Validation of an algorithm for predicting the remaining useful life, for a model with linear degradation

G Grebenişan^{1,3}, N Salem², S Bogdan¹, D C Negrău¹

¹University of Oradea, Romania ²Zarqa University, Jordan

³grebe@uoradea.ro

Abstract. This paper aimed to validate a working tool, component of the Predictive Maintenance Toolbox TM, produced by Matlab (MathWorks), in the case of a procedure for monitoring the operation of mechanical systems, in order to diagnose a failure of the process and to estimate the remaining useful life (RUL). This toolbox provides toolsets, materialized in function files, for labeling data, designing condition indicators, and estimating a parameter named the remaining useful life of a machine. You can analyze and label machine data imported from local files, cloud storage, and distributed file systems. The algorithm suggested by Matlab (software owned by MathWorks) was used in detail to process part of the data set provided freely by NASA through The Prognostics Data Repository, The Prognostics Center of Excellence (PCoE) at Ames Research Center. Of the 4 data sets, only one was used for this paper. Each data set is composed of 3 working files, in text format, for training, test and algorithm validation, and solution statement, respectively. The results obtained confirm the validity of the computer-assisted training system, diagnostics, prognosis, and validation tools, on a statistical basis, in the case of consistent databases.

1. Introduction

Monitoring the operation status of industrial equipment provides data on their condition, [1], [2]. Any failure, or deterioration of the state of the installations, can be detected, and preventive measures can be taken within an appropriate cycle to avoid catastrophic failures. This is done by monitoring parameters such as vibration, solid wear in the oil, noise emission, etc. Changes to these parameters help detect the spread of defects, diagnose the causes of the problem, and anticipate failure. Maintenance can be supported, so corrective actions can be planned accordingly. Applying condition monitoring in machines and installations leads to savings in maintenance costs and improved availability and safety, [3], [4].

The main function of monitoring the operating condition of a mechanical system, machine, or installation is to provide an almost accurate diagnosis of the condition of the machine and its rate of change, so that preventive measures can be taken at a given time. Knowledge of the condition of the machines can be obtained by selecting a parameter that indicates the deterioration of the condition of the machine. The value of this parameter can be measured periodically or continuously. In some cars, the deterioration of the conditions develops so fast that it is possible that there are only a few seconds between the detection of faults and the total failure. In such cases, continuous or online monitoring with an automatic shutdown of the system or machine is recommended.

2. Fault detection and diagnosis

When a process error occurs, it must be detected as soon as possible. The fault detection system must indicate that something is wrong in the process. After detection, the fault is diagnosed, then is isolated, and an attempt is made to detect the cause of the fault. Typically, the techniques used to detect and diagnose defects are divided into two general categories: estimation methods and pattern recognition methods, [1], [3], [5], [6].

Fault detection and diagnosis is currently a significant issue in process automation, [5, 6]. Screening and diagnostic methods based on pattern recognition and expert systems, respectively, have been suggested to solve the problem of singularity (anomaly, [6]), as defined by a process failure. Various methods and techniques have been suggested in the literature, such as model-based techniques such as pattern recognition, neural networks, with a lot of possible architectures for diagnosing errors. The neural network was the first automatic technique for learning the various situations that define a system failure.

Usually, an anomaly is considered that non-compliant state that does not respect the development of expected behavior, therefore, the problem of detecting an anomaly is reduced to find, in the available data set, those data models that do not follow the rules, which are out of general description process. As a general approach, these anomalies, or non-compliant models, are often referred to as exceptional values, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants in different fields of application, [7]. Of these, anomalies and outliers are two terms most commonly used in the context of anomaly detection, sometimes interchangeably. Examples of anomaly detection are found in a wide range of applications, such as credit card fraud, insurance or health care detection, cybersecurity intrusion detection, critical security breach detection, and military surveillance of enemy activities. The importance of anomaly detection is since data anomalies translate into relevant, often critical, information that can be used in a wide variety of fields of application.



