# Echo State Network for the Remaining Useful Life Prediction of a Turbofan Engine

Marco Rigamonti[1], Piero Baraldi[2], Enrico Zio[3], Indranil Roychoudhury[4], Kai Goebel[5], and Scott Poll[6]

[1, 2, 3]*Energy Department, Politecnico di Milano, Via Ponzio 34/3, Milan, 20133, Italy*
*marcomichael.rigamonti@polimi.it*
*piero.baraldi@polimi.it*
*enrico.zio@polimi.it*

[3] *Chair System Science and the Energy Challenge, Fondation Electricité de France (EDF), CentraleSupélec, Université Paris Saclay, Grande Voie des Vignes, 92290 Chatenay-Malabry, France*
*enrico.zio@ecp.fr*

[4] *Stinger Ghaffarian Technologies, Inc., NASA Ames Research Center, Moffett Field, California 94035*
*indranil.roychoudhury@nasa.gov*

[5, 6]*NASA Ames Research Center, Moffett Field, California 94035*
*kai.goebel@nasa.gov*
*scott.poll@nasa.gov*

## ABSTRACT

Among the various data-driven approaches used for RUL prediction, Recurrent Neural Networks (RNNs) have certain prima facie advantages over other approaches because the connections between internal nodes form directed cycles, thus creating internal states which enables the network to encapsulate dynamic temporal behavior and also to properly handle the noise affecting the collected signals. However, the application of traditional RNNs is limited by the difficulty of optimizing their numerous internal parameters and the significant computational effort associated with the training process. In this work, we explore the use of the Echo State Network (ESN), a relatively new type of Recurrent Neural Network (RNN). One of the main advantages of ESN is the training procedure, which is based on a simple linear regression. Unlike traditional RNNs, ESNs can be trained with fairly little computational effort, while still providing the generalization capability characteristic of RNNs. In this paper, we use Differential Evolution (DE) for the optimization of the ESN architecture for RUL prediction of a turbofan engine working under variable operating conditions. A procedure for pre-processing of the monitored signals and for identification of the onset of acceleration of degradation (i.e., the so-called *elbow point* in the degradation trend) will be shown. The datasets used to validate the approach have been taken from the NASA Ames Prognostics CoE Data Repository. These datasets were generated using a turbofan engine simulator, based on a detailed physical model that allows input variations of health-related parameters under variable operating conditions and records values from some specific sensor measurements. The results obtained on these data confirm the ESN's capability to provide accurate RUL predictions.



## 1. INTRODUCTION

Prognostics and Health Management (PHM) can help to achieve the operational reliability and safety requirements of engineered systems in a cost-effective way. A system's failure can be anticipated by an accurate prediction of the future evolution of a system's degradation state, resulting in increasing safety and reliability while, at the same time, reducing the overall maintenance costs (Bonissone, Xue, & Subbu, 2011). Therefore, PHM is currently receiving a lot of attention from industries such as aerospace, military, transportation, and energy production (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006; Pecht, 2008).

Data-driven methods, which rely on historical data and do not require physics-based models, are increasingly becoming more attractive (Bonissone et al., 2011). Among data-driven techniques for Remaining Useful Life (RUL) prediction, Recurrent Neural Networks (RNNs) are the most promising due to their capability of representing the dynamics of the degradation evolution (Lukoševičius & Jaeger, 2009) and their ability to encapsulate dynamic temporal behavior. While feedforward Artificial Neural Networks (ANNs) provide only a direct functional mapping

between input and output data (Samanta & Al-Balushi, 2003), the recurrent nature of RNNs, obtained by using feedback connections between the neurons of a layer and those of the preceding layers (Moustapha & Selmic, 2008), allow for the handling of noisy data and the processing of dynamic information. Several types of RNNs have been used in literature for prognostic purposes. In (Tse & Atherton, 1999), the authors developed a RNN-based prognostic system that used vibration data for predicting machine deterioration evolution. In (Samanta & Al-Balushi, 2003), a RNN was applied to the RUL prediction of a helicopter drivetrain system gearbox and the results were compared to those provided by a Support Vector Regression (SVR) approach. The author of (Heimes, 2008) proposed a RNN, whose architecture was optimized by means of an evolutionary algorithm and whose weights were set using an Extended Kalman Filter-based algorithm, and applied it to the RUL prediction of turbofan engines working under variable operating conditions. The proposal to train an Infinite Impulse Response-Locally Recurrent Neural Network (IIR-LRNN) online for modeling the dynamics of a next-generation nuclear reactor was presented in (Zio, Broggi, & Pedroni, 2009). An Adaptive Recurrent Neural Network (ARNN), whose weights are adaptively optimized using the recursive Levenberg-Marquardt (RLM) method, was proposed in (Liu, Saxena, Goebel, Saha, & Wang, 2010) and applied to the RUL prediction of Lithium-ion batteries. (Mahli, Yan, & Gao, 2011) proposed a modified RNN that was applied to the multi-step long-term prediction of bearing defect progression.

The main challenges for developing practical applications of RNNs are: i) the slow and computationally intensive training procedure, which also cannot guarantee the final convergence of the algorithm towards an accurate and robust model (Lukoševičius & Jaeger, 2009); and ii) the lack of guidelines for the definition of the RNN architecture (i.e., number of hidden layers, number of neurons in the hidden layers, etc).

In order to overcome these problems, a new approach for RNN training called Reservoir Computing (RC) was proposed in (Jaeger, 2001). RC involves randomly creating a RNN, called *Reservoir*, which remains unchanged during the training and is passively excited by the input signal, maintaining in its state a nonlinear transformation of the input history. The desired output signal is then generated as a linear combination of the neuron's signals produced by the input excited reservoir. The coefficients of the linear combination are the only parameters of the network that are optimized by using the teacher signal as a target (Lukoševičius & Jaeger, 2009).

Among RC approaches, Echo State Network (ESN) is one of the most interesting due to its intrinsic dynamic properties, its generalization capability and its fast training procedure. In practice, ESN consists of a large reservoir of sparsely connected neurons, whose output weights are obtained performing a linear regression of the teacher outputs on the reservoir internal states, that in turn depend on the received input history. The obtained ESN preserves the modeling capability typical of RNNs, while requiring a considerably shorter and less computationally intensive training process. Of particular interest is the so-called *echo state property*, which postulates that the effect of initial conditions should gradually vanish as time passes (Yildiz, Jaeger, & Kiebel, 2012). Although ESNs have been extensively investigated and used for the prediction of chaotic time series (Jaeger & Haas, 2004; Shi & Han, 2007; Li, Han, & Wang, 2012), they have so far seen limited use for RUL prediction of industrial systems (Peng, Wang., Wang, Liu, & Peng, 2012), (Morando, Jemei, Gouriveau, Zerhoumi, & Hissel, 2013), (Fink, Zio, & Weidmann, 2013). In (Peng et al., 2012a) the authors developed a prognostic model based on multiple ESN sub-models for the RUL prediction of turbofan engines. An ESN-based approach for the prediction of the RUL of industrial Fuel Cells was developed in (Morando et al., 2013), whereas (Fink et al., 2013) proposed a hybrid approach combining ESN and Conditional Restricted Boltzmann Machines (CRBM) for predicting the occurrence of railway operation disruptions. With respect to these works, a critical problem was setting the ESN architecture parameters, such as the size of the dynamical reservoir, the spectral radius, the connectivity, and input and output scaling and shifting factors. These parameters heavily influence the ESN modeling capability.

In order to overcome this problem in non-prognostic applications, parameters optimization has been carried out using, for example, a Particle Swarm Optimization (PSO) algorithm in (Rabin, Hossain, Ahsan, Mollah, & Rahman, 2013), and Genetic Algorithms (GAs) in (Ferreira & Ludermir, 2009) and in (Ferreira, Ludermir, & De Aquino, 2013).

Several types of GAs have been used for the structural optimization of Artificial Intelligence (AI) tools (Yan, Duwu, & Yongqing, 2007; Qu & Zuo, 2012; Vukicevic, Jovicic, Stojadinovic, Prelevic, & Filipovic, 2014), based on different strategies to generate variations of the parameter vectors and accepting a new parameter vector if and only if it reduces the value of the objective function of the optimization. This runs the risk of becoming trapped in a local minimum. In order to overcome this, (Storn & Price, 1997) proposed a simple heuristic method, called Differential Evolution (DE), which employs the difference of two randomly selected parameter vectors as the source of random variations for a third parameter vector. This approach has been shown to converge with an improved chance of finding the global minimum, regardless of the initial system parameter values. A DE approach has been successfully applied in (Heimes, 2008) for the automatic tuning of the parameters of a traditional RNN used for the

RUL prediction of turbofan engines. To the best of our knowledge, DE has not yet been applied to the optimization of an ESN architecture.

