

Prediction of the Remaining Useful Life of Aircraft Systems via Web Interface

<https://doi.org/10.3991/ijoe.v16i04.11873>

Daniel Azevedo^(✉), Bernardete Ribeiro, Alberto Cardoso
University of Coimbra, Coimbra, Portugal
dazevedo@student.dei.uc.pt

Abstract—In this work a web-based tool is presented for the simulation of a Prognostics and Health Management (PHM) system used for exploring and testing different machine learning experimental scenarios with the goal of predicting the Remaining Useful Life (RUL) of aircraft systems. With this tool, the user can select a set of options like the datasets to use, its size, the machine learning method to apply for the RUL prediction and the metrics used for comparing the results. The proposed datasets correspond to public data extracted from a model which aims to simulate a Turbofan Engine of an aircraft. Also, three different State of the Art machine learning techniques are made available to be applied and tested: a Similarity-based, a Neural Network-based and an Extrapolation-based approach. The results obtained by the different approaches can be graphically compared in the web interface. As the methods are executed remotely, the user incurs no computational costs, which constitutes an advantage of using this tool. This web tool aims to be a user-friendly interface used for simulating online experiments regarding the RUL prediction.

Keywords—Aircraft Maintenance, Machine Learning, Prognostics and Health Management, Remaining Useful Life

1 Introduction

Aircraft play an important role in the modern society, as they are used more and more as a way of transport. According to the statistics from Flightradar24 [1], every day more than 100 000 flights are performed. As several flights are performed by the same airplanes, degradation and wastage is accumulated in the different subsystems of the airplane. Thus, in order to guarantee that the flights are performed successfully and safely, it is necessary to assure that the different aircraft subsystems are healthy, this means, that they do not have any failure or anomaly that prevents its operation.

Aircraft maintenance has an important role in this matter, as it is responsible to assure that the required conditions for the aircraft to fly are complied with. One important responsibility of the maintenance team concerns the parts replacement.

There are different strategies for executing the replacement of the different aircraft components or parts. One of the most common approaches is to use a preventive approach, where the interventions are performed at fixed interval times [2]. Nevertheless, using this strategy, the replacement time is not optimized.

Another promising and optimized approach for the components replacement is the Condition Based Maintenance (CBM) approach, in which the replacement is based on the real health condition of the aircraft components [3]. Using this strategy, the goal is to monitor the degradation of the different components and predict its future behavior. The health assessment is done through an intelligent analysis of the data retrieved from the different aircraft sensors. For prognostic purposes, the output of the CBM approach corresponds to the RUL of the component being analyzed, which is the number of hours/cycles remaining of useful life, that is, the remaining time that the component is estimated to be able to function without any fault or anomaly interfering with the normal operation [4]. There are different machine learning methods or techniques used to compute the RUL [5]. Hence, in order to visualize and compare the RUL obtained from different approaches applied over the same dataset, an online tool is proposed. Over the years, different web-based platforms have been developed with the goal of online experimentation, applied in many different contexts [6].

In this work, a web interface (<http://planum.dei.uc.pt:8080>) is developed for the simulation of a PHM model, where the goal is to predict the RUL of specific aircraft systems. Three state of the art machine learning techniques are considered: a Similarity-based approach [7], a Neural Network-based approach [8] and an Extrapolation-based approach [8]. Using this tool, a visual and comparative analysis of the RULs obtained by the different approaches can be performed. Furthermore, in the web interface, the user can setup the experimental scenario by configuring some options for the model simulation, and, this way, compare the results using different configurations. The web tool is composed by two main components: the *Configuration Window* and the *Results Window*.

2 Configuration window

The *Configuration window* is the first to be shown to the user. In this window, the user can configure and customize certain options for the execution of the PHM methodology, which aims to estimate the RUL of each trajectory in the dataset.

