Method Article

A neural network framework for similarity-based prognostics

Oguz Bektas\textsuperscript{a,1,*}, Jeffrey A. Jones\textsuperscript{a}, Shankar Sankararaman\textsuperscript{b,2}, Indranil Roychoudhury\textsuperscript{c}, Kai Goebel\textsuperscript{d,e}

\textsuperscript{a}Warwick Manufacturing Group, University of Warwick, Coventry, CV4 7AL, United Kingdom
\textsuperscript{b}Pricewaterhouse Cooper, San Jose, CA, 95110, United States
\textsuperscript{c}Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, Moffett Field, CA, 94035, United States
\textsuperscript{d}NASA Ames Research Center, Moffett Field, CA, 94035, United States
\textsuperscript{e}Luleå Technical University, Division of Operation and Maintenance Engineering, Luleå, Sweden

\begin{abstract}
Prognostic performance is associated with accurately estimating remaining useful life. Difficulty in accurate prognostic applications can be tackled by processing raw sensor readings into more meaningful and comprehensive health condition indicators that will then provide performance information for remaining useful life estimations. To that end, typically, multiple tasks on data pre-processing and predictions have to be carried out such that tasks can be assessed using different methodological aspects. However, incompatible methods may result in poor performance and consequently lead to undesirable error rates.

The present research evaluates data training and prediction stages. A data-driven prognostic method based on a feed-forward neural network framework is first defined to calculate the performance of a complex system. Then, the health indicators are used in a similarity based remaining useful life estimation method. This framework presents a conceptual prognostic protocol that overcomes challenges presented by multi-regime condition monitoring data.

© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
\end{abstract}

\begin{articleinfo}
\textbf{Keywords:} Similarity based RUL calculation, Artificial neural networks, Data-driven prognostics
\textbf{Article history:} Received 8 November 2018; Accepted 15 February 2019; Available online 20 February 2019
\end{articleinfo}

\footnote{Corresponding author. \textit{E-mail address: o.bektas@warwick.ac.uk} (O. Bektas).}

\footnote{Part of the original work (Bektas et al., 2008) was completed while the author was an exchange visitor at the Universities Space Research Association at NASA Ames Research Center, whose kind hospitality is gratefully acknowledged.

The original work (Bektas et al., 2008) was performed while being employed at SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035, USA.}

https://doi.org/10.1016/j.mex.2019.02.015
2215-0161 © 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
**Specifications Table**

| Subject Area | Engineering |
|--------------|-------------|
| More specific subject area: | Prognostics and Health Management |
| Method name: | Feed Forward Neural Network Filtering for Similarity-Based Prognostics |
| Name and reference of original method | Oguz Bektas, Jeffrey A. Jones, Shankar Sankararaman, Indranil Roychoudhury, Kai Goebel. A Neural Network Filtering Approach for Similarity-Based Remaining Useful Life Estimations: The International Journal of Advanced Manufacturing Technology, 2018. [1] |
| Resource availability | The method is evaluated by PHM08 challenge data from the competition held at the 1st international conference on Prognostics and Health Management (PHM08) (Goebel, 2008). The dataset is publicly available at Data Repository of NASA Prognostics Center of Excellence. [1] |

**Method details**

The complexity in accurately estimating remaining useful life (RUL) has led to a rise in the number of algorithms and methods in the prognostic literature. The significance of these methodologies relies on their potential to estimate the evolution of degradation conditions in time. Prognostics provide sufficient time for condition-based maintenance operations and notification for necessary actions. Many prognostic applications proposed for various domains can be found in the literature, such as composite materials [2], aircraft actuators [3], turbofan engines [4–7], electronic components [8], etc.

In this article, the motivation is to propose a hybrid framework of data processing and RUL estimation by using various complex system indicators and multi-dimensional sensor measurements. The main research focus is based on the use of data pre-processing methods in order to increase the prognostic performance. Fig. 1 demonstrates the main stages of the protocols which has introduced a conceptual framework to overcome the challenges of multi-regime degradation behavior.

