NARX Time Series Model for Remaining Useful Life Estimation of Gas Turbine Engines

Oguz Bektas[1], Jeffrey A. Jones [2]

[1,2] *Warwick Manufacturing Group, University of Warwick, Coventry, UK*
O.Bektas@warwick.ac.uk
J.A.Jones@warwick.ac.uk

ABSTRACT

Prognostics is a promising approach used in condition based maintenance due to its ability to forecast complex systems' remaining useful life. In gas turbine maintenance applications, data-driven prognostic methods develop an understanding of system degradation by using regularly stored condition monitoring data, and then can automatically monitor and evaluate the future health index of the system. This paper presents such a technique for fault prognosis for turbofan engines. A prognostic model based on a nonlinear autoregressive neural network design with exogenous input is designed to determine how the future values of wear can be predicted. The research applies the life prediction as a type of dynamic filtering, in which training time series are used to predict the future values of test series. The results demonstrate the relationship between the historical performance deterioration of an engine's prior operating period with the current life prediction.

1. INTRODUCTION

Critical engineering systems, such as gas turbines, require dynamic maintenance planning strategies and predictions in order to reduce unnecessary maintenance tasks. A Condition Based Maintenance (CBM) decision-making strategy based on the observation of historical condition measurements can make predictions for the future health conditions of systems and this capability of predictions makes CBM desirable for the systems reliability, maintenance, and overall operating costs (Jardine, Lin, & Banjevic, 2006). The predictions on the health level of the system can be provided to maintenance scheduling and planning. This is especially useful in demanding applications, where the maintenance must be performed safely and economically for an entire lifetime.

One of the key enablers in CBM is the prediction, which leds to the transfer of data from the present monitoring into prognostics. CBM can ensure impending failure diagnosis and equipment health prognosis by obtaining periodic data from system indicators (Peng, Dong, & Zuo, 2010).

Prognostics can be defined as the process of predicting the lifetime point at which a component or a system could not complete the proposed function planned during its design (Pecht, 2008). The amount of time from the current time to the point of a system's failure is known as Remaining Useful Life (RUL) (Galar, Kumar, Lee, & Zhao, 2012). The concept of RUL has been widely applied as a competitive strategy to improve maintenance planning, operational performance, spare parts provision, profitability, reuse and product recycle (Si, Wang, Hu, & Zhou, 2011). The prediction of RUL is the principal goal of machine prognostics and this paper, therefore, evaluates the prognostics as the calculation of RUL for turbofan engine systems.

Data-driven prognostics are more effective methods in gas turbine prognostic applications because of the simplicity in data finding and consistency in complex processes (Heng, Zhang, Tan, & Mathew, 2009). They are also of particular importance because of the ability to integrate innovative and conventional approaches by generating inclusive prognostic methods over a wide-ranging data series.

The most commonly practiced data-driven prognostic methods in the literature are the Artificial Neural Networks (ANNs) (Schwabacher & Goebel, 2007; Heng et al., 2009). ANNs are computational algorithms inspired by biological neural networks of the brain and are used as machine-learning systems made up of data processing neurons, which are the units to connect through computation of output value by the input data. They learn by example by identifying the unique output with many past inputs values (Byington, Watson, & Edwards, 2004).

Neural networks are effective applications to model engineering tasks consisting of a broad category of nonlinear dynamical systems, data reduction models, nonlinear regression and discriminant models (Sarle, 1994). In some complex engineering applications, the observations from the system may not include precise data, and the desired results may not have a direct link with the input values. In such cases, ANN is a powerful tool to model the system without knowing the exact relationship between input and output data (Murata, Yoshizawa, & Amari, 1994). In particular, when a longer horizon with multistep ahead long term predictions is required, the recurrent neural networks play an important role in the dynamic modelling task by behaving as an autonomous system, and endeavouring to recursively simulate the dynamic behaviour that caused the nonlinear time series (Haykin & Li, 1995; Haykin & Principe, 1998; Menezes & Barreto, 2008). Recurrent Neural Network structure applies the target values as a feedback into the input regressor for a fixed number of time steps (Sorjamaa, Hao, Reyhani, Ji, & Lendasse, 2007),

Multi-step long term predictions with dynamic modelling are suitable for complex system prognostic algorithms since they are faster and easy to calculate compared to various other prognostics methods. The recurrent neural networks, therefore, have been widely employed as one of the most popular data-driven prognostics methods and a significant number of studies across different disciplines have stated the merits of them by introducing different methodologies.