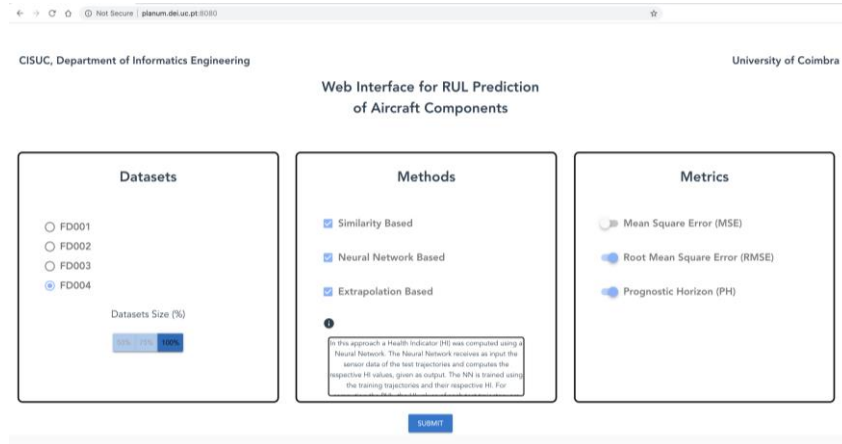


Fig. 1. Configuration Window

The Configuration *Window* is divided into three parts: *Datasets*, *Methods* and *Metrics*, as illustrated in Figure 1.

2.1 Datasets

In the web tool, four different datasets are proposed: **FD001**, **FD002**, **FD003** and **FD004**. These were extracted from the Turbofan Engine Degradation Data Set [9], that is publicly available in the NASA’s Prognostics Data Repository [10]. The four datasets have the same structure and organization, each one contains synthetic data of 21 sensors from a turbofan engine, obtained from a simulation of a model developed to simulate the behavior of an aircraft turbofan engine [11]. These sensors concern to measures of pressure, temperature and velocity of the different parts of the engine.

Each dataset contains several trajectories (flight sequences of the same airplane) and in each of them, the sensors values reflect the presence and evolution of a system fault. The dataset is divided into a train dataset, where the data reflects all the fault behavior, and a test dataset, where only the initial part of the fault behavior is reflected in the sensors data. The goal is to predict the RUL, that is, the number of remaining useful cycles, in each test trajectory, before the system total failure, due to the evolution of the initial fault. As each trajectory has different degradation patterns, a different RUL is obtained. The four datasets were simulated over different conditions, thus the complexity in predicting the fault evolution is different, the FD001 dataset is the easiest in predicting the RUL, and the FD004 is the most challenging.

2.2 Implemented methods

As observed, three different methods are proposed for the RUL computation: **Similarity-based approach**, **Neural Network-based approach** and **Extrapolation-based approach**. In each method, the RUL is computed based in the Health Indicator (HI)

values. The HI reflects the health condition of the system, and in this work, it is represented as a health percentage that varies between 0 and 1, where 0 means the system is unhealthy (due to failure or degradation) and 1 means the system is totally healthy.

Similarity-based approach: In this approach, a set of training trajectories is used to compute the RUL of a test trajectory by measuring the similarity amongst them, and then based on the results, calculate a weighted RUL. Firstly, the set of the training trajectories most similar to the test trajectory is determined, by comparing their degradation behavior using the Euclidean distance. The selected trajectories are the ones with lower distance to the test trajectory. After selecting the training trajectories, these are used for estimating the future degradation behavior of the test trajectory. Then the RUL is computed, based on the obtained degradation patterns.

Figure 2 illustrates an example of the RUL computation for a specific trajectory. Each red curve corresponds to a different training trajectory. The test trajectory is represented by the blue points, whose future degradation wants to be predicted. The RUL of the system associated to the trajectory corresponds to the difference between the last known trajectory cycle and the future cycle where the HI value will be 0.

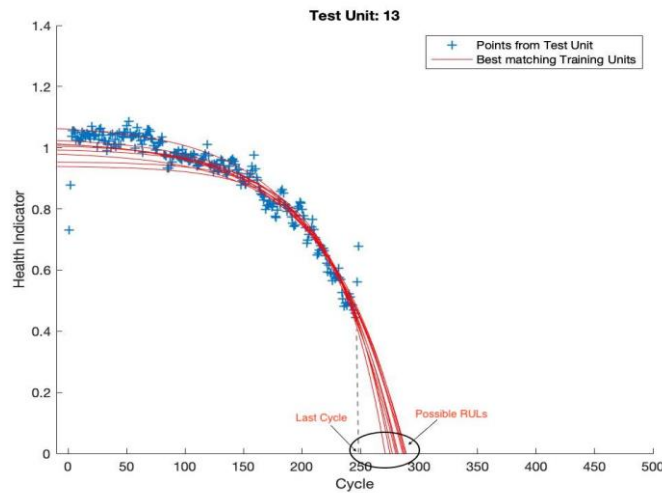


Fig. 2. RUL estimation using the Similarity Approach

As observed in Figure 2, each similar trajectory has a different RUL associated, which composes a set of possible RULs for the test trajectory. The RUL for the test trajectory is calculated using a weighed sum, represented by the following formula:

$$RUL_{test\ traj.} = \left(\frac{13}{23} \cdot \min(RUL_{pool}) + \frac{10}{23} \cdot \max(RUL_{pool}) \right)$$

Where RUL_{pool} corresponds to the set of RULs obtained by the similar trajectories.

