

# Unsupervised Kernel Regression Modeling Approach for RUL Prediction

Racha Khelif, Simon Malinowski, Brigitte Chebel - Morello and Noureddine Zerhouni

FEMTO - ST Institute, 24 rue Alain Savary, Besanon, 25000, France

*racha.khelif@femto-st.fr*

## ABSTRACT

Recently, Prognostics and Health Management (PHM) has gained attention from the industrial world since it aims at increasing safety and reliability while reducing the maintenance cost by providing a useful prediction about the Remaining Useful Life (RUL) of critical components/system. In this paper, an Instance-Based Learning (IBL) approach is proposed for RUL prediction. Instances correspond to trajectories representing run-to-failure data of a component. These trajectories are modeled using Unsupervised Kernel Regression (UKR). A historical database is used to learn a UKR model for each training unit. These models fuse the run-to-failure data into a single feature that evolves over time and hence allow the construction of a library of instances. When unseen sensory data arrive, the learned UKR models are used to construct the test degradation trajectories. RUL is deduced by comparing the test degradation trajectory to the library of instance. Only the most similar train instances are kept for RUL prediction. The proposed approach was tested and compared to approaches that apply linear regression and PCA to model the library of instances. Results highlight the benefit of using UK compared to other approaches.

## 1. INTRODUCTION

Industrial systems are becoming more and more complex. Maintaining them is thus becoming costly and difficult. Prognostics and Health Management aims at reducing such maintenance costs while increasing systems security and reliability. In a PHM process, prognostic is a central activity where the common task is to predict the remaining life before failure of the examined equipment. As defined by the 2004 International Organization for Standardization (ISO, 2004), prognostics is an estimation of time to failure and risks of one or more existing or future failure modes.

Racha Khelif et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In general prognostic approaches can be classified into three classes: model-based, data-driven and hybrid approaches.

Model-based approaches study and model the degradation of the component by relying on the physical laws describing the damage propagation. This type of approaches gives accurate prognostics results. However, building such models for complex systems is a hard task especially in the absence of an adequate knowledge about the physical degradation phenomena.

Data-driven approaches, on the other hand, offer an appealing alternative to perform prognostics due to their ability to learn models from historical data. They are based on statistical and learning techniques and give the prediction output directly from the condition monitoring data. They offer a tradeoff between precision, complexity and implementation costs. Unlike model-based approaches which are application specific, data-driven approaches have a wider framework of applications. They can be applied on different systems as long as the assumptions related to the implemented approach are satisfied. However, the prediction outcome resulting from such approaches is less accurate.

Hybrid approaches are a combination of both data-driven and model-based approaches. The combined approach inherits the merits of both approaches while reducing the associated inconveniences. To increase the accuracy and the prediction performance, the physical model is studied and validated offline using model-based techniques and then models parameters are updated online using data-driven techniques.

As we do not have any prior knowledge about the physical degradation model of the monitored component, in this paper, we propose the use of a data-driven approach for RUL prediction. The approach is known under the name Instance Based Learning (IBL). The problem with this approach is to find an instance formalization that is able to estimate the RUL of a component while using the entire available sensory data.

There exist two types of instance formalizations: supervised

and unsupervised formalizations. (Wang, Yu, Siegel, & Lee, 2008) for example used a supervised formalization of instances by applying linear regression. They proposed to learn a regression model of the damage by taking into account only the boundaries of the sensory data. (Mosallam, Medjaher, & Zerhouni, 2013) on the other hand, used an unsupervised formalization by applying principal component analysis.

We select the unsupervised formalization of instances and we propose the use of unsupervised kernel regression for this purpose. We compare the performance of the latter to both PCA and linear regression. Instances in our approach are thus obtained using unsupervised kernel regression. UKR allows modeling the latter without any assumptions about the components health status or the degradation model. The proposed method constructs a library of instances by fusing the run-to-failure data into a single feature that is faithful to the sensory data representing the damage propagation. Test instances are matched to the library of instances using a similarity measure and the RUL is estimated by using the end of life values of the retrieved best matches. This approach is compatible with any applications satisfying these assumptions:

- Run-to-failure data is available.
- Test components are assumed to go through the same degradation process as train components.
- Sensory data capture the health status evolution.
- Component level prognostics.

