



HAL
open science

Reducing arbitrary choices in model building for prognostics: an approach by applying parsimony on an evolving neuro-fuzzy system.

Mohamed El-Koujok, Rafael Gouriveau, Nouredine Zerhouni

► To cite this version:

Mohamed El-Koujok, Rafael Gouriveau, Nouredine Zerhouni. Reducing arbitrary choices in model building for prognostics: an approach by applying parsimony on an evolving neuro-fuzzy system.. Microelectronics Reliability, 2011, 51 (2), pp.310-320. 10.1016/j.microrel.2010.09.014 . hal-00526022

HAL Id: hal-00526022

<https://hal.science/hal-00526022v1>

Submitted on 15 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reducing arbitrary choices in model building for prognostics: an approach by applying parsimony principle on an evolving neuro-fuzzy system

Mohamed El-Koujok, Rafael Gouriveau*, Nouredine Zerhouni

*FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM,
Automatic Control and Micro-Mechatronic Systems Department
24, rue Alain Savary, 25000 Besançon, France
{mohamed.elkoujok, rgourive, zerhouni}@ens2m.fr*

Abstract

Failure prognostics requires an efficient prediction tool to be built. This task is as difficult as, in many cases, very few knowledge or previous experiences on the degradation process are available. Following that, practitioners are used to adopt a “trial and error” approach, and to make some assumptions when developing a prediction model: choice of an architecture, initialization of parameters, learning algorithms... This is the problem addressed in this paper: how to systematize the building of a prognostics system and reduce the influence of arbitrary human intervention? The proposition is based on the use of a neuro-fuzzy predictor whose structure is partially determined, on one side, thanks to its evolving capability, and on the other side, thanks to parsimony principle. The aim of the approach is to automatically generate a suitable prediction system that reaches a compromise between complexity and accuracy capability. The whole proposition is illustrated on a real-world prognostics problem concerning the prediction of an engine health.

Keywords: prognostic, prediction, neuro-fuzzy system, parsimony criteria

1. Introduction

The high costs in maintaining complex equipments make necessary to enhance maintenance support systems and traditional concepts like preventive and corrective strategies are progressively completed by new ones like predictive and proactive maintenance [1]. In this context, prognostic reveals to be a very promising maintenance activity as it should permit to improve safety, plan successful missions, schedule maintenance, reduce maintenance costs and down time [2]. Also, industrials show a growing interest in this thematic which becomes a major research framework as mentioned in recent papers dedicated to Condition-Based Maintenance “CBM” [3, 4, 5].

A central problem can be pointed out: the accuracy of a prognostics system is related to its ability to approximate and predict the degradation of equipment:

starting from a “current situation”, a prognostics tool must be able to forecast the “future possible situations”. From the research point of view, a wide variety of approaches can be used for that purpose [5, 6, 7, 8]. However, choosing an efficient technique depends on classical constraints that limit the applicability of the tools: available data-knowledge-experiences, complexity and dynamic of systems, available monitoring devices, implementation requirements (precision, computation time...). Also, it can be difficult to provide effective models of dynamic systems including the inherent uncertainty of prognostic. Following that and assuming that it can be very difficult even impossible to provide a model of the system under study, data-driven approaches are been increasingly applied to prognostics (mainly techniques from artificial intelligence AI). These approaches use real data (like on-line gathered with sensors or operator measures) to approximate and track features revealing the degradation of components and to forecast the global behavior of a system. Indeed, in many applications, measured input/output data is the

*Corresponding author, Tel.: +33 (0)3 81 40 27 96, Fax.: +33 (0)3 81 40 28 09

major source for a deeper understanding of the system degradation. Within these data-driven approaches, neuro-fuzzy (NF) systems appear to be very promising prognostic tools: NFs learn from examples and attempt to capture the underlying relationships among data, even if they are unknown or hard to describe (by a learning process). NFs are computationally effective techniques and are thereby well suited for practical problems, where it is easier to gather data than to formalize the behavior of the system being studied. Actual developments confirm the interest of using NFs in forecasting applications [9, 10, 11, 12].

