Prognostics by classifying degradation stage on Lambda architecture

Jinhyuck Choi
R&D Center
Korea Gas Technology Corporation
Daejeon, Republic of Korea
Email: zinyugi@gmail.com

Jinwoo Lee
R&D Center
Korea Gas Technology Corporation
Daejeon, Republic of Korea
Email: ns0331@kogas-tech.co.kr

Won Jeong Cho
R&D Center
Korea Gas Technology Corporation
Daejeon, Republic of Korea
Email: chowj@kogas-tech.co.kr

***Abstract*—To enhance the reliability and availability of an asset in its life, predicting the remaining useful life of an asset is strongly encouraged by assessing the extent of deviation or degradation of the asset's monitored parameters from its expected normal operating conditions. Although intelligent fault prognostic techniques such as machine learning and artificial neural networks have been applied in modern industries, application in actual industrial conditions requires that the forecasting process is revealed and more descriptive. To investigate the issue and increase the accuracy, this paper proposes an additional technique that can be further applied to any recent intelligent prognostic methods. The proposed method consists of two steps. First, the entire training set is divided into several degradation stages before regression using a heuristic approach and then the regression results are synthesized for each stage. The proposed method will increase the monotonicity of the predictive parameters, thus helping improve the predictive model's accuracy. To demonstrate the hypothesis, real condition monitoring data of high-pressure LNG pump and acceleration experimental data of a rotating machine is used for an experiment. Moreover, a system in which the proposed method can be appropriately executed is introduced with Lambda architecture. Finally, by demonstrating that the proposed method is capable of parallel computing, it is proven suitable for use in the proposed large-scale distributed processing system.**

I. Introduction

Status monitoring and Prognostics and Health Management (PHM) are directly related to facility maintenance activities such as determining the current status of facilities and or the equipment repair time. As an essential activity that must be accompanied by long-term operation, its role has become increasingly important in recent decades as in [1], such that industrial fields have increased not only the productivity, reliability, and stability of their equipment.

Most importantly, rotating machine typically plays a very significant role throughout the whole plant. Therefore, components that can detect abnormal symptoms are installed in accordance with Condition-Based Maintenance (CBM) as in [2]. Since vibrations change when a mechanical fault occurs or when an inner part deteriorates, it is well-known in many studies that vibration monitoring is essential for detecting the symptoms of mechanical defects such as wear, malfunction, noise, and structural damage.

Naturally, the collected vibration data is used by artificial intelligence techniques to predict the remaining useful life of rotating machinery as training data, so many intelligent prognostic techniques based on machine learning have been developed. The well-known methods are the linear regression, the regression tree, ENSEMBLE, Support Vector Machine (SVM), Gaussian Process Regression (GPR), and Artificial Neural Network (ANN) as in [3], [4], [5], [6], [7].

However, there are several issues with the actual facilities of the national infrastructure industry such as public energy plants. First, the amount of fault data is relatively small compared to the normal state's data because of excessive maintenance for safety reasons; this will likely result in an over-fitting prediction model while training and to make it difficult to determine the general fault type. Furthermore, if there are several hidden steps between the initial state and the final defect state, the defect detection procedure will vary with various conditions. In addition, it is difficult to extract predictive parameters that have monotonic characteristics across the entire period from initiation to defect.

In this paper, the assumption that several stages of degradation exist throughout the entire life of the machine is the main key to solving the mentioned issues. In particular, the features of degradation stages will be remarkable for determining what type of fault occurs and which prediction model is best suited for new incoming data. In addition, by dividing the interval into smaller parts that are clustered from their own features, the monotonicity of the predictive parameters can be increased and the performance can be further improved over the single regression across the entire life. In short, the proposed method consists of three steps; the first step divides the entire life of the machine into several sequential degradation stages before regression. Dividing the degradation stages has been previously mentioned in many papers as prediction methods based on the probability estimation of the health state as in [8], [9]. The second step is to estimate the regression models in each degradation stage. Various regression methods can be applied in this step such as those mentioned above. Lastly, the estimated regression values in each degradation stage are synthesized as the final remnant useful life. In addition, this paper introduces a predictive maintenance system model that utilizes the proposed method and mainly consists of three process parts: data acquisition, fault diagnosis, and the remnant useful life prognosis. In addition, a data processing

978-1-7281-6286-7/20/$31.00 ©2020 IEEE

framework called *Lambda* architecture is introduced on which the introduced predictive maintenance system can operate appropriately. The architecture is implemented using *Hadoop* as in [10]. Moreover, the proposed method improves the prediction accuracy by finding degradation stages through several tests; those main predictive parameters show better monotonicity. Additionally, parallel computing improves the additional computing cost consumed by the computation for determining the degradation stage.

In Section II, the mentioned predictive maintenance system and *Lambda* architecture on which the predictive maintenance system operates appropriately are introduced. Section III gives a detailed description of the proposed prognostic approach. Finally, Section IV shows the results of predicting the remnant useful life and those of parallel computing.

