

ProgMachina: Feature Extraction and Processing Package for Prognostic Studies [†]

Tarek Berghout ^{1,*}, Mohamed Benbouzid ^{2,3} and Jaouher Ben Ali ^{4,5}¹ Laboratory of Automation and Manufacturing Engineering, University of Batna 2, Batna 05000, Algeria² UMR CNRS 6027 IRDL, University of Brest, 29238 Brest, France; mohamed.benbouzid@univ-brest.fr³ Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China⁴ Laboratory of Signal Image and Energy Mastery (SIME), National Higher School of Engineers of Tunis, University of Tunis, 5 Av. Taha Hussein, Tunis 1008, Tunisia; benali.jaouher@essths.u-sousse.tn

* Correspondence: t.berghout@univ-batna2.dz

[†] Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: <https://ecsa-10.sciforum.net/>.

Abstract: Prognostic studies of industrial systems essentially focus on health deterioration analysis that has recently been oriented toward data analytics and learning systems. In general, real degradation phenomena suffer from complex drifted data in which degradation patterns are hidden and change over time. Accordingly, such a process requires a well-structured processing and extraction mechanism to reveal such patterns, which facilitates the transition to other model reconstruction and investigation tasks. In this context, to provide additional simplicity of data processing in the field, a complete software package is designed and grouped into a single function that is fully automated and does not require human intervention. The package named ProgMachina (i.e., prognostic machine) provides a featured list of processed features from a life cycle that passed through denoising, filtering, outlier removal, and scaling process to ensure data significance in terms of degradation. The package allows using a time window with a specific overlap to ensure that the scanning process of all possible degradation patterns is properly done. Additionally, an exponential function is used to identify a corresponding health index of degraded signals. Besides, a set of well-known metrics is used to assess degradation of extracted features. Data visualization and many previous experiments on machines show the effectiveness of such a methodology in terms of obtained prediction accuracy and degradation assessment. The package is designed with Matlab software and made available online to be exploited in similar fields.

Keywords: degradation; feature extraction; health index; machine learning; prognostics and health management; remaining useful life

Citation: Berghout, T.; Benbouzid, M.; Ali, J.B. ProgMachina: Feature Extraction and Processing Package for Prognostic Studies. *Eng. Proc.* **2023**, *56*, x. <https://doi.org/10.3390/xxxxx>

Academic Editor(s): Name

Published: 15 November 2023



Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, prognostic studies rely heavily on data analysis and learning systems for condition monitoring rather than highly complex traditional physics-based modeling. Physics-based modeling is primarily needed when the systems under study are both safety-critical and financially expensive, rarely fail under working conditions, and cannot be subjected to real deterioration or accelerated aging laboratory experiments. However, physics-based modeling is used for generative modeling and is also hybridized with learning systems to ensure efficient predictions. In this case, the acquisition, extraction, and processing of run-to-failure data are crucial steps for data analysis and reconstruction of the learning model [1]. When it comes to building a learning model for system prognostics, run-to-failure is usually a challenge of complexity and data drift, while degradation patterns are hidden and buried with ever-changing noise and different distortion patterns, respectively, resulting from harsh system operating conditions. In this case, training

a learning model with such data will certainly mislead the predictions and over-fit the model [2]. In this context, the need for a well-structured feature extraction and processing methodology is urgent to ensure that data is well presented in terms of providing a reliable source of information to improve the performance of the learning model.


In the literature, many paths have been proposed, including most importantly, denoising [3], extraction [4], and outlier removal [5]. Accordingly, since these methodologies are proven to be necessary for progressive degradations analysis in terms of prognostics studies, the main goal of this paper is to combine them as a single and full package as an important contribution to facilitating such a complex process. In this case, this paper introduces ProgMachina; a full package designed specifically to deal with such run-to-failure data features passing via different important steps. Each of these steps is used to uncover and extract degradation patterns from row data of entire life cycles. The package also allows for releasing an exponentially deteriorating health index for the intended life cycle.

This paper is organized as follows: Section 2 represents the package descriptions, its main features, and its relationship with run-to-failure data besides some illustrative examples. Section 3 is specifically dedicated to introducing the impact of this package on prognostics studies, while the conclusion part is dedicated to limits and future improvements of the package.