In this work, we discuss a prognostic approach based on the development of an optimized ESN model. The major novelties of the work are: i) the optimization of the ESN architecture using a DE approach, and ii) the use of the ESN for the prediction of the RUL of a degrading system.

The proposed approach has been verified with respect to a case study concerning the prediction of the RUL of a fleet of turbofan engines working under variable operating conditions. Data describing the evolution of 21 signals during the engine lives have been taken from the NASA Ames Prognostics CoE Data Repository (Saxena, Goebel, Simon, & Eklund, 2008). The prognostic results have been compared to those obtained by using other AI techniques, such as Extreme Learning Machine (ELM) (Huang, Zhu, & Siew, 2006) and the Fuzzy Similarity-based approach (FS) (Zio, Di Maio, & Stasi, 2010).

The remaining part of the paper is organized as follows: Section 2 illustrates the proposed prognostic approach. Section 3 shows the case study and the available data. Section 4 presents the pre-processing procedure, Section 5 illustrates the proposed approach for the elbow point identification. Section 6 presents the methods used within the proposed prognostic approach, their application to the case study, and compares the results to those obtained with other AI techniques. Finally, in Section 7, some conclusions and remarks are drawn.

## 2. PROBLEM FORMULATION AND APPROACH

The objective of this work is to develop a data-driven approach for the RUL prediction of a fleet of industrial components. The main challenges to be tackled are due to the noisy nature of real industrial data and to the intrinsic behavior variability among the fleet components, which can be caused by differences in the manufacturing process and in the environmental conditions. The proposed procedure is based on the following 3 steps: *i)* Pre-processing, *ii)* Elbow Point Detection, and *iii)* Prognostics. Figure 1 illustrates the proposed approach and can be summarized as follows:

- *i) Data Pre-processing*: This step is applied to the available raw data. It entails a) data normalization, which reduces the effects of the variable operating conditions on the signals, b) data filtering, which reduces the noise of the normalized signals, and c) the selection of prognostic signals based on the computation of prognostic measures. The data processing methods will be directly illustrated in the case study reported in Section 4.
- *ii) Elbow point detection*: This step of the procedure identifies the time instant at which the component degradation becomes observable, which we refer to as the *elbow point*. The Z-test based method used for the elbow point identification will be illustrated in Section 5. In the remaining part of the procedure, we will consider for each trajectory only the data subsequent to the time instant of the elbow point detection.
- *iii) Prognostics:* This step of the procedure predicts the component RUL. It is based on the analysis of the prognostic signal values after the elbow point. The method relies on ESN whose architecture is optimized using DE. Section 6.1 will briefly illustrate the ESN theory, while the DE method and its application will be presented in Section 6.2.
    - In the following section, we discuss the turbofan engine data we use as a case study for illustrating our approach. Then we describe the prognostic approach shown in Figure 1 in some detail and use the case study as a running example to exemplify each of the three steps.

| | **Pre-Processing** → | **Detection** → | **Prognostics** |
|---|---|---|---|
| **INPUT** | • Raw Data | • Significant prognostic signals | • Prognostic signals after elbow point |
| **METHOD** | • Normalization<br>• Exponential Filter<br>• Prognostic Measures | • Z-test | • Differential Evolution<br>• Echo State Network |
| **OUTPUT** | • Normalized and filtered significant prognostic signals | • Elbow point | • RUL prediction |

Figure 1. Block diagram of the proposed prognostic approach.

## 3. CASE STUDY

The proposed approach is verified with respect to the prediction of the RUL of a fleet of turbofan engines working under continuously varying operating conditions. The data used in this paper have been taken from the NASA Ames Prognostics CoE Data Repository (Saxena et al., 2008), and consists of 260 run-to-failure trajectories. Each trajectory is a 24-dimensional time series of different length, formed by 21 signals measured by sensors and 3 signals referring to the turbofan engines operating conditions (Altitude, Mach Number and Throttle Resolver Angle, TRA). These latter three signals indicate six different operating conditions, which significantly influence the values of the other 21 measured signals. Table 1 summarizes the main characteristics of the dataset.

Table 1. Dataset Characteristics.

| | |
|---|---|
| Number of Trajectories | 260 |
| Maximum Length | 378 |
| Minimum Length | 128 |
| Number of Signals | 21 |
| Number of Operating Conditions | 6 |

According to this, it is worth noting that methods for prognostics under variable operating conditions have been proposed by (Gasperin, Boskoski, & Juricic, 2011), (Hu, 2015), (Heimes, 2008) and (Peng et al., 2012a). (Gasperin et al., 2011) proposed an algorithm for the on-line estimation of the parameters of a varying physics-based model, which is then used for the RUL prediction of a gearbox under non-stationary operating conditions. In (Hu, 2015) a Particle Filter-Based approach for the estimation of the effects of the working condition on a physics-based degradation model and for the simultaneous prediction of the system RUL has been proposed. (Heimes, 2008) developed a RNN for the RUL prediction of turbofan engines working under variable operating conditions. Finally, in (Peng et al., 2012a), a prognostic approach based on multiple ESN sub-models for the RUL prediction of turbofan engines has been proposed.

### 3.1. C-MAPPS Dataset

These data have been generated using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) model that receives as input an evolving health indicator (i.e., a parameter representing the degradation level of an engine component) and provides as output the values of the signals influenced by the corresponding input health condition (Frederick, De Castro, & Litt, 2007). In particular, the C-MAPPS simulation model allows simulating the effects of faults and deterioration in any of the engine rotating components showed in Figure 2 on the monitored signals (i.e., the C-MAPPS output). Unlike the data used for the 2008 PHM Challenge, the data considered in this work have been taken from the "train_FD002.txt" file of the C-MAPPS dataset 2 (Saxena et al., 2008), which is characterized by the occurrence of a single failure mode only (i.e., the degradation of the High Pressure Compressor (HPC) of the engine). In order to better contextualize the development of the present work, it is important to point out that several works have been carried out on the 4 available CMAPPS datasets, and an exhaustive and complete survey about them is provided by (Ramasso & Saxena, 2014). In particular, this latter highlights that the majority of the works were considering the CMAPPS dataset 1, which is characterized by one operating condition and one failure mode. Just few works, such as (Peng, Xu, Liu, & Peng, 2012b; Li, Qian, & Wang, 2013; Zao & Willet, 2011; Ramasso, 2014; Wang, 2010), took into account the CMAPPS datasets 2 (i.e., the dataset used in this work) and 4, which, on the contrary, are characterized by 6 variable operating conditions.

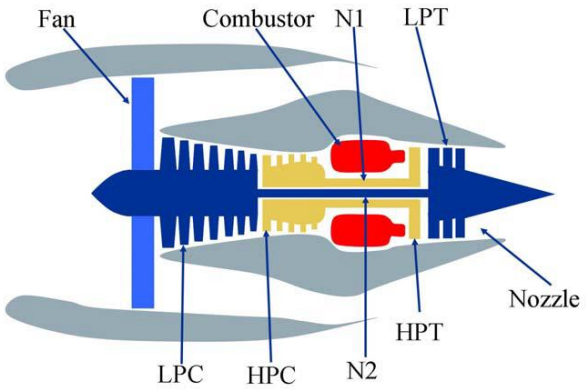



Figure 2. Turbofan Engine (Saxena et al., 2008)

In order to simulate realistic data, each of the run-to-failure trajectories considers an engine characterized by a different initial level of wear. Furthermore, process and measurement noises have been added during the simulation: in particular, the process noise was added to the input health indicator and therefore percolated through system dynamics, whereas the random measurement noise was added to the output signals. This multistage noise contamination resulted in complex noise characteristics often observed in real data, thus posing a realistic challenge to the accurate prediction of the engines' RUL. Furthermore, it is worth noting that no information is provided within the dataset about the real nature of the available signals: for example, it is not specified if a signal is representative of temperature, pressure, etc. The only provided information is about the engine operating conditions, which are described by three signals, (i.e., Altitude, Mach Number and TRA). Notice that the operating conditions significantly influence the values of the other 21 measured signals, and it may be difficult to distinguish the signal patterns trends that are due to

component degradation from those that are due to the changing operating conditions.

Thus, in order to perform an initial exploration of the available data, we have clustered the operating conditions by applying the Fuzzy C-means algorithm (Bezdek, Ehrlich, & Full, 1984) to the three corresponding signals. This allowed us to identify the different behaviors of the signals according to the corresponding operating condition, which are shown in Figure 3 with respect to Signal 11, whereas Figure 4 shows the values of Altitude, Mach Number and TRA in the six clustered operating conditions.

