BM scenarios.

2.3.1 IID Case

Let T represent the residual life of the component at time t_n, and T satisfies $y_{t_n+T} = D$. Then the cdf of T given Y can be written as

$$\begin{aligned} P(T \le t|Y) &= P(y_{t_n+T} \ge D|y) \\ &= 1 - P(y_{t_n+T} \ge D|y) \\ &= 1 - P\{Z < \frac{D - \tilde{\mu}(t+t_n)}{\sqrt{\tilde{\sigma}^2(t+t_n)}}\} \\ &= P\{Z \ge \frac{D - \tilde{\mu}(t+t_n)}{\sqrt{\tilde{\sigma}^2(t+t_n)}} = \phi(g(t)) \end{aligned} \quad (21)$$

where Z is a standard normal random variable.

It can be shown that the domain of the residual life T is $(-\infty, \infty)$,which is not true in our case since the residual life should always be a non-negative value. Adding a constraint $T \ge 0$, we get a truncated cdf of T as

$$\begin{aligned} P\{T \le t|Y, T \ge 0\} &= \frac{P[0 \le T \le t|Y]}{P[T \ge 0|Y} \\ &= \frac{\phi((g(t))g'(t)}{1 - \phi(g(0))} \end{aligned} \quad (22)$$

And we can further get the pdf of the residual life

$$f_{T|Y,T\ge 0}(t) = \frac{\phi(g(t))g'(t)}{1 - \Phi(g(0))} \quad (23)$$

where $\phi(\cdot)$ is the pdf of a standard normal random variable.

2.3.2 BM Case

Let T represent the residual life of the component at time t_n, and T satisfies $y_{t_n+T} = D$. We can find the cdf of T given Y as follows:

$$\begin{aligned} P(T \le t|Y) &= P(y_{t_n+T} \ge D|y) \\ &= 1 - P(y_{t_n+T} \ge D|y) \\ &= 1 - P\{Z < \frac{D - \tilde{\mu(t)} - Y_{t_n}}{\sqrt{\tilde{\sigma}^2(t)}}\} \\ &= P\{Z \ge \frac{D - \tilde{\mu}(t)}{\sqrt{\tilde{\sigma}^2(t)}} = \phi(g(t)) \end{aligned} \quad (24)$$

where Z is a standard normal random variable. It can be shown that the domain of residual life T given the cdf above is $(0, \infty)$, so it is easy to obtain the pdf of T as

$$f_{T|Y,T\geq 0}(t) = \phi(g(t))g'(t) \tag{25}$$

where $\phi(\cdot)$ is the pdf of a standard normal random variable.

3 Implementing the model and result analysis

In this section, we apply the degradation models presented above to the degradation signals of 26 bearings that we have run to failure under accelerated testing conditions. There are another 25 bearings' degradation signal data that will be used to derive the priors of the model parameters (intercept and slope).

The bearing degradation signal was collected and processed every 2 minutes, i.e. $t_{i+1} - t_i = 2$, for all $i \geq 0$.

3.1 Derive the prior

The priors we used for the inverse Gamma distribution are kind of Non-informative. As the Bayesian theory show that the Jeffrey's prior for a normal variance is a inverse Gamma distribution with parameters $a = 0$ and $b = 0$. Therefore, the smaller the parameters the smaller information contained in the prior. Thus we just put very small number for the prior a and b in our model since we don't have much information for it.

The priors we used for the model parameters β is informative. We used the real failure time of another 25 bearings to estimate the slope parameter from a one-parameter Bernstein Distribution, which is also called inverse Normal distribution. And use the first degradation signal of the 25 bearings to estimate the distribution of the intercept, which is also a normal distribution. Additionally, the slope and intercept are assumed to be independent to each other.

3.2 Implementing the model

The prior information is then used to estimate the residual life distribution of the component being monitored. Subsequent degradation-based sensory information is the used to update these residual life distributions, in real-time. The Figure 1 show the mean and variance of the sequence of posterior intercept and the sequence of posterior slope, where the X-axis is the time t_k.