Neural network-based approach: In this approach, a Multilayer Perceptron (MLP) is used for training a neural network capable of computing the HI based on the sensors

data. The network is trained using the training dataset, and it is simulated with the test dataset, the output corresponds to the HI values of each test trajectory. Then, an extrapolation of the HI is performed for estimating the RUL of the test trajectory.

Figure 3 illustrates the HI values computed by the Neural Network (NN), represented by the blue points. The HI values are extrapolated, and the RUL value is computed by calculating the difference between the last cycle (last blue point) and the cycle where the HI will reach 0.

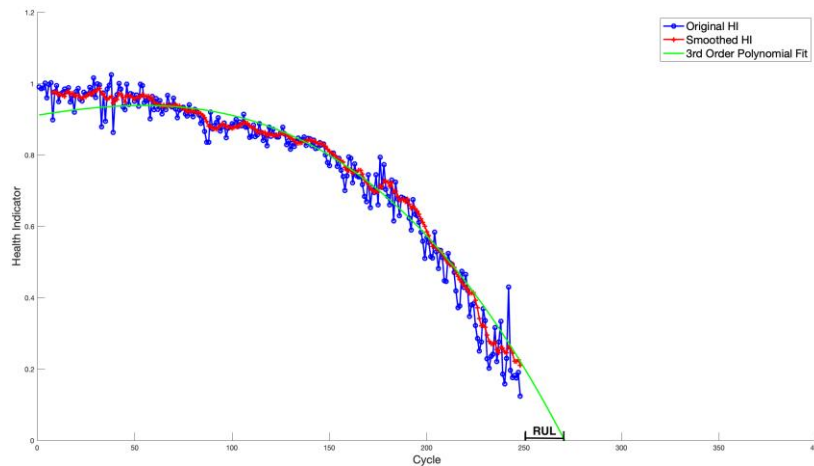


Fig. 3. RUL estimation using Neural Network Approach

Extrapolation-based approach: This last approach is identical to the previous one, but in this case, the HI values are computed in a more simplistic way. Using the training dataset, an exponential model is fitted, such that the sensors data have an assigned weight, to best represent the HI value (target). Using this model, the HI values for the test trajectories are computed. Then, similarly to the previous approach, the HI values are extrapolated, and the RUL of the trajectory is calculated.

Figure 4 illustrates the RUL prediction for a specific test trajectory.

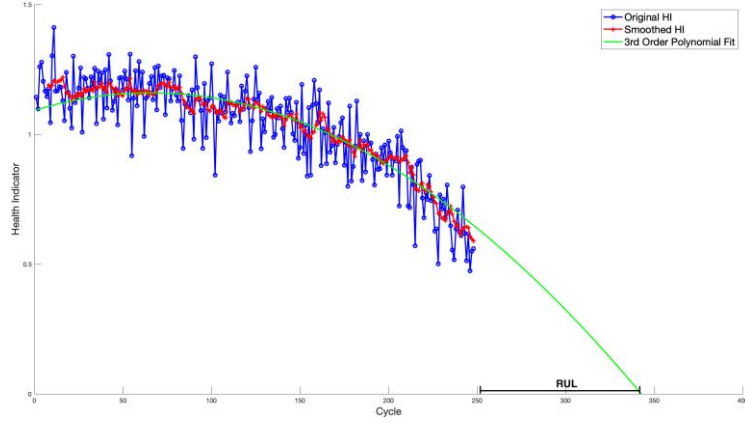


Fig. 4. RUL estimation using an Extrapolation Approach

2.3 Metrics

As depicted in Figure 1, three different metrics can be chosen for comparing the results obtained: Mean Square Error (MSE), Root Mean Square Error (RMSE) and Prognostic Horizon (PH). These metrics compute the accuracy in the estimation of the RUL of each test trajectory, as the RUL ground truth is provided.

Mean Square Error (MSE): The MSE value corresponds to the average of the squares of the difference between the estimated RULs and the true RULs. It is calculated using the following equation:

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (1)$$

Where n is the number of trajectories, \hat{y}_i is the calculated RUL for the i -th trajectory and y_i the RUL ground truth of the i -th trajectory.