Nevertheless and in spite of the capabilities of NFs systems, building a NFs model for prediction is not a trivial task: various fuzzy structures can be used, the nature and quantity of inputs, as well as the form of the membership's functions have to be chosen, different learning algorithms exist, a random initialization (or expert made) must be done... Moreover, these choices are critical as they directly have an impact on both, the accuracy of the predictions of the system, and on its complexity. Following that, practitioners are used to adopt a "trial and error" approach in order to obtain good predictions whereas avoiding over-parametrization of the system, and there is no systematic way of building a suitable prediction system.

In this context, the paper deals with the definition of a NF prognostic system for which any assumption is necessary, i.e. a system that enables to reduce the influence of human intervention. The proposition is based on the use of a neuro-fuzzy predictor whose structure is partially determined, on one side, thanks to its evolving capability, and on the other side, thanks to parsimony principle. The underlying idea of parsimony is that there is a clear tradeoff between reducing the dimension of the parameter space at the expense of increasing the residual error of prediction. Both are related to the structure of the predictive model and the aim of the approach is thereby to automatically generate a system that reaches a compromise between complexity and accuracy of predictions.

The paper is organized in four main parts. First, the concept of "prognostic" is clarified and replaced within maintenance strategies. This part enables to point out that the efficiency of a prognostic system is highly dependent on its ability to perform "good" predictions. In the second part, the use of Takagi-Sugeno neuro-fuzzy systems for prognostic is justified and the ways of building such models are discussed as well as the learning aspects that influence the prediction performances. An evolving neuro-fuzzy system is pointed out and presented. This NF system is self-built and enables to

make predictions whose accuracy is independent from the user. Following that, the next section is dedicated to the proposition of a method to balance complexity and generalization capability. The procedure is based on the used of parsimony principle in order to define the number of inputs of the predictor in an automatic form. Various parsimony criteria are proposed and discussed thanks to experiments on an industrial prediction benchmark. The proposition is also discussed by comparing the NF system with the classical Auto Regressive eXogenous approach (ARX). In the last part, the whole procedure is finally applied on a real-world prognostics problem concerning the prediction of an engine health.

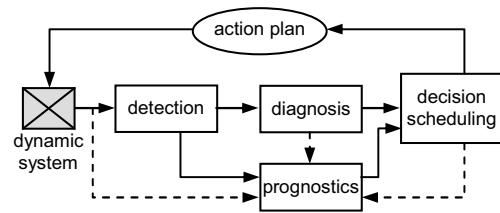
2. Background: prediction as a critical step of prognostic

2.1. Prognostic and intelligent maintenance systems

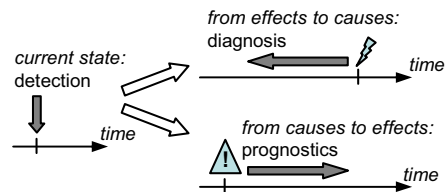
Maintenance activity combines different methods with tools and techniques to reduce costs while increasing reliability, availability and security of equipments. Thus, one usually speaks about fault detection, failures diagnosis, and response development (choice and scheduling of preventive/corrective actions). Briefly, these steps correspond to the need, firstly, of "perceiving" phenomena, secondly, of "understanding" them, and finally, of "acting" consequently. However, rather than understanding a phenomenon which has just appeared like a failure (a posteriori comprehension), it is convenient to "anticipate" its manifestation in order to take adequate actions as soon as possible. This is what can be defined as the "prognostic process". The relative positioning of "detection", "diagnostic", "prognostic" and "decision / scheduling" in the IMS (Intelligent Maintenance Systems) framework can be depicted as proposed in Figure 1a. From the phenomenological point of view, the complementarity of these processes can be explained as follows Figure (1b):

- detection aims at identifying the operating mode of the system, i.e. its current state, assuming that a failure occurred,
- diagnostic permits to isolate and to identify the component that has ceased to operate (past propagation: from effects to causes),
- prognostic deals with the prediction of the future(s) state(s) of the system (future propagation: from causes to effects).