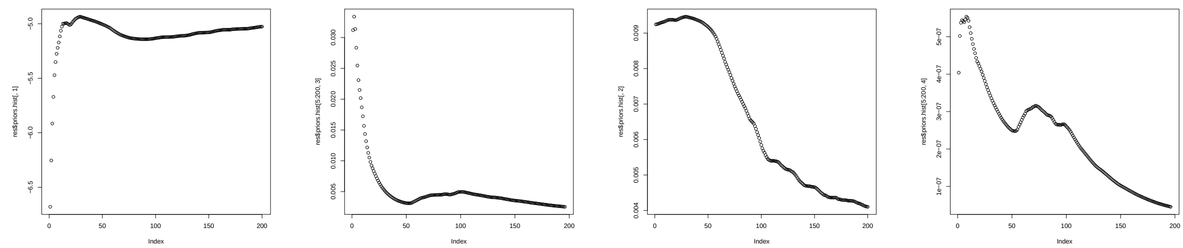

Figure 1: The Posterior Distribution of β. From left to right: (1). Mean Posterior Intercept; (2). Mean of Posterior Slope; (3). Variance of Posterior Intercept; (4). Variance of Posterior Slope

Figure 2 illustrates the evolution of the revised residual life distributions as more in-situ sensory information is used in the updating methodology.

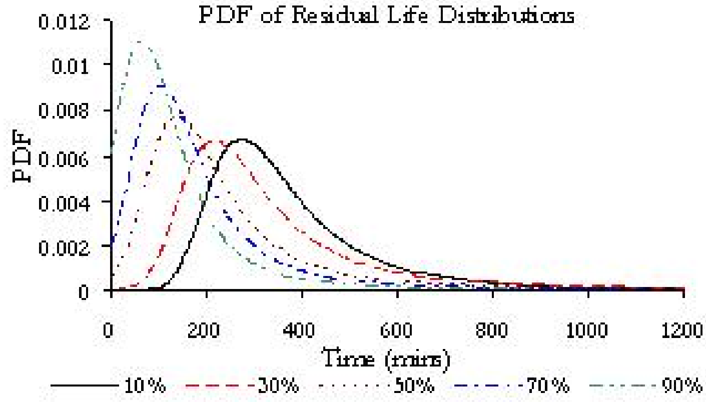

Figure 2: The Interval Plot

3.3 Estimate the residual life

Now we have the RLD function of the residual life. With this distribution, we can calculate some point estimator of the residual life T. The candidates include the mean, median, and so on. The Median finally is chosen as an approximate value of the residual life. There are two reasons for choosing the Median. First, the distribution of the residual life is highly skewed. Under these conditions, it is common to use the Median as a measure of central tendency. The second reason for choosing the Median is that the Mean does not exist. Although the distribution of the signal is Normal, the residual life distribution, however, it is similar to the Bernstein distribution (Ahmad and Sheikh 1984 [3]).

Also, in IID error case, we used a "conservative" policy when apply the median as an estimator of residual life. Since that in practice, there is a situation that the estimate is not always decreasing with respect to time. To eliminate the flactuation, at each time point, we will choose the smallest estimate before this point. But in Brownian Motion case, this policy is not necessary.

3.4 Results Analysis

We will compare the estimated residual life with the real failure time. The difference between predicted failure times and actual bearing failure times are used to compute prediction errors in order to evaluate the performance of the proposed degradation methodologies. For each sampling epoch, the percentage difference between the actual and expected bearing failure times is computed using expression (15) and equals

$$D_h^i = \frac{(m_h^i + t_h^i) - FT_h}{FT_h} \tag{26}$$

where, D_h^i is the prediction error associated with bearing, B_h, computed at sampling epoch i; FT_h is the actual failure time of B_h; m_h^i is the current total operating time of B_h at sampling epoch i; m_h^i is the median of the residual life distribution of B_h computed at the ith sampling epoch; and t_h^i is the time at epoch i for bearing B_h.