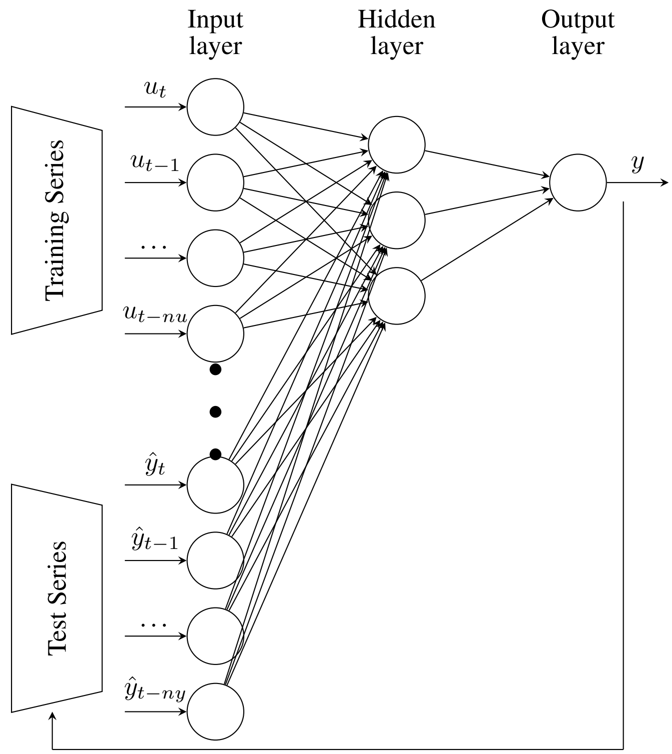
However, ANN multi-step predictions in prognostic applications can be quite challenging when only a few time series or a little previous knowledge about the degradation process is available and the failure point is expected to happen in the longer term. The greater interest in neural networks is the accomplishment of learning but it is not always possible to train the network as desired. The results at multi-step long-term time series predictions may be ineffective and this is generally more evident in the time series having exponential growths or decays.

In this paper, a Nonlinear AutoRegressive neural network with eXogenous inputs (NARX) is designed to make future multistep predictions of an engine from past operational values. The model learns successfully to make predictions of time series that consists of performance related parameters.

The designed prognostic method is modeled by using turbofan engine simulation C-MAPSS datasets from NASA data repository (Saxena & Goebel, 2008). The results demonstrate the relationship between the historical performance deterioration of an initially trained subset and the RUL prediction of a second test subset.

2. MODELING AND PREDICTION WITH NARX TIME SERIES

A NARX model is a recurrent dynamic network which relates the current value of a time series with the trained and predicted past values of the driving (exogenous) series. The model includes feedback connections enclosing several layers of the network and these connections form the externally determined series that generates the series of predictions (Beale & Demuth, 1998).



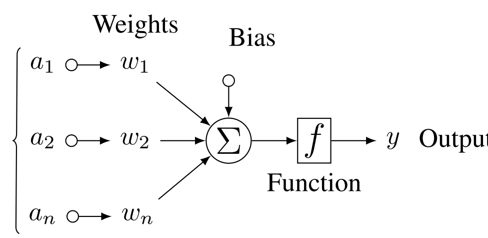


Figure 1. NARX network with the modelling of Series-parallel and Parallel modes

NARX can be extended to the multivariable case when the scalars are replaced by vectors in appropriate cases and multiple tapped delay lines are structured from the target series of the network (Siegelmann, Horne, & Giles, 1997). This extension would provide that each delay line is used for a following single output as seen on Figure 1.

Specifically, the defining equation for the operation of NARX model is:

$$y(t+1) = f(y_{t-1}, y_{t-2}, ..., y_{t-n_y}, u_{t-1}, u_{t-2}, ..., u_{t-n_u}) \tag{1}$$

where u_n and y_n stand for, respectively, the externally determined training input and target variables to be explained and predicted, while t denotes the discrete time step (Menezes & Barreto, 2008; Siegelmann et al., 1997). This equation is implemented by using a feedforward neural network to approximate the function where the previous values of the exogenous input signal along with the previous values of the output signal regress together to achieve the next values of $y_{(t)}$. The model can be expressed as in equation 2;

$$y_{t+1} = f(\boldsymbol{y}_{(t)}; \boldsymbol{u}_{(t)}) \tag{2}$$

The back-propagation network structure in Figure 1 describes that NARX model is applied in two different modes (Menezes & Barreto, 2008).

- **Series-parallel (SP) mode**, or also called as Open Loop mode, only applies to actual values of the target series in order to form the regressor of target series. SP mode is used for network training between the target variables and the principal components (input series).

$$\hat{y}(t+1) = \hat{f}(\boldsymbol{y}_{\boldsymbol{sp}(t)}; \boldsymbol{u}_{(t)}),$$
$$\hat{y}(t+1) = \hat{f}(y_{t-1}, y_{t-2}, ..., y_{t-n}, u_{t-1}, u_{t-2}, , u_{t-n}) \tag{3}$$

- **Parallel (P) mode**, or also called as Closed Loop mode, forecasts the next values of target series by using the feedbacks from the regressor of target series. The multi-step ahead long term predictions are performed in parallel mode after both input and target series are trained in SP mode.

$$\hat{y}(t+1) = \hat{f}(\boldsymbol{y}_{\boldsymbol{p}(t)}; \boldsymbol{u}_{(t)}),$$
$$\hat{y}(t+1) = \hat{f}(\hat{y}_{t-1}, \hat{y}_{t-2}, ..., \hat{y}_{t-n}, u_{t-1}, u_{t-2}, , u_{t-n}) \tag{4}$$

The state variables of the network, which correspond to x_n on the bottom of Figure 1, are defined to be the memory elements such as the set of time delay operators. A network has a bijective function (one to one correspondence) between these state variables and the node activations because each of the node values is stored at each time (Siegelmann et al., 1997).

