

RUL prediction based on a new similarity-instance based approach

Racha Khelif, Simon Malinowski, Brigitte Morello, Noureddine Zerhouni.
FEMTO - ST Institute, 24 rue Alain Savary, 25000 Besançon, France
Email: racha.khelif@femto-st.fr

Abstract— Prognostics is a major activity of Condition-Based Maintenance (CBM) in many industrial domains where safety, reliability and cost reduction are of high importance. The main objective of prognostics is to provide an estimation of the Remaining Useful Life (RUL) of a degrading component/ system, i.e. to predict the time after which a component/system will no longer be able to meet its operating requirements. RUL prediction is a challenging task that requires special attention when modeling the prognostics approach. This paper proposes a RUL prediction approach based on Instance Based Learning (IBL) with an emphasis on the retrieval step of the latter. The method is divided into two steps: an offline and an online step. The purpose of the offline phase is to learn a model that represents the degradation behavior of a critical component using a history of run-to-failure data. This modeling step enables us to construct a library of health indicators (HI) from run-to-failure data. These HI's are then used online to estimate the RUL of components at an early stage of life, by comparing their HI's to the ones of the library built in the offline phase. Our approach makes use of a new similarity measure between HIs. The proposed approach was tested on real turbofan data set and showed good performance compared to other existing approaches.

Keywords—*prognostics; RUL prediction; Instance Based Learning; similarity measures.*

I. INTRODUCTION

In industrial world, breakdowns can have huge impact on the system's performance and may lead to disastrous situations where both human's life and systems are put in jeopardy. In order to avoid such situations, maintenance strategies shifted from traditional *fix-it-when-it-breaks* to *predict-prevent* maintenance schemes in the form of Condition-Based Maintenance (CBM) first and Prognostics and Health Management (PHM) later. CBM makes use of machinery run-time data to determine the machinery condition and hence its current fault/failure condition, which can be used to schedule repair and maintenance prior to breakdown. PHM refers specifically to the phase involved with predicting future behavior, including remaining useful life, in terms of current operating state and the scheduling of required maintenance action to maintain systems health [1]. The core of PHM process is prognostics.

Prognostics as defined by the international standard organization is “the estimation of time to failure and risk for one or more existing future failure modes” [2]. It is thereby a promising activity that aims at reducing the costly systems downtime and maximizing the useful operating life of

equipment by avoiding unnecessary replacements and thus increases the system's reliability.

In general, prognostics approaches can be categorized under three categories; model-based, data-driven and hybrid approaches [3, 4]. Model-based prognostics build a mathematical model that is directly related to the physical process that influences the health of the component. The model can be described by dynamic systems such as nonlinear equations, differential equations, state space models and solved accordingly. These approaches provide the most accurate and precise tool for prognostics applications. However, it is usually hard to construct physical models for complex systems, because we do not have the necessary knowledge of the physical degradation phenomena. Model-based approaches are application specific and cannot be generalized [5, 6]. Data-driven approaches, on the other hand, attempt to derive models from routinely collected monitoring data instead of building the models based on comprehensive system physics and human expertise. They are based on statistical and learning techniques built upon the historical records and produce prediction output directly in terms of condition monitored data. Data-driven approaches are easier to implement compared to model-based approaches but give less accurate results. They hence offer a trade-off between precision, complexity and implementation cost. Hybrid approaches combine both previously described approaches. Usually for these approaches, the physical model is validated offline and data-driven techniques are then used to update the model's parameters to increase the accuracy.

Thanks to the easiness of implementation and their ability to learn models from historical data, data-driven approaches offer an appealing tool to perform prognostics. Artificial Neural Networks (ANN) is the most widely used data-driven approach for prognostics due to its accuracy [7, 8]. However, neural networks lack the ability to explain and interpret their results. They are black-box systems where the relationship between the inputs and outputs is not known. Statistical approaches are also popular in the field of prognostics. However, they are usually based on assumptions that are not suitable for real world industrial applications [9, 10]. In this paper, instance based learning (IBL) is developed as a machinery prognostics approach. This method is able to give good prediction results without any prior knowledge about the components characteristics or the degradation model. The degradation model is directly learned from the data. We propose to enhance the retrieval step of IBL by developing a

new similarity measure algorithm that takes into account the whole history data while giving more importance to late observations. This similarity measure is used to retrieve the K most similar instances, represented by health indicator trajectories in the library, and locate the time position of the new experience (instance) in order to predict its RUL. Our approach can be used in any application following these hypotheses:

- Run to failure historical data available.
- The history of training data starts with components in good health and ends when the components reach failure condition that is the end of life criteria.
- Test components are assumed to exhibit the same degradation behavior.
- Variable contexts were not addressed in this work.
- Observation signals i.e. sensory data should capture the health status evolution through time.