The remaining of this paper is organized as follows: section 2 details the proposed approach. Section 3 describes the experimental validation and the obtained results. Finally section 4 concludes the paper.

## 2. RUL PREDICTION APPROACH

The proposed approach predicts the remaining useful life of a new component based on already seen examples. That is learned instances.

IBL approaches for RUL prediction usually go through three main steps as depicted in Figure 1; instance formalization, retrieval step and RUL prediction. The purpose of the instance formalization step is to construct a library of instances that characterize the health status evolution of components. At the retrieval step, a similarity test is conducted to retrieve the most similar instances that are present in the library and related to the problem instance. Once these instances are identified, the information present in them is then used for RUL prediction.

In our proposed approach, instances are formalized as degradation trajectories modeled using unsupervised kernel regression. The method is divided into two steps: an offline and an online step. Offline, a UKR model is learned from each train instance, where a train instance is an instance that goes

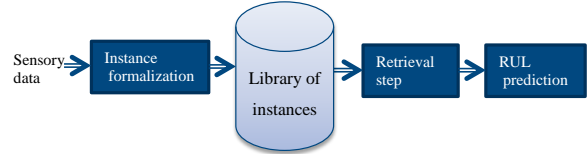


Figure 1. General IBL approach for RUL prediction.

through the whole degradation process. These learned models are used to fuse the multidimensional run-to-failure data into a single feature that depicts the evolution of the health status of the component. Hence, this modeling step enables the construction of a library of train instances that are faithful to the sensory data reflecting the degradation propagation. Online, each of the learned UKR models will be used to reconstruct a test degradation trajectory for the considered test unit. For a single test unit, all the reconstructed trajectories are compared to the train trajectories present in the library of instances. RUL is deduced by keeping only the train trajectories that are close to the test instance. The proposed approach is summarized in Figure 2 and will be further explained hereafter.

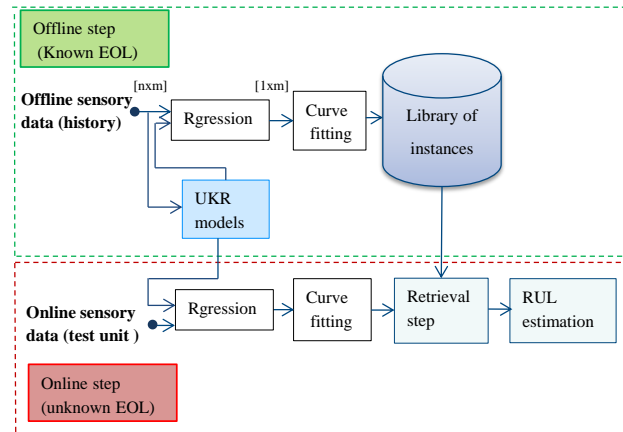


Figure 2. The general proposed approach. IBL approach for RUL prediction.

### 2.1. Instance Formalization

Instances are formalized as a one dimensional signal that is a faithful and compact representation of the multidimensional sensory data related to the degradation process. These degradation trajectories are modeled using unsupervised kernel regression.

UKR is a recent approach that is used to obtain a faithful latent dimensional representation  $X=(x_1, x_2, \dots, x_N)$ ,  $[qxN]$  of the set of observed data (sensory data in our case)  $Y=(y_1, y_2, \dots, y_N)$ ,  $[pxN]$ . The method was proposed by Meinecke and Klanke as an unsupervised formulation of the Nadaraya-Watson estimator. The idea is to generalize the estimator to the unsuper-

vised case of function learning (Meinicke, Klanke, Memisevic, & Ritter, 2005). In the supervised case the estimator realizes a continuous generalization of the functional relationship between two random variables  $X$  and  $Y$  as described in Eq.(1)

$$y = f(x) = \sum_{i=1}^N y_i \frac{K_H(x - x_i)}{\sum_j K_H(x - x_j)} \quad (1)$$

Where  $K_H$  is the kernel density.