According to all this, it is obvious that prognostic can not be seen as a single maintenance task. Also, the



-a- Detection, diagnosis, prognostics and scheduling activities in Intelligent Maintenance Systems



-b- Detection, diagnosis and prognostics - the phenomenological aspect

Figure 1: Prognostics within IMS

whole aspects of failure analysis and prediction must be viewed as a set of activities that all must be performed. This aspect is highlighted within the Condition-Based Maintenance (CBM) concept. According to CBM practitioners, various activities, ranging from data collection through the recommendation of specific maintenance actions, must be carried out to perform predictive maintenance (and thereby improve maintenance's performances). Generally, a CBM system is seen as the integration of seven layers, one of them being that of "prognostic" (see Figure 2 for a distributed CBM architecture [13]). A brief description of each layer is given hereafter [14].

- Layer 1: Sensor Module. It provides the CBM application with digitized sensor or transducer data.
- Layer 2: Signal Processing Module. It performs signal transformations and feature extractions, reduction and selection.
- Layer 3: Condition Monitoring Module. It compares on-line data with expected values of system's parameters.
- Layer 4: Health Assessment Module. It determines if the system has degraded. It also generates a diagnostic record and suggests fault possibilities.
- Layer 5: Prognostic Module. It predicts the future condition of the monitored system.

- Layer 6: Decision Support Module. It provides recommended actions to fulfill the the mission.
- Layer 7: Presentation Module. It can be built into a regular human-machine interface.

Although prognostic is recognized as a key process in maintenance strategies, the CBM system analysis enables to point out that prognostic is not sufficient in itself but must be seen as a part of a more global process. Nevertheless, next parts of this paper are only dedicated to prognostic.

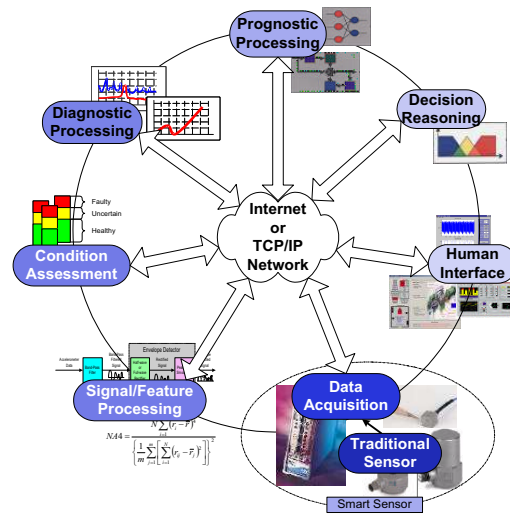


Figure 2: Architecture of Condition-Based Maintenance [13]

2.2. Concept of prognostic

Prognostic is traditionally related to fracture mechanics and fatigue. It started to be brought up by the modal analysis community as a field of interest [15]. In this "meaning", prognostic is called the prediction of a system's lifetime. Prognostic can also be defined as a probability measure: a way to quantify the chance that a machine operates without a fault or failure up to some future time [16]. This "probabilistic prognostics value" is all the more an interesting indication as the fault or failure can have catastrophic consequences (e.g. nuclear power plant). Some authors introduce prognostic as a process that allows the *a priori* reliability modeling and thereby enables to estimate the remaining time to underpass a limit fixed by the practitioner or by past experiences [17, 18]. Finally, although there are some divergences in literature, prognostic can be defined as proposed by the International Organization for Standard-

ization: “prognostic is the estimation of time to failure and risk for one or more existing and future failure modes” [19]. In this acceptance, prognostic is also called the “prediction of a system’s lifetime” as it is a process whose objective is to predict the remaining useful life (*RUL*) before a failure occurs given the current machine condition and past operation profile [5].