This paper is organized as follows. Section 2 presents related work, section 3 outlines the proposed method for RUL prediction, section 4 describes the experimental verification and results and finally, section 5 concludes the paper.

II. RELATED WORK

IBL application in prognostics is relatively new (beginning of the 2000s) [11-15]. IBL re-utilizes the experience gained from solving similar instances to solve new problem instances. Hence, in such approaches, learning occurs as a natural outcome of solving problems; when a problem is successfully solved, the experience is retained in order to solve similar problems in the future. The idea is to exploit the only available knowledge, by learning from experience. This type of approaches for RUL prediction follows three main steps; instance retrieval from the training data set, prediction through local model and aggregation of local predictions [11].

In [12], Xue et al. proposed a fuzzy instance based prognostic approach for RUL prediction. The approach starts by building local fuzzy models for test engines. The fuzzy model defines a cluster of peers in which each of these peers is a similar instance to the given engine with comparable operational characteristics. The final RUL estimate of the given engine was obtained by aggregating the RULs of similar training instance via a similarity weighted average. Mosallam et al. in [13] modeled the monitoring data as trajectories that characterize the lifecycle of the component using principal component analysis. For the problem component, the most similar trajectory is retrieved based on the Euclidean distance, and its RUL is directly considered as the RUL for the new component. Wang et al. [14] created a library of degradation models for PHM 2008 challenge turbofan data using linear regression and then used those models to construct health indicators for each engine. Final RUL for the test engines were estimated through a weighted sum of RULs of the most similar instances. Zio et al. [15] proposed a similarity-based approach for prognostics using a fuzzy point-wise similarity defined for degradation trajectories. The distance score between two trajectories based on a fuzzy membership function that maps the difference between trajectories

elements into membership. Weights to the distance score are assigned such that the smaller the distance is the larger the weight given to the corresponding trajectory. RUL is obtained as a weighted sum of RULs of the similar instances. Ramasso et al. in [11] proposed a method that jointly predicts observations (continuous states) and health states (discrete states) in order to predict the remaining useful life. Instead of aggregating RULs of the most similar instances, the observation trajectories are aggregated to predict the future observations. Those observations are classified as health states and RUL is predicted as the time transition from the degrading to fault state. The retrieval phase of the algorithm is based on a Euclidean distance measure where only the last block of the test observation trajectory is considered.

Instance retrieval step is of high importance in the IBL approach as retrieving inappropriate instances will lead to completely wrong predictions. It is based on measuring the distance between the training data and test data. This distance is directly used to select the similar instances from the training data or is converted to a similarity score that determines the similar instances.

In most of the discussed IBL-prognostics approaches, the information contained in the up-to-date condition monitoring data is not fully used to decide upon the similarity. At the retrieval step, the similarity is either set by using a vector of attributes that characterizes the instance instead of the instance data [12], or by using a feature vector of only the latest measurements [11, 15]. To address this, [13, 14] considered the complete history of observations with a point-wise Euclidean distance between the test and train trajectories. These latter truly fully used the up-to-date condition monitoring data. However, all observations had the same influence on the distance measurement while it is known that late observations are more critical since fault is most likely to occur by the end of life of the component and thus should be accorded more importance (weight). In this paper a new similarity measure is proposed to improve the retrieval step of IBL and thus improve the prediction performance. The algorithm makes use of the whole history of data while giving more weight to late observations.