As stated in (Memisevic, 2003), the difference between the supervised and the unsupervised regression lies in the usage of the input variables. In the supervised case this becomes a problem of estimating a functional relationship of the input and the related output variable using samples of the latter. In the unsupervised case, the input variable space is considered missing and needs to be estimated together with the functional relationship by finding the sample set of outputs that gives the minimum reconstruction error. In order to derive the unsupervised counterpart of the estimator, (Meinicke et al., 2005) use the same functional form of the Nadaraya-Watson kernel regression estimator, but treat the missing input data as parameters. This set of parameters  $X=x_i$  serves as a lower dimensional latent representation of the original dataset  $Y=y_i$ . The UKR function becomes:

$$\begin{cases} b_i(x; X) = \frac{K(x-x_i)}{\sum_j K(x-x_j)} \\ y = f(x; X) = \sum_i y_i b_i(x; X) = Yb(x; X) \end{cases} \quad (2)$$

Where  $b_i(x; X)$  contains the kernel-based latent basis function and  $f(x; X)$  is the UKR function.

The objective of unsupervised function as defined by Meinicke et al. (2005) is to find a suitable realization of the mapping between the latent domain and the original data domain together with an associated latent representation. This can be reduced to a problem of finding a suitable latent mixture density  $p$ .

$$p(x; X) = \frac{1}{N} \sum_{i=1}^N K(x - x_i) \quad (3)$$

With that latent density model, the UKR function can be completely specified without any further parameters. After having defined the UKR model, the training phase of UKR consists in minimizing the reconstruction error,  $R$ , which is the error between the original observed data and the data points reconstructed from the latent variable vectors  $x_i$

$$R(X) = \frac{1}{N} \sum_{i=1}^N \|y_i - f(x_i; X)\|^2 \quad (4)$$

The concept of UKR is an appropriate choice in our application since the output variable space to which we do not have access, as we do not have any prior information about the degradation evolution, is not required to perform the regres-

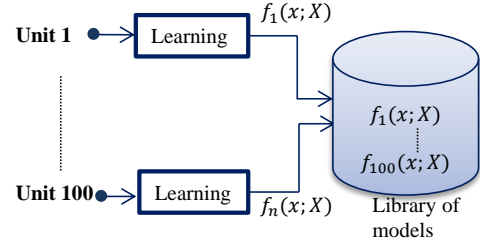


Figure 3. Learning how to formalize instances.

sion. The output of the regression model is a compact representation of the input data that keeps the resulting information loss at minimum.

As it can be seen from figure 3, from each training unit, a model is learned and saved in a library of models. This library is later used to formalize train and test instances. See figure 4.

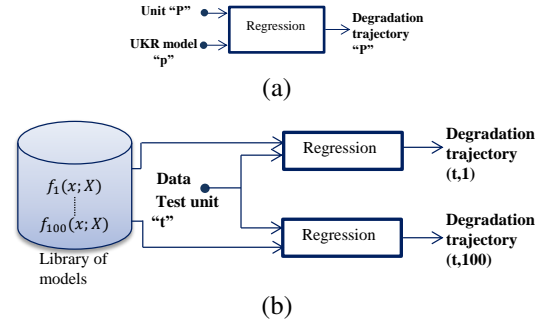


Figure 4. Instance formalization. (a) For a training unit. (b) For a test units.

For a train instance, the corresponding UKR model is known and directly used to construct the degradation trajectory. As for a test instance, the corresponding model is not known but assumed to be one of the models presents in the library. In order to identify the right model, all the models of the library are used. This results in “n” - number of UKR models- test trajectories for a single test unit. At the retrieval step only the appropriate models are kept.

The obtained trajectories using UKR are further processed to produce a smoother output. Figure 5 presents the obtained trajectory after curve fitting.

## 2.2. Retrieval Step

In IBL, the retrieval step is of high importance. Retrieving unrelated instances will result in a large margin of prediction error. In order to obtain an estimation of the RUL of a given test instance, the train instances (instances with known End of Life values) similar to the test instance are retrieved. This is done by conducting a similarity test between test and train tra-

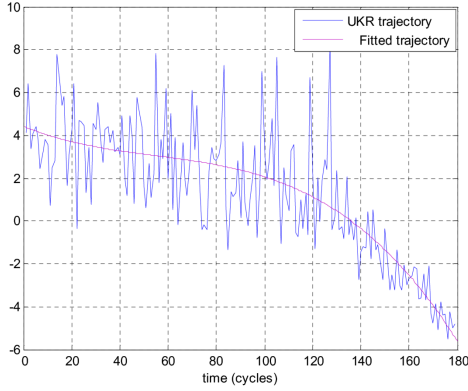


Figure 5. UKR degradation trajectory.

jectories. In most of the available IBL prognostic approaches, the historical data is not entirely used to set this similarity, it is either set by a vector of features characterizing the instance instead of the actual instance data (Xue et al., 2008), or by using only the last measurements (Ramasso, Rombaut, & Zerhouni, 2013), (Zio, Di Maio, & Stasi, 2010), (Mosallam et al., 2013) and (Wang et al., 2008) took into consideration the whole historical data. However, with giving the same weight to all observations while it is known that late observations are of higher importance as failure of components occurs at late ages.