Figure 3, Figure 4, Figure 5, Figure 6 (See the very end of this paper) present the interval plot of the prediction errors for all twenty-five bearings (from 26th to 50th) evaluated at nine degradation percentiles (after the actual failure time is observed) for the exponential degradation model with IID error assumption, exponential degradation model with BM error, linear degradation model with IID error, and linear degradation model with BM error, respectively.

By analyzing the prediction errors, we observe that all the estimator predict the residual life pretty well at 90% of the actual life time of the bearing. When turn to more detail, we have 2 observed conclusions.

First, when we compare the exponential degradation model and the linear degradation model, the prediction errors show that the exponential model is better than the linear model because the range of the predict error of the exponential model is around -25%∼5%, while the range for linear model is around -45%∼-5%. The linear degradation model tends to underestimate the predicted residual life even at the very end side of the bearing's lifetime.

Second, both the IID error assumption and the BM error assumption are good. But in exponential degradation model case, the BM error assumption is slightly better than the IID assumption because the shape of the prediction error curve is concave for BM error, while it is convex for IID error. Thus when only look at the second half of a bearing's life, we prefer to use BM to estimate the residual life, though it will underestimate a little bit in most case.

4 Conclusions

Degradation signals have been widely used to improve our understanding of the underlying physics-of-failure. In most applications, degradation signals are often correlated with the degree of damage. A common approach is to model the path of the degradation signal using random coefficients/stochastic models.

In this paper, we improved [1](Gebraeel et al.) by assuming that the variance of the error term is random, while this variance is assumed as a fixed constant in [1]. Additionally, a non-informative prior for the variance is applied.

To assess the validity of our degradation methodology, we perform a two-dimensional analysis of the prediction results. The first criterion is related to identifying the appropriate functional form for modeling the degradation signal of the application under consideration in this case-study. Secondly, we evaluate the difference between using a IID Error assumption and BM error assumption.

Another very important topic in this area is the optimal stopping problem. In the real world, we can acquire data to perform the analysis. But the problem is that the estimate of the residual life varies all the time when we add new sample observations in. Also, data is not free in most real case, which means there will be a cost for acquiring new data. For the above two reasons, people would like to choose an appropriate time to stop the research and get an reliable estimate of the residual life based on the cost. There will be a balance between the accuracy and the cost people want to pay. Then the question is to find this balance in some sense.

Besides the optimal stopping rules, future research directions also include the feasibility of using other models and other failure time distributions for the sensory-updating methodology.

A R Codes: Description

As introduced, there are four different models for the data: linear trend with error *independently and identically distributed* (IID) (with mean zero and variance σ^2), linear trend with error following Brownian Motion (with variance σ^2), exponential trend with IID error, exponential trend with error following Brownian Motion.

And it is assumed that the model parameters (intercept and slope) follow bivariate normal distribution, and the variance σ^2 of error follows Inverse Gamma distribution (either informative or noninformative).

We also try some method to correct data, taking the envelope of the original data as the new input data.

The R codes implement the prediction of the residual life at different history time points by specifying these parameters mentioned above for each group of data (corresponding each bearing). The predicted sequence of residual life values (based on data accumulated till different time point), and the prediction errors can be returned for graphical and statistical analysis.

B R Codes

B.1 Sample Codes

The sample codes is used to compute the prediction errors for Bearing 28 that used to perform the interval plot for the exponential degradation signal with IID error model.

```
csv <- read.csv("C:\\166project\\B28.csv",header=T)
y <- csv[,2]
t <- csv[,1]
if (t[1]==0) {t <- t + 2}
X <- matrix(c(rep(1,length(y)),t),byrow=F,nrow=length(y))
res <- RLPrediction(y, X, ifExp=T, ErrDist="IID")
```