2. ProgMachina Package Description

ProgMachina is a function designed in Matlab software to deliver well-processed run-to-failure data with a corresponding health index (HI) ready to feed a learning system for training and evaluation. Table 1 gives further details about the metadata of the package. ProgMachina allows the acquisition of run-to-failure dataset per life cycle (i.e., a single degradation unit from normal operating conditions to a complete failure of the system), which is organized vertically as (observations, channels (i.e., different sensors measurements)) only and uses them to generate an extracted and well-prepared list of features and corresponding HI. According to previous literature [3–5], ProgMachina follows specific steps of extraction, denoising, and outlier removal as main steps of uncovering hidden degradation patterns in provided lifecycles, while, smoothening, filtering, and scaling brings further enhanced representations and building strange connection and correlation between data samples. Accordingly, this section is dedicated to exploring such steps in detail.

Table 1. Important metadata of ProgMachina.

Package Name	ProgMachina
Current code version	v1.0.0
Permanent link to the software	https://doi.org/10.5281/zenodo.8174085
Software code languages, tools, and services used	Matlab
Compilation requirements	Matlab \geq r2023a
Logo	

2.1. Features Extraction

A set of well-used features in the literature is included in ProgMachina. These features include mean, standard deviation (Std), skewness, kurtosis, peak to peak, square root of the arithmetic mean (RMS), crest factor, shape factor, impulse factor, margin factor, energy, mean value spectral kurtosis (SKMean), standard deviation of spectral kurtosis (SKStd), spectral kurtosis of skewness (SKSkewness), spectral kurtosis of kurtosis (SK Kurtosis). More details of these features background and their mathematical background can be found in the following references [4–7]. These features are extracted per each time window that overlaps all over the signal and have been selected as they are well-known

signal descriptors and used for such slowly evolving degradation processes analysis while reducing problems complexity and prevent from information loss [7]. It should be mentioned that these features need further analysis of whether they describe a degradation mechanism or not. In this case, metrics like Monotonicity, Tenability, Prognosability, and Robustness (MTPR) are well investigated for such purposes [7]. Accordingly, ProgMachina also includes such metrics to further provides insights about the degradation ability of extracted features and also to provide further information about feature selections. The goal of measuring MTPR is to guarantee that the signal is monotonic to specific degradation trends and given in a meaningful way through the degradation path reflecting actual system health. While, prognosability, is mainly indicating the possibility of separating faulty and healthy degradation patterns.

2.2. Denoising

Slowly evolving degradation processes are well-known with the complex dynamics resulting in very complex feature space with higher levels of noise with unknown sources [8]. In this context, the collected features unquestionably need to be subject to a noise reduction procedure. ProgMachina offers an empirical Bayesian wavelet transformation to create more reliable representations by reducing the amount of noise of such features. This method successfully minimizes the effect of noise in the feature space by combining a Cauchy prior with a posterior median threshold rule [9,10]. This process is accomplished by including the default “`wdenoise(---)`” Matlab function.

2.3. Outlier Removal

Besides the existence of noise in recorded signals as well as per extracted features, different random pulses of higher magnitude disturbances can be found in such a degradation process. Therefore, an outlier remover is necessary to eliminate/reduce their effects on the recorded data. Subsequently, the denoised characteristics of the entire tire life cycle will be further processed using an outlier removal tool. This distinct outlier removal approach was implemented to distinguish differences in data characteristics. Removal of outliers was done by default, using a moving median function “`rmoutliers(---)`” [11].

2.4. Smoothing and Filtering

Additional filtering and smoothing processes are required to further enhance signal quality and provide further appearance to degradation patterns. While the Markov-switching dynamic regression model models data, we employed the state probabilities of the active latent states in the regime transition for smoothing “`smooth(---)`”. It conducts reverse recursion after doing forward recursion [12]. For the filtering process, an additional step of median filtering is involved to further enhance signal quality “`medfilt1(---)`”.

2.5. Scaling and Health Index Identification

A min-max normalization in the range $[0,1]$ is used as usual to unify features measurements scales after each process of denoising and smoothing and outlier removal. From such data, ProgMachina defines a deteriorating HI according to an exponential function (see Equation (2) from [13]).

2.6. Illustrative Example

As an illustrative example, a life cycle of bearing dataset generated from a mathematical model is used in this case [14]. The dataset represents a vibration run-to-failure measurement. Figure 1a is an example of vibration measurement while degradation grows exponentially. ProgMachina is used to find out both features in Figure 1b and HI in Figure 1c. The extracted features are smoother, cleaner, and even representative of degradation than the original row signal and the HI signal. This is demonstrated by the fact that Figure 1d,e,g shows values closer to 1 for all features. This means that the features reflect the degradation

mechanism. Meanwhile, Figure 1f provides further instructions for feature selection if dimensionality reduction is required.