III. RUL PREDICTION BASED ON IBL

The proposed approach as described in Fig. 1, predicts the remaining useful life before failure of components based on already lived experiences. The idea is to match the test component's current degradation behavior represented by a health indicator trajectory to a library of training HI's and then use only the most similar ones to directly estimate the RUL. This library of HI's is constructed offline by extracting smooth monotonic signals from sensory data. These generated trends represent the progression of components health status and model the degradation evolution. They are thus called health indicators. To do so, linear regression was applied on the sensory data. The regression model learnt offline is then used online on the test sensory data to generate test health indicators in the same manner. The test HI is then compared to the library of training HI's by means of the new similarity measure.

proposed in the paper. To predict the RUL, only those training instances that best matched the test instance are kept.

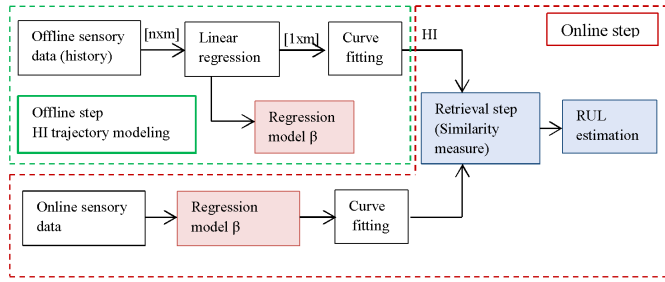


Fig. 1. Overall approach scheme.

A. Health indicator trajectory modeling

Multi-dimensional sensory data are fused to better visualize and assess the degradation in the component's performance. In [13], the authors used principal component analysis to reduce the high-dimensional space of features into a single-dimensional space. The generated signal was called a health index. Benkedjough et al. in [16] applied ISOMAP for the same purpose. Despite both techniques being well established dimensionality reduction tools, the generated reduced signal, although considered as a health indicator by the authors in both [13] and [16] do not faithfully model the degradation process and thus cannot be considered as a health indicator. The signal only provides a compact lower dimensional representation of features obtained using fusion of data tools. Some of the selected features during the fusion process may not be appropriate for characterizing the health status of the component. In addition fusing without considering the health condition of the component may result in an irrelevant health indicator. Both papers [13] and [16] generate health indicators without any consideration on the component health during the fusion process and no assumptions on the degradation evolution. In addition, no model is learned behind. The same reduction steps are followed for the test data which makes both approaches slow and less appropriate for online applications. In [17], logistic regression was used to convert the multi-dimensional features into HI's with the assumption that the healthy state will produce an output, HI, equals to one and the faulty state will give an output equals to zero. As found by Wang et al. in [14], logistic regression will distort the original degradation patterns of the system. The logistic curve is flat when the values approaches 0 or 1; therefore the HI's produced by logistic regression are less sensitive near the early and end life of the system than in the middle life, which may lead to larger prediction error when extrapolating the HI curve. To preserve the original patterns in the signal/features, the authors proposed a linear regression model used as a performance assessment [14]:

$$y = \alpha + \beta_1 x_1 + \dots + \beta_N x_N + \varepsilon \quad . (1)$$

Where $x=(x_1, x_2, \dots, x_N)$ is the observation vector, y is the generated health indicator, (α, β_i) are the regression model parameters learned offline and used both offline and online to convert sensory data into health indicators. The proposed approach is used in this paper to construct a library of health indicators. The regression model is trained by taking data from healthy and near failure conditions components and assigning the corresponding outputs with '1' and '0' respectively.

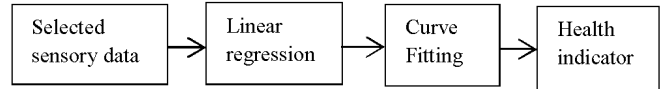


Fig. 2. Health indicator modeling.

As shown in Fig. 2, these health indicators are further processed to produce a smoother output. Fig. 3 presents the health indicators obtained after curve fitting.

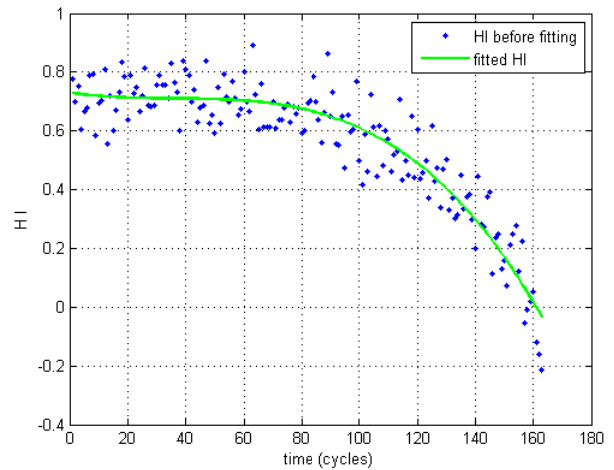


Fig. 3. Health indicator extracted from the selected sensory data by using linear regression and curve fitting.