The state variables at the next time step are expressed as

$$a_i(t+1) = z_i(t) \tag{5}$$

The target series is assigned arbitrarily to be equal to the first node in the network and therefore;

$$y(t) = z_1(t) \tag{6}$$

The state variables of the NARX model includes two delay lines on the input and target series. The corresponding formula to calculate the state variables is reformed as

$$a(t+1) = \begin{cases} u(t) & i = n_u \\ y(t) & i = n_u n_y \\ a_{i+1}(t) & 1 \leq i < n_u \quad and \quad n_u < i < n_u + n_y \end{cases} \tag{7}$$

when the delays correlate with the values at time t, it is denoted by

$$a(t+1) = [(y_{t-1}, y_{t-2}, ..., y_{t-n}, u_{t-1}, u_{t-2}, ..., u_{t-n})] \tag{8}$$

A NARX network is formed of a Multilayer Perceptron (MLP) which takes the input state variables as a window of past input and output values and computes the current output. Neurones, which are the building blocks of a neural network, evaluate and process these input state variables. The nodes in the hidden layer are performed by the function of

$$y = \sigma\left(\textstyle\sum_{i=1}^{n} w_i a_i + b\right) \tag{9}$$

where the fixed real-valued weights (w_i) are multiplied by state variables (x_i) with the adding of bias b. Thus, the neuron's activation (y) is obtained as a result of the nodes and the nonlinear activation function of neurons (σ) (Barad, Ramaiah, Giridhar, & Krishnaiah, 2012; Krenker, Kos, & Bešter, 2011). The activation function used in the model is the sigmoid function denoted as

$$\sigma_{(a)} = \frac{1}{1 + e^{-a}} \tag{10}$$

It is accepted in this study that y corresponds to assumed values of gas turbine performance, and u is the input data set consisting of multiple multivariate time series from a certain type of engine. The state variables of input values used in the nodes are represented by x. It is noted that the errors in data training and forecasting do not allow the current value of the output variables to be predicted exactly by using historical data.

3. APPLICATION OF ALGORITHM

Neural network performance is related to the primary steps of the network design process. In this work, the forecasting setup of NARX consisted of the following structures:

1. Data Collection and Preparation
2. Open Loop Setup
3. Network Configuration
4. Network Training
5. MSE (mean square error) validation
6. Switching to Closed Loop Mode
7. Using the Closed Loop design with trained network to predict future outputs of test subset
8. Evaluation of the result with corresponding RUL values.

The input data sets consist of multiple multivariate time series measured from the engine. The time series are divided into sequential subsets of data points, typically consisting of successive measurements made over a time interval. It is expected from the model to train itself from the previous state and make further predictions by using this data.

3.1. Data Exploration & Preparation

The C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is formed of multi-various time series assembled into training and test subsets. At the start of each series, the variables start in normal operational conditions with a case-specific unknown initial wear which is considered normal (Saxena & Goebel, 2008).

Training time series have full operational periods which end at the failure point due to the wear. However, the test subsets are curtailed some time before they reach the system failure. The challenge is to predict the remaining time interval between the end of each test set and the actual failure time. A data vector corresponding to true RUL values of the test data are given separately and so that, the results can be validated with the true RUL values of test subsets.

Each measurement in both data subsets is a snapshot of data which is taken during a single operational cycle. Although the measurements are not named, it is known that they correspond to different variables (Saxena & Goebel, 2008). The exponential growth in wear during the operations is the key assumption used to train the network, to calculate remaining useful life and to decide on the threshold point for multistep time series predictions. Therefore, this paper only uses the variables in which an exponential growth or decays occurs. There are 14 different measurements used in the model for training and prediction. These measurements are in different value ranges and moreover, the time series are very noisy and unsuitable for NARX closed loop prediction. It is therefore required to extract new information from the data set and transform it into a comprehensible structure for further forecasting use.

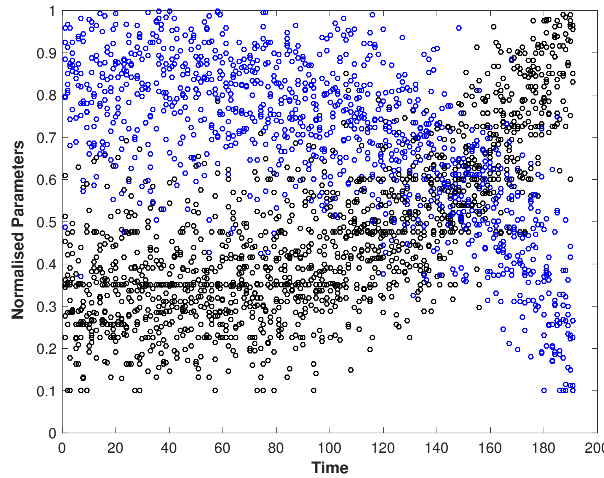


Figure 2. Time Series After Normalisation

- **Normalisation and Transformation:**

  The raw time series needs to be reformatted in a given array. An unity-based normalisation (feature scaling) method is used to bring all these values into the range of [0, 1] and later, the range of values are restricted slightly due to avoid multiplication by zero which would cause failure in NARX training.