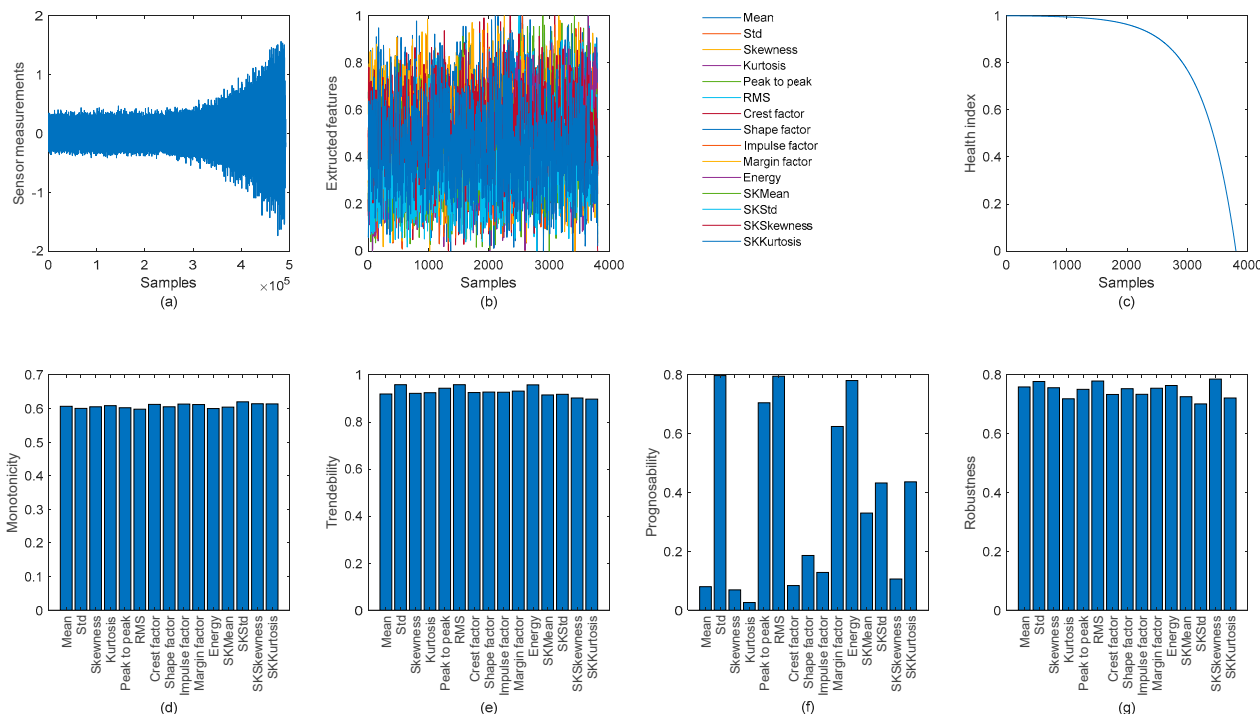


Figure 1. ProgMachina package inputs and outputs: (a) raw vibration data; (b) Extracted and processed features; (c) Identified health index.

3. ProgMachina Impact

ProgMachina as an easy-to-use single package is expected to draw several advantages to prognostics studies including the most important ones listed as follows:

- Bringing more simplicity in learning model reconstruction;
- Needs less human intervention as everything is automatically done;
- Easily investigating variability of new features by simply adding them to the open source code;
- Providing a reliable source of information, especially by involving outlier removal and denoising;
- Make the feature selection process further simplified by studying signal degradations metrics;
- Spending more time on developing learning systems rather than processing.

4. Conclusions

This paper has introduced ProgMachina; a full package for feature extraction and processing degradation signals recorded from slowly evolving degradation processes. It flows different important steps of extraction, denoising, outlier removal, smoothing, filtering, and scaling to reach quality signals that can be used to feed learning systems and ready for investigations. We should mention that ProgMachina is built based on a limited set of both time domain and frequency domain feature extraction and processing. Therefore, future opportunities in improving such a package are to consider other features and further signal processing tools to produce better quality and clean data with better illustration of degradation. It is also important to consider tempo-frequency domain. Additionally, we can use methods of attribute reduction to define health indicators in a new space whose evolution will surely be more linear.