B. Similarity Measure

In order to obtain an estimation of the remaining useful life of a test instance only the most similar instances are used. This is done by conducting a similarity test between train instances i.e. instances with defined end of life values and represented by their HI's trajectories and the test instance i.e. an instance that did not reach the end of life criteria yet and represented by a HI as well. The aim of the similarity measure is to attribute a similarity score to each train instance and locate the present time, time to start the prediction, on the train instance trajectory time axis. The proposed similarity measure takes into account the whole history of the test instance while giving more importance to late cycles since it is more likely to observe failure at the late ages of life of the equipment.

```

- Divide the test trajectory into N-windows.
   $W_{k, \text{test}} = \{y_{\text{test}}\}_t^{t+T_K}$ ,  $k=1 \dots N$ 
- Divide the train trajectories into M overlapping windows
   $W_{k, \text{train}} = \{y_{\text{train}}\}_t^{t+T_K}$ ,  $k=1 \dots M$ 
- Set ix=1 % initialize the scan starting point.
  For i=1 : N
    For j=ix :M % scan the test window over the train trajectory.
      %  $D_{ij}$  is the Euclidean distance between
      % test window « i » and train window « j ».
       $sim_{i,j} = \exp\left(\frac{-D_{ij}}{\lambda}\right)$ 
      %  $\lambda$  is a relaxing factor set to enlarge the
      % similarity constraint.

      If  $sim_{i,j} >$  threshold
        Increment the number of similar windows.
        ix=i % start the next scan at the current train window.
      end
    end
  end

  % w(i) is the weight associated to the
  % test window i.
  % N.S.W is the number of
  % similar windows.
  % N.W is the total number of
  % windows in the train trajectory.
  % S.C is the similarity score.
   $S.C = \frac{N.S.W}{N.W} \cdot \sum_{i=1}^N w(i) \cdot \max(sim_{i,j})$ 

```

Algorithm 1. Similarity measure algorithm.

Algorithm 1 describes the similarity measure algorithm. It starts by dividing the test HI trajectory into N windows. Each window in the test trajectory is scanned over the train HI trajectory. The first similar window detects the start of similarity and the last similar window indicates the end of similarity and thus the current position on the train HI trajectory. Each window in the test trajectory is given a weight in a way that favors late windows. This similarity measure will evaluate the similarity between two trajectories and produce a score that is equal to “one” in the case of a perfect match and “zero” for completely non-similar trajectories. Since it is very unlikely to observe the exact same test instance in the train instance library, an additional factor; λ , called a relaxing factor is proposed. This factor allows a margin of difference between train and test trajectories. The bigger λ is the larger the margin. In prognostics, early predictions are preferable than late ones. For this reason long trajectories are penalized by dividing the similarity score by the total number of windows in the train trajectory. This number is larger for longer trajectories.

Fig. 4 gives an example of the best and worst matches for a given test instance.

C. RUL estimation

To predict the remaining useful life of a test instance, the train instances having similar degradation patterns depicted by the HI trajectories are used as a basis for RUL estimation. Each HI in the library can produce one estimated RUL for the test instance:

$$RUL_i = EOL_i - t_{pi} \quad . (2)$$

Where EOL_i is the end of life of train instance “i” and “ t_{pi} ” is the present time location on the train instance trajectory defined by the similarity measure as described in Fig. 5.