Figure 1.- Normal regions and anomalies data, [3]

Figure 1 illustrates the scattering of anomalies in a simple set of data represented in a twodimensional graph. There are two well-defined regions, in which the data are grouped, each representing the normal behavior of a system, or systems, while the scattered, discordant data, sufficiently distant from the two typical regions are singularities, or anomalies of this data set, [8]. It is essential that these anomalies are not treated as "noisy". Noises are generally defined as obstacles in the work of analysis of phenomena and processes and must be treated as component parts of the mathematical model of the process, [1, 2, 3, 4, 9, 10, 11]. Considering a linear model, which expresses the relation between the time series of the inputs, X_t , and the time series of the outputs, Y_t , in the form:

$$\mathbf{Y}_{t} = \boldsymbol{\upsilon} \left(\mathbf{B} \right) \mathbf{X}_{t} + \mathbf{N}_{t} \tag{1}$$

where, $\upsilon(\mathbf{B}) = \upsilon_0 + \upsilon_1 \mathbf{B} + \upsilon_2 \mathbf{B}^2 + ...$, is the transfer function, expressed polynomially, $\upsilon(\mathbf{B}) = \frac{\Omega(\mathbf{B})}{\delta(\mathbf{B})} = \delta^{-1}(\mathbf{B})\Omega(\mathbf{B})$ respectively, represented as a ratio of mathematical operators, and which denotes the dynamic relationship between outputs and inputs, in which **B** is a *backward shift operator* (given an observable time series z_t , with the components: $: z_1, z_2, \dots, z_N$, then $Bz_t = z_{t-1}, B^m z_t = z_{t-m}$, while N_t , represents the filtered component of the noise superimposed over the input signal, a_t , and which influences (transforms) the z_t component as follows:

$$\mathbf{z}_{t} = \mu + \mathbf{a}_{t} + \psi_{1}\mathbf{a}_{t-1} + \psi_{2}\mathbf{a}_{t-2} + \dots = \mu + \psi(\mathbf{B})\mathbf{a}_{t}$$
(2)

where μ is a parameter that generally expresses the "level" of z_t , and a_t is a sequence of random components, weighted, after the weighted operator $\psi(\mathbf{B}) = \mathbf{1} + \psi_1 \mathbf{B} + \psi_2 \mathbf{B}^2 + ...$, called the filter transfer function (this is different in definition and mathematical model, of anomalies, singularities, etc .: $\mathbf{N}_t = \psi(\mathbf{B})\mathbf{a}_t$, so that the analysis of the behavior of a process, or system, imposes the determination, both of the transfer function, $\upsilon(\mathbf{B})$, and of the transfer function of the filter, $\psi(\mathbf{B})$, [1], [2], [9], [10], An iterative numerical procedure is presented in [12], starting from the analysis of two classes of outliers, generated by dynamic models of exceptional interventions, at unknown moments of time: innovative outlier (IO) and an additive outlier (AO). Starting from a stochastic process model, x_t , following an autoregressive-integrated-moving average (ARIMA) model (possibly with the characteristics **p**- a positive integer indicating the degree of the nonseasonal autoregressive polynomial, **d**- a non-negative integer indicating the degree of nonseasonal integration in the linear time series, **q**-a positive integer indicating the degree of the nonseasonal known), [1], [13], [14]:

$$\phi(\mathbf{B})\alpha(\mathbf{B})\mathbf{x}_{t} = \theta(\mathbf{B})\mathbf{a}_{t} \tag{3}$$

where **B** is the *backward shift operator*, defined above; $\phi(\mathbf{B}) = (\mathbf{1} - \phi_1 \mathbf{B} - \phi_2 \mathbf{B}^2 - \dots - \phi_p \mathbf{B}^p)$; and $\theta(\mathbf{B}) = (\mathbf{1} - \theta_1 \mathbf{B} - \theta_2 \mathbf{B}^2 - \dots - \theta_q \mathbf{B}^q)$; there are two polynomials whose roots lie outside the circle with unit radius $\alpha(\mathbf{B}) = (\mathbf{1} - \mathbf{B})^{d_1} (\mathbf{1} - \mathbf{B}^s)^{d_2}; \mathbf{d} = \mathbf{d}_1 + \mathbf{s}\mathbf{d}_2$; and \mathbf{a}_t has been defined above, has the definition as a function of $\mathbf{N}_t = \psi(\mathbf{B})\mathbf{a}_t$. The model of an exceptional external intervention, [10], [12], is mathematically represented by a dynamic model:

$$z_{t} = \frac{\omega(B)}{\beta(B)} \zeta_{t}^{(T)} + x_{t}$$
where
$$\zeta_{t}^{(T)} = \begin{cases} 1 & \text{for} \quad t = T \\ 0 & \text{otherwise} \end{cases}$$
(4)