In this work, we use a similarity measure that considers the whole observation data with giving more weights to late ones. Figure 6 illustrates how to conduct this similarity test for a single test unit "p" when "n" train instances are available.

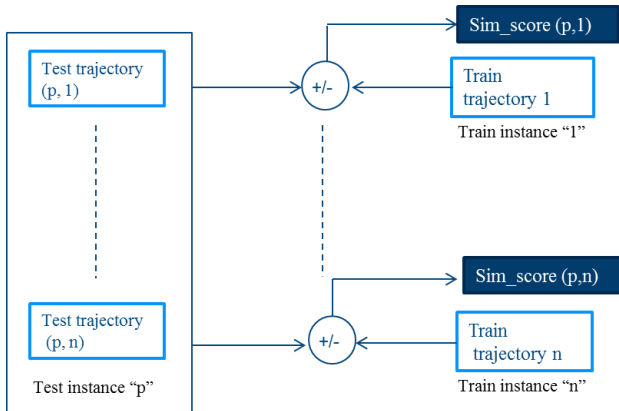


Figure 6. Retrieval step for a single test instance.

For each train instance, a single trajectory is constructed using the UKR model learned from that train instance. As for test instances, the testing unit consists of n test trajectories each constructed using one of the UKR models learned offline. As shown in Figure 6, each test trajectory is compared to its peer train trajectory that is the train trajectory con-

structed using the same UKR model. The sign +/- on the figure represents the computation of a similarity score between the two trajectories. This score is obtained by conducting a similarity measure as follows: The examined trajectories are divided into windows. Each window in the test trajectory is scanned throughout the entire train trajectory. The purpose of doing this is to find the trajectories with the highest similarity scores. The similarity between windows and thereby trajectories is based on the Euclidean distance, where late windows are given more importance since failure occurs at the late ages of life of the component. The final similarity score for each train trajectory is a value that is between 'zero' and 'one'. Zero indicating complete dissimilarity and one indicating a perfect match. The described similarity measure is formalized in algorithm 1 and illustrated in figure 7.

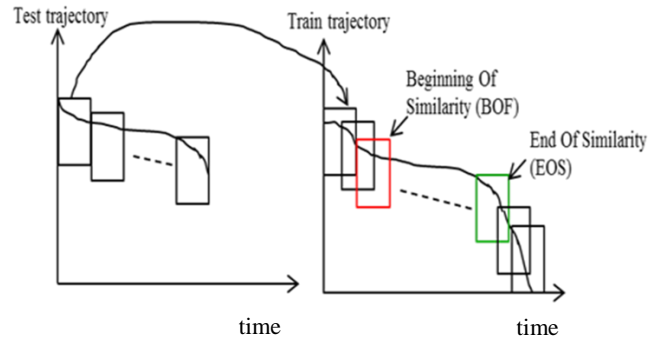
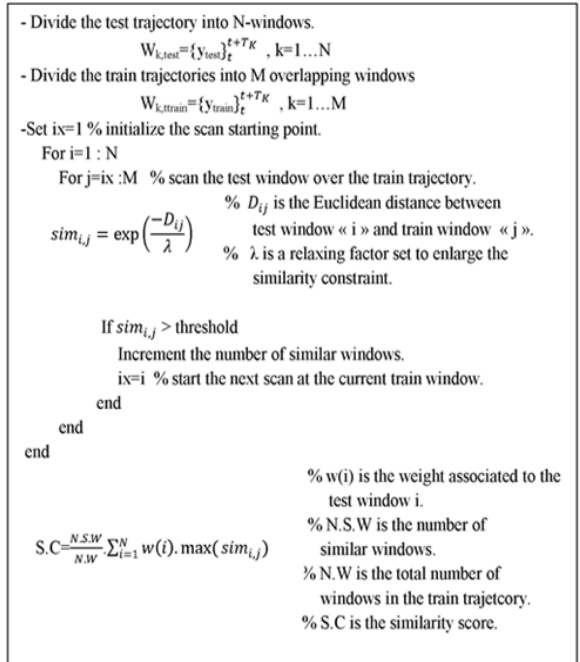


Figure 7. Proposed similarity measure.