The science of prognosis is predicated on system ageing and monotonic damage accumulation, and it is possible to correlate sensor behavior with signs of ageing to estimate the remaining useful life of systems [9]. However, the multi-dimensional characteristics in raw data does not provide useful

![Fig. 1. Flowchart representing the methodology stages.](image-url)
Fig. 2. Clustered Regimes for Multi-Regime Time Series.
information to measure the monotonic damage accumulation. Further data processing is needed to provide useful information for remaining useful life estimations.

The first step of multi-dimensional data processing is to identify the operational regimes that can be found by finding the number of clusters in the operational settings.

$$f(x_p) = \text{arg}_{k=1,...,K}^\text{find}(x = k)$$  \hspace{1cm} (1)

where $\text{arg}_{k=1,...,K}^\text{find}(x = k)$ defines the index of the occurrence of regimes, $k$, in the data $x$.

Fig. 2 provides an illustrative example of the multi regime sensors after regime assignment. In this sample, a multiple regime normalization method can carry out adjustments by returning raw condition monitoring measurements into a common scale (for example the z-score). To compute a z-score for each regime, it is required to know the mean ($\mu$) and standard deviation ($\sigma$) of the regime population of each sensor to which a data point belongs.

$$z = \frac{x - \mu}{\sigma}$$  \hspace{1cm} (2)

This multiple regime normalization process is applied to all regimes separately; the standardized sensors are reassembled to produce a single-regime condition monitoring dataset. To produce a single health output from these normalized sensors, the multiple readings are aggregated by taking the mean of all sensor measurements at each time step.

$$s_i = \frac{1}{n} \sum_{j=1}^{n} z_{ji}$$  \hspace{1cm} (3)

where $s$ is the single health output and $n$ is the number of sensors.

Even though these aggregated outputs are smoother than the normalized sensors, they are still noisy and there is a risk that when they are included in the training stage, the network might learn from the noise. Therefore, a two-term power series fitting model is applied to describe the relationship between the health output and a Health Indicator (HI) to be used in the neural network training (see Fig. 3 Adjusted Cycle and HI).

$$y = a s^b + c + \varepsilon$$  \hspace{1cm} (4)
Fig. 4. Feed Forward Neural Network Training Design.
here an approximation to a power-law distribution $x^b$ includes two fitting terms $a$ and $c$. The fitted HIs calculated by this fitting model exhibits only increasing values with minimal wear increase levels in early stages.

Artificial Neural Network (ANN) training is intended to accommodate the uncertainty of measurements. The parameters from the multiple regimes are handled by a network mapping process between the raw sensor values and the assigned output HI at the model identification stage. This allows the current and upcoming data to be normalized individually. The function fitting neural network is a two-layer feedforward method including a sigmoid transfer function ($f_h$) in the hidden layer and a linear transfer function in the output layer ($f_0$). This adaptive filtering model is critical for taking into account the standardization issues as well as the consistency in damage information. The general equation representing this network model is denoted by:

$$y(t) = f[x(t)] = f_0 \left\{ b + \sum_{h=1}^{nh} w_h f_h \left( b_h + \sum_{i=1}^{n_i} w_{hi} x_{(t-i)} \right) \right\}$$

where the state variables ($x$) are multiplied by fixed real-valued weights ($w$) and a bias ($b$) is added. The neuron’s activation is obtained as a result of summation of the nodes and the nonlinear activation function ($f$) is applied to this sum [10,11].