### 3.2. Dataset Partition

First, the available dataset made of 260 trajectories is divided into 3 subsets: *i*) a training set, *ii*) a test set, and *iii*) a validation set. The training set, which comprises 70 trajectories randomly selected, is used for ESN training. The test set, which comprises 60 trajectories randomly selected, is used only within the DE application for evaluating the prognostic performance of the network architectures. Finally, the validation set, comprising the remaining 130 trajectories, is used to evaluate the prognostic performance on trajectories, which have never been used during the prognostic model development, allowing the comparison with the performance of other prognostic approaches. The partition of the dataset is shown in Table 2.

Table 2. Dataset Partition

| Dataset | Number of Trajectories |
|---|---|
| Original | 260 |
| Training Set | 70 |
| Test Set | 60 |
| Validation Set | 130 |

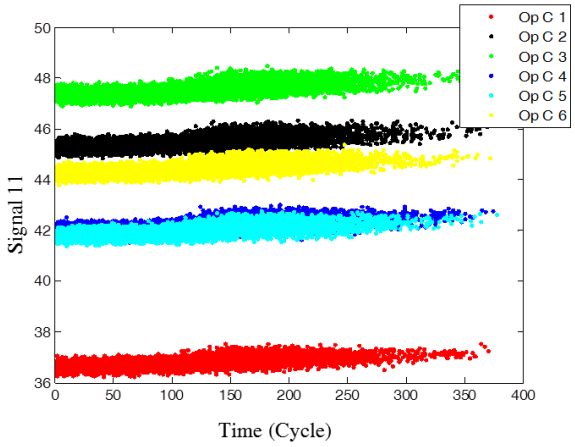



Figure 3. Values of Signal 11 in the 6 operating conditions

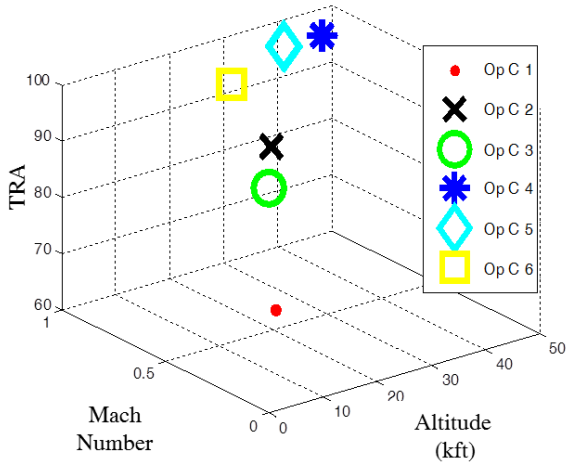



Figure 4. Values of Mach Number, Altitude, and TRA in the 6 operating conditions.

## 4. DATA PREPROCESSING

A three-step data preprocessing procedure has been applied to *i)* take out the influence of the operating conditions on the signal behavior, *ii)* reduce the signal noise, and *iii)* select the "prognostic signals", i.e., identify from among the 21 measured signals those containing information useful for the prognostic task and, thus, to be used as ESN inputs. With respect to step *i)*, signal values have been normalized taking into account signal ranges in the different operating conditions. Considering $N_{TR}$ run-to-failure trajectories, comprising $S$ signals, and $C$ different possible operating conditions varying during the whole life of the component, data are normalized by applying:

$$x_s^{norm}(t) = \frac{x_s^c(t) - \mu_s^c}{\sigma_s^c} \quad (1)$$

where $x_s^{norm}(t)$ represents the $s$-th normalized signal at the time instant $t$, $x_s^c(t)$ the $s$-th signal measured when the system is in the $c$-th operating condition, and $\mu_s^c$ and $\sigma_s^c$ are the mean and the standard deviation values of the $s$-th signal in the $c$-th operating condition.

These parameter values have been computed taking into account all the $N_{TR}$ =70 trajectories belonging to the training set. With respect to step *ii)*, noise has been reduced by applying an exponential filter. For ease of comprehension, the effects of steps *i)* and *ii)* of the data preprocessing procedure are shown in Figure 5.

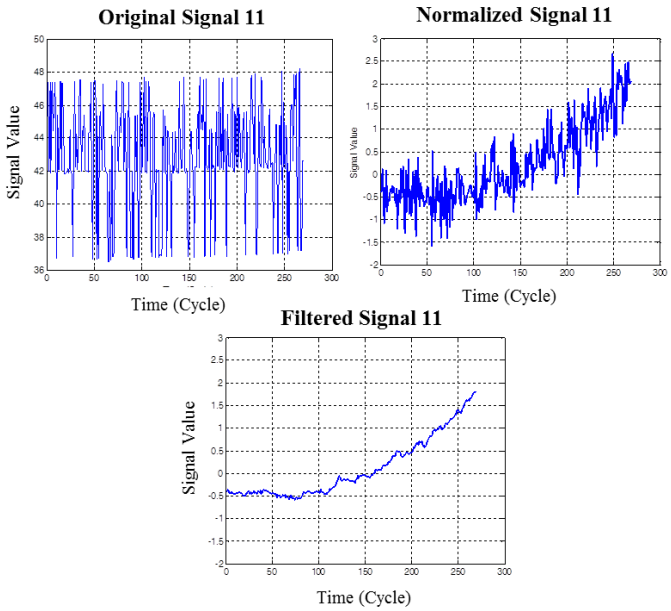



Figure 5. (Upper-left): Original behavior of Signal 11; (Upper-right): Normalized behavior of Signal 11; (Bottom): Filtered behavior of Signal 11.

The upper-left plot in Figure 5 shows the original behavior of Signal 11 during one transient. The upper-right plot shows the normalized behavior of Signal 11. Finally, the bottom plot shows the de-noising effects of the filtering step of the procedure.

### 4.1. Selection of Prognostic Signals

With respect to step *iii)*, the idea is to evaluate how well the signal represents the degradation process in order to eliminate those signals which do not adequately represent the degradation process and, thus, can reduce the ESN accuracy by carrying misleading and/or meaningless information. To this end, we consider the three prognostic measures of *Monotonicity*, *Prognosability*, and *Trendability* as proposed in (Coble, 2010).

Monotonicity is defined as:

$$\text{Mono}_s = mean_{N_{TR}}\left(\left|\frac{\#pos\, d/dx}{T^n - 1} - \frac{\#neg\, d/dx}{T^n - 1}\right|\right), \quad (2)$$

which is the average difference between negative slopes and positive slopes of the *s*-th signal within the $N_{TR}$ trajectories. A Monotonicity value close to 1 indicates a very monotonic signal, whereas a value close to 0 indicates a non-monotonic signal.

The Prognosability measure is defined as:

$$\text{Progno}_s = \exp\left(-\left(\sigma_s^{fail} \Big/ \left|\mu_s^{fail} - \mu_s^{Healthy}\right|\right)\right) \quad (3)$$

This measure focuses on the values of the *s*-th signal at the times of the components failures, and indicates how much they are spread with respect to the average variation of the signal during its whole life. Prognosability measures close to 1 tell that the failure values of the considered signals are similar, whereas measures close to 0 indicate that the failure values of the considered signals are very different between each other, thus complicating the prognostic task. Finally, Trendability of the *s*-th signal is defined as the minimum value of the correlation coefficients computed among all the $N_{TR}$ trajectories:

$$\text{Trend}_s = \min\left(\left|corrcoeff_{ij}\right|\right) i \neq j, i, j = 1,..,N_{TR}. \quad (4)$$

Intuitively, trendability represents how much the trajectories of the same signal are characterized by similar functional behavior. This measure can be used in this case study, since we are considering a fleet of components affected by the same failure mode, which is expected to be described by a similar functional behavior of the signals.

In order to evaluate different features considered for a specific task, the three prognostic measures have been aggregated into one parameter indicating the signal representativeness of the degradation process:

$$\text{Rep}_{\text{deg}} = w_m \cdot \text{Mono} + w_p \cdot \text{Progn} + w_t \cdot \text{Trend}, \quad (5)$$

where $w_m$, $w_p$, and $w_t$ represent the weights of Monotonicity, Prognosability, and Trendability, respectively. In this work, in order to give more importance to the Prognosability, $w_p$ has been set equal to 0.8, whereas $w_m$ and $w_t$ have been set equal to 0.05 and to 0.15, respectively. Notice that, depending on the objective of the study, different weights can be used. For example, if one were to consider components such as batteries, which may experience some degree of self-repair during their non-use periods, then using monotonicity of the degradation feature might not lead to desired results. Once the representativeness of the degradation process of a signal has been evaluated, the best set of features for the prognostic task can be identified.

Imposing a selection threshold equal to 0.8 (i.e., a signal is selected only if $Rep_{deg} > 0.8$), we have been able to identify the 6 most significant signals (out of the 21 signals available per trajectory), which correspond to Signals 2, 3, 4, 11, 15 and 17. Figure 6 shows the run-to-failure evolution of the 6 selected signals in trajectory #157.