B.2 R functions

```
#22S:166 Project
library(MASS)

######################################################
###                Functions Needed:               ###
######################################################

#1. function needed to update posteriors (Beta & sigma2)
##  for priors: Beta~normal, sigma2~inverse gamma
post.NormIG <- function(muBeta,SigmaBeta,a,b,i,X,y,sigma2) {
    betahat <- ginv(t(X) %*% X) %*% t(X) %*% y
    meanBeta.po <- t((t(betahat) %*% t(X) %*% X + t(muBeta)
                   %*% ginv(SigmaBeta)) %*% ginv(t(X)
                   %*% X + ginv(SigmaBeta)))
    varBeta.po <- sigma2*(ginv(t(X) %*% X + ginv(SigmaBeta)))
    a.po <- a+(i/2)
    b.po <- b+0.5*( t(y-X%*%betahat)%*%(y-X%*%betahat) +
            (t(betahat)%*%t(X)%*%X%*%betahat) +
            (t(muBeta)%*%ginv(SigmaBeta)%*%muBeta) -
            (t(meanBeta.po)%*%(t(X)%*%X+ginv(SigmaBeta))%*%meanBeta.po))
    b.po <- b.po[1,1] #b.po must be transformed to scalar form
    list(meanBeta.po=meanBeta.po, varBeta.po=varBeta.po,
         a.po=a.po, b.po=b.po)
}


#2. functions needed to update posteriors (Beta & sigma2)
##  for priors: Beta~normal, sigma2~noninformative Inverse Gamma
post.NormNoninfoIG <- function(muBeta,SigmaBeta,a,b,i,X,y,sigma2) {
    betahat <- ginv(t(X)%*%X)%*%t(X)%*%y
    meanBeta.po <- betahat
    varBeta.po <- sigma2*ginv(t(X)%*%X)
    a.po = (i-1)/2
    b.po = 0.5*(t(y-X%*%betahat)%*%(y-X%*%betahat))
```

```
    b.po <- b.po[1,1] #b.po must be transformed to scalar form
    list(meanBeta.po=meanBeta.po, varBeta.po=varBeta.po,
         a.po=a.po,b.po=b.po)
}


#3. functions needed to update posteriors (Beta & sigma2)
##  for priors: Beta~normal, sigma2~noninformative
post.NormNoninf <- function(muBeta,SigmaBeta,X,y,sigma2) {
    betahat <- ginv(t(X) %*% X) %*% t(X) * y
    meanBetaNorm <- t((t(betahat) %*% t(X) %*% X / sigma2 +
                    t(muBeta) %*% ginv(SigmaBeta)) %*% ginv(t(X)
                    %*% X / sigma2 + ginv(SigmaBeta)))
    varBetaNorm <- ginv(t(X) %*% X / sigma2 + ginv(SigmaBeta))
    list(meanBeta.po=meanBeta.po, varBeta.po=varBeta.po)
}


#4. functions needed to update posteriors (Beta & sigma2)
##  for priors: Beta & sigma2~noninformative
post.Noinf <- function(muBeta,SigmaBeta,X,y,sigma2) {
    meanBeta.po <- ginv(t(X) %*% X) %*% t(X) %*% y
    varBeta.po <- ginv(t(X) %*% X / sigma2)
    list(meanBeta=meanBeta.po, varBeta=varBeta.po)
}


#5. function needed to compute CDF of residual life
##  gt: function g(t), see Nagi(2005)--Residual Life...
gt <- function(t,tk,meanBetapo,varBetapo,sigma2,D){
  XX <- t(c(1,(t+tk)))
  mu <- XX %*% meanBetapo
  var <- (XX) %*% varBetapo %*% t(XX) + sigma2
  gt <- (mu - D) / sqrt(var)
}


###################################################################