Fig. 4 shows such feed-forward network training functions which take a set of input vectors (raw multidimensional condition monitoring data), and then another set of target vector (HI). Neurons, which are connected with coefficients (weights), constitute the neural network structure and evaluate the input state variables. Where a network function is trained with a single trajectory input and
output, an alternative network that is trained with other input and output values, might result in different estimations due to different weight and bias coefficients. To that end, as seen on Fig. 4, multiple neural networks with their inputs and outputs from different trajectories are trained to filter the raw data and each trained function is stored in a network library with “nl” number of trained functions. This library receives all novel trajectories and produces multiple estimates for each inserted input trajectory. Fig. 5 shows a set of network library estimations that result in similar exponential growth patterns. Each line here is an estimated output of the same raw training input resulting from a certain trained network function in the library. There is a consistency between the results from different functions and similar HI estimations starting and ending at parallel wear levels are assigned adequately.

As there are multiple network estimations for a single input “x”, a final HI equation adapted from the moving average method is derived as follows:

$$ S_l = \begin{cases} 
\frac{1}{p+1} \sum_{q=1}^{p} \sum_{j=1}^{n} y_{iq}^{NN} & \text{if } i - p < 0 \\
\frac{1}{p + (l - i + 1)} \sum_{q=1}^{p+(l-i+1)} \sum_{j=1}^{n} y_{iq}^{NN} & \text{if } i + p > l \\
\frac{1}{2p + 1} \sum_{q=-p}^{p} \sum_{j=1}^{n} y_{iq}^{NN} & \text{if } i - p \geq 0 \text{ and } i + p \leq l
\end{cases} $$

(6)

where the final HI (s) is the mean of (p) moving average of (n) number of estimations and (l) is the length of trajectories. The window size of moving average is a numeric duration including the parameters in the current location plus previous and upcoming neighbors.

After both stages of training are achieved, the estimated health indicators are stored to calculate the pairwise distance relations for the remaining useful life predictions. A similarity-based remaining useful life estimation model is used to identify the best matching training HI units for each test HI unit. The similarity-based estimation algorithm used in this article predicts the future behavior of the systems only when there is sufficient training data to map out the damage space. HI derived from the network estimation must provide a realistic representation of system performance.

RUL estimation is applied as a tool to match the test cases with the full operational training periods. The pairwise distance between the two cases is given as a similarity measure which can be expressed as the difference between two vectors.

Fig. 6 shows the moving pairwise distance calculation between a test trajectory and two full sets of training observations. The similarity for each training baseline is initially calculated at time step “1” and then, the testing curve is moved step by step to the end of the training curve to find the best-matching location that can be in the later part of the curve. Thereby, the moving pairwise distance is calculated throughout the training baseline by the following equation:

$$ d(te, tr)_{ij} = \sum_{j=1}^{n_{te}} \left( \sum_{i=1}^{n_{tr}} (te_{i} - tr_{i+1})^{2} \right)^{1/2} $$

(7)

where \( n_{tr} \) and \( n_{te} \) respectively corresponds to the length of the training and test trajectories. Once the test cases are moved, the location of the best similar segments is identified by the minimum pairwise distance value.

$$ L_{te, tr} = \arg \min d(te, tr)_{ij} = \min(d(te, tr)_{1}, d(te, tr)_{2}, ..., d(te, tr)_{n_{tr} - n_{te}}); j = 1, 2, ..., (n_{tr} - n_{te}) $$

(8)

Then, these locations are used for the calculation of RUL for each baseline (training trajectory).

$$ RUL_{te, tr} = n_{tr} - (n_{te} + L_{te, tr}) $$

(9)

The final RUL is found by averaging multiple minimum distance values of baselines so that the multiple estimates could decrease the risk of biased multistep ahead estimations.
Method validation

The algorithm is tested on PHM08 data challenge provided by the Prognostics Center of Excellence at NASA Ames, and it achieved the current overall leading score in the literature [1,12]. While the methods are designed through the application of different protocols for multi-step-ahead predictions, error based prognostic metrics are dynamically used to measure the performance of the model. The prognostic metrics, and their impact on the technical prognostic requirements, gain particular importance in the way that they complement each other. Therefore, the presented methods will be based on both the modelling and performance evaluation of prognostics in terms of technical metrics.