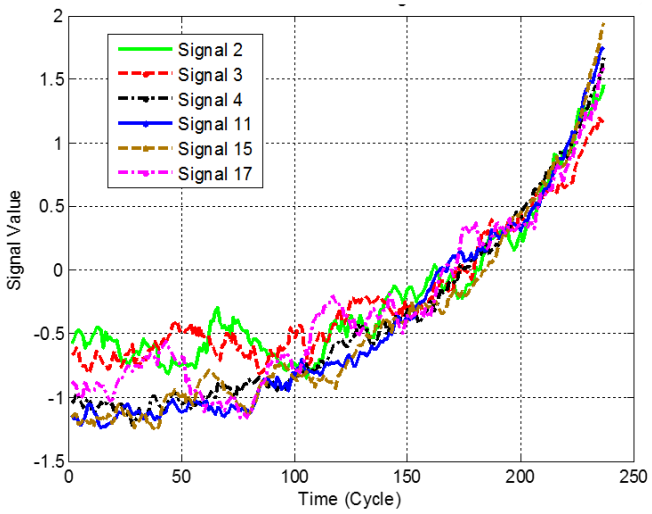



Figure 6. Run-to-failure evolution of the 6 selected signals in trajectory #157.

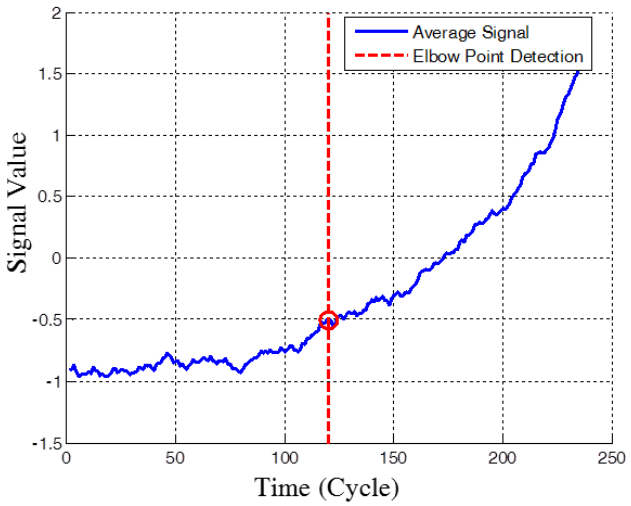



Figure 7. Elbow point identification for trajectory #157.

## 5. ELBOW POINT DETECTION

The premise of this work is that there is an accelerated degradation phase which is succinctly different from the phase of "ordinary" degradation. Such an acceleration can be brought on by a fault condition or by late-phase degradation mechanisms (which typically have properties of exponential growth). In either case, since the beginning of the accelerated degradation phase manifests at random time, in order to properly focus on the underlying physical relationship between the evolution behavior of the degrading signals and the corresponding decreasing component RUL, one needs to detect the onset of the accelerated degradation process. To that end one needs to find the *elbow point* in the signals, i.e. the time instant at which the degradation changes from an ordinary degradation phase to an accelerated degradation phase. Here, the 6 selected signals have been averaged to reduce the noise and the Z-Test proposed in (Daigle, Roychoudhury, Biswas, Koutsoukos, Patterson-Hine, & Poll, 2010) has been applied for change detection. This heuristic solution allows obtaining a satisfactory detection of the elbow point for each trajectory. In the remaining part of the paper, every time we refer to a specific trajectory, we consider only the data subsequent to the detected elbow point time instant and prognostics is applied only after the identification of the elbow point. Figure 7 shows the identified elbow point for the average of the six prognostic signals present in trajectory #157, which is represented by the dashed vertical line.

## 6. PROGNOSTICS

We use ESN for predicting the RUL. In the following, we first describe Echo State Networks, and then talk about how Differential Evolution can be used for optimizing the architecture of the ESN.

### 6.1. Echo State Networks

In this section, we briefly describe the ESN used for carrying out the prognostic step. Input to the model are measurements of signals correlated to the component degradation state, whereas the model output is the component RUL.

Figure 8 shows the generic architecture of an ESN, where the *reservoir*, i.e., a RNN used as a nonlinear temporal expansion function, is separated from the readout, which is the only part of the ESN to be trained (Lukoševičius & Jaeger, 2009). Some guidelines for producing good reservoirs are presented in (Jaeger, 2001) and (Jaeger, 2002) where, motivated by the intuitive goal of producing a rich set of dynamics, it is suggested to generate big, sparsely- and randomly-connected reservoirs. In practice, this means that the reservoir dimension $N$ should be sufficiently large, with a number of connections ranging from tens to thousands, dependent on the complexity of the task. The weight matrix $W$ is sparse, with connectivity value $C$, i.e. the fraction of internal neurons connected to each other that can vary from several to 50. Also, the weights of the connections are usually randomly generated from a uniform distribution symmetric around the zero value.

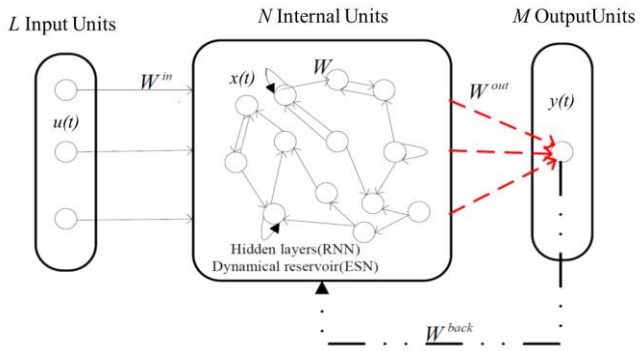



Figure 8. Basic architecture of the ESN (Peng et al., 2012a).

In this work we consider a discrete-time ESN with $L$ input units receiving at time $t$ the current signal measurements $u(t)=(u_1(t),\ldots,u_L(t))$; $N$ internal network units whose internal states are represented by $x(t)=(x_1(t),\ldots,x_N(t))$; and one output unit producing the output signals $y(t)=RUL(t)$. The activation of internal units $x(t)$ at time $t$ is obtained using:

$$x(t) = f\,(W^{in}u(t) + Wx(t-1) + W^{back}\,y(t-1)), \qquad (6)$$

where $f=(f_1 \ldots f_N)$ are the internal units activation functions, which are typically sigmoidal, $W^{in}=(w_{ij}^{in})$ is the $N\times L$ input weights matrix, $W=(w_{ij})$ is the $N\times N$ internal weights matrix, and $W^{back}=(w_{ij}^{back})$ is a $N\times M$ output feedback weights matrix. The input weights $W^{in}$ and the output feedback weights $W^{back}$ are usually dense and randomly generated from a uniform distribution. In order to deal with a specific task, both $W^{in}$ and $W^{back}$ can be scaled: the scaling of $W^{in}$ ($IS$) and shifting of the input ($IF$) depend on how much nonlinearity of the processing unit the task needs. If the inputs are close to 0, the sigmoidal neurons tend to operate with activations close to 0, where they are essentially linear, while inputs far from 0 tend to drive them more towards saturation where they exhibit more nonlinearity; the same idea drives the choice of the output scaling ($OS$) and shifting ($OF$), whose values affect the range of the trained $W^{out}$ and might lead to an unstable condition. Finally, the scaling of $W^{back}$ ($OFB$) is, in practice, limited by a threshold at which the ESN starts to exhibit an unstable behavior, i.e., the output feedback loop starts to amplify the output entering into a diverging generative mode (Jaeger, 2001).

The output equation of ESN is:

$$y(t)=f_{out}\,(W^{out}\,(u(t),x(t),y(t-1))), \qquad (7)$$

where $f_{out}=(f_{out}^1 \ldots f_{out}^M)$ are the output unit activation functions, which are typically linear, and $W^{out}=(w_{ij}^{out})$ is the $M\times(L+N+M)$ output weights matrix.

ESN training attempts to find optimal values for $W^{out}$ and is performed according to the procedure reported in Appendix A, which is based on the use of a Least Squares linear regression to minimize the error between the network output and a target signal on a set of training data. An important characteristic of ESNs is the echo state property (Jaeger, 2001), which states that the effect of a previous state $x(t)$ and a previous input $u(t)$ on a future state $x(t+k)$ should vanish gradually as time passes, and not persist or even get amplified. For most practical purposes, the echo state property is assured if the reservoir weight matrix $W$ is scaled so that its spectral radius ($SR$) $\rho(W)$ (i.e., the largest absolute eigenvalue of $W$) satisfies $\rho(W)<1$. Once the ESN has been trained, it can be used to predict the output $y(t)$ by applying Eq. (6) first, and then Eq. (7) to the input $u(t)$.