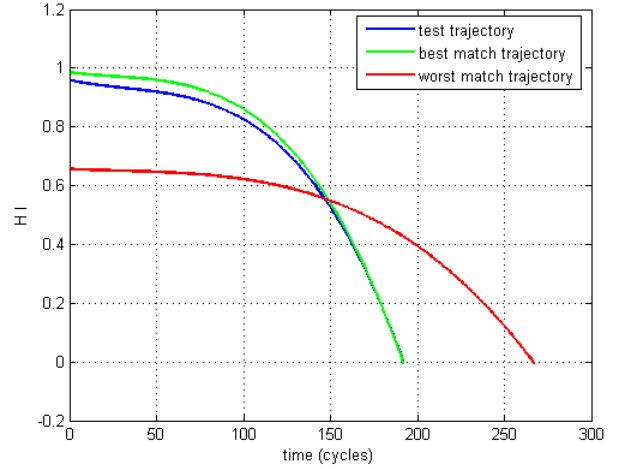


Fig. 4. Example of a test trajectory best and worst matches.

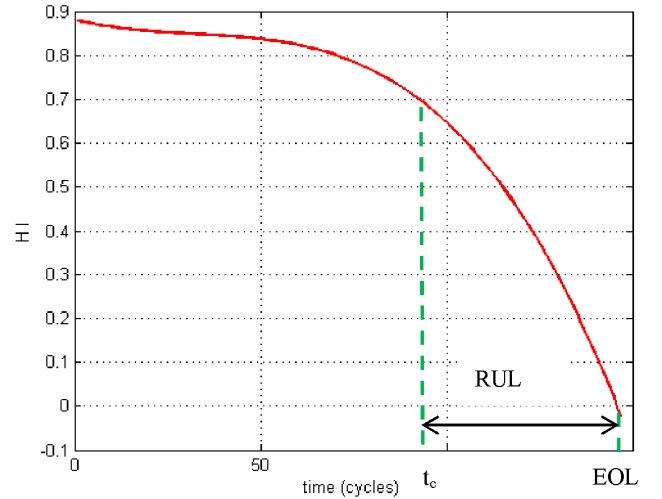


Fig. 5. RUL estimation.

The final RUL of the test instance can be estimated through a weighted sum of the obtained RULs of the k-nearest train instances, that is the k trains instances with the highest similarity scores.

$$RUL = \sum_{i=1}^k w_i \cdot RUL_i \quad . (3)$$

$$w_i = \frac{S.C_i}{\sum_{i=1}^k S.C_i} \quad . (4)$$

Weights assigned to the train instance depend on the similarity degree between the test and train instances whereas k, number of selected neighbors, is application dependent.

IV. APPLICATION AND RESULTS

A. Turbofan data

In order to assess the capability of the proposed approach for RUL prediction, the challenge data set of diagnostics and prognostics of machine faults from the first international conference of PHM [18] was considered. This dataset is a multiple multivariate time series signals (26 features) contaminated with measurement noise. Each time series is from a different engine of the same complex system. Each engine is operating normally at the start and develops a fault at some point leading to system failure.

In this paper, the first experiment of the challenge was used. In order to train and test the prognostic approach, the text files “train_FD001” and “test_FD001” were used. Both files are composed of 100 times series representing the degradation behavior of 100 engines. It should be noted that in the test file, the time series ends up some time prior to failure. The objective is to predict the number of remaining operational cycles before failure in the testing dataset. Each instance either for test or train consists of 26 variables. However, only 5 were used accordingly to [11].

B. Performance Evaluation

To assess the prediction, we define the prediction error for a given prediction by (5):

$$E = \text{Actual RUL} - \text{Predicted RUL} \quad (5)$$

In this paper, an interval $I = [-10, +13]$ set by [11] was considered to assess the performance. A prediction is considered correct if its corresponding error falls within the interval I . This interval is considered a severe condition compared to the literature [19].

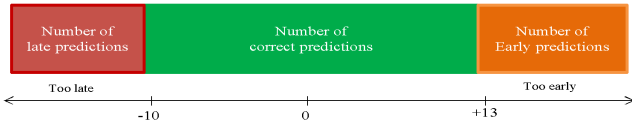


Fig. 6. Metrics of performance assessment [11].

C. Prognostic results

For all the testing process the following parameters values were used; $W=30$, the window size, $\lambda=0.25$, relaxing factor, and $k=7$, number of used neighbors. This value was chosen since it gave better results compared to other values.