###               Begin Here ............                       ###
###################################################################


###RLPrediction: function used to predict residual life
RLPrediction<-function(y, X, ifExp=T, ErrDist="IID", Prior=1, ifEnvelope=F){

#input parameters:
#--y: data
#--X: design matrix
#--ifExp: specify if the model is exponential (T: exp, F: linear)
#--ErrDist: error distribution ("IID" or "BM")
#--Prior: prior dist of Beta & sigma2(1:Norm&IG, 2:Noinfo&IG, 3:Norm&Noinfo,
#         4: Noinfo&Noinfo)
```

```
#--ifEnvelope: specify if the data need to be "enveloped" (T or F)

###output results
#--rl.med: median prediction of residual life
#--rl.meda: conservative(adjusted) median prediction of predicted redual life
#--rl.exp: truncated expectation prediction of residual life
#--rl.expa: conservative(adjusted) truncated expectation prediction of residual life
#--err.med: error ratios for median
#--err.meda: error ratios for adjusted median
#--err:exp: error ratios for expectation
#--err.expa: error ratios for adjusted expectation
#--priors.hist: matrix of values of priors (col1: meanBeta1, col2:meanBeta2,
#          col3: varBeta1, Col4: varBeta2, Col5: corrBeta1&2)


#Step 1: Initialize Data Settings: M, D, y, and parameter priors

    M <- 5000    #large number when truncate
    D <- 0.025
    t <- X[,2]
    maxk <- length(t)


    if (ifEnvelope) { # take the envelope of the data
        for(i in 2:maxk)
        y[i]=max(y[i-1],y[i])
    }

    if (ifExp) { # for exponential model
        D <- log(D)
        y <- log(y)
    }
    if ((Prior==1) || (Prior==2)) {
        a.pr <- 10
        b.pr <- 0.002
        sigma2 <- b.pr/(a.pr-1)
    }
    else {
        sigma2 <- 0.095253869
    }

    m1 <- 0.00924345006881;   # mu of the slope Bern
    v1 <- 0.00000209244286;   # var of the slope Bern
    m2 <- log(0.001);         # mu of the intercept
    v2 <- 0.83858266088337;   # var of the intercept

    #in BM case, we need to make the error terms iid n(0,1)
    #then follow the same way as in IID case
    if (ErrDist=="BM") {
        sqt1 <- sqrt(t[1])
        y[1] <- y[1] / sqt1
        X[1,1] <- 1 / sqt1
```

```
        X[1,2] <- sqt1
        for(i in 2:maxk) {
            sqti <- sqrt(t[i]-t[i-1])
            y[i] <- (y[i] - y[i-1]) / sqti
            X[i,1] <- 0
            X[i,2] <- sqti
        }
    }

    meanBeta <- c(m2,m1)
    varBeta <- matrix(c(v2,0,0,v1),nrow=2,byrow=T)


#Function of posteriors: arguments include prior parameter and new data

#Step 2: Start a new loop to updata numEachTime data each time.

    #Initialize: Xnew, sigma2pr, meanBetapr, varBetapr
    sigma2pr <- sigma2
    meanBetapr <- meanBeta
    varBetapr <- varBeta
    median <- numeric()
    medianAdj <- numeric() #conservative sequence of medians
    exp <- numeric()