**Example: ESN Input Creation**

The ESN input are the 6 selected signals and 6 synthetic signals, $x_s^n$, which simulate the component behavior in healthy conditions. These 6 synthetic signals have been created according to the following equation:

$$x_s^n(t) = \mu_s^{n,Healthy} + \nu(t) \qquad \nu(t)\sim N\left(0,\sigma_s^{n,Healthy}\right) \qquad (8)$$

where $\mu_s^{n,Healthy}$ represents the average of the $s$-th signal during the healthy state of the $n$-th component degradation trajectory (i.e., before the detection of the elbow point) and $\sigma_s^{n,Healthy}$ represents the standard deviation of the $s$-th signal in the healthy state.

The reason behind the use of these 6 synthetic signals is to facilitate the ESN in identifying the difference between the current and the expected signal values in healthy conditions, which is an indicator of the component degradation and can help in the prediction of the component RUL.

### 6.2. Differential Evolution for ESN Architecture Optimization

A difficulty that is typically encountered during the development of ESN is the setting of the parameters characterizing the architecture of the network, such as the size of dynamical reservoir $N$, the spectral radius $SR$, the connectivity $C$, the input units scaling $IS$, the input units shift $IF$, the output units feedback $OFB$, the output units scaling $OS$ and the output units shift $OF$. According to (Jaeger, 2002), "the success of the modeling task of an ESN depends crucially on the nature of the excited dynamic" that depends on the network structure. Somewhat unsatisfying is that (Jaeger, 2002) also states that "the successful application of an ESN approach involves a good judgment on the dynamic excited inside the reservoir", and this judgment ability can only grow with the experimenter's

personal experience. In order to overcome this difficulty, in this work we propose to apply Differential Evolution (DE) for the optimization of the ESN architecture and of the network parameters, thus allowing a more methodical determination of the ESN architecture.

**Multi-Objective Differential Evolution**

DE is a parallel, direct, genetic-algorithm-based search method which utilizes a population of *NP* parameter vectors $x_{i,G}$, $i = 1, 2, \ldots, NP$, called *chromosomes*, for an iterative search of optimal solutions with respect to some objective functions. The initial vector population is sampled randomly from a uniform probability distribution covering the parameter domain space. The objective of the method is to identify the best chromosomes, i.e., the best parameters vectors that lead to the optimal values of the objective functions. Depending on the application, the DE can be single-objective, or multi-objective. In practice, the DE approach is based on a three-step procedure: *i) mutation*, which generates new parameter vectors by adding the weighted difference between two population vectors to a third vector, where each of these three vectors has been randomly selected; *ii) crossover*, which mixes the mutated vector parameters with those of another predetermined vector, the target vector, to yield the so-called trial vector; and *iii) selection*, which evaluates the objective functions of the trial vector and, if their values are better than those obtained with the target vector, keeps the trial vector in the population for the new generation in replacement of the target one. Since each chromosome in the population must serve once as the target vector to be compared to a trial vector, *NP* competitions take place in one generation. Details on DE theory and application can be found in Appendix B and in (Storn & Price, 1997).

**DE Application**

We resort to a DE Multi-Objective (MO) approach considering three different prognostic performance indicators as objective functions of the optimization: *i)* the Cumulative Relative Accuracy (CRA) (Saxena, Celaya, Saha, Saha, & Goebel, 2010), *ii*) the Alpha-Lambda (α–λ) metric (Saxena et al., 2010), and *iii*) the Steadiness Index (SI) (Olivares, Cerda Muñoz, Orchard, & Silva, 2013). In the following paragraphs, the objective functions are defined with reference to the RUL prediction of a single degradation trajectory. Then, in the application, the three global objective functions are calculated (as averages) over a set of test trajectories.

The ***Cumulative Relative Accuracy*** is the normalized weighted sum of Relative Accuracy (RA) values, computed at specific $t_\lambda$ time instances.

$$CRA = \frac{1}{|p_\lambda|} \sum_{i \in p_\lambda} w(r(i)) RA_\lambda , \tag{9}$$

where $w(r(i))$ is a weight factor function of the RUL, $p_\lambda$ is the set of all time instants at which a RUL prediction is made for a degradation trajectory, $|p_\lambda|$ is the cardinality of the set and $RA_\lambda$ is defined as the relative error of the RUL prediction at time $t_\lambda$. In this work, all the weight factors $w(r(i))$ have been set equal to 1, $p_\lambda$ is constructed using 9 time instants corresponding to 10%, 20%,…, 90% of the component life, and $RA_\lambda$ is defined by:

$$RA_\lambda = \frac{\left| \hat{RUL}_\lambda - RUL_\lambda \right|}{RUL_\lambda}, \tag{10}$$

where $\hat{RUL}_\lambda$ is the predicted RUL at time $t_\lambda$ and $RUL_\lambda$ is the ground truth for RUL at time $t_\lambda$. Small values of $RA_\lambda$ indicate more accurate predictions.

***α-λ Metric*:** The *α-λ* metric is defined as a binary metric that evaluates whether the prediction accuracy at specific time instant $t_\lambda$ falls within specified *α*-bounds, which are expressed as percentage of the actual $RUL_\lambda$ at $t_\lambda$.

$$\alpha - \lambda_{t_\lambda} = \begin{cases} 1 \;\; if \; (1-\alpha) \cdot RUL_\lambda \le \hat{RUL}_\lambda \le (1+\alpha) \cdot RUL_\lambda \\ 0 \;\; Otherwise \end{cases} \tag{11}$$

where $\lambda$ refers to the $t_\lambda \in p_\lambda$ instant at which the prediction is performed and $\alpha$ is the percentage value defining the acceptance confidence bounds. In this work, the $\alpha$ value has been set equal to 20%. Furthermore, it has to be pointed out that, for a single prediction, the *α-λ* accuracy results in a binary vector of 9 elements: therefore, in this work we consider the average of the 9 obtained elements as the value of the considered objective function to be maximized:

$$\overline{\alpha - \lambda} = \frac{1}{|p_\lambda|} \sum_{\lambda=1}^{|p_\lambda|} \alpha - \lambda_{t_\lambda} . \tag{12}$$

***Steadiness Index*:** the SI measures the volatility of the expected value of the failure time prediction $\bar{T}$. It is defined by:

$$SI_t = \sqrt{\mathrm{var}(\bar{T}_{(t-\Delta t):t})}, \tag{13}$$

where $\Delta t$ is the length of a sliding time window. In order to focus on the stability of the end-of-life prediction over the whole component life, in this paper we take $\Delta t$ equal to the trajectory length.

The three metrics selected in this paper have been chosen due to their representativeness of the ESN prediction accuracy and stability. In fact, the $\overline{\alpha - \lambda}$ metric indicates how many times, on average, the RUL prediction falls within two relative confidence bounds; the CRA metric provides an average estimation of the RUL prediction relative error; and finally, the SI metric provides an

indication of how stable is the prediction of the component end of life during the whole monitoring process. Notice that the CRA metric, differently from the $\overline{\alpha-\lambda}$ metric, quantifies the amount of the error and, being a relative measure, tends to enlarge errors made at the end of the system life. Since $\overline{\alpha-\lambda}$ and CRA estimate the RUL accuracy from a different point of view, we require the optimization of both of them. Steadiness is optimized to facilitate maintenance decisions, which would be hampered by unstable RUL predictions.

**Example: DE Application and Best Solution Identification**

We have applied the DE algorithm for identifying the best ESN architecture for our three prognostic objectives. To this end, we have considered a DE population of *NP*=200 chromosomes formed by 8 parameters representing the ESN architecture and varying in the ranges reported in Table 3.

Table 3. ESN Parameters Search Space

| Parameter | Min Value | Max Value |
|---|---|---|
| *N* | 10 | 700 |
| SR | 0.05 | 1 |
| C | 0.01 | 0.5 |
| IS | $10^{-7}$ | 1 |
| IF | -1 | 1 |
| OFB | $10^{-7}$ | 1 |
| OS | $10^{-7}$ | 1 |
| OF | -1 | 1 |

Figure 9 illustrates the Pareto front corresponding to the DE last population: each solution of the Pareto front is optimal, since no other superior solutions have been obtained when all the three objectives (i.e., CRA, $\overline{\alpha-\lambda}$ and SI) are considered (Zitzler & Thiele, 1999).

In order to identify a trade-off solution among those belonging to the Pareto front, we resorted to the TOPSIS method (Chen & Hwang, 1992), based on the computation of the solution relative closeness to the optimal ideal solution (Opricovic & Tzeng, 2004). Notice that the selected compromise solution, which is represented in the three subfigures of Figure 9 by the squared marker (and which will be considered in the remaining part of the paper) is characterized by the second best SI, the third best CRA, and by the second worst $\alpha$–$\lambda$ value: however, regardless of the single objective function, the selected solution is the closest to the ideal one. The values of the parameters characterizing the ESN, which has been selected as best compromise solution, are reported in Table 4.