To evaluate the performance of the approach, leave one out cross validation was first performed on the train_FD001 file at different critical times t_c , time to start the prediction. 99 instances were used as the training dataset and the remaining instance was used as the testing dataset. t_c was adapted according to the trajectory’s length. It is expressed as a percentage of the whole trajectory’s length. Fig. 7 shows that the larger t_c the better the prediction performance. Fig. 8 illustrates the evolution of the error i.e. the difference between the estimated and real RUL at each time step. For this example

a good estimate of the RUL is obtained at $t_c=0.25\%$ length of trajectory that is $t=47$ cycles and 141 cycle ahead.

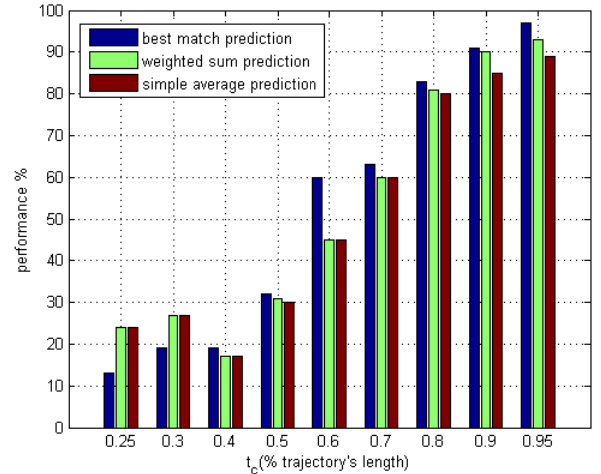


Fig. 7. RUL prediction performance for different t_c values.

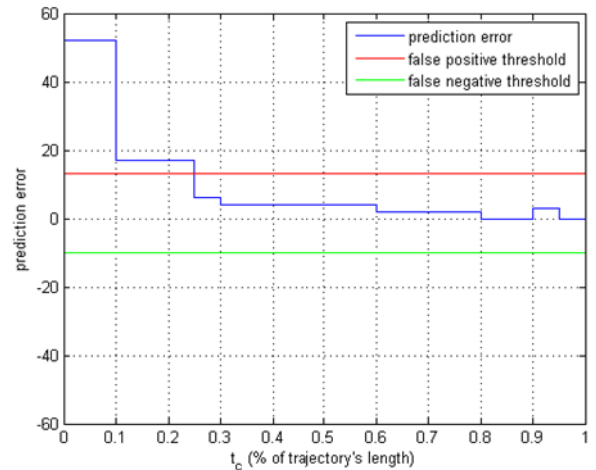


Fig. 8. Error evolution between real and estimated RUL.

A second test was performed considering the same challenge however, this time the text file “test_FD001” was used as the testing dataset. The file is composed of 100 instances with unknown RULs. The actual RUL values are found in the text file “RUL_FD001”. Table I summarizes the approach’s performance on the whole test units along with results obtained by other approaches. For comparison reasons, the method proposed by Wang et al. [14] was implemented and tested on the same test dataset considering both the sensor selection proposed by the authors and the one proposed by [11]. The number of neighbors, k , was set equal to 7 as in our approach. Using the sensor selection proposed in [11] gave better results. In [7] unfortunately, the prognostic model was only tested on the first 15 elements from the test file, an adequate comparison is not possible.

Compared to other approaches, the proposed similarity instance based approach for RUL prediction had a

better correct prediction rate. The new proposed similarity measure enhanced the number of correct predictions by 4%.

TABLE I. PREDICTION PERFORMANCE EVALUATION.

approach	Correct %	Early %	Late %	remark
Current approach	54	18	28	Tested on 100 test units.
[11] Ramasso et al. (2012)	53	36	11	Tested on 100 test units.
[7] Javed et al. (2013)	53	27	20	Tested only on 15 test units.
[14] Wang et al. (2008)	44	19	37	Sensor selection proposed by the authors.
[14] Wang et al. (2008)	50	19	31	Sensor selection used in this paper.

V. CONCLUSION

This paper presented a prognostics approach based on instance based learning with a new similarity measure. Linear regression was used to model the degradation behavior of components as health indicator trajectories. Online, the piece HI trajectory is compared to the library of HI's constructed offline to directly estimate the RUL. The method was demonstrated on the challenge data set of diagnostics and prognostics of machine faults which is known for its complexity and nonlinear nature. Results showed that the new similarity algorithm was able to enhance the retrieval step of IBL and thus RUL prediction performance. Our future work will focus on the preprocessing step by investigating features that might better discriminate instances.