II. System and architecture

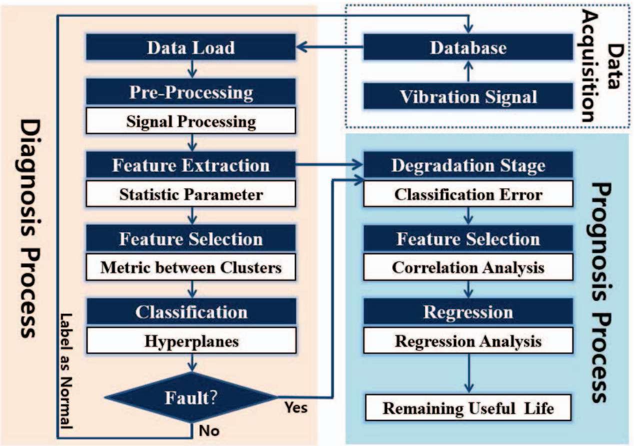


Fig. 1. The Predictive Maintenance System (PMS)

A. Predictive Maintenance System (PMS)

We define the Predictive Maintenance System (PMS) as in Fig. 1 and [11]. The Predictive Maintenance System (PMS) process using machine learning is mainly divided into three categories as shown Fig. 1. First, monitoring data such as vibrations are obtained from a sensor or an indicator at a facility and stored in a well-defined database that allow users to know where and when the vibrations occur. Then, in the *Diagnosis Process*, the obtained vibration data is classified into several groups to determine the state of the facility. For example, a pump in operation will progress in the closest state among the four: rotor bar defects, friction defects, bearing defects, and normal. If the state of the facility is progressing in a fault, the residual time to the fault is predicted by a prognostic model in the *Prognosis Process*. Otherwise, the data determined to be in a normal status is stored in the database and labeled as normal. To carry out this process, the diagnostic and prognostic models should be trained in advance. The following describes the procedure of training the diagnostic model.

1) Steps of training the diagnostic model:

- Data loading: Load raw data labeled with a type of status or fault.
- Pre-processing: Pass through signal processes such as *Fourier* transform and wavelet.
- Feature extraction: Calculate statistical values in time and frequency domains as feature.
- Feature selection: Select features according to the density of clusters in each features.
- Classification: Train a hyperplane for classification by using supervised learning.

The following steps for training prognostic model are the same as those for training the diagnostic model until the step of determining the degradation stages as in *2)* of Section II.A. As mentioned above, the step of determining the degradation stages is performed before the regression step. After determining the degradation stages, to estimate the regression model from the observed vibration data and the corresponding operating time, the main predictive factor of the observation is extracted by correlation coefficient analysis. The following describes the proposed procedure of training the prognostic model.

2) Steps of training the prognostic model:

- Data loading: Load a set of time series continuous raw data for a specific fault labeled with the operating time.
- Pre-processing: Pass through signal processes such as *Fourier* transform and wavelet.
- Feature extraction: Calculate statistical values in time and frequency domains as features.
- Determination of degradation stages: Find the degradation stages that have the lowest classification error.
- Extraction of the main predictive factors: Find the best features using correlation coefficient analysis.
- Regression: Estimate the regression model in each degradation stage.

B. Data processing framework with Lambda architecture

To implement the predictive maintenance system introduced in Section II.A, a data processing framework is established with appropriate software as in Fig. 2; here, PMS means the introduced predictive maintenance system. The applied data-processing architecture is *Lambda* architecture designed to handle massive quantities of data by taking advantage of both *Real-time Processing* and *Batch Processing*, as shown in Fig. 2. The predictive maintenance system requires obtaining two types of data, streaming-type sensing data such as vibrations and event-type data such as failure and maintenance data. However, streaming-type vibration data is too bulky to accumulate in its original form. Therefore, the original waveform of the vibration data is converted into a more compact form to train or evaluate the proposed diagnostic and prognostic models. Like pre-processing and feature extraction steps in *1)* and *2)* of Section II.A, the immediate processing of a constantly incoming streaming data passes through *Real-time Processing* in Fig. 2. In addition, in *Real-time Processing*, new incoming data is evaluated by trained models to determines whether a failure has occurred and how long the facility's useful life can be maintained. As sensing and event data accumulate, to obtain diagnostic and prognostic models for the above mentioned evaluations, gathering training datasets, feature selection, and training models are executed and the results are restored in *Batch Processing*. Since the waveform of sensing datasets comes from various sites in real-time and the targets of training prognostic models are the entire lifecycle of all facilities, the amount of calculation is enormous. Therefore, cluster-computing and parallel computing are needed on addition to an appropriate data processing framework with *Lambda* architecture as in [10].

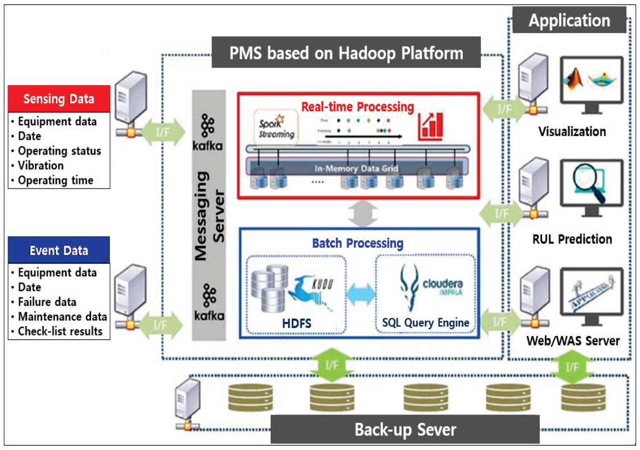


Fig. 2. Implementation of PMS using Hadoop

The *Hadoop* ecosystem, which is a well-known software framework, is used to establish *Lambda* architecture. In the receiving data section, *Kafka*, the messaging server, first collects real-time data and delivers it on demand. Then *Apache Spark*, which is a cluster-computing framework, treats the constantly incoming sensing data such as the pre-processing, extract features, and evaluates the remnant useful life in *Real-time Processing*. In addition, *Spark* executes cluster-computing to train the diagnostic and prognostic models in *Batch Processing*. Moreover, it gathers data to train diagnostic and prognostic models and searches for an appropriate prognostic model for evaluation; *Apache Impala*, which is a query engine that runs on *Apache Hadoop* is used. In addition, *Impala* enables users to issue low-latency SQL queries to store data like the prediction results or the trained models in the Hadoop distributed file system (HDFS) that data management software uses to distribute data to each cluster node.