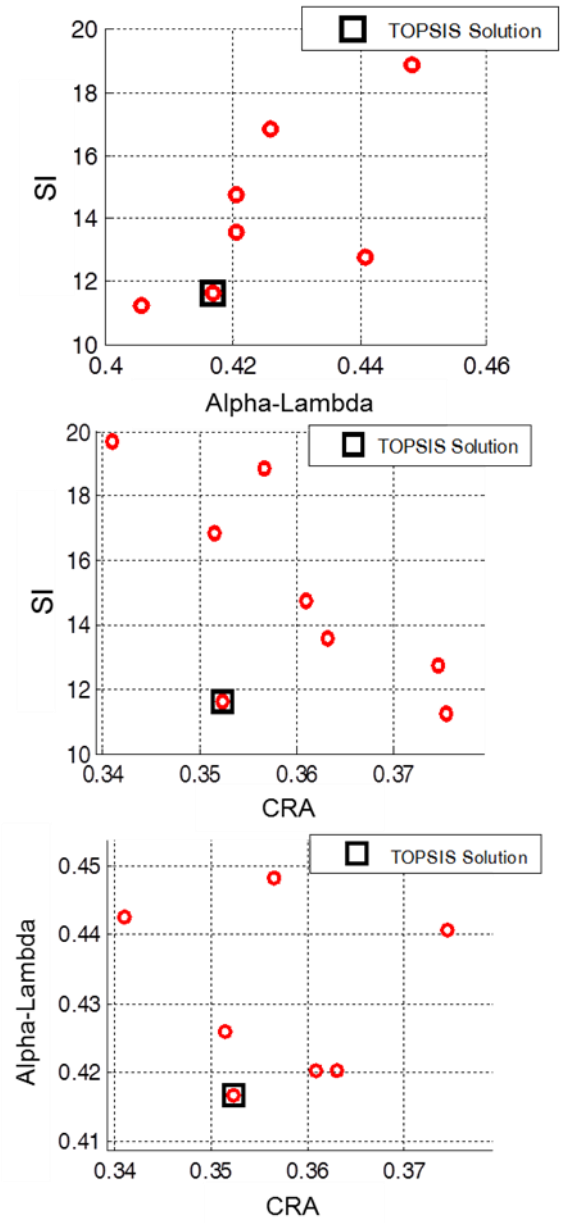



Figure 9. Bi-directional representation of the Pareto front of the optimal solutions.

Notice that the selected ESN is characterized by a small number of reservoir neurons (*N*), is relatively highly connected (*C*), has a large spectral radius value (*SR*), and the effects of the output feedback into the reservoir (*OFB*) have been limited by scaling them with a low factor. The obtained ESN parameters, especially the limited number of neurons and the high connectivity value, imply that the network properties required for accomplishing the prognostic task in the case considered do not entail large non-linear and dynamical capability (Jaeger, 2002).

Table 4. Structure parameters of the best compromise network

| N | SR | C | IS | IF | OFB | OS | OF |
|---|---|---|---|---|---|---|---|
| 15 | 0.92 | 0.43 | 0.45 | -0.99 | $3.5 \cdot 10^{-5}$ | $3.8 \cdot 10^{-3}$ | 0.69 |

### 6.3. RUL Prediction

After the identification of the parameters reported in Table 4, a 10-fold Cross Validation (CV) has been performed in order to robustly evaluate the predictive performance of the ESN with respect to the three prognostic metrics described in Section 6.2. The ESN predictive performance is also compared to that of an Extreme Learning Machine-based (ELM) (Huang et al., 2006; Fink, Weidmann, & Zio, 2014) and a Fuzzy Similarity-based approach (FS) (Zio et al., 2010). ELM has been chosen for the comparison since the concept behind it is similar to that of ESN, but they do not exploit recurrent connections (Huang et al., 2006), whereas the FS-based approaches has been chosen because it has been shown to be able to provide satisfactory RUL predictions in several different prognostic applications (Bonissone & Varma, 2005; Zio et al., 2010). In order to compare the prognostic performance of these methods, we resorted to the same prognostic metrics used during the DE Optimization described in Section 6.2, which have been used also in (Peng et al., 2012b) and (Li et al., 2013). It has to be pointed out that, differently from the PHM Challenge 2008, in this work we considered only the training part of the CMAPPS dataset 2, which consists only of run-to-failure trajectories. Furthermore, since we had the possibility to evaluate the prognostic performance at any time of the validation trajectories, we considered the overall performance of the developed prognostic model, without resorting to the timeliness measure, which on the contrary has been used for the final performance evaluation of the PHM Challenge 2008 (Ramasso & Saxena, 2014) and which considers only the RUL prediction at a single prefixed time for each degradation trajectory. Table 5 reports the results of the comparison: for each method we reported the metrics average value computed over the 10 CV iterations and its standard deviation. Notice that the proposed DE-ESN approach outperforms both the FS and the ELM approaches in all the considered prognostic metrics. In particular, the CRA value shows that the ESN provides an average relative error on the RUL prediction which is 5% lower than that of the FS, and 11% lower than that of the ELM**;** the SI value shows that the stability of the component end-of-life prediction provided by the ESN is the most satisfactory, although it is close to that provided by the FS. Finally, with respect to the $\alpha$–$\lambda$ metric, the average value shows that the ESN approach is able to provide RUL predictions that, in 38% of cases lie within relative boundaries equal to the 20% of the corresponding real RUL, whereas the same values for both the FS and the ELM performance are significantly lower. An analysis of the $\alpha$–$\lambda$ metric shows that even if the performance of the three investigated methods are comparable at the beginning of the components life (i.e., with respect to $t_{10\%}$, $t_{20\%}$ and $t_{30\%}$ time instants), the ESN RUL predictions are clearly outperforming those of the other two methods when time instants closer to the component end of life, such as $t_{60\%}$, $t_{70\%}$ and $t_{80\%}$, are considered. This confirms the superiority of the proposed DE-ESN method with respect to both ELM and FS, for this case study.

Table 5. Comparison among the prognostic performances provided by DE-ESN, FS, and ELM

| | | DE-ESN | FS | ELM |
|---|---|---|---|---|
| **Relative Accuracy** | | **0.37 ± 0.03** | **0.42 ± 0.03** | **0.48 ± 0.04** |
| **Steadiness** | | **12.4 ± 1.2** | **12.7 ± 0.7** | **15.3 ± 2.2** |
| $\alpha$–$\lambda$ | **10%** | 0.44 ± 0.03 | 0.43 ± 0.04 | 0.43 ± 0.03 |
| | **20%** | 0.43 ± 0.04 | 0.45 ± 0.04 | 0.41 ± 0.04 |
| | **30%** | 0.43 ± 0.03 | 0.43 ± 0.05 | 0.38 ± 0.03 |
| | **40%** | 0.38 ± 0.04 | 0.38 ± 0.03 | 0.34 ± 0.03 |
| | **50%** | 0.39 ± 0.05 | 0.36 ± 0.03 | 0.29 ± 0.04 |
| | **60%** | 0.39 ± 0.04 | 0.31 ± 0.03 | 0.27 ± 0.04 |
| | **70%** | 0.40 ± 0.04 | 0.28 ± 0.03 | 0.26 ± 0.04 |
| | **80%** | 0.37 ± 0.04 | 0.26 ± 0.04 | 0.25 ± 0.03 |
| | **90%** | 0.23 ± 0.03 | 0.18 ± 0.04 | 0.19 ± 0.03 |
| **Average $\alpha$–$\lambda$** | | **0.38** ± **0.04** | **0.34** ± **0.04** | **0.31** ± 0.04 |

Figure 10 shows the three RUL predictions obtained for a representative degradation trajectory of the validation set. As expected, the ESN RUL prediction (i.e., the solid line) is closer to the true RUL (i.e., the dotted straight line) than that provided by both ELM (the dashed line), which is largely overestimating, and FS (the dashed-dotted line), which is slightly overestimating. Nonetheless, it should be noted that – compared to the FS – the ESN method has higher variability throughout the prediction interval.

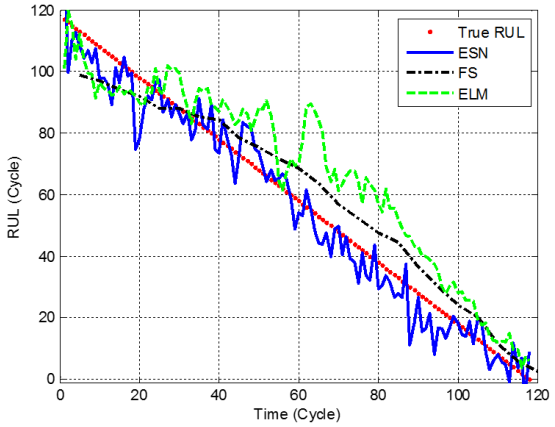



Figure 10. RUL prediction of trajectory #157 obtained with ESN, FS, and ELM.

## 7. CONCLUSION AND FUTURE WORK

In this work, we have proposed an approach for the development of a prognostic model for industrial components working in variable operating conditions, based on ESN for RUL prediction. ESN has been chosen due to the capability of dealing with time-varying signals, its generalization capability, and ability to handle noisy data. The ESN training procedure is considerably shorter and less computationally intensive than that of other recursive models such as RNN. In order to overcome the main drawbacks of ESN, i.e., the need of expert opinion for the design of its architecture and the setting of its several characteristic parameters, we have proposed to use a DE algorithm for the automatic optimization of the ESN. In fact, although the ESN creation requires the expert setting of several parameters, the proposed procedure relying on the DE optimization allows the user to obtain an ESN tailored and optimized for the specific task. The approach has been applied to a case study concerning the RUL prediction of turbofan engines, taken from the NASA Ames Prognostics CoE Data Repository, and the performance of the proposed method has been compared to those provided by an ELM-based and a FS-based approach. The results show the superior capability of the ESN in generalizing the behavior of similar degrading components, resulting in more accurate, stable and reliable predictions of the components' RUL. As part of future work, we would like to look into the variability characteristics of the output, extend this approach to handle multiple faults, and apply this approach to the test data that was provided as part of the PHM 2008 Data Challenge. Furthermore, we aim to develop an ensemble strategy for the component RUL prediction: the idea is to exploit different ESN models that, being tailored to different characteristics of the available trajectories, allow for better focus on the characteristics of the monitored trajectory, resulting in a more accurate and robust RUL prediction.

## REFERENCE


Bezdek, J. C., Ehrlich, R., Full, W., (1984). FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10.2: pp. 191-203.

Bonissone, P. P., Varma, A., (2005). Predicting the best units within a fleet: prognostic capabilities enabled by peer learning, fuzzy similarity, and evolutionary design process. *The 14th IEEE International Conference on Fuzzy Systems, 2005* (pp. 312-318).

Bonissone, P.P., Xue, F., Subbu, R., (2011). Fast meta-models for local fusion of multiple predictive models. *Applied Soft Computing Journal*, 11 (2), pp. 1529-1539.

Chen, S.J., Hwang, C.L., (1992). *Fuzzy Multiple Attribute Decision Making: Methods and Applications*. Springer- Verlag, Berlin.

Coble, J. B. (2010). Merging data sources to predict remaining useful life–an automated method to identify prognostic parameters. *Doctoral dissertation*.

Daigle, M.J., Roychoudhury, I., Biswas, G., Koutsoukos, X.D., Patterson-Hine, A., Poll, S., (2010). A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 40 (5), art. no. 5504182, pp. 917-931.

Ferreira, A.A., Ludermir, T.B. (2009). Genetic algorithm for reservoir computing optimization. *Proceedings of the international joint conference on neural networks* – IJCNN 2009, Atlanta (pp. 811–815).

Ferreira, A.A., Ludermir, T.B., De Aquino, R.R.B., (2013). An approach to reservoir computing design and training. *Expert Systems with Applications*, 40 (10), pp. 4172-4182.

Fink, O., Zio, E., Weidmann, U., (2013). Predicting time series of railway speed restrictions with time-dependent machine learning techniques. *Expert Systems with Applications*, 40 (15), pp. 6033-6040.

Fink, O., Weidmann, U., Zio, E., (2014). Extreme learning machines for predicting operation disruption events in railway systems. *Safety, Reliability and Risk Analysis: Beyond the Horizon - Proceedings of the European Safety and Reliability Conference, ESREL* 2013, pp. 1781-1787.

Frederick, D., DeCastro, J., Litt, J., (2007). User's Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS). *NASA/ARL*, Technical Manual TM 2007-215026.

Gasperin, M., Boskoski, P., Juricic, D., (2011). Model-based prognostics under non-stationary operating conditions. In *Annual Conference of the Prognostics and Health Management Society* (pp. 831-853).

Heimes, F.O., (2008). Recurrent neural networks for remaining useful life estimation. *2008 International Conference on Prognostics and Health Management*, PHM 2008, art. n. 4711422.

Hu, Y., (2015). Development of prognostics and health management methods for engineering systems operating in evolving environments. *PhD Thesis, Politecnico di Milano*, 2015.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70 (1-3), pp. 489-501.

Jaeger, H., (2001). The echo state approach to analyzing and training recurrent neural networks. *Technical Report GMD Report 148*, German National Research Center for Information Technology.

Jaeger, H., (2002). A Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the Echo state network approach. *Technical Report GMD Report 159*, German National Research Center for Information Technology.

Jaeger, H., Haas, H., (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304 (5667), pp. 78-80.

Li, D., Han, M., Wang, J., (2012). Chaotic time series prediction based on a novel robust echo state network. *IEEE Transactions on Neural Networks and Learning Systems*, 23 (5), art. no. 6177672, pp. 787-797.

Li, X., Qian, J., Wang, G. (2013). Fault prognostic based on hybrid method of state judgment and regression. *Advances in Mechanical Engineering*, 2013(149562), 1-10.

Liu, J., Saxena, A., Goebel, K., Saha, B., Wang, W., (2010). An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. *Annual Conference of the Prognostics and Health Management Society*, PHM 2010.

Lukoševičius, M., Jaeger, H., (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3 (3), pp. 127-149.

Malhi, A., Yan, R., Gao, R.X., (2011). Prognosis of defect propagation based on recurrent neural networks. *IEEE Transactions on Instrumentation and Measurement*, 60 (3), art. no. 5710193, pp. 703-711.

Morando, S., Jemei, S., Gouriveau, R., Zerhouni, N., Hissel, D., (2013). Fuel Cells prognostics using echo state network. *IECON Proceedings (Industrial Electronics Conference)*, art. no. 6699377, pp. 1632-1637.

Moustapha, A. I., Selmic, R. R., (2008). Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection. *IEEE Trans. Instrum. Meas.*, vol. 57, no. 5, pp. 981–988.

Olivares, B.E., Cerda Muñoz, M.A., Orchard, M.E., Silva, J.F., (2013). Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena. *IEEE Transactions on Instrumentation and Measurement*, 62 (2), art. no. 6302189, pp. 364-376.

Opricovic, S., Tzeng, G.-H., (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, 156 (2), pp. 445-455.

Pecht, M.G., (2008). Prognostics and Health Management of Electronics. *Prognostics and Health Management of Electronics*, pp. 1-315.

Peng, Y., Wang, H., Wang, J., Liu, D., Peng, X., (2012a). A modified echo state network based remaining useful life estimation approach. *PHM 2012 - 2012 IEEE Int. Conf. on Prognostics and Health Management: Enhancing Safety, Efficiency, Availability, and Effectiveness of Systems Through PHM Technology and Application*, Conference Program, art. no. 6299524.

Peng, Y., Xu, Y., Liu, D., Peng, X. (2012b). Sensor selection with grey correlation analysis for remaining useful life evaluation. In *Annual Conference of the PHM Society*.

Qu, J., Zuo, M.J., (2012). An LSSVR-based algorithm for online system condition prognostics. *Expert Systems with Applications*, 39 (5), pp. 6089-6102.

Rabin, M.J.A., Hossain, M.S., Ahsan, M.S., Mollah, M.A.S., Rahman, M.T., (2013). Sensitivity learning oriented non-monotonic multi reservoir echo state network for short-term load forecasting. *2013 International Conference on Informatics, Electronics and Vision*, ICIEV 2013, art. no. 6572692.

Ramasso E., Saxena, A., (2014). Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. *International Journal of Prognostics and Health Management*, , 2014, 5 (2), pp.1-15.

Ramasso, E., (2014). Investigating computational geometry for failure prognostics. *International Journal of Prognostics and Health Management*, 2014, 5(5), pp 1-18.

Samanta, B., Al-Balushi, K., (2003). Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical System Signal Processing*, vol. 17, no. 2, pp. 317–328.

Saxena, A., Goebel, K., Simon, D., Eklund, N., (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *2008 International Conference on Prognostics and Health Management*, PHM 2008, art. no. 4711414.

Saxena, A., Celaya, J., Saha, B., Saha, S., Goebel, K., (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management*, 1 (1).

Shi, Z., Han, M., (2007). Support vector echo-state machine for chaotic time-series prediction. *IEEE Transactions on Neural Networks*, 18 (2), pp. 359-372.

Storn, R., Price, K., (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11 (4), pp. 341-359.

Tse, P.W., Atherton, D.P., (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. *Journal of Vibration and Acoustics, Transactions of the ASME*, 121 (3), pp. 355-362.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., Wu, B., (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, Wiley, New York.

Vukicevic, A.M., Jovicic, G.R., Stojadinovic, M.M., Prelevic, R.I., Filipovic, N.D., (2014). Evolutionary assembled neural networks for making medical decisions with minimal regret: Application for predicting advanced bladder cancer outcome. *Expert Systems with Applications*, 41 (18), pp. 8092-8100.

Wang, T., (2010). Trajectory similarity based prediction for remaining useful life estimation. *PhD Thesis, University of Cincinnati*, 2010.

Yan, T., Duwu, D., Yongqing, T., (2007). A new evolutionary neural network algorithm based on improved genetic algorithm and its application in power transformer fault diagnosis. *2nd International Conference on Bio-Inspired Computing: Theories and Applications*, BICTA 2007, art. no. 4806406, pp. 1-5.

Yildiz, I.B., Jaeger, H., Kiebel, S.J., (2012). Re-visiting the echo state property. *Neural Networks*, 35, pp. 1-9.

Zio, E., Broggi, M., Pedroni, N., (2009). Nuclear reactor dynamics on-line estimation by Locally Recurrent Neural Networks. *Progress in Nuclear Energy*, 51 (3), pp. 573-581.

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), pp. 482-491.

Zitzler, E., Thiele, L., (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3 (4), pp. 257-271.

## BIOGRAPHIES

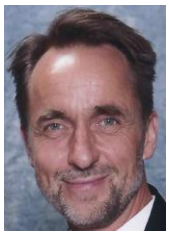
**Marco Rigamonti** (MSC in Nuclear engineering. Politecnico di Milano, December 2012) is pursuing his PhD in Energetic and Nuclear Science and Technology at Politecnico di Milano (Milan, Italy). He is co-author of 4 works accepted for publications on international journals.

**Piero Baraldi** (PhD in nuclear engineering, Politecnico di Milano, 2006) is professor of Nuclear Engineering at the department of Energy at the Politecnico di Milano. He has been Technical Committee Co-chair of the European Safety and

Reliability Conference, ESREL2014, and Technical Programme Chair of the 2013 Prognostics and System Health Management Conference (PHM-2013). He is co-author of 2 books and more than 100 papers on international journals and proceedings of international conferences.

**Enrico Zio** (Nuclear Engineer Politecnico di Milano (1991); MSc in mechanical engineering, University of California, Los Angeles, UCLA (1995); PhD in nuclear engineering, Politecnico di Milano (1995); PhD in Probabilistic Risk Assessment, Massachusetts Institute of Technology, MIT (1998); Full professor, Politecnico di Milano (2005-); Director of the Chair on Complex Systems and the Energy Challenge at Ecole Centrale Paris and Supelec, Fondation Europeenne pour l'Energie Nouvelle – EdF (2010-present); Chairman of the European Safety and Reliability Association-ESRA (2010- present. He is co-author of seven books and more than 250 papers on international journals.

**Indranil Roychoudhury** received the B.E. (Hons.) degree in Electrical and Electronics Engineering from Birla Institute of Technology and Science, Pilani, Rajasthan, India in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, Tennessee, USA, in 2006 and 2009, respectively. Since August 2009, he has been with SGT, Inc., at NASA Ames Research Center as a Computer Scientist. His research interests include hybrid systems modeling, model-based diagnostics and prognostics, distributed diagnostics and prognostics, and Bayesian diagnostics of complex physical systems. Dr. Roychoudhury is a Senior Member of the IEEE and a member of the Prognostics and Health Management Society.



**Kai Goebel** received the degree of Diplom-Ingenieur from the Technische Universitat Munchen, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is currently the Technical Area Lead of the Discovery and Systems Health Technology Area at NASA Ames Research Center. He also coordinates the Prognostics Center of Excellence and is the Technical Lead for Real-Time Safety Modeling with NASA's SMART-NAS project. Prior to joining NASA in 2006, he was a Senior Research Scientist at General Electric Corporate Research and Development Center since 1997. He was also an Adjunct Professor of the Computer Science Department at Rensselaer Polytechnic Institute, Troy, NY, between 1998 and 2005 where he taught classes in Soft Computing and Applied Intelligent Reasoning Systems. He has carried out applied research in the areas of real time monitoring, diagnostics, and prognostics and he has fielded numerous applications for aircraft engines, transportation systems, medical systems, and manufacturing systems. Dr. Goebel holds 18 patents and has co-authored more than 300 technical papers in the field of Prognostics Health Management. He is currently member of the board of directors of the Prognostics and Health Management Society and Associate Editor of the International Journal of Prognostics and Health Management.

**Scott Poll** is the Deputy Lead of the Discovery and Systems Health (DaSH) Technical Area in the Intelligent Systems Division at NASA Ames Research Center. He also leads the Diagnostics and Prognostics Group within DaSH. He has conducted research to benchmark diagnostic algorithms in detecting and isolating faults in an experimental testbed. He has also conducted research in detection, isolation, accommodation, and situational awareness of aircraft flight control system failures. Prior to that, he was a researcher and assistant project director for a multi-phase wind tunnel test program of a cargo transport aircraft. He received the BSE degree in Aerospace Engineering from the University of Michigan, and the MS degree in Aeronautical Engineering from the California Institute of Technology.

## Appendix A – ESN Training Procedure

ESN Training Procedure can be described as follows (Jaeger, 2001)

Assume a teacher input $u(t)$ and a teacher output $d(t)$.

- Generate an untrained network, where $W^{in}$, $W$, and $W^{back}$ are randomly established.
- Define a matrix $W_1 = W / |\lambda_{\max}|$, where $\lambda_{max}$ is the spectral radius of $W$, then the spectral radius of $W_1$ will be $\rho(W_1)=1$.
- Initialize the network parameters, i.e. the size of dynamical reservoir $N$, the spectral radius $SR$, the connectivity $C$, the input units scaling $IS$, the input units shift $IF$, the output units feedback $OFB$, the output units scaling $OS$, and the output units shift $OF$.
- For times $t=0...T$, drive the network by feeding the input $u(t)$ and by teacher-forcing the teacher output $d(t-1)$. Collect the input unit and the network state into a state collecting matrix $B$. In the end, one has obtained a state collecting matrix of size $(T-1)\cdot(L+N)$. Considering an ESN with linear output weights, collect the teacher output $d(t)$ into a teacher collection matrix $C$, to end up with a teacher collecting matrix of size $(T-1)\cdot M$.
- To obtain the output weights apply the Least Squares theory to the obtained matrixes:

$$(W^{out})^{Tr}=B^{-1}C, \quad (1A)$$

where *Tr* denotes transpose: transpose $(W^{out})^{Tr}$ to $W^{out}$ to obtain the desired trained output weights.

## Appendix B – DE Algorithm

DE's basic strategy can be described as follows (Storn & Price, 1997):

### Mutation

For each target vector $x_{i,G}$, $i$ = 1, 2, . . ., *NP*, a mutant vector is generated according to:

$$v_{i,G+1} = x_{r1,G} + F\cdot(x_{r2,G} - x_{r3,G}) , \quad (1B)$$

where the random indexes $r_1$, $r_2$, and $r_3 \in \{1, 2, . . , NP\}$ are integer and mutually different, and the parameter *F*, which is a real and constant factor which controls the amplification of the differential variation $(x_{r2,G} - x_{r3,G})$, is positive.

### Crossover

In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. To this end, the trial vector:

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, . . , u_{Di,G+1}) \quad (2B)$$

is formed according to:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1}, if\,(randb(j) \le CR \vee j = rnbr(i)) \\ x_{ji,G}, if\,(randb(j) > CR \vee j \ne rnbr(i)) \end{cases} \quad (3B)$$

where $j$ = 1, 2, . . , *D*; *randb(j)* is the *j*-th evaluation of a uniform random number generator with outcome $\epsilon$ (0, 1). *CR* is the crossover constant $\epsilon$ (0, 1) which has to be determined by the user; *rnbr(i)* is a randomly chosen index $\epsilon$ {1, 2, …, D} which ensures that $u_{i,G+1}$ gets at least one parameter from $v_{i,G+1}$ (Storn & Price, 1997).

### Selection

To decide whether or not the trial vector $u_{i,G+1}$ should become a member of generation *G+1*, its corresponding fitness function is compared to that corresponding to the target vector $x_{i,G}$: if vector $u_{i,G+1}$ yields a smaller fitness function value than $x_{i,G}$, then $x_{i,G+1}$ is set equal to $u_{i,G+1}$; otherwise, the old value $x_{i,G}$ is retained (Storn & Price, 1997).