  The curves of engine data series are sometimes inconsistent to wear like trajectories, acting like exponential decays as seen on Figure 2, so the decaying curves (blue) are required to be transformed into exponentially growing curves (black).

  The formula unifying both rescaling and transformation is given as:

$$f(x_{(i)}) = \begin{cases} \frac{x_{(i)} - x_{min}}{x_{max} - x_{min}} & for \quad c < 0 \\ \left(x_{max} - \frac{x_i - x_{min}}{x_{max} - x_{min}} \right) + x_{min} & for \quad c \geq 0 \end{cases} \tag{11}$$

- where, c corresponds to the difference between the mean of early time series and the mean of late series, represented as;

$$c = \frac{1}{d}\sum_{i=1}^{d} x_i - \frac{1}{d}\sum_{i=(L-d+1)}^{L} x_i \quad (12)$$

The length of the time series is L, while d is the constant arbiter factor that determines the amount of series used in the calculations of mean. Both train and test subsets are combined into a transient dataset for normalisation and transformation processes in order to assign the threshold point consistent with each other. Subsequently, they are divided into two discrete subsets again.

- **Filtering:**

  The exponential growth of the wear in time series is very noisy because of the system and environmental settings. The series are required to be filtered by reducing the information that is ineffectual or confusing to the model. The polynomial fitted curves are used for exponential curve fitting process returning the coefficients for a k^{th} degree polynomial which best fits in a least-squares manner and calculates the threshold point of failure as closely as possible .

  The polynomial regression is used to filter the nonlinear relationship between the noisy raw values and the corresponding dependent variable which is modelled as an k^{th} degree polynomial (Anderson, 2011). In this study, 9^{th} and 4^{th} degree polynomial regressions are used respectively for train and test subsets because only the higher degree polynomial regression can detect the failure threshold point in training subset. The main aim of this analysis is to estimate the expected values of dependent vectors in terms of the values of independent vectors.

  Generalising from a raw vector to a k^{th} degree polynomial, the general polynomial regression model yields to

$$y_{f_i} = a_0 + a_1 x_i + a_2 x_i^2 + ... + a_k x_i^2 \quad \text{(i=1,2,...,k)} \quad (13)$$

  where, x is independent variable, y is dependent variable and a is response vector. This expression can be written in matrix form as

$$\begin{bmatrix} y_{f_1} \\ y_{f_2} \\ y_{f_3} \\ \vdots \\ y_{f_n} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ 1 & x_3 & x_3^2 & \cdots & x_3^k \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} \quad (14)$$

  In pure matrix notation, the equation is given by;

$$\vec{y_f} = \mathbf{X}\vec{a} \quad (15)$$

After the polynomial regression is received, the filtered data can be achieved by the following formula

$$x_f(i) = \left(y_{f_i} g + x_i\right) / (g+1) \quad (16)$$

where g corresponds to multiplication variable which is selected in a way that it ensures consistency and relevance of the time series for network training. In other words, it reduces the noise into an acceptable range that NARX model can perform, train and predicts well.

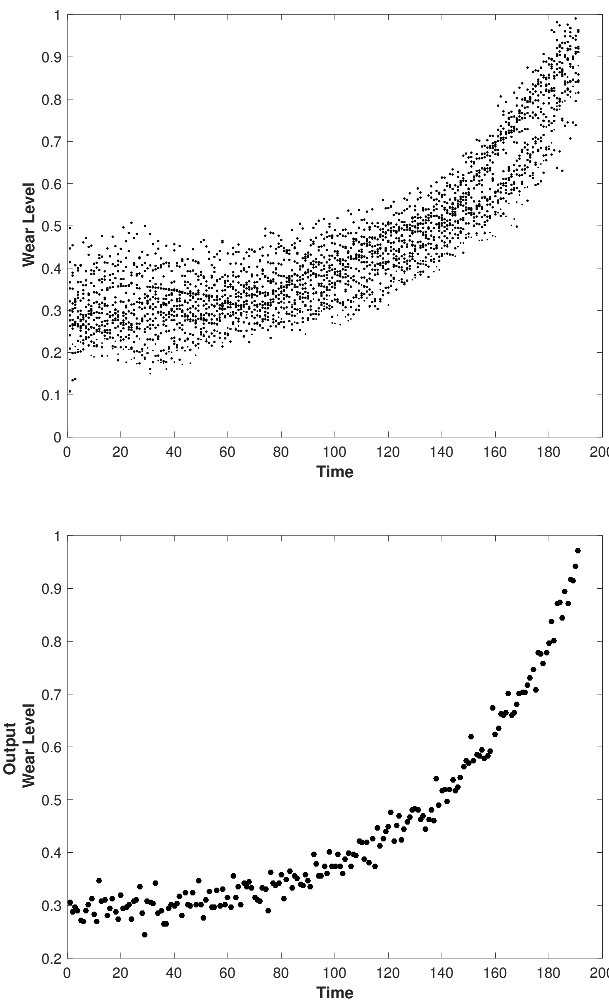


Figure 3. Input And Output Variables

The normalised, and transformed series coherent with the wear growth pattern of the assets performance are shown in Figure 3.