III. Proposed method

The proposed method finds the hidden degradation stage prior to the prediction process as *2)* in Section II.A. Then, the remnant useful life is predicted using the regression model that is obtained by extracting the predictive factors that have monotonic characteristics at each degradation stage. The challenge remains in how to determine the optimal degradation stages where features can be distinguished. While there are many ways of doing this such as clustering; this paper presents an iterative and intuitive approach. The idea of how to determine the optimal degradation stages is to find the degradation stage that minimizes the classification error. First, to evaluate classification errors for comparison, a set of time series observations is divided into several groups, and the divided groups are given their own labels. To make a classifier, supervised learning is executed by using portion of the data set with the labeled groups. By using the calculated classifying model, the classification error about the rest other than the data used for learning or the entire data is evaluated. For every case of dividing to groups, according to the above procedure, classification errors are obtained in each case. Finally, the case that has the minimum classification error among them is selected as the optimal degradation stage. To minimize the number of cases required to calculate the classification error, this paper uses a binary search technique. For example, when a set of n time series observations is divided into k degradation stages, the number of cases to classify is n combination k, ${}_n\mathrm{C}_k$, whose complexity is $\mathrm{O}(n^k)$. However, the number of calculations can be greatly reduced by applying the binary search technique which has complexity of $\mathrm{O}(n)$, because the complexity of $(n-1)+(n-3)+...+(n-2log_2k+1)$ is $O(log_2k \cdot n) = O(n)$. The binary search technique is an iterative way of dividing the previous one into two sub-parts as the division level increases. As the level increases, the target data set is only divided into two subsets, and each subset is divided again into two sub-subsets in the same manner as shown in Fig. 3. Partitioning is performed until a certain constraint condition is satisfied. For example, the constraint can be the minimum number of observations in a stage or the tolerance of feature values in a stage. This paper set up the minimum number of observations as a constraint. As mentioned above, the binary search technique has lower complexity and the advantage that calculating the classifier is simpler than the multi-classifier for all groups. The detail of the proposed method is described for each training and evaluating the prediction model below.

A. In training

In this section, the training methods of the diagnostic and prognostic models are described. The steps of training the diagnostic and prognostic models are briefly described in *1)* and *2)* in Section II.A,. Until the features extraction step, the procedures of training the diagnostic and prognostic models are the same except for data loading. In the data loading diagnostic step, datasets labeled with type of fault type or normal state are loaded. Meanwhile, a data set of time series observations is loaded in the data loading step of the prognostics. The steps in *1)* and *2)* of Section II.A are described in detail in the follow subsections.

1) Pre-processing: After the data loading step, to extract features, the loaded data passes through pre-processing in

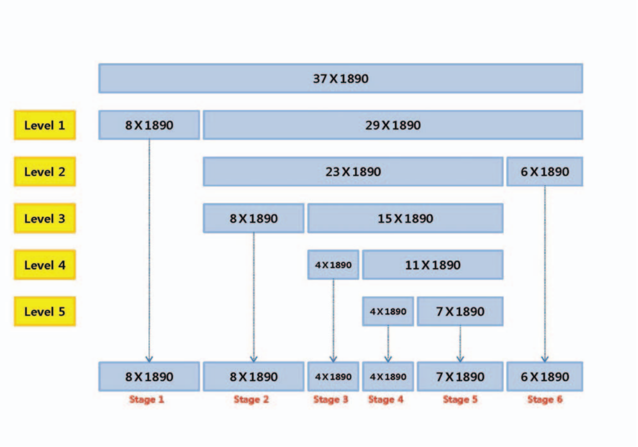


Fig. 3. Determination of degradation stages by using binary search

TABLE I
PARAMETERS OF TIME AND FREQUENCY DOMAINS

No.	Time domain parameters (12)	No.	Frequency domain parameters (9)
P_1	Mean(MN)	P_{13}	$\sum_i^N f_i s(f_i) / \sum_i^N s(f_i)$
P_2	Root Mean Square (RMS)	P_{14}	$\sqrt{\sum_i^N (f_i - P_{13})^2 s(f_i)/(N-1)}$
P_3	Shape Factor (SF)	P_{15}	$\sqrt{\sum_i^N f_i^2 s(f_i) / \sum_i^N s(f_i)}$
P_4	Crest Factor (CF)	P_{16}	$\sqrt{\sum_i^N f_i^4 s(f_i) / \sum_i^N f_i^2 s(f_i)}$
P_5	Skewness (SKEW)	P_{17}	$P_{15}^2 \cdot \sqrt{\sum_i^N s(f_i) / \sum_i^N f_i^4 s(f_i)}$
P_6	Kurtosis (KURT)	P_{18}	P_{13}/P_{14}
P_7	Kurtosis (KURT)	P_{19}	$\sum_i^N (f_i - P_{13})^3 s(f_i)/(P_{14}^3 N)$
P_8	Entropy Estimation Error (EEE)	P_{20}	$\sum_i^N (f_i - P_{13})^4 s(f_i)/(P_{14}^4 N)$
P_9	Lower-Bound of Histogram (LB)	P_{21}	$\sum_i^N \lVert f_i - P_{13} \rVert s(f_i)/(P_{14} N)$
P_{10}	Upper-Bound of Histogram (UB)		
P_{11}	Standard Deviation (STD)		
P_{12}	Normal Negative log-likelihood (NNL)		

the same way for diagnostic and prognostic models. The loaded waveform of the vibration data passes through several signal processing such as Discrete Wavelet Transform (DWT), Hilbert-Huang Transform (HHT) and Fast Fourier Transform (FFT). By using DWT, raw wave data is separated into two other waveforms whose frequency spectrum only includes low or high field. When the level of DWT is l_w, the number of separated waveform at the deepest leaf is 2^{l_w}. Therefore, the summation of the separated waveform at all leafs is $2^{l_w+1}-1$. In addition, each $2^{l_w+1}-1$ waveform has a pair envelop by HHT. Finally, $2(2^{l_w+1}-1)$ separate waveforms are obtained.