Root Mean Square Error (RMSE): This metric is obtained using the previous metric. As the name suggests, it corresponds to the root of the MSE value. Mathematically, it is calculated using the following equation:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (2)$$

Prognostic Horizon (PH): This metric is more suited for prognostic evaluations, as it analyzes the prognostics performed over time and not only on the last prediction.

According to the equation (3), this metric calculates the difference between the time index when the prediction satisfies the α -bound criteria and the time index corresponding to the End of Life (EoL). The equation is the following:

$$PH = t_{EoL} - t_{i_\alpha} \quad (3)$$

Where t_{EoL} corresponds to the time index of EoL and t_{i_α} is the first time index where the α -bound is satisfied.

3 Results Window

On the completion of the setup scenario for the training and testing of the model, the *Results Window*, illustrated in Figure 5, is displayed.

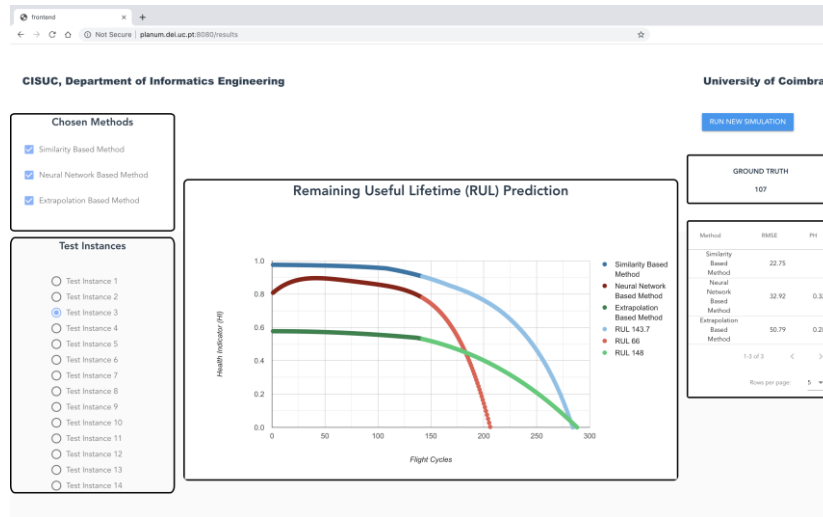


Fig. 5. Results Window

This window presents the results (RUL estimations) of the application of the trained model on the test dataset. The aim of this window is to provide a relevant environment for reaching conclusions regarding the accuracy of the RUL estimations using the different machine learning methods. On the left-hand side, the user may choose which methods' results should be displayed and also which trajectory degradation should be displayed. According to these options, in the center pane, the degradation prediction of the chosen trajectories is illustrated, in terms of the HI. Each line in the graph is subtitled with the type of method associated and also the RUL obtained according to that degradation behavior. In the right pane, the ground truth is indicated, which is the true RUL value of the trajectory being displayed, and also the accuracy metrics obtained when applying the test dataset. These metrics were selected in the setup window. Combining all displayed information, the user can perform his/her analysis and reach conclusions regarding the suitability of each method for the RUL prediction. To perform a new simulation, the user can click on the "Run New Simulation" button.

4 Discussion of Results

Table 1 presents the results obtained by running each method in the four datasets.

Table 1. RMSE Error obtained after applying the three methods on the dataset

	Turbofan Dataset			
	<i>FD001</i>	<i>FD002</i>	<i>FD003</i>	<i>FD004</i>
Similarity Approach	19.87	22.65	21.40	22.75
NN Approach	25.63	31.16	26.58	32.92
Extrapolation Approach	25.15	38.77	55.18	43.85

As observed, the **Similarity method** yields the best results. This was already expected as it was based in the approach with the best score obtained at the PHM 2008 Challenge Competition. Also, the fact that degradation curves present irregular patterns increases this approach’s suitability. Regarding the **NN approach** and the **Extrapolation approach** the results weren’t very interesting, especially due to the method used for the RUL computation. Thus, it can be concluded that the extrapolation method is not well suited for representing and forecasting irregular curves, also, the low amount of data in certain trajectories contributed to the poor results obtained.

5 Conclusion

The web interface presented and described in this work comprises a useful tool for experimenting, testing and comparing different PHM systems configured by the user. Different options for configuration are provided, like the size of the dataset, the methods to use for the RUL prediction and the metrics to apply for the results comparison.