The exponential growth is generally linked with the performance of main engine components and can also be assumed to be equivalent to the parameters of aircraft engine modules such as efficiency and flow level. The interval start and end points represent a full operation period between a certain point at a set performance level and the eventual threshold limit, which can be determined by the end user according to their standards. In this study, the threshold limit is determined according to the ending value of training output series.

- **Output assignment:**
  After the input variables are normalised, transformed and filtered, the target series are assigned as the median of all input values at the same time step. The corresponding formula separating the higher half of input variables at the same time step from the lower half is expressed by the following equation (Hogg & Craig, 1995).

  According to the order statistics;

$$x_{f_1} = \min_j \tilde{y}_{tj}, x_{f_2}, , x_{f_{N-1}}, x_{f_N} = \max_j \tilde{y}_{tj} \quad (17)$$

  the statistical median of the Input variables at time step j is defined by

$$y_{tj} = \begin{cases} x_{f_{(N+1/2)}} & \text{if } N \text{ is odd} \\ \frac{1}{2}\left(x_{f_{N/2}} + x_{f_{1+N/2}}\right) & \text{if } N \text{ is even} \end{cases} \quad (18)$$

3.1.1. Recurrence relation

Since the target vector representing the wear level of the system is detected by the median values of input series, the wear growth model can be applied to learn the pattern from historical data and to estimate RUL time until the pattern exceeds threshold point.

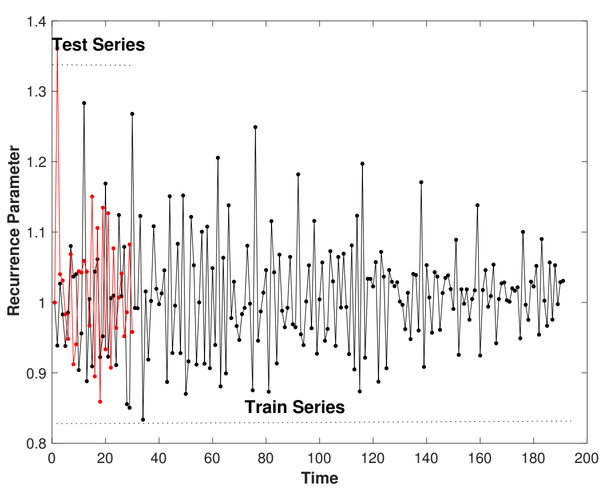


Figure 4. Transformed Data

Although the NARX model can accomplishment the network training in open loop mode, it cannot produce predictions of multi-step long-term time series when the vectors have exponential growths as are present in these series. Therefore, a recurrence relation equation is used to redefine the exponential series into a form that the model can perform well. Each further series are defined as a function of the preceding values.

$$x_{r(i)} = \begin{cases} 1 & \text{for} \quad i = 1 \\ x_{f(i)}/x_{f(i-1)} & \text{for} \quad i \geq 0 \end{cases} \quad (19)$$

where x_r (or y_r according to the data type) corresponds to initial principal components u_n or the target variables y_n which are used for network training and forecasting in Figure 1.

The input and output series trajectories take form as in Figure 4 after the exponential data is redefined according to above equation. Subsequently, the predicted data vector is reinstated to its original exponential form after closed loop network makes multistep predictions.

3.2. Network Configuration and Training

The first step to structuring the network structure is the configuration of input and output series. Inputs are situated as a $14 \times t$ cell array of 1×1 matrices which represents a dynamic data of 't' time steps of 14 different variables. The output, or target, is also a $1 \times t$ cell array of 1×1 matrices and only represents t time steps of 1 variable. In order to reduce error numbers between trained outputs and targets, multiple open-loop networks are designed in a double loop structure over the increasing number of hidden nodes of an outer loop and random weight initialisations of an inner loop (Heath, 2015).

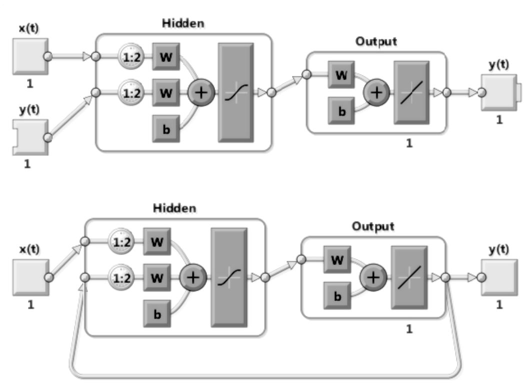


Figure 5. Closed and Open Loop designs in NARX

- **Mean Squared Error :** The loops terminates when minimum desired error occurs. The standard reference Mean Squared Error is used to calculate the performance of the networks. It represents the average squared difference between achieved output values from open loop mode and initial target values.

$$MSE = \frac{1}{t}\sum_{i=1}^{t}\left(\check{y}_{r(i)} - y_{r(i)}\right)^2 \tag{20}$$

At the end of both loops, the least erroneous network is converted to closed loop mode. For validation, a predetermined MSE goal is used to choose the best open loop design mode.

Figure 5 illustrates the open and closed loop modes in NARX design. The network is firstly trained by the introduction of input series x and target values y. The tapped delay lines are decided with two delays (1:2) for both the input and the targets so training begins after these points. In each time of neural network training, the results can give different solutions because of different initial weight w and bias values b. Following the training between input and target, NARX uses back-propagation for multi step ahead predictions by using the stored network function.