The significance of the parameter $\zeta_t^{(T)}$ is: the moment when the intervention takes place on the process x_t , and the respective operators $\omega(\mathbf{B}) = (\omega_0 - \omega_1 \mathbf{B} - \omega_2 \mathbf{B}^2 - ... - \omega_s \mathbf{B}^s)$; are two polynomials depending on **B**, whose ratio $\omega(\mathbf{B}) / \beta(\mathbf{B})$ characterizes the dynamic behavior of the intervention. Proving that these interventions can cause pronounced bias in the procedure for calculating and estimating correlations, partial autocorrelations, and autoregressive moving average (ARMA) parameters, it is obvious the need to find a method to identify and determine these interventions, respectively eliminate their effects. Assuming that the two classes of interventions in the time series model, innovational outlier (IO) and an additive outlier (AO), are characterized, each by the following models (the operators used here, $\theta(\mathbf{B}), \phi(\mathbf{B}), \alpha(\mathbf{B})$ have been previously defined), [10], [12], [15]:

-a dynamic model for the innovative outlier (IO)

$$\mathbf{z}_{t} = \frac{\boldsymbol{\theta}(\mathbf{B})}{\boldsymbol{\phi}(\mathbf{B})\boldsymbol{\alpha}(\mathbf{B})}\boldsymbol{\omega}\boldsymbol{\zeta}_{t}^{(\mathrm{T})} + \mathbf{x}_{t}$$
(5)

- a dynamic model for the additive outlier (AO)

$$\mathbf{z}_{t} = \boldsymbol{\omega}\boldsymbol{\zeta}_{t}^{(\mathrm{T})} + \mathbf{x}_{t} \tag{6}$$

If the last relations are rewritten, in terms of a random sequence, a_t , we obtain:

- for innovational outlier (IO)

$$\mathbf{z}_{t} = \frac{\boldsymbol{\theta}(\mathbf{B})}{\boldsymbol{\phi}(\mathbf{B})\boldsymbol{\alpha}(\mathbf{B})} \left\{ \mathbf{a}_{t} + \boldsymbol{\omega}\boldsymbol{\zeta}_{t}^{(\mathrm{T})} \right\}$$
(7)

- for additive outlier (AO)

$$\mathbf{z}_{t} = \frac{\boldsymbol{\theta}(\mathbf{B})}{\boldsymbol{\phi}(\mathbf{B})\boldsymbol{\alpha}(\mathbf{B})} \mathbf{a}_{t} + \boldsymbol{\omega}\boldsymbol{\zeta}_{t}^{(\mathrm{T})}$$
(8)

Thus, analyzing the last two models, it is found that additive outlier (AO), affects the time series model only at the level of observation **T**, while innovational outlier (IO), is an exceptional intervention, that means an unusual event, at time **T**, but which affects all observations, z_T , z_{T+1} , z_{T+2} ..., which

follows, from that moment, through the $\frac{\theta(B)}{\phi(B)\alpha(B)}$ term.

3. Experimental setup

Datasets are delivered, [11], [16], [17], [18], in directories and files consisting of several time series. Each set of data is further divided into training and testing subsets. Each time series comes from a different engine - that is, the data can be considered to be from a group of engines of the same type. The experiment begins with each engine having a certain degree of initial wear and different levels of manufacturing accuracy classes that are not known to the user. This wear and the level of manufacture are considered normal; i.e., it is not considered a state of error. There are three condition settings that have an important effect on engine performance and functional characteristics. These operational

settings are included in the data (operational setting 1; operational setting 2; operational setting 3. The data is contaminated with sensor noise.

Each engine runs typically at the beginning of each time series and, according to the experimental scenario, develops a failure at some point during the series. In the training set, the defect increases in amplitude, eventually generating system failure. In the test set, the time series ends shortly before the system fails. The aim of the research is to predict the number of operating cycles remaining before the failure of the test set, i.e., the number of operating cycles after the last cycle in which the engine will continue to run. A vector of true remaining useful life (RUL) values, which was also provided for the test data. The data is provided as a text file structured as tables with 26 columns of numbers, separated by spaces, cell arrays, and column vectors, in txt files. Each row is a snapshot of the data taken during a single operational cycle, and each column is a different variable. The columns correspond to the variables, to the table header of the data structure:

1) unit number

2) time, in cycles3) operational setting 1

4) operational setting 2

5) operational setting 3

6) sensor measurement 1

7) sensor measurement 2

...

26) sensor measurement 26

The experimental scheme and the appropriate scenario belong to NASA, The Prognostics Center of Excellence (PCoE) at Ames Research Center, Turbofan Engine Degradation Simulation Data Set, [11], [16].

4. Results and Discussions

Data used here were received from NASA's database, The Prognostics Center of Excellence (PCoE) at Ames Research Center, Turbofan Engine Degradation Simulation Data Set, [16], were preprocessed in Matlab(all the file functions designed for this algorithm was used in original form, same as definitions and statements were used in original form, [17], [18]), using the specific *loadData* helper function, which converts the text file, data in tables, in cell array files as well as in vector files. The basic file, "*degradationData*", is a cellular structure, with 100 cells, arranged vectorially in a single column. Each cell represents a table of numerical data, arranged in different numbers of rows (samples) and 26 columns. Rows represent a time sequence, i.e., a set of 26 values collected at a time, value according to the second column, "time". The columns represent the variables defined in Section "3. Experimental setup": 1) unit number; 2) time, in cycles; 3) operational setting 1; 4) operational setting 2; 5) operational setting 3; 6) sensor measurement 1; 7) sensor measurement 2; ...; 26) sensor measurement 26. Preprocessing in Matlab is a continuous process, so it is inappropriate to call it "pre", as definitions of variables, their processing, are performed throughout the workflow.

The converted data, from the table to the cell structure, "*degradationData*" with dimensions: 249x1 cells, is partitioned (using the *cvpartition* function) into data needed for training (200 of the 249 cells of the *degradationData* time series). A partition of this primary data (49 cells from *degradationData*) will be used for the validation process, to evaluate the performance of the procedure/algorithm.

In figure 2 and figure 3, one can analyze the plots of data evolution for a group of 7 sensors/graph, on 2 samples, and 5 samples, respectively. This analysis is not very performant or efficient because there

r 1	500 H M	110 0					
ensor_3 sensor 2 senso	450	ANNO MAL	MMU MM		MANANA ANA	MWWWWWW	
	0	50	100	150	200	250	300
		allhout	IMAN A MANA	And Weak	MMMAA	MMM	
	0	50	100	150	200	250	300
	1600 MAN	ANNA ANNA AN	4MAM A MANA	my My My apr	North And	WWWW	
4 S	0	50	100	150	200	250	300
insor		ahala ana an	AMAM A MANN	Art MM Wald	MAN AND AND	MMMM	
ŝ	0000	50	100	150	200	250	300
		al and a second	MAAMM	MMMM	hor Market	WWW	
	0 ñ	50	100	150	200	250	300
7		ahahah	MARAMAN	MMMM	WWWWWWW		
	й ₀	50	100	150	200	250	300
ensor	400	al and a second	MAN A MAN	MMMM	hum how	MMM	
š	0	50	100	150	200	250	300
				time			

a)-sensors #1 to #7



b)-sensors #8 to #14



c)- sensors #15 to #21 **Figure 2**. Graphical representation of data evolution for a group of sensors, on 2 samples



a)-sensors #1 to #7







are no obvious trends in data measurements. To have a clearer image, in mind, of the evolution/trend of the chart, to be able to highlight and extract clear trends of degradation, respectively marking the occurrence of any trend of failure, the three settings of the operations will be used (different, for each sensor). It should be noted that each member of the ensemble contains 3 operating conditions:

"op_setting_1", "op_setting_2" and "op_setting_3". First, the data will be extracted from each cell of the *degradationData* structure (249 cells, each cell is a table with 49473 rows and 26 columns), as column vectors, then these vectors will be concatenated into an extended table with 43352 rows and 26 columns. Then, from this table, with 49473 rows (samples) the data corresponding to the 3 columns of the operating conditions: "op_setting_1", "op_setting_2" and "op_setting_3" are extracted, and next will be grouped in an array with 49473 rows and 3 columns, by vertical concatenation.

Considering two working regimes: "clustering" and "normalization", there are three graphical representations captured in figures 4, 5, and 6, respectively, which can simplify the analysis of the evolution trend of the data collected from each sensor. Thus, figures 4 and 5 are valid for data processing in clustering mode, and the last of these two shows that the small distances between the different operating points, the centroids, coincide with the operating points (6 regimes, respectively, distinct operating points). The K-means algorithm is used to automatically locate the 6 clusters. Repeating the algorithm 5 times, and the results are identical: 0.377212.



Figure 4. The 6 operating points in clustering mode



Figure 5. The simultaneous plot of the clustering results and the identified cluster centroids (with "x" marker)

Figure 6 shows the trend of evolutions, mostly positive (increasing graph), of data collected from sensors # 2, # 3, # 4, #8, #9, #11, #13 and # 17. Graphical representations are now, after data normalization, more expressive and easy to be analyzed. This is the most important gain of the normalization operation.



Figure 6. Using the data normalized by the working regime, the degradation trends for some sensor measurement s





b)- sensors #8 to #14



c)- sensors #15 to #21

Figure 7 concentrates on two subfigures, the most trendable measurements, which in fact show a strongly increasing trend, in the evolution graph. Once the data has been normalized, the graphical representation is more obviously expressive.

Figure 7. The selected most

measurements

trendable

sensor



b)- sensorsTrended #9, #11, #13 and # 17

Next, we'll go to the creation of a health indicator, from fusing the measured data. A system is supposed to start behaving poorly, starting from a state of operation in accordance with the rules within which it was designed, called a state of "health". If one considers the state of health a function, the value 1 is assigned to the state of health at the beginning, and to the failure, the value zero is assigned to it. It can be considered that the health condition has a linear evolution, the degradation being linearly between 1 and 0. The measured and collected data will be fused with the 21 sensors in a so-called health indicator, based on similarity. Several models and fusing techniques can be used in this case, [11], [13], [17], [18]. Figure 8 shows the linear representation of evolutions in three cases: 5 samples, 25 samples, and 100 samples.



The slope of the lines in Figure 8 is consistent with the rate of degradation of the systems represented by the data collected from the sensors. Linear regression, therefore, can be chosen as a model for Health

Condition Indicator, in which the data for characterizing the regressors are those corresponding to the sensors with the highest trendability: # 2, # 3, # 4, #8, #9, #11, #13 and # 17, plots in Figure 9.



Figure 9. Plots of the fused Health Condition Indicator for training data in three cases: 5 samples; 25 samples and 50 samples

One may repeat the data normalization process and the sensor fusion process for the validation data set, figure 9.





Using the training data, one can now build a model of Remaining Useful Life with residuals-based that is compared for fit compared to fused data, mathematically using a second-order polynomial curve. The fit error is calculated as the difference between the training data in the structure of the health indicator of the machine and the estimation data of the same health indicator of the machine. The *residualSimilarityModel* Matlab function is used to estimate the remaining RUL of a component with a similarity model based on the residual comparison. This method can be applied in this case because it has data sets that characterize degradation profiles for a set of similar components, with the same specifications, and the dynamics of the degradation process are known. This stage of the algorithm consists of two distinct steps: configuring the model (residualSimilarityModel) and comparing the data obtained in this model with the fused trained data. To similarity RUL model may evaluate in three partial data sets will be used, i.e., samples of 50%, 70%, and 90% of the previously determined validation set, to predict its RUL, figures 11 to 16.



In figure 17 is represented the histogram of the error between estimated RUL and true RUL for each breakpoint with a probability distribution when the evaluation was made for all validation data.

5. Conclusion

In practice, a lot of applications and calculation systems, algorithms, and elaborate schemes for predicting the remaining useful life are used, [3], [4], [11], [16], [17], [18]. By definition, widely accepted, degradation models extrapolate past behavior to predict a future state. This type of RUL calculation fits a linear model, most often, or exponentially adapted to the degradation profile of a condition indicator (also called health indicator), taking into account the degradation profiles of the user as a whole. Then, for the training validation, the degradation profile of the test component is used to statistically calculate the time remaining until the indicator reaches a prescribed threshold. These models are most useful when there is a known value of the status indicator indicating a fault. Of the two available types of degradation models: the linear degradation model (*linearDegradationModel*) and the exponential degradation model (*exponentialDegradationModel*), the first model was used in the paper, as the collected data have a linear degradation profile (using statistical procedures) and the degradation type is not cumulative. The computational algorithm used here briefly states that after the degradation model is created, or the data are preprocessed so that it meets the degradation profile, the model is initialized using historical health data of an assembly with similar components, such as many machines,

or equipment, manufactured to the same specifications, to do this, a specific fit function is used, after which appropriate calculations can determine it, predicted RUL of similar components using *predictRUL*.

References

- [1] Isermann R, Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance, ISBN-10 3-540-24112-4, Springer, Berlin, 2008.
- [2] Wang L and Robert X. Gao (Eds.) Condition Monitoring and Control for Intelligent Manufacturing, ISBN 978-1-84628-269-0, 2006, Springer-Verlag.
- [3] Grebenişan G, Bogdan S, Salem N and Negrău D C, A brief assessment of outliers and malfunctions detecting techniques with an application on lubricant condition monitoring, Mat. Sc. and Eng. 568 (2019) 012039 IOP Publishing doi:10.1088/1757-899X/568/1/012039.
- [4] Grebenişan G, Bogdan S, Salem N, and Negrău D C A neural networks approach of process fault diagnosis using time series collected data through oil condition monitoring, Mat. Sc. and Eng. 568 (2019) 012079 IOP Publishing doi:10.1088/1757-899X/568/1/012079.
- [5] Ortiz D, Byington C, Patrick R, Ture C, Farnach J, Moffatt J, Combined Lubrication Monitor for On-Line Gearbox Health Assessment, doi: <u>10.1109/AERO.2011.5747563</u>, 2011, Aerospace Conference, IEEE Xplore.
- [6] Teng H S, Chen K and Lu S C-Y, Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns, Conference Proceedings on Research in Security and Privacy, 1990.., 1990, IEEE Xplore DOI: 10.1109/RISP.1990.63857
- [7] Chandola, V., Banerjee, A., and Kumar, V. 2009. Anomaly detection: A survey. ACM Comput. Surv., 2009, 58 pages. <u>http://doi.acm.org/10.1145/1541880.1541882</u>
- [8] Hodge V J. and Austin J, A Survey of Outlier Detection Methodologies, Artificial Intelligence Review 22: 85–126, 2004, Kluwer Academic Publishers, doi: 10.1023/B:AIRE.0000045502.10941.a9
- [9] Box G E. P., Jenkins G M., Reinsel G C., Ljung G M., Time Series Analysis: Forecasting and Control, Fifth Edition, 2015, John Wiley & Sons.
- [10] Box G E. P. & Tiao G. C. (1975) Intervention Analysis with Applications to Economic and Environmental Problems, Journal of the American Statistical Association, 70:349, 70-79, DOI: <u>10.1080/01621459.1975.10480264</u>.
- [11] Saxena A., Goebel K., Simon D. and Eklund N., "Damage propagation modeling for aircraft engine run-to-failure simulation," 2008 International Conference on Prognostics and Health Management, Denver, CO, 2008, pp. 1-9, doi: 10.1109/PHM.2008.4711414.
- [12] Chang, Ih, George C. Tiao, and Chung Chen. "Estimation of Time Series Parameters in the Presence of Outliers." *Technometrics*30, no. 2 (1988): 193-204. doi:10.2307/1270165.
- [13] Statistics and Machine Learning Toolbox[™] User's Guide, last accessed on February 2020, <u>https://www.mathworks.com/help/releases/R2018a/pdf_doc/stats/stats.pdf</u>,.
- [14] Abraham B, Box G. E. P., "Bayesian Analysis of Some Outlier Problems in Time Series", 1979, Biometrika 66(2):229-236, DOI: 10.1093/biomet/66.2.229
- [15] Rousseeuw P J., Leroy A M., "Robust Regression and Outlier Detection", John Wiley & Sons.
- [16] The Prognostics Center of Excellence (PCoE) at Ames Research Center, Turbofan Engine Degradation Simulation Data Set, <u>https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognosticdata-repository/#turbofan</u>, May 2020 last accessed.
- [17] Predictive Maintenance Toolbox[™], May 2020 last accessed, <u>https://www.mathworks.com/products/predictive-maintenance.html</u>.
- [18] Similarity-Based Remaining Useful Life Estimation, May 2020 last accessed, <u>https://www.mathworks.com/help/predmaint/ug/similarity-based-remaining-useful-life-estimation.html?s_tid=srchtitle,</u>.