Whatever the point of view of all the above cited authors is, all acceptations of the “prognostic’s concept” have as common point the predictive aspect of prognostic: a future situation must be caught. This obviously supposes that the current state of the system can be identified and assessed (practically, it is the synthesis of a detection process and of a measured data of the system). Moreover, definitions of prognostic are focused on the failure notion (“termination of the ability to perform a required function”), which implies that the “prognostic activity” is associated with a degree of acceptability. Following that and according to previous works, prognostic should be based on assessment criteria, whose limits depend on the system itself and on performance objectives [20, 21]. Figure 3a can be used to illustrate this assumption, where the predicted situation at time “ $t+dt$ ” can be considered as a critical one because of the degradation limit. Thus, prognostic could be split into 2 sub-activities: a first one used to predict the evolution of a situation at a given time, and a second one that allows to assess this predicted situation with regards to an evaluation referential. Let’s resume (Figure 3b):

- identification: a situation is captured by the detection process and additional current measures,
- prediction: the situation is forecasted in time,
- assessment: the situation is evaluated with performance criteria,
- prognostic: the predicted situation is assessed.

In addition, a central problem appears: the accuracy of a prognostic system is related to its ability to approximate and to predict the degradation of equipment; thus the prediction phase is a critical one. A look at prognostic metrics enables to point out this aspect.

2.3. Prognostic metrics

There is no general agreement as to an appropriate and acceptable set of metrics that can be employed in prognostic applications, and researchers and maintenance practitioners are still working on this domain [8, 22, 23]. However, various measures emerge from literature and are presented hereafter.

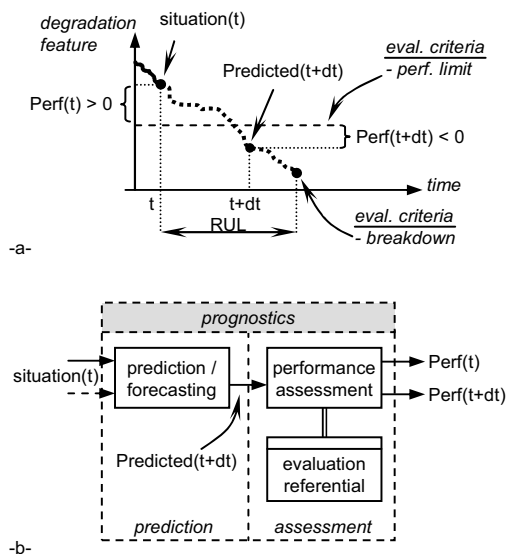


Figure 3: Prognostics as a prediction and assessment process [20]

As for any industrial task, prognostic can be evaluated at least in two ways.

1. The main objective of prognostic is to provide the efficient information that enables the underlying decision process, i.e., the choice of maintenance actions. So, a first set of metrics are those that quantify the risks incurred by the monitored system. This kind of metrics can be called the prognostic measures.
2. Assuming that prognostic is in essence an uncertain process, it is useful to be able to judge from its “quality” in order to imagine more suitable actions. In this way, prognostic system performance measures can be constructed.

The main and the most used prognostic measure is the predicted time to failure (*TTF*), also called the remaining useful life (*RUL*). In addition, a *confidence* measure can be built to indicate the degree of certitude of the future predicted failure time. By extension, and considering that practitioners can be interested on assessing the system with regard to any performance limit, *RUL* and confidence can be generalized, as shown in Figure 4a. On this latter, *TT_{xx}* refers to the remaining time for the component, sub-system or system to overpass the performance limit *Perf_{xx}*, and *Conf_{xx}T* is the confidence with which can be taken the asset *TT_{xx} > T*.

The performance of any prognostic method can be assessed through some measures, like timeliness, accuracy and precision. The *timeliness* of the predicted

time to failure is the relative position of the probability density function (pdf) of the prediction model along the time axis with respect to the occurrence of the failure event. This measure evolves as more data are available and reveals the expected time to perform preventive actions. According to [24], one has to define two different boundaries for the maximum acceptable late and early predictions (Figure 4b). The *accuracy* measures the closeness of the predicted value to the actual one. It has an exponential form and is as higher as the error between the predicted value of *TTF* and the real one is smaller. *Precision* reveals how close predictions are clustered together and is a measure of the narrowness of the interval in which the remaining life falls. Precision depends on the variance of the predicted results for many experiments. The complementarity between accuracy and precision is illustrated in Figure 4c. More details on some practical issues regarding accuracy, precision and confidence of the *RUL* estimations can be found in [25].