Bayesian regularisation is used to minimise a combination of squared errors and weights in network training. This approach can be implemented well in generalisation for difficult and noisy datasets.

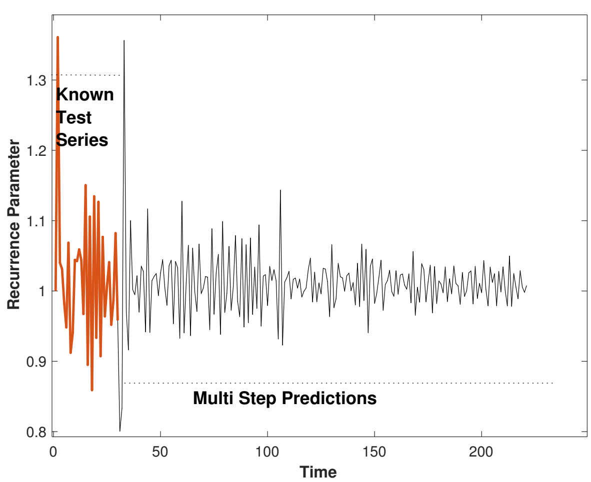


Figure 6. Predicted Multistep Variables

3.3. Multi Step Prediction

Test subsets have ended some time prior to failure occurrence. This means that there is no real data to train the remaining time series. To that end, the closed loop mode replaces the feedback input values of the test subset's values with a direct connection from the test targets. The algorithm is trained in open loop network simulation with the train subset, and then converted into a closed loop mode to make multistep predictions by including only the external test subset's inputs.

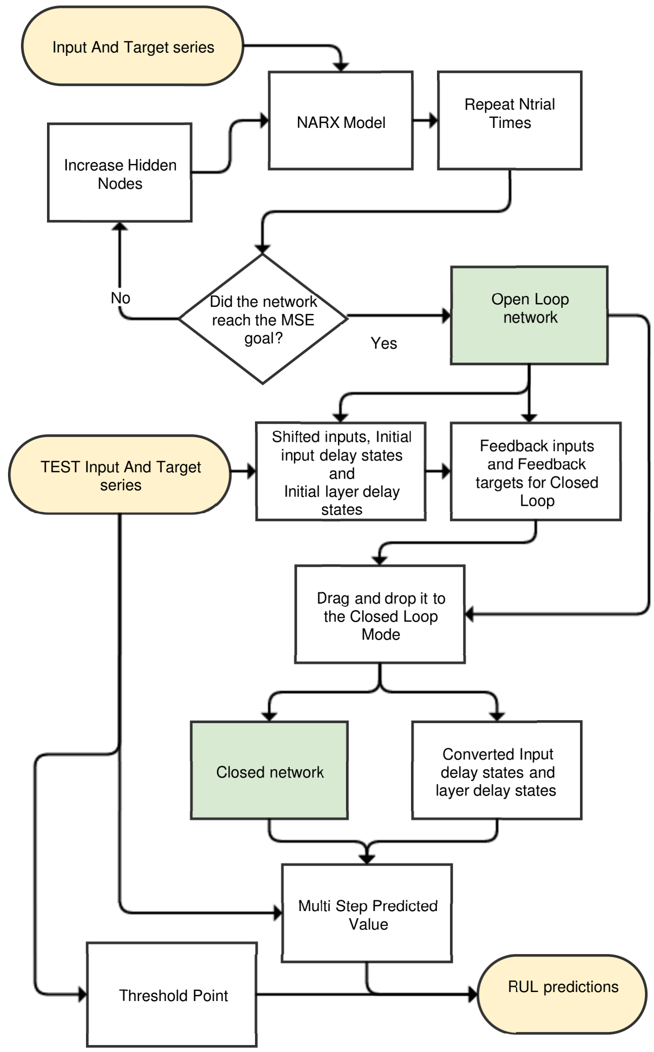


Figure 7. Flowchart representing the process

The input and output series are complicated and required to be simplified so both series can be shifted as required steps in order to fill the input and layer delays. Then, the feedback outputs are formed into a new form conducive to defining closed-loop parameters as seen on the flowchart in Figure 7. As a result of this process, a vector of multistep predictions of target series can be calculated by the closed loop network and test parameters. The predictions made by using internal inputs can be as long as the training series. Figure 6 illustrates the multi step predictions of the recurrence relation. A reverse calculation is required to reinstate the wear shaped