    expAdj <- numeric() #conservative sequence of truncated expectations
    medianlast <- 10000
    explast <- 10000
    varMed <- numeric()
    varE <- numeric()
    betaLog1 <- numeric()
    betaLog2 <- numeric()
    betaLog3 <- numeric()
    betaLog4 <- numeric()
    betaLog5 <- numeric()

    #beginning of loop from the 1st datum to the last datum
    for(i in 1:maxk){
        tk <- t(i)
        if (i==1) {Xnew <- t(X[1:i,])}
        else {Xnew <- X[1:i,]}
        Ynew <- y[1:i]


#Step 3: Using function to compute the posterior

        if (Prior==1) { #beta~normal, sigma2~inverse gamma
            res <- post.NormIG(meanBetapr,varBetapr,
                   a.pr,b.pr,i,Xnew,Ynew,sigma2)
            meanBetapo <- res$meanBeta.po
            varBetapo <- res$varBeta.po
            a.po <- res$a.po
            b.po <- res$b.po
```

```
            sigma2 <- b.po /(a.po-1)
        }
        if (Prior==2) { #beta~normal, sigma2~noninformative inverse gamma
            res <- post.NormNoninfIG(meanBetapr,varBetapr,
                   a.pr,b.pr,i,Xnew,Ynew,sigma2)
            meanBetapo <- res$meanBeta.po
            varBetapo <- res$varBeta.po
            a.po <- res$a.po
            b.po <- res$b.po
            sigma2 <- b.po /(a.po-1)
        }
        if (Prior==3) { #beta~normal, sigma2~noninformative
            res <- post.NormNoninf(meanBetapr,varBetapr,Xnew,Ynew,sigma2)
            meanBetapo <- res$meanBeta.po
            varBetapo <- res$varBeta.po
        }
        if (Prior==4) { #beta~noninformative, sigma2~noninformative
            res <- post.Noninf(meanBetapr,varBetapr,Xnew,Ynew,sigma2)
            meanBetapo <- res$meanBeta.po
            varBetapo <- res$varBeta.po
        }

        betaLog1 <- c(betaLog1,meanBetapo[1])
        betaLog2 <- c(betaLog2,meanBetapo[2])
        betaLog3 <- c(betaLog3,varBetapo[1,1])
        betaLog4 <- c(betaLog4,varBetapo[2,2])
        betaLog5 <- c(betaLog5,varBetapo[1,2])


#Step 4: Use posteriors to calculate CDF of RLD by function rldIIDcdf or rldBMcdf

        if (ErrDist=="BM") {
            for(tt in 1:M){
              cdfs[tt] <- pnorm(gt(tt,tk,meanBetapo,varBetapo,sigma2,D))
            }
        }
        else {
            cdf0<-pnorm(gt(0,tk,meanBetapo,varBetapo,sigma2,D))
            cdfs <- numeric()
            for(tt in 1:M){
              cdfs[tt] <- (pnorm(gt(tt,tk,meanBetapo,varBetapo,sigma2,D))-cdf0)
                                    / (1-cdf0)
            }
        }


#Step 5: Calculate Median / Truncated Expectation

        #Median
        pr <- 0
        medianTmp <- 0
        while(pr < 0.5){
```

```
          medianTmp <- medianTmp + 1
          pr <- cdfs[medianTmp]
        }
        if (abs(cdfs[medianTmp]-.5) > abs(cdfs[medianTmp-5]-.5)) {
          medianTmp <- medianTmp - 1
        }


        #Truncated Expectation