The design of the web interface is meant to be simple and clear. The HI evolution, which represents the degradation behavior, and the respective RUL values are presented in a graph easy to understand and interpret. For the same test trajectory, the different degradation trajectories generated by the different chosen methods are illustrated in the same graph so that their comparison can be performed. Although only three state of the art machine learning methodologies are provided for computing the RUL, in the future, other methods, like Support Vector Machines (SVM) and Random Forests, may be added to diversify the RUL prediction. Hereafter, other datasets may be added by the user in order to experiment and apply the PHM system in other contexts besides aircraft field. Also, regarding future work, specific parameters, known as hyperparameters, may be added to the interface, as a way to have more control and customization of the models created for the RUL prediction.

6 Acknowledgement

This work was partially financed by national funds through the FCT – Foundation for Science and Technology, I.P., within the project CISUC -UID/CEC/00326/2019.

7 References

- [1] Flightradar24.Live Flight Tracker - Real-Time Flight Tracker Map. Flightradar24. [online] Available at: <https://www.flightradar24.com/data/statistics> [Accessed 27 Aug. 2019].
- [2] Jianhui Luo, M. Namburu, K. Pattipati, Liu Qiao, M. Kawamoto and S. Chigusa, “Model-based prognostic techniques [maintenance applications]”, Proceedings AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference. Anaheim, CA, USA, 2003. <https://doi.org/10.1109/autest.2003.1243596>
- [3] Jong-Ho Shin, Hong-Bae Jun, On condition based maintenance policy, Journal of Computational Design and Engineering, Volume 2, Issue 2, 2015, pp. 119-127.
- [4] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Dong-Hua Zhou, Remaining useful life estimation – A review on the statistical data driven approaches, European Journal of Operational Research, Volume 213, Issue 1, 2011, pp 1-14. <https://doi.org/10.1016/j.ejor.2010.11.018>
- [5] Chen Xiongzi, Yu Jinsong, Tang Diyin and Wang Yingxun, “Remaining useful life prognostic estimation for aircraft subsystems or components: A review”, IEEE 2011 10th International Conference on Electronic Measurement & Instruments, Chengdu, 2011. <https://doi.org/10.1109/icemi.2011.6037773>
- [6] Henke K., Tabunshchik G., Wuttke H., Vietzke T., Ostendorff St. (2014). “Using Interactive Hybrid Online Labs for Rapid Prototyping of Digital Systems”, iJOE. <https://doi.org/10.3991/ijoe.v10i5.3994>
- [7] T. Wang, Jianbo Yu, D. Siegel and J. Lee, “A similarity-based prognostics approach for Remaining Useful Life estimation of engineered systems”, 2008 International Conference on Prognostics and Health Management, Denver, CO, 2008, pp. 1-6. <https://doi.org/10.1109/phm.2008.4711421>
- [8] Riad, A.M., Elminir, H.K., & Elattar, H.M. (2010). “Evaluation of Neural Networks in the Subject of Prognostics As Compared To Linear Regression Model”, International Journal of Engineering & Technology.
- [9] A. Saxena and K. Goebel (2008). “Turbofan Engine Degradation Simulation Data Set”, NASA Ames Prognostics Data Repository (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA. <https://doi.org/10.18178/ijsp.4.3.245-251>
- [10] Ti.arc.nasa.gov. (2019). *Prognostics Center - Data Repository*. [online] Available at: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository>.
- [11] Frederick, D, & DeCastro, J. & Litt, J. (2007). User's Guide for the Commercial Modular Aero-P propulsion System Simulation (C-MAPSS). NASA Technical Manuscript.

8 Authors

Daniel Azevedo is a researcher at CISUC, Dep. of Informatics Engineering of the University of Coimbra, DEI, Polo II – UC, 3030-290 Coimbra, Portugal (email: dazevedo@student.dei.uc.pt).

Bernardete Ribeiro is Full Professor at the Dep. of Informatics Engineering of the University of Coimbra, DEI, Polo II – UC, 3030-290 Coimbra, Portugal and the Director of CISUC (email: bribeiro@dei.uc.pt).

Alberto Cardoso is Assistant Professor at the Dep. of Informatics Engineering of the University of Coimbra, DEI, Polo II – UC, 3030-290 Coimbra, Portugal (email:alberto@dei.uc.pt).

Article submitted 2019-12-04. Resubmitted 2020-01-12. Final acceptance 2020-01-12. Final version published as submitted by the authors.