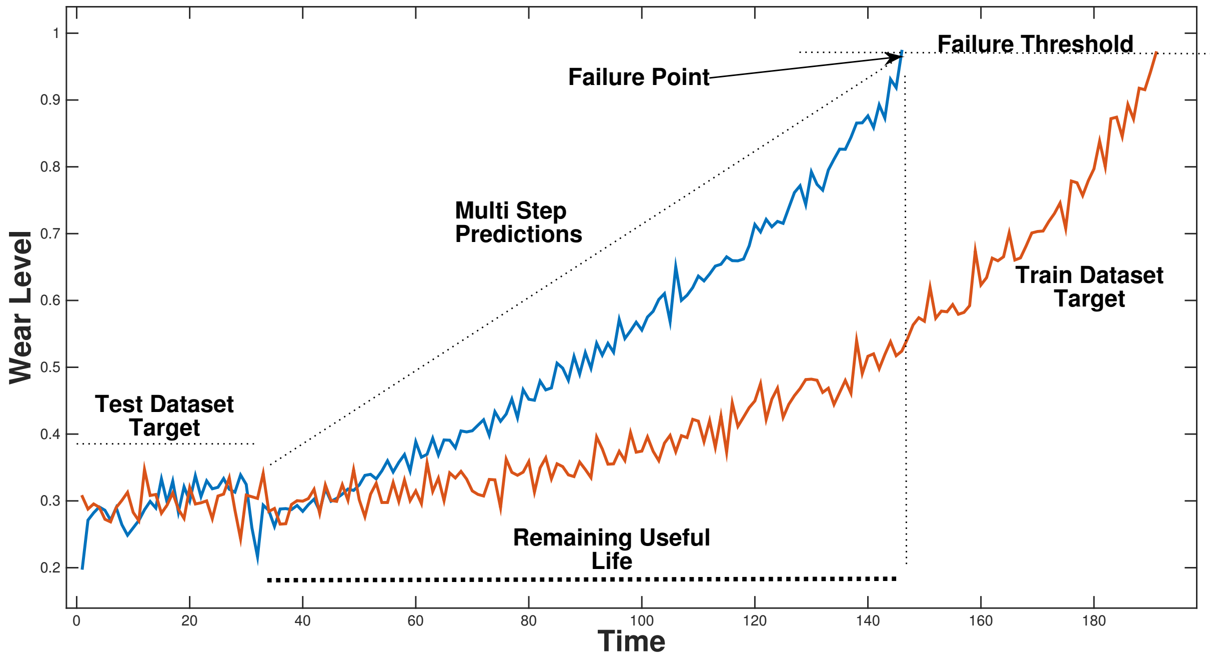


Figure 8. Remaining useful life multi step estimation

exponential growth as seen on Figure 8.

$$y_{p_{(i)}} = \begin{cases} y_{t\,(i)} \times y_{r_{(i)}} & \text{for} \quad i = 1 \\ y_{p_{(i-1)}} \times y_{r_{(i)}} & \text{for} \quad i \geq 0 \end{cases} \tag{21}$$

4. ANALYSIS OF MODEL

Based on the previous section, the engine wear level prognostic analysis is carried out using dynamic network modelling. An initial wear threshold point is specified before the model is started. This point basically corresponds to where training subset's target values has the maximum value. It is assumed that the test subsets needs to fail at a similar failure point found in training data. Figure 8 presents the result of multistep data prediction for the test subset of the first operational sequence of the first dataset. Here, x-axis represents the time index (cycles) showing the assumed number of flights whereas y- axis is the wear index based on normalised parameter levels.

The test subset in Fig 8 consist of 30 steps. The future trajectories after this point are calculated and predicted by closed loop mode. The training subset, on the other hand, is formed of 191 steps and the end of the series represents the failure point. Since NARX model allows to make predictions as long as the data steps trained in the open loop mode, there can only be 191 future predictions to find the remaining useful life of test subset.

The normalised wear level for both subsets has started close to each other so the values of initial wear for both subsets are parallel in the beginning. Remaining useful life is calculated as the time interval between the end of test subset and the point where the prediction value exceeds the value of training subset target vector. In Figure 8, where the future wear growth at gas turbine performance is predicted, RUL estimation corresponds 116 cycles. The true remaining useful life for the test engine subset was given as 112 steps separately in the dataset. The absolute deviation between true and predicted values is therefore reasonably small.

In Table 1, the estimated RUL for the first twenty sequence is shown along with the true RUL and absolute deviation of predictions. The future remaining life predictions of NARX seems promising for employing past and present time series while some calculations may suffer from computational complexity in particular to the cases having dissimilar initial wear levels. The prediction of engine RUL from Figure 8 is found to be satisfactory, however, the algorithm may provide a lower performance than that have shown very close results due to the different levels of scattering of the data and the incongruity of test and train series.

Table 1. Results of first 20 sequences

Sequence	True RUL	Pred RUL	Abs Deviation
1	112	116	4
2	98	112	14
3	69	43	**26**
4	82	79	3
5	91	88	3
6	93	111	18
7	91	93	2
8	95	107	12
9	111	118	7
10	96	93	3
11	97	85	12
12	124	78	**46**
13	95	84	11
14	107	98	9
15	83	97	14
16	84	100	16
17	50	52	2
18	28	39	11
19	87	113	**26**
20	16	26	10

Among all these cases, RUL predictions of 6 cases could be calculated in a close-range to true remaining useful life, in which the deviation is less than ten steps. The results of case 3, 12 and 19 have resulted in less performance. The difference at starting levels of variables between the train and test variables are relatively higher in these cases.

The number of early predictions is 8 while late predictions are 12. However, it should be mentioned here that early in time predictions are less risky as compared to late in time predictions because late predictions may cause catastrophic results during operations. On the other hand, an early in time prediction may cause a significant economic burden when the failures may not necessarily terminate with life-threatening conditions (Saxena, Goebel, Simon, & Eklund, 2008).