2) Feature extraction: For the separate waveform extracted during pre-processing, in the feature extraction step, 12 features of the time domain are calculated from the waveform and nine features of the frequency domain are calculated from the power spectrum obtained by FFT. As with the pre-processing step, this step is performed in the same manner for diagnostic and prognostic models. Table I describes 12 parameters of the time domain and nine parameters of the frequency domain as features, in which f_i is the amount of the i_{th} frequency variable $s(f_i)$ in the frequency domain and N is the number of discretized variables in the frequency domain.

Since the range and variance differ for each feature value, sometimes a feature exists with a large extent, but a clustering property that does not work well. Therefore, in the feature extraction step, scaling is needed to adjust the range and variance of all features, such as the normalization.

3) Feature selection: When training the diagnostic model, various features are extracted from the pre-processing and feature extraction steps to extract as many features as possible. However, there are too many features to classify the groups because a diagnosis classifier leads to poor fault diagnostics performance at high dimensions. Therefore, to reduce the dimensionality of the feature space, effective features are selected using the Distance Evaluation Technique (DET) as in [12]. To select the effective features, all features are prioritized with a DET score. DET is a measure of the sensitivity of each feature vector i expressed as the ratio of the density in a class d_i to the mean of the center distance d'_i of different classes, as d'_i/d_i. In the order of feature vectors with the highest ranking, an appropriate number of features is selected.

4) Classification: In the classification step of training the diagnostic model, various techniques can be used for supervised learning using the group's name as a label. For example, The Support Vector Machine (SVM) with $Gaussian$ kernel which is the Radial Basis Function (RBF) can be used. When more than two groups are used, the one-against-all or one-against-one technique can be used for the multi-class classification.

5) Determination of degradation stages: In the determination of degradation stages step in training the prognostic model, the binary search technique is used to find the degradation stage that minimizes the classification errors. At the division level of the binary search technique, to find a point as a boundary that divides a time series data set into two classes that have the minimum classification error, two-class classifications are independently executed according to each dividing point in the data set. First, after pre-processing and feature extraction in the same manner of diagnosis, as a dividing point, a set of time series observations is divided into two classes that have their own label. In the same manner of diagnosis, the best features are selected by DET for the two divided classes and supervised learning is executed about the classes. In this study, one-to-one SVM is used for supervised learning about two classes in the case of dividing, because there are just two classes. In addition, the RBF kernel is used. By using the rest of training data or all data of the time series observations, the trained classifier is tested. The test is a closed test for calculating the classification error E that is

$$E = \frac{FP + FN}{TP + TN + FP + FN}. \tag{1}$$

In Eq. (1), TP is true positive, TN is true negative, FP is false positive and FN is false negative. After calculating the classification error for all possible cases in a data set of a division level, the point with the minimum classification error is selected as the boundary that divides the set into two subsets. After determining the dividing point, the selected dividing point, the profile of the selected features and two classifiers are stored to estimate the probability of entering to each degradation stage when new data arrives. Since there are two classes such as the left and right sides, there can be two classifiers even though only one is used for classification. The classifier is obtained in a division by making the left or right side true, respectively. The classifier is in the form of a $Gaussian$ distribution that is a linear summation of the normal distributions formed as $N(x, \mu, \sigma)$ with mean μ, variance σ and random variable x, with bias. As the division level increases, the datasets are divided into two smaller subsets until they satisfy the requirement that the number of data in a subset is the preset minimum number. After completing the division, all paths belonging to a degradation stage are stored. For example, the path of Stage 1 in Fig. 3 is $_{lv1}P_{1st}(left)]$ which means the left side of the first section at division level 1. Meanwhile, the path of Stage 3 is $[_{lv1}P_{1st}(right), _{lv2}P_{1st}(left), _{lv3}P_{1st}(right), _{lv4}P_{1st}(left)]$.

6) Extraction of main predictive factors: After determining the degradation stages, the main predictive factors in each degradation stage are selected, respectively. The factors are calculated by correlation coefficient analysis like in Eq. (2), which is used to estimate the regression model.

$$r = \frac{\sum(x_j - \bar{x})}{\sqrt{\sum(x_j - \bar{x})^2 \sum(y_j - \bar{y})^2}}, \quad x_j \in X,\ y_j \in Y, \tag{2}$$

where x_j is an independent variable in X that is a set of values of a feature dimension and y_j is the corresponding time in Y that is a set of the operating time and $\bar{x}$ and $\bar{y}$ are the expectations of X and Y. The feature that has the maximum absolute value of r is selected as the main predictive factor and the other features that have the next highest scores can be used in regression. The profiles of selected predictive factors are stored apart from the selected features in the determination of degradation stage step.