REFERENCES

- [1] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess and B. Wu, "Intelligent fault diagnosis and prognosis for engineering systems," John Wiley & Sons, Inc. 2006.
- [2] ISO13381-1, Condition monitoring and diagnostics of machines - prognostics - Part1: General guidelines. International Standard, ISO,2004.
- [3] J. Lee, F.Wu, W. Zhao, M. Ghaffari, L. Liao and D. Siegel, "Prognosticsandhealthmanagementdesignforrotary machinerysystems—Reviews,methodologyandapplications," Mechanical Systems and Signal Processing, vol. 42, pp. 314–334, 2013.
- [4] Y. Peng, M. Dong and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: a review," Int J Adv Manuf Technol, vol.50, pp. 297–313, 2010.
- [5] Y. LI, T. R. Kurfess, and S. Y. liang, "Stochastic prognostics for rolling element bearings," Mechanical Systems and Signal Processing, vol. 14, no. 5, pp. 747-762, 2000.
- [6] Oppenheimer, C.H., and Loparo, K.A., 2002, "Physically based diagnosis and prognosis of cracked rotor shafts," Component and Systems Diagnostics, Prognostics, and Health Management II, Vol. 4733, Bellingham, pp. 122-132.
- [7] K. Javed, R. Gouriveau and N. Zerhouni, "Novel failure prognostics approach with dynamic thresholds for machine

- degradation," in Proceeding of the IEEE Industrial Electronics Society, IECON'13, 2013.
- [8] G.Yu , H. Qiu, D. Djurdjanovic and Jay Lee, "Feature signature prediction of a boring process using neural network modeling with confidence bounds," Int J Adv Manuf Technol, vol. 30, pp. 614–621, 2006.
- [9] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni and G. Tripot, "A Data-Driven Failure Prognostics Method based on Mixture of Gaussians Hidden Markov Models," Reliability, IEEE Transaction, vol. 66, no. 2, pp. 491 – 503.
- [10] M. Dong, D. He, P. Banerjee and J. Keller, "Equipment health diagnosis and prognosis using hidden semi-Markov models," Int J Adv Manuf Technol, vol. 30, pp. 738–749, 2006.
- [11] E.Ramasso, M. Rombaut, and N.Zerhouni," Joint prediction of observations and states in time-series: a partially supervised prognostics approach based on belief functions and KNN," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, pp. 1-14, 2012.
- [12] F. Xue, P. Bonissone, A.Varma,W. Yan, N. Eklund, K. Goebel, "An Instance-Based Method for Remaining Useful Life Estimation for Aircraft Engines," J Fail. Anal. and Preven.vol.8, pp. 199–206, 2008.
- [13] A. Mosallam, K. Medjaher and N. Zerhouni, "Bayesian Approach for Remaining Useful Life Prediction," Chemical Engineering Transactions, vol.33, pp. 139-144, 2013.
- [14] T. Wang, J. Yu, D. Siegel, and Jay Lee, "a similarity-based prognostics approach for remaining useful life estimation of engineered systems," in Proceeding of the IEEE Int. Conf. on Prognostics and Health Management, pp. 1-6, 2008.
- [15] E. Zio, F.D. Maio and M.Stasi, "a data-driven approach for predicting failure scenarios in nuclear systems," Annals of Nuclear Energy, vol. 37, no.4, pp. 482-491, 2010.
- [16] T. Benkedjouh, K. Medjaher, N. Zerhouni and S.Rechak, "Remaining useful life estimation based on nonlinear feature reduction and support vector regression," vol. 26, pp. 1751–1760, 2013.
- [17] J. Van, M. Koc, and J. Lee, "A prognostic algorithm for machine performance assessment and its application," Production Planning Control, Vol. 15, No.8, 2004, pp. 796-801.
- [18] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in porceeding of the Int. Conf. on Prognostics and Health Management, Denver, CO, USA, pp. 1–9, 2008.
- [19] K. Goebel and P. Bonissone, "Prognostic information fusion for constant load systems," in 7th Annual Conference on Information Fusion, pp. 1247–1255, 2003.