Algorithm 1. Similarity measure algorithm.

By the end of the retrieve step, the most similar instances to the train instance are identified based on their similarity scores and kept aside for later use.

### 3. RUL PREDICTION

For a given test instance, RUL is predicted using the retrieved train instances. As described in figure 8, the library of instances contains instances with known end of life values. Once an online instance arrives, that is an instance with an unknown end of life value, a similarity test between instances is conducted using the approach described in this paper. RUL is then deduced using the EOLs of the best match instances.

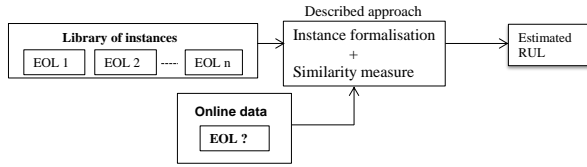


Figure 8. Proposed similarity measure.

For each retrieved train instance, RUL is calculated as the difference between the end of life of the latter and the end of similarity. Eq.(5)

$$RUL(i) = EOL_i - EOS_i \quad (5)$$

Where  $EOL_i$  is the end of life of the train instance  $i$  and  $EOS_i$  is the end of similarity which also indicates the current location on the train instance and is set by the similarity measure.

The predicted test RUL is obtained as either a simple average of RULs of best match instances, Eq. (6) or a weighted sum, where weights are obtained based on the similarity score of the best match instances, Eq. (7).

$$MeanpredictedRUL = \frac{1}{k} \cdot \sum_{i=1}^k RUL(i) \quad (6)$$

$$WeightedsumpredictedRUL = \sum_{i=1}^k w(i) \cdot RUL(i) \quad (7)$$

where,

$$w(i) = \frac{SimScore(i)}{\sum_{i=1}^k SimScore(i)}$$

## 4. APPLICATION AND RESULTS

### 4.1. Data Representation

The challenge dataset of diagnostics and prognostics of machine faults from the first international conference of PHM (Saxena, Goebel, Simon, & Eklund, 2008) was used to evaluate and assess the performance of the proposed approach.

This dataset simulates the damage propagation of aircraft gas turbine engines. It consists of 26 features which are multiple multivariate time series signals. Each time series represents a different engine from the same complex system. At the beginning, each engine is operating normally but ends up developing a fault prior to failure.

Among the available datasets, dataset 1 was used. This dataset is characterized by one operating condition and one fault mode. The training file is composed of 100 time series representing the damage propagation of 100 units. Each unit in this file goes through the whole degradation process. The test file is composed of 100 time series as well. However, these time series end up some time prior to failure. Hence, the objective is to predict the remaining useful life for each test unit. Among the 21 sensors, only 5 were used accordingly to (Ramasso et al., 2013),(Wang et al., 2008).

### 4.2. Evaluation Metric

To evaluate the performance of the proposed approach, the percentage of acceptable predictions is considered as an evaluation criteria.

A prediction is considered correct if its corresponding error, Eq. (9) falls with the range of acceptable errors (Ramasso et al., 2013), (Goebel & Bonissone, 2005). In this paper, the interval was set as  $I = [-10, 13]$ . The interval is asymmetric as early predictions i.e. predictions with positive errors, are preferable in prognostics and hence more tolerable compared to late ones. Figure 9 illustrates this interval.

The performance is then calculated as the percentage of the overall correct predictions.

$$Error = ActualRUL - PredictedRUL \quad (8)$$

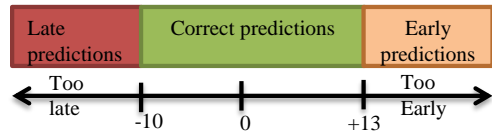


Figure 9. Evaluation Metrics.

### 4.3. Results and Discussion

To estimate the remaining useful life of the test unit a UKR model was learned from each unit in the training file. The entire 100 units of the test file were used for testing. It should be noted here that test trajectories have different lengths. That is each test unit has a different prediction horizon.