Before the results of the presented algorithm are sent for validation, the model is tested by reconstructed secondary test datasets which were randomly selected from full size training trajectories [13]. The length of units in reconstructed database is equal to the final test units from the original file to ensure that similar behavior of test trajectories is obtained.

Acknowledgements

The material is based upon work supported in part by NASA under award NNX12AK33A with the Universities Space Research Association.

We would like to show our gratitude to the University of Warwick for funding to cover the publication fee.

References

[1] O. Bektas, J.A. Jones, S. Sankararaman, I. Roychoudhury, K. Goebel, A neural network filtering approach for similarity-based remaining useful life estimation, Int. J. Adv. Manuf. Technol. (2018), doi:http://dx.doi.org/10.1007/s00170-018-2874-0.
[2] J. Chiachio, M. Chiachio, S. Sankararaman, A. Saxena, K. Goebel, Condition-based prediction of time-dependent reliability in composites, Reliab. Eng. Syst. Saf. 142 (2015) 134–147, doi:http://dx.doi.org/10.1016/j.ress.2015.04.018.
[3] C.S. Byington, M. Watson, D. Edwards, Data-driven neural network methodology to remaining life predictions for aircraft actuator components, IEEE Aerosp. Conf. Proc. (2004), doi:http://dx.doi.org/10.1109/AERO.2004.1368175.
[4] T. Wang, J. Yu, D. Siegel, J. Lee, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, 2008 Int. Conf. Progn. Heal. Manag. PHM 2008 (2008), doi:http://dx.doi.org/10.1109/PHM.2008.4711421.
[5] F.O. Heimes, Recurrent neural networks for remaining useful life estimation, 2008 Int. Conf. Progn. Heal. Manag. PHM 2008 (2008), doi:http://dx.doi.org/10.1109/PHM.2008.4711422.
[6] L. Peel, Data driven prognostics using a kalman filter ensemble of neural network models, 2008 Int. Conf. Progn. Heal. Manag. PHM 2008 (2008), doi:http://dx.doi.org/10.1109/PHM.2008.4711423.
[7] O. Bektas, J.A. Jones, W.M. Group, NARX time series model for remaining useful life estimation of gas turbine engines, Eur. Conf. Progn. Heal. Manag. Soc. 2016, (2016).
[8] B. Saha, J.R. Celaya, P.F. Wysocki, K.F. Goebel, Towards prognostics for electronics components, IEEE Aerosop. Conf. Proc. (2009), doi:http://dx.doi.org/10.1109/AERO.2009.4839676.
[9] S. Uckun, K. Goebel, P.J.F. Lucas, Standardizing research methods for prognostics, 2008 Int. Conf. Progn. Heal. Manag. PHM 2008 (2008), doi:http://dx.doi.org/10.1109/PHM.2008.4711437.
[10] S.G. Barad, P.V. Ramaiyah, R.K. Giridhar, G. Krishnaiah, Neural network approach for a combined performance and mechanical health monitoring of a gas turbine engine, Mech. Syst. Signal Process. 27 (2012) 729–742, doi:http://dx.doi.org/10.1016/j.ymssp.2011.09.011.
[11] A. Krenker, J. Bester, A. Kos, Introduction to the artificial neural networks, Artif. Neural Networks - Methodol. Adv. Biomed. Appl. (2011), doi:http://dx.doi.org/10.5772/15751.
[12] O. Bektas, An Adaptive Data Filtering Model for Remaining Useful Life Estimation, The University of Warwick, Coventry, 2018.
[13] O. Bektas, J.A. Jones, S. Sankararaman, I. Roychoudhury, K. Goebel, Reconstructing secondary test database from PHM08 challenge data set, Data Br. 21 (2018) 2464–2469, doi:http://dx.doi.org/10.1016/j.dib.2018.11.085.