Author Contributions: Conceptualization, T.B.; methodology, T.B. and M.B.; validation, T.B., M.B. and J.B.; formal analysis, T.B., M.B. and J.B.A.; investigation, T.B.; resources, T.B.; data curation, T.B., M.B. and J.B.A.; writing—original draft preparation, T.B.; writing—review and editing, T.B., M.B. and J.B.A.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This work used data from https://figshare.com/articles/dataset/Simulated_Bearing_Degradation_Data_mat/12554690. Necessary files to reproduce the findings of this work can be downloaded at: <https://doi.org/10.5281/zenodo.8174085>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Berghout, T.; Benbouzid, M. A Systematic Guide for Predicting Remaining Useful Life with Machine Learning. *Electronics* **2022**, *11*, 1125. <https://doi.org/10.3390/electronics11071125>.
2. Berghout, T.; Benbouzid, M. PrognosEase: A Data Generator for Health Deterioration Prognosis. *SoftwareX* **2023**, *23*, 101461. <https://doi.org/10.1016/j.softx.2023.101461>.
3. Berghout, T.; Mouss, L.H.; Kadri, O.; Saïdi, L.; Benbouzid, M. Aircraft Engines Remaining Useful Life Prediction with an Adaptive Denoising Online Sequential Extreme Learning Machine. *Eng. Appl. Artif. Intell.* **2020**, *96*, 103936. <https://doi.org/10.1016/j.engappai.2020.103936>.
4. Ben, J.; Harrath, S.; Bechhoefer, E.; Benbouzid, M. Online Automatic Diagnosis of Wind Turbine Bearings Progressive Degradations under Real Experimental Conditions Based on Unsupervised Machine Learning Online Automatic Diagnosis of Wind Turbine Bearings Progressive Degradations under Real Experimental Con. *Appl. Acoust.* **2018**, *132*, 167–181. <https://doi.org/10.1016/j.apacoust.2017.11.021>.
5. Berghout, T.; Benbouzid, M.; Amirat, Y. Towards Resilient and Secure Smart Grids against PMU Adversarial Attacks: A Deep Learning-Based Robust Data Engineering Approach. *Electronics* **2023**, *12*, 2554. <https://doi.org/10.3390/electronics12122554>.
6. Berghout, T.; Mouss, M.-D.; Mouss, L.; Benbouzid, M. ProgNet: A Transferable Deep Network for Aircraft Engine Damage Propagation Prognosis under Real Flight Conditions. *Aerospace* **2022**, *10*, 10. <https://doi.org/10.3390/aerospace10010010>.
7. Qiu, G.; Gu, Y.; Chen, J. Selective Health Indicator for Bearings Ensemble Remaining Useful Life Prediction with Genetic Algorithm and Weibull Proportional Hazards Model. *Meas. J. Int. Meas. Confed.* **2020**, *150*, 107097. <https://doi.org/10.1016/j.measurement.2019.107097>.
8. Saxena, A.; Goebel, K.; Simon, D.; Eklund, N. Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation. In Proceedings of the 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 6–9 October 2008; pp. 1–9.
9. Donoho, D.L. De-Noising by Soft-Thresholding. *IEEE Trans. Inf. Theory* **1995**, *41*, 613–627. <https://doi.org/10.1109/18.382009>.
10. Johnstone, I.M.; Silverman, B.W. Needles and Straw in Haystacks: Empirical Bayes Estimates of Possibly Sparse Sequences. *Ann. Stat.* **2004**, *32*, 1594–1649. <https://doi.org/10.1214/009053604000000030>.
11. Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A Review on Outlier/Anomaly Detection in Time Series Data. *arXiv* **2020**, arXiv:2002.04236v1.
12. Kim, C.-J. Dynamic Linear Models with Markov-Switching. *J. Econom.* **1994**, *60*, 1–22. [https://doi.org/10.1016/0304-4076\(94\)90036-1](https://doi.org/10.1016/0304-4076(94)90036-1).
13. Berghout, T.; Mouss, L.-H.; Bentrucia, T.; Benbouzid, M. A Semi-Supervised Deep Transfer Learning Approach for Rolling-Element Bearing Remaining Useful Life Prediction. *IEEE Trans. Energy Convers.* **2022**, *37*, 1200–1210. <https://doi.org/10.1109/TEC.2021.3116423>.
14. Koceila, A.; Mouchaweh, M.S.; Cornez, L.; Chimentin, X. Simulated Bearing Degradation Data. **2020**. <https://doi.org/10.6084/m9.figshare.12554690.v2>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.