7) Regression: Regression is a statistical analysis and prediction technique that represents a representative model of a causal relationship between two variables as a single line or curve. In this paper, the linear regression model is completed using the operation time data that belongs to a degradation stage created through the regression step in 2) of Section II.A and the main predictive factors of those stages are extracted from the extraction of main predictive factors in 2) of Section II.A. The selected predictive factors and the corresponding operation times are considered independent and dependent variables, respectively. For the dependent variable y, estimating the relationship with one or more independent variables x_j yields

$$y = \alpha + B \cdot X + \epsilon, \tag{3}$$

where α is a constant variable, B is an inclination vector and X is a n dimensional independent variables vector where elements x_j are variable about the selected predictive factors. Ep. (3) is obtained in each degradation stage by applying Least Square Estimation (LSE), whose the coefficients α and B are obtained as

$$\alpha = \bar{y} - B \cdot \bar{X},\ B = \frac{\sum(X_i - \bar{X})(y_i - \bar{y})}{\sum(X_i - \bar{X}) \cdot (X_i - \bar{X})}, \tag{4}$$

where $\bar{x}$ and $\bar{y}$ are the expectations of X and y, respectively, and X_i is a set of predictive factors of the i_{th} observation.

B. In evaluating

This section describes how to evaluate fault diagnosis and predict the remnant useful life when new data comes in. First, new wave data x goes through pre-processing and feature extraction as shown in 1) of Section II.A. After feature extraction, some features of the feature vector are selected according to the previously-stored profile of the selected features, and the current state of the incoming data is determined by using the stored diagnostic classifier. If the data is progressed in a fault, the process of predicting the residual time to the fault proceeds. First, regarding the feature vector obtained from pre-processing and feature extraction, the probabilities of entering each degradation stage are determined by the paths to the degradation stages and the stored binary classifiers at the division level of each path. In detail, let the probability that a feature vector $\mathbf{x}$ will enter the degradation stage k is $g_k(\mathbf{x})$. To calculate $g_k(\mathbf{x})$, $_lp_{i,left}$ and $_lp_{i,right}$ that are the probabilities that x will enter the left or right side of the i_{th} section in the division level l are calculated by the selected profile of features and two binary classifiers which is mentioned in 5) of Section III.A. According to the profile of the features, some features are selected from the feature vector $\mathbf{x}$ and the profiles differ for each division. The probabilities $_lp_{i,left}$ and $_lp_{i,right}$ are calculated from the function values a and b that are the value of the selected features assigned to both classifiers. If both a and b are positive or negative, the probabilities $_lp_{i,left}$ and $_lp_{i,right}$ can be determined as a ratio of the corresponding value to the sum of the two values. In addition, if the sign of the two values a and b differ, the probability of each is set to 0 and 1 as the following items.

- $a, b \leq 0 : {_lp_{i,left}} = \frac{a}{a+b}$, $_lp_{i,right} = \frac{b}{a+b}$
- $a, b \leq 0 : {_lp_{i,left}} = \frac{b}{a+b}$, $_lp_{i,right} = \frac{a}{a+b}$
- $a > 0, b < 0 : {_lp_{i,left}} = 1$, $_lp_{i,right} = 0$
- $a < 0, b > 0 : {_lp_{i,left}} = 0$, $_lp_{i,right} = 1$

Finally, the probability $g_k(\mathbf{x})$ is multiple of all probabilities $_lp_i$ in S_k that the set of probabilities according to the path to the degradation stage k stored from the determination of the degradation stages, such as

$$g_k(\mathbf{x}) = \prod_{l=1}^{M} {_lp_i}, \quad {_lp_i} \in S_k,\ M = n(S_k), \tag{5}$$

where $n(S_k)$ is the number of elements of S_k. Since at all division levels $_lp_{i,left} + {_lp_{i,right}} = 1$, $\sum g_k(\mathbf{x}) = 1$ holds. Furthermore, the estimation of the local remnant useful life $\hat{y}_k$ in the degradation stage k is calculated by the predictive factors $\mathbf{x}'$ and the regression model y_k that is Eq. (3) obtained from the degradation stage k.

Finally, the Remnant Useful Life (RUL) of the observation data x represents the summation multiplies of the already evaluated $g_k(\mathbf{x})$ and estimated stage 1 to K, such as $\hat{y}_k(\mathbf{x}')$

$$RUL = \sum_{1}^{K} g_k(\mathbf{x})\hat{y}_k(\mathbf{x}'). \qquad (6)$$

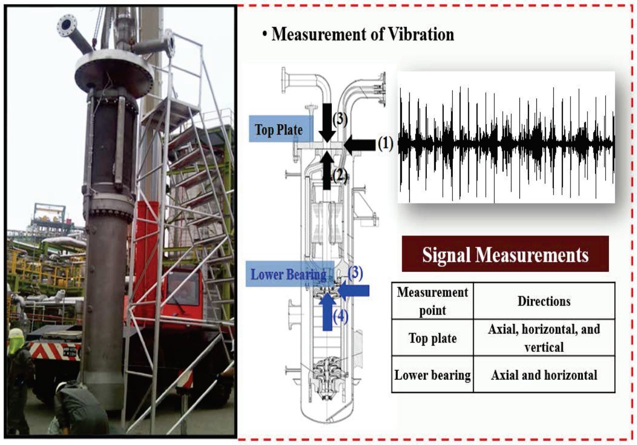
IV. Experiment

The purpose of this experiment can be summarized in three parts, whether the accuracy is improved by dividing the degradation stages according to the proposed method, whether the predictive factors derived from each degradation stage are monotonic and visually confirmed to have sufficient characteristics for new data to find a suitable prediction model in the future, and whether the proposed method is capable of parallel computing.