        expTmp <- 0
        for(j in 1:M){
          expTmp <- expTmp + (1-cdfs[j])
        }

#Step 6: Apply conservative policy if IID

        median <- c(median,medianTmp)
        if(medianlast <= medianTmp){
          medianAdj <- c(medianAdj,medianlast)
        }else{
          medianAdj <- c(medianAdj,medianTmp)
          medianlast <- medianTmp
        }

        exp <- c(exp,expTmp)
        if(explast <= expTmp){
          expAdj <- c(expAdj,explast)
        }else{
          expAdj <- c(expAdj,expTmp)
          explast <- expTmp
        }


#Step 7: Calculate Loss function (Numerical Var of Estimator of Residual Life)
        varMedian <- 0
        for(j in 2:M){
        #  varMedian <- varMedian + (j-medianlast)^2 * (cdfs[j] - cdfs[j-1])
          varMedian <- varMedian + (j-medianTmp)^2 * (cdfs[j] - cdfs[j-1])

        }
        varMed <- c(varMed,varMedian)

        varExp <- 0
        for(j in 2:M){
        #  varExp <- varExp + (j-explast)^2 * (cdfs[j] - cdfs[j-1])
          varExp <- varExp + (j-expTmp)^2 * (cdfs[j] - cdfs[j-1])
        }
        varE <- c(varE,varExp)

#Step 8: Update the Prior using Posterior
#       meanBetapr <- meanBetapo
#       varBetapr <- varBetapo
```

```
#       if ((Prior==1) || (Prior==2)) {
#           a.pr <- a.po
#           a.pr <- a.po
#       }
    }       # LOOP END


#Step 9: Compute the error ratios of the predicted values of residual life.

    # creat the time points  to compute the everage 5-15%, 15-25%,...
    tpoint<-seq(.05, .95, .1)
    # choose time tinterval as 5% 15%...95% of the failure time
    # ERLD(median,..) in the nine intervals: 5-15%, 15-25%,..., 85-95%
    tinterval=round(tpoint*maxk)

    tmedian<-numeric()
    tmedianAdj<-numeric()
    texp<-numeric()
    texpAdj<-numeric()

    # tnow: time point of 10%, 20%,..., 90% of index failure time without tnull
    tnow<-t[round(maxk*seq(.1,.9,.1))]

    for(i in 1:9) {
        tmedian[i] <- mean(median[tinterval[i]:tinterval[i+1]])
        tmedianAdj[i] <- mean(medianAdj[tinterval[i]:tinterval[i+1]])
        texp[i]<-mean(exp[tinterval[i]:tinterval[i+1]])
        texpAdj[i]<-mean(expAdj[tinterval[i]:tinterval[i+1]])
    }

    medianErr <- (tmedian+tnow-t[maxk])/t[maxk]
    medianAdjErr <- (tmedian+tnow-t[maxk])/t[maxk]
    expErr <- (texp+tnow-t[maxk])/t[maxk]
    expAdjErr <- (texpAdj+tnow-t[maxk])/t[maxk]

#ending: set the returned values
    priors <- matrix(rep(0,5*maxk),nrow=maxk)
    priors[,1] <- betaLog1
    priors[,2] <- betaLog2
    priors[,3] <- betaLog3
    priors[,4] <- betaLog4
    priors[,5] <- betaLog5
    list(rl.med=median, rl.meda=medianAdj, rl.exp=exp, rl.expa=expAdj,
         err.med=medianErr, err.meda=medianAdjErr, err.exp=expErr,
         err.expa=expAdjErr, priors.hist=priors)

} #end of function (RLPrediction)
```