Throughout the whole testing, the same set of parameters of the similarity measure was used, the size of windows was set to 30, the overlap to 15,  $\lambda$  was set to 1 and the threshold to

0.8. this set of parameters is user-defined and determine how strict is the similarity measure.

Figure 10 shows the predicted and real RUL values for the 100 test unit, using UKR with a simple average of best match RULs.

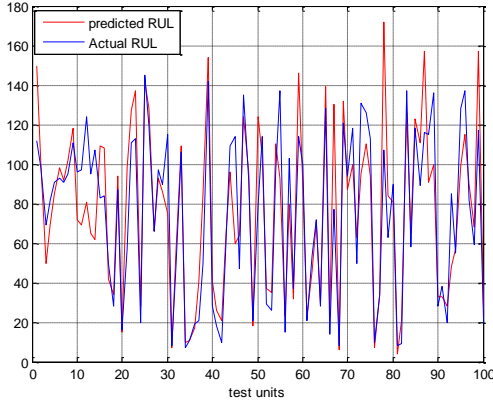


Figure 10. Actual and predicted RUL values for the test units.

The performance of our approach based on UKR was compared to PCA and linear regression. To do this, the UKR modeling step in the general approach, Figure 2, was replaced by PCA and linear regression.

As a first alternative to UKR, and for comparison reasons, PCA was used to fuse the sensory data into a one dimensional signal. This step was repeated offline and online. The obtained fitted online signal was compared to the library of instances constructed using PCA and RUL was calculated as described in section 2.3.

The second alternative was to model the degradation trajectories using linear regression. As proposed by (Wang et al., 2008) a regression model was trained offline by considering two states of the component; healthy and faulty. A component is considered healthy at the beginning of its life and faulty at the end of its life. Only sensory data representing the healthy and faulty states were used to train the model. The regression model was used both offline and online to fuse the sensory data. The fitted online trajectory was compared as well to the library of instances following the same approach described in this paper.

The performance of UKR was compared to PCA, since in this work UKR was used as dimension reduction tool and PCA is the most widely used and understood dimension reduction tool. Linear regression on the other hand was used by (Wang et al., 2008) for the same datasets and proved to be efficient for damage modeling on this dataset.

Results obtained using UKR based modeling approach, PCA and linear regression are shown in figures 11 and 12. Fig-

ure 11 depicts the obtained results using a simple average of RULs of best match instances while figure 12 depicts the obtained results using a weighted sum of the latter. Both methods had almost equal performance with slight preference of the weighted sum method.

Figure 13 depicts the performance difference between UKR linear regression and PCA according to the selected number of neighbors. The graph shows better performance of UKR.

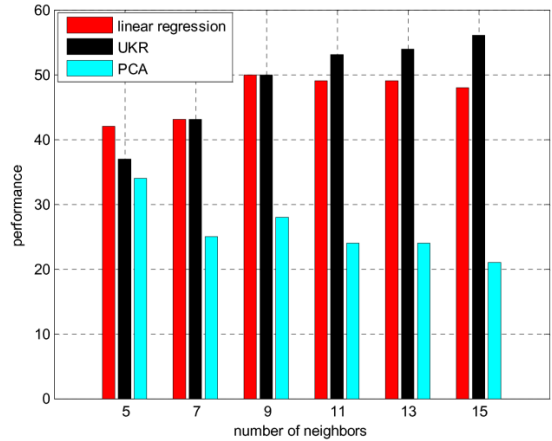


Figure 11. Obtained results using simple average of best match RULs.

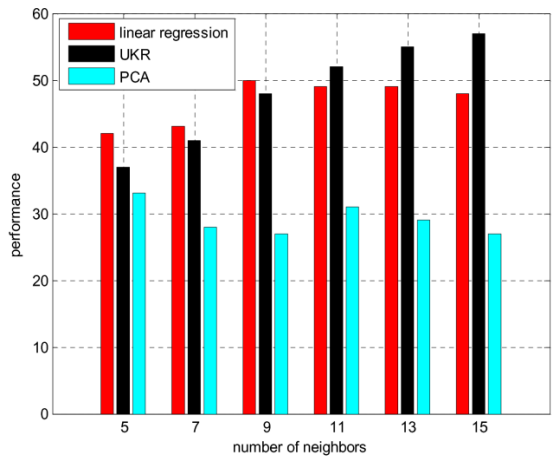


Figure 12. Obtained results using a weighted sum of best matches RULs.