A. Experiment details

The proposed prognostic method was tested on two subjects, the first is vibration data of a Liquid Natural Gas (LNG) high-pressure pump that operates at the Korea Gas (KOGAS) corporation's plant, another is vibration data of an accelerated experiment of a bearing generated by the NFS I/UCR center of Intelligent Maintenance Systems (IMS). The first subject is an LNG high pressure pump that is a motor-driven vertical pump that rotates at 3,858 rpm, as shown on the left side of Fig. 4. Thirty-seven observations were obtained during the operating time of 5,916 hours. When the pump was overhauled, the reason for the failure was deemed to be a crack in the rotor bar. In the first case, observations were measured by sampling at 12,480 Hz in three directions: axial, horizontal, and vertical on the top plate, as shown on the right side of Fig. 4. Each observation has 4,096 samples. The second subject is experimental equipment that has four bearings installed on a shaft. Here, 2,156 observations were obtained during the operating time of 345.27 hours. As the end of the test-to-failure experiment, the inner race's defect occurred in the third bearing and the roller element defect occurred in the fourth bearing. In this case, observations were measured with sampling at 20 kHz from two directions: the axial and horizontal of four points. Each observation had 20,480 samples and there were three tests.

First, for the LNG high-pressure pump, the prediction accuracy of RUL was measured while changing the number of degradation stages with the Root Mean Square of Error (RMSE), Mean of Absolute value of Error (MAE), and the Maximum Error (ME). Second, for the accelerated experiment of bearing, the prediction accuracy of RUL was measured for various prediction methods such as the proposed method, the linear regression, the regression tree, ENSEMBLE, Support Vector Machine (SVM), and Gaussian Process Regression (GPR), with RMSE, MAE, and ME. Finally, for the accelerated experiment with the bearing, the execution time was measured for each level while changing the number of workers. Windows 10 with an Intel i7 four-core chip was used as the environment and the MATLAB tool's libraries such as DWT, SVM, GPR, ENSEMBLE and so on was used for testing.



Measurement point	Directions
Top plate	Axial, horizontal, and vertical
Lower bearing	Axial and horizontal

Fig. 4. Vibration measurement of LNG high pressure pump

B. Results

1) Results of RUL prediction according to the number of degradation stages: Table II displays the results of RUL prediction according to the number of degradation stages for the KOGAS LNG high-pressure pump. As mentioned in $2)$ in Section II.A and Section III.A, the steps of training the prognostic model were executed about 37 observations. In the step of pre-processing, the level of DWT was three. Since the number of observations is too small to separate training and evaluation data from the original data set, 37 observations used for training are reused in evaluation mentioned in Section III.A. As shown in Table II and Fig. 5, the error decreases as the number of stages increases. This shows that the splitting regression period improves the prediction accuracy.

TABLE II
Results of RUL prediction according to the number of degradation stages for KOGAS case

Num of stages	RMSE	MAE	ME
10	10.99 hours	4.83 hours	42.52 hours
6	107.73 hours	67.11 hours	284.99 hours
5	114.49 hours	80.54 hours	284.99 hours
4	178.66 hours	123.53 hours	486.27 hours
3	370.52 hours	294.61 hours	878.87 hours
2	606.02 hours	506.88 hours	1384.30 hours
1	2071.24 hours	1676.67 hours	4574.77 hours
Total operating time : 5,916 hours			

2) Comparison of results of RUL prediction according to methodologies: For the accelerated experiment of bearing of IMS, the prediction accuracy of RUL was measured for the proposed method and the other various prediction methods. Prior to comparison, it was confirmed whether dividing the

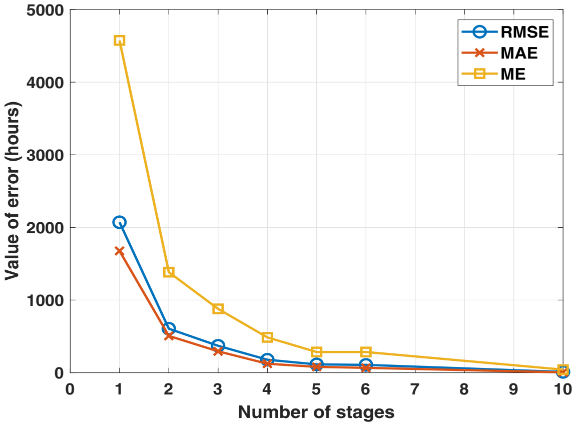


Fig. 5. Error estimation of RUL prediction for LNG high pressure pump

regression interval increases the monotony of the predictive parameters. This experiment was conducted in the same manner as the above KOGAS case with three DWT level. Here, 70 percents of 2,156 observations was used for training and the rest was used for evaluation. Fig. 6 presents the scaled values of the main factors derived for each degradation stage along with the actual remaining time to the fault. In Fig. 6, the black dotted line is a graph of the main predictive factor of the entire regression period. About the main predictive factors of the degradation stages in Fig. 6, the correlation coefficient r of Eq. (2) was compared with those of the main predictive factor in the entire regression period. In addition, the monotonicity of the main predictive factors of degradation stages and those of the entire regression period were compared. The computation of monotonicity uses this formula:

$$monotonicity = \left| \sum_{n=1}^{N} \frac{sgn(x(n+1) - x(n))}{N-1} \right|, \qquad (7)$$

where $x(n)$ represents n_{th} value, and N is the number of values. Table III presents the results of the absolute values of the correlation coefficient and the monotonicity about the main predictive factors of the degradation stages and the entire period that are calculated in each degradation stage. It shows that dividing regression period improves the correlation coefficient and the monotonicity of the main predictive factor.