To sum up, the developed model seems to exhibit promising results at multi-step long term time series predictions for exponential wear growths. The training of network could accomplish learning as desired while training performance is substantially increased by recurrence input data use and the loop designs in closed loop mode. However, poor performance on the calculations may happen occasionally since the future prediction can never be absolutely accurate especially when the asset is very complex. The deficiencies exhibited by network adequacy and the functionality of the approach for remaining useful life calculations, such as exponential growths and unstable operating conditions or inability to obtain satisfactory multistep calculations, can be eradicated if data sets are trained multiple times and the highest performance network is applied for forecasting. The suggested model also demonstrates the point that there may be more reasonable modifications for open and closed loop based NARX model depending on distinctive implementations and developments.

5. CONCLUSION

This paper presents a data driven prognostic framework for gas turbine applications by adapting existing nonlinear autoregressive neural network with external input methods. The NARX model is based on the conversion of open and closed loop modes to predict multistep exponential growth behaviours from measured historical information.

An initial dataset is trained in the model to learn the engine behaviour and a second subset is used to predict future wear level. The application of the suggested technique shows that NARX is able to detect the unknown RUL effectively and can predict the exponential wear level of the system at multistep long terms.

REFERENCES

Anderson, T. W. (2011). *The statistical analysis of time series* (Vol. 19). John Wiley & Sons.

Barad, S. G., Ramaiah, P., Giridhar, R., & Krishnaiah, G. (2012). Neural network approach for a combined performance and mechanical health monitoring of a gas turbine engine. *Mechanical Systems and Signal Processing*, *27*, 729–742.

Beale, M., & Demuth, H. (1998). Neural network toolbox. *For Use with MATLAB, User's Guide, The Math Works, Natick*, 1–6.

Byington, C. S., Watson, M., & Edwards, D. (2004). Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *Aerospace conference, 2004. proceedings. 2004 ieee* (Vol. 6, pp. 3581–3589).

Galar, D., Kumar, U., Lee, J., & Zhao, W. (2012). Remaining useful life estimation using time trajectory tracking and support vector machines. In *Journal of physics: Conference series* (Vol. 364, p. 012063).

Haykin, S., & Li, X. B. (1995). Detection of signals in chaos. *Proceedings of the IEEE*, *83*(1), 95–122.

Haykin, S., & Principe, J. (1998). Making sense of a complex world [chaotic events modeling]. *Signal Processing Magazine, IEEE*, *15*(3), 66–81.

Heath, G. (2015, March). *Narnet tutorial on multistep ahead predictions.* [Online forum comment] https://uk.mathworks.com/matlabcentral /newsreader/view thread/338508.

Heng, A., Zhang, S., Tan, A. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing*, *23*(3), 724–739.

Hogg, R. V., & Craig, A. T. (1995). *Introduction to mathematical statistics: 5th ed.* Macmillan.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Krenker, A., Kos, A., & Bešter, J. (2011). *Introduction to the artificial neural networks*. INTECH Open Access Publisher.

Menezes, J. M. P., & Barreto, G. A. (2008). Long-term time series prediction with the narx network: an empirical evaluation. *Neurocomputing*, *71*(16), 3335–3343.

Murata, N., Yoshizawa, S., & Amari, S.-i. (1994). Network information criterion-determining the number of hidden units for an artificial neural network model. *Neural Networks, IEEE Transactions on*, *5*(6), 865–872.

Pecht, M. (2008). *Prognostics and health management of electronics*. Wiley Online Library.

Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, *50*(1-4), 297–313.

Sarle, W. S. (1994). Neural networks and statistical models.

Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository*.

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–9).

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium* (pp. 107–114).

Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation–a review on the statistical data driven approaches. *European Journal of Operational Research*, *213*(1), 1–14.

Siegelmann, H. T., Horne, B. G., & Giles, C. L. (1997). Computational capabilities of recurrent narx neural networks. *Systems, Man, and Cybernetics, Part B Cybernetics, IEEE Transactions on*, *27*(2), 208 215.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, *70*(16), 2861–2869.

BIOGRAPHIES

Oguz BEKTAS is a Doctoral student at WMG, University of Warwick under the supervision of Dr Jeffrey Jones. His research interests focuse on PHM for aircraft engines, fault detection, diagnosis and training of data driven prognostic methods with changeable external inputs. By forecasting the remaining useful life for multi-step-ahead predictions, he is developing a relational understanding of gas turbine performance behaviours. His research benefits from a diverse set of disciplines in the prognostic area including Machine Learning, Recurrent Neural Networks, Regression Analysis and Collaborative Data Mining. He is using an innovative data-driven methodology by combining multi-sourced collaborative techniques with data mining and dynamic Design of Time Series to achieve higher performance levels for prognostic calculations.

Jeffrey Jones Dr Jeffrey Jones is an Associate Professor at WMG with research interests in technical asset management, and dependability. The focus of this work is the use of models, simulations and artificial intelligence to help companies support their assets in various ways. These concepts have been applied at all levels from large multi-faceted systems down to MEMS device and device physics.