References

[1] Gabraeel,N., Lawley,M., Li,R., Ryan,J.(2005) *Residual-life distributions from component degradation signals: A Bayesian approach. IIE Transactions*, 37, 543-557

[2] Morris H. DeGroot, *Optimal Statistical Decision* New York, McGraw-Hill [1969, c1970]. Math Lib QA279.4.D43

[3] M. Ahmad and A. Sheikh, *Bernstein reliability model: derivation and estimation of parameters* Reliability Engineering, vol. 8, 1984, pp. 131-148.

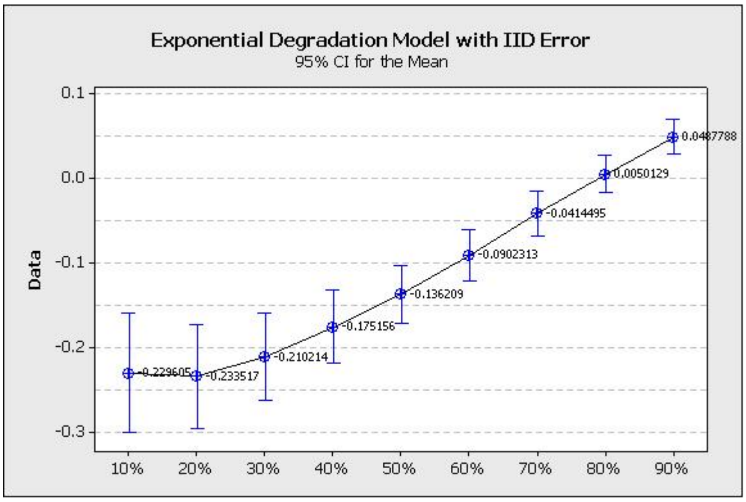

Figure 3: The Interval Plot: Exponential Degradation Signal with IID error model

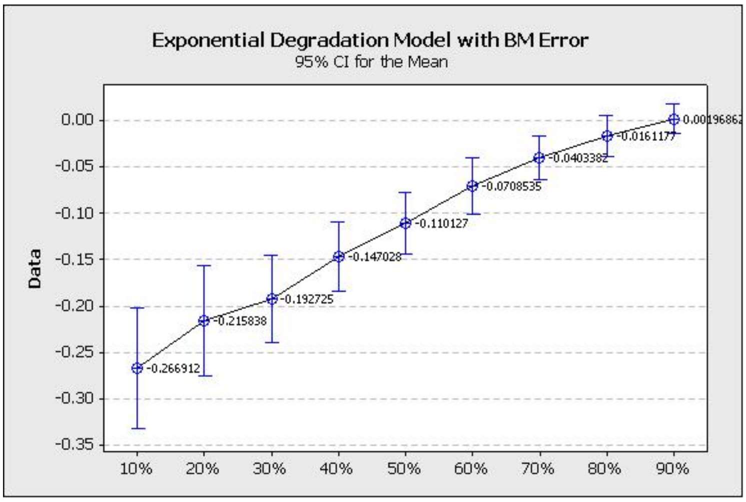

Figure 4: The Interval Plot: Exponential Degradation Signal with BM error model

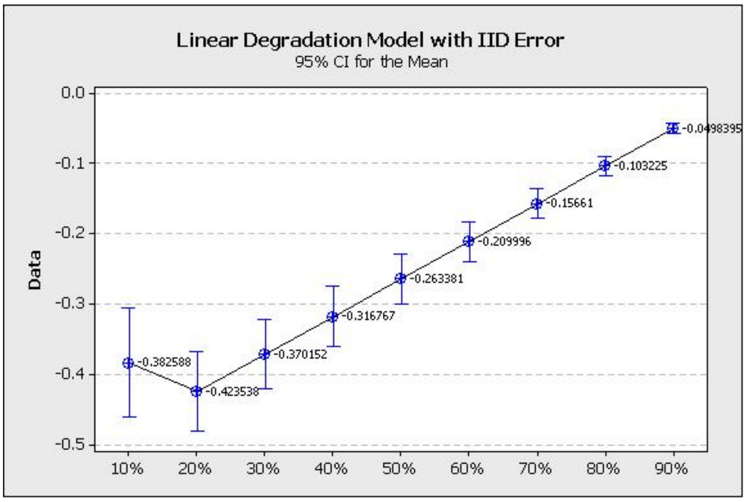

Figure 5: The Interval Plot: Linear Degradation Signal with IID error model

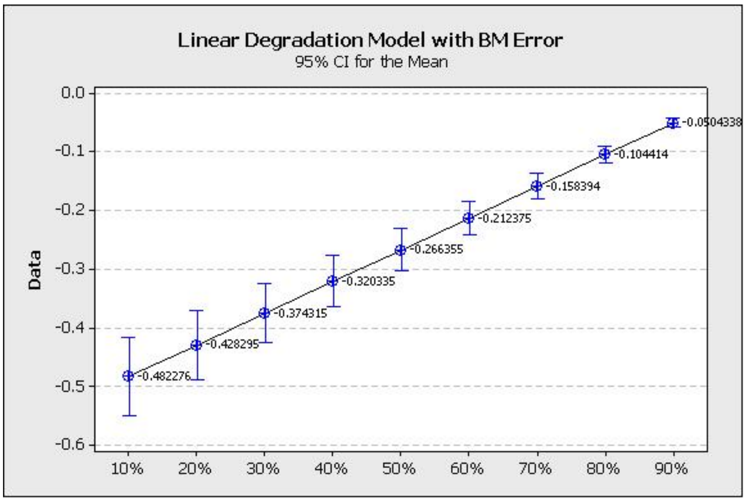

Figure 6: The Interval Plot: Linear Degradation Signal with BM error model