In addition, the main predictive factors of degradation stages in Fig. 6 have different profile and distribution as shown in Table IV and Fig. 7 because which features are selected as the main predictive factors is determined according to the degradation stage. Table IV presents the identification number of feature of the main predictive factor, mean, and Standard Deviation (STD) of the values in each degradation stage. Fig. 7 is a graph for distribution of the values. When looking at the profile and the distribution of each factor, it seems to have sufficient characteristics for identification.

Furthermore, Fig. 8 shows the results of RUL prediction in the case of the accelerated experiment of bearing as a graph. In the legend of Fig. 8, the actual RUL is the remaining time to the fault and the estimated RUL is the result of prediction by the proposed method. The long gap of the graph in Fig. 8 indicates the downtime.

Table V presents the results of RUL prediction according to several methods such as the proposed method, the linear regression, SVM, GPR, regression tree, and ENSEMBLE for the experimental equipment of IMS. A observation of the experimental equipment of IMS that has eight channels went through the pre-processing and the feature extraction mentioned in *1*) and *2*) in Section III.A, and 5040 features were obtained. About the obtained 5040 feature vectors of 2,156 observations, RUL prediction was performed by the linear regression, SVM, GPR, regression tree, and ENSEMBLE. In addition, experiments in which Principal Component Analysis (PCA) was applied to these methods were additionally performed. Here, 70 percents of 2,156 observations was used for training and the rest was used for evaluation. The proposed method divides the entire regression period into six degradation stages. Otherwise, the other regression method used did not divide the regression section. As shown in Table V, the proposed method mostly yields better accuracy than the other methods except for regression tree and ENSEMBLE. Compared to ENSEMBLE, RMSE and MAE are lower, but ME is higher. Otherwise, compared to regression tree, RMSE and MAE are higher, but ME is lower. Since the propose method used simple linear regression in each degradation stage, if a different regression method is used or the number of stages increases, there is room for improvement in accuracy.

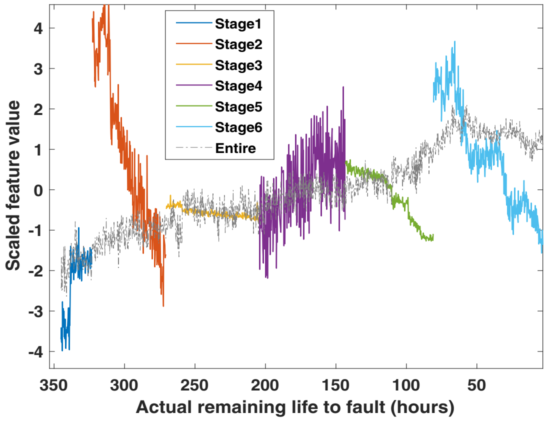


Fig. 6. Distribution of values of main predictive parameters

3) Comparison of computation time according to the number of workers: For RUL prediction about the accelerated experiment of IMS, the execution time was measured for each level while changing the number of workers. Table VI is the results of computation time according to the number of workers, in which the unit is seconds. As shown in Table VI, the computation time decreases depending on the number of workers. In addition, the slope of the decrease seems to reduce as the number of workers increases.

TABLE III
COMPUTING THE CORRELATION COEFFICIENT AND THE MONOTONICITY

	$\|r\|$		monotonicity	
Section	divided	original	divided	original
Stage 1	.81	.29	.31	.07
Stage 2	.94	.46	.90	.12
Stage 3	.84	.17	.22	.02
Stage 4	.60	.41	.00	.02
Stage 5	.96	.68	.95	.14
Stage 6	.94	.37	.30	.01

TABLE IV
PROFILE OF THE MAIN PREDICTIVE FACTOR

Degradation stage	Num of feature	Mean	STD
Stage 1	98	-2.48	0.85
Stage 2	940	0.77	1.92
Stage 3	4422	-0.57	0.11
Stage 4	4675	0.07	0.89
Stage 5	4283	-0.27	0.63
Stage 6	3830	0.68	1.27

TABLE V
COMPARISON OF RUL PREDICTIONS BY METHODOLOGY FOR IMS CASE

Used method	RMSE	MAE	ME
Proposed Method	9.23 hours	6.17 hours	62.46 hours
Linear Regression	34.33 hours	28.10 hours	98.48 hours
Regression Tree	8.36 hours	5.37 hours	89.57 hours
SVM	19.84 hours	10.47 hours	192.73 hours
ENSEMBLE	10.00 hours	8.42 hours	33.86 hours
GPR	23.37 hours	10.87 hours	161.25 hours
Linear Regression (PCA)	84.32 hours	12.53 hours	2118.10 hours
Regression Tree (PCA)	14.32 hours	7.76 hours	145.43 hours
SVM (PCA)	97.07 hours	83.36 hours	190.69 hours
ENSEMBLE (PCA)	15.16 hours	11.93 hours	75.06 hours
GPR (PCA)	21.67 hours	8.99 hours	455.79 hours
Total operating time : 345.27 hours			

TABLE VI
COMPUTATION TIME BY PARALLEL COMPUTING

	Num of workers			
	Single core	Two cores	Three cores	Four cores
Division Level 1	2.96×10^2 sec	1.86×10^2 sec	1.42×10^2 sec	1.12×10^2 sec
Division Level 2	1.72×10^2 sec	1.10×10^2 sec	83.7 sec	69.3 sec
Division Level 3	71.2 sec	46.4 sec	34.2 sec	30.3 sec
Division Level 4	28.3 sec	19.2 sec	14.2 sec	12.4 sec
Etc.	34.8 sec	27.4 sec	20.5 sec	21.5 sec
Total	6.02×10^2 sec	3.88×10^2 sec	2.95×10^2 sec	2.45×10^2 sec
Difference (single - multi-cores)	-	2.14×10^2 sec	3.08×10^2 sec	3.57×10^2 sec
Ratio (n cores/single)	100%	64.49%	48.90%	40.77%

V. CONCLUSION

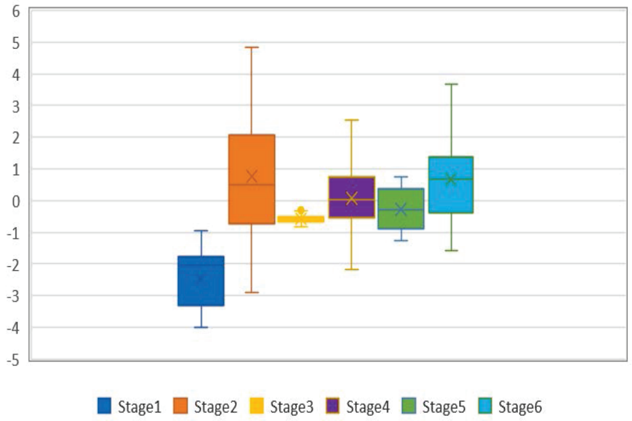


Fig. 7. Distribution of the main predictive factors

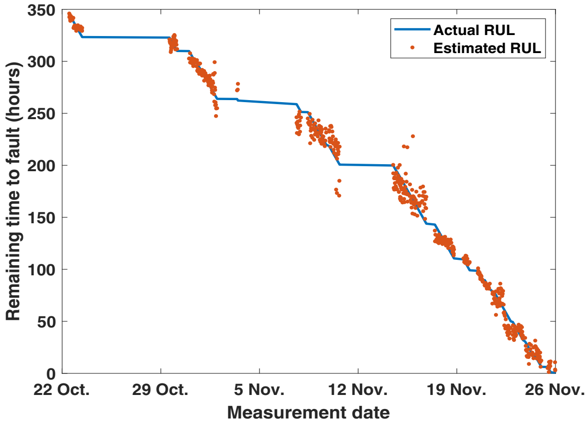


Fig. 8. Results of RUL prediction in IMS case

Through experimentation, the proposed method is shown to improve the monotonicity of the predictive parameters and help improve the accuracy of the predictive model. Furthermore, the measured maximum error can be used as tolerance when determining the timing of the breakdown maintenance. In addition, the profile and distribution of predictive factors in each degradation stage can be used to find a suitable prediction model as the characteristics of the prediction model, when new observations arrive. As shown in Fig. 8, it can be inferred that the error increases at the boundary at which the interval is separated. This is expected to be a similar boundary problem to many discretizing methods. In the future, a study on the boundary problem and a study of the findings for a predictive model will be conducted using the distribution characteristics of the prediction factors.

REFERENCES

[1] K. TSUI, N. Chen, Q.Zhou, Y. Hai and W. Wang, *Prognostics and Health Management: A Review on Data Driven Approaches. Mathematical Problems in Engineering*. Hindawi Publishing Corporation Mathematical Problems in Engineering Volume, ID 793161, 2015.

[2] *Condition monitoring and diagnostics of machines vibration condition monitoring*, Part1. General procedures. ISO, 13373-1, 2002.

[3] S. Dong and T. Luo, *Bearing degradation process prediction based on the PCA and optimized LS-SVM model.* Measurement, 9(3143-3152), 2013.

[4] Z. Zhang, X. Gu, Y. Xie, Z. Wang, Z. Wang and K. Chakrabarty, *Diagnostic system based on support vector machines for board-level functional diagnosis*. Proceedings, IEEE European test symposium(ETS) 1-6, 2012.

[5] D. Dabrowski, *Condition monitoring of planetary gearbox by hardware implementation of artificial neural networks*. Measurement, vol.91, pp. 295-308, 2016.

[6] A. Prasad, L. Iverson and A. Liaw, *Newer Classification and Regression Tree Techniques: Bagging and Random Forests for Ecological Prediction*. ECOSYSTEMS, pp. 181-199, 2016.

[7] T. Fan and W. Zhao, *Ensemble of model-based and data-driven prognostic approaches for reliability prediction*. 2017 Prognostics and System Health Management Conference (PHM-Harbin), pp. 1-6, 2017.

[8] H. Kim and A. Tan, *Bearing fault prognosis based on health state probability estimation*. Expert System Application, 5 (5200-5213), 2012

[9] R. Singleton, E. Strangas and S. Aviyente, *Discovering the hidden health states in bearing vibration signals for fault prognosis*. 40th Annual Conference of the IEEE Industrial Electronics Society(IECON), Dallas, TX, pp. 3438-3444, 2014.

[10] A. Batyuk and V. Voityshyn, *Streaming Process Discovery for Lambda Architecture-Based Process Monitoring Platform*. 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, pp. 298-301, 2018.

[11] H. Kim, S. Hwang, A. Tan, J. Mathew and B. Choi, *Integrated approach for diagnostics and prognostics of HP LNG pump based on health state probability estimation*. Journal of Mechanical Science and Technology, 26 (11), pp. 3571-3585, 2012.

[12] H. Kim and T. Jeon, *Evaluation of feature extraction techniques for intelligent fault diagnostics of ressure LNG pump*. Proceedings of the 10th World Congress on Engineering Asset Management, pp. 553-562, 2015.