The results show clearly higher performance of UKR based modeling approach compared to both PCA and linear regression modeling. This superior performance can be explained by the following main two reasons; absence of any type of modeling while using PCA, and using only portions of the training data to train the regression model while applying linear regression.

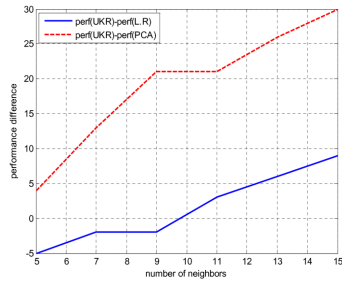


Figure 13. Performance difference between UKR, PCA and linear regression.

The approach is built on instance based learning where the similarity between train and test instances is of high importance. In the absence of any learned model, that is applied to both train and test instances as it is the case for PCA, finding and detecting such a similarity is rare (not always an option) since the instances were not modeled in the same way. This is why PCA had the worst performance compared to linear regression and UKR. As for linear regression, although a unique model was used for both test and train instances the model was learned using only a portion of the training data while neglecting the rest. This slightly affected the performance of the linear regression leading to worse performance than the proposed UKR-based approach for higher number of neighbors. As it can be seen from figures 8 and 9, changing the number of neighbors affects the performance of the prediction. The prediction performance for both approaches varies from 42% to 50% for the linear regression approach and from 38% to 57% for the UKR approach. The best prediction performance value using the linear regression approach is 50% and it is obtained by considering 9 neighbors while the best prediction performance for the UKR approach is 57% obtained using 15 neighbors. Comparing the best prediction performances of both approaches UKR seems to be better as it gives the highest overall performance.

## 5. CONCLUSION

This paper presented a prognostic approach for RUL prediction based on instance based learning and unsupervised kernel regression. UKR was used to model the degradation trajectories without any prior knowledge about the health state of the component. Online, the piece of trajectory constructed using the UKR models learned offline, is compared to a library of degradation trajectories. RUL is then estimated directly using the retrieved best match trajectories.

The approach was demonstrated on the challenge dataset of diagnostics and prognostics of machine faults. Results showed better performance of UKR modeling compared to PCA. As for linear regression, the performance difference is in favor of UKR for higher number of neighbors. Our future work will focus on further enhancing the instance formalization and the

similarity measure by adding to the temporal aspect of trajectories a frequency aspect and considering the frequency difference when setting the similarity score.

## ACKNOWLEDGMENT

The authors would like to acknowledge the European Regional Development Fund (FEDER) who funded this work as a part of a project named **ALTIDE: Aid to Lifecycle Traceability for Intelligently Developed Equipment**.

## REFERENCES

- Goebel, K., & Bonissone, P. (2005). Prognostic information fusion for constant load systems. In *Information fusion, 2005 8th international conference on* (Vol. 2, pp. 9–pp).
- ISO. (2004). *Condition monitoring and diagnostics of machines, prognostics part 1: General guidelines* (Vol. ISO/IEC Directives Part 2; Tech. Rep. No. ISO13381-1). International Organization for Standardization.
- Meinicke, P., Klanke, S., Memisevic, R., & Ritter, H. (2005). Principal surfaces from unsupervised kernel regression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(9), 1379–1391.
- Memisevic, R. (2003). *Unsupervised kernel regression for nonlinear dimensionality reduction*. Unpublished doctoral dissertation, University of Bielefeld.
- Mosallam, A., Medjaher, K., & Zerhouni, N. (2013). Bayesian approach for remaining useful life prediction. *Chemical Engineering Transactions*, 33, 139–144.
- Ramasso, E., Rombaut, M., & Zerhouni, N. (2013). Joint prediction of continuous and discrete states in time-series based on belief functions. *Cybernetics, IEEE Transactions on*, 43(1), 37–50.
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–9).
- Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).
- Xue, F., Bonissone, P., Varma, A., Yan, W., Eklund, N., & Goebel, K. (2008). An instance-based method for remaining useful life estimation for aircraft engines. *Journal of Failure Analysis and Prevention*, 8(2), 199–206.
- Zio, E., Di Maio, F., & Stasi, M. (2010). A data-driven approach for predicting failure scenarios in nuclear systems. *Annals of Nuclear Energy*, 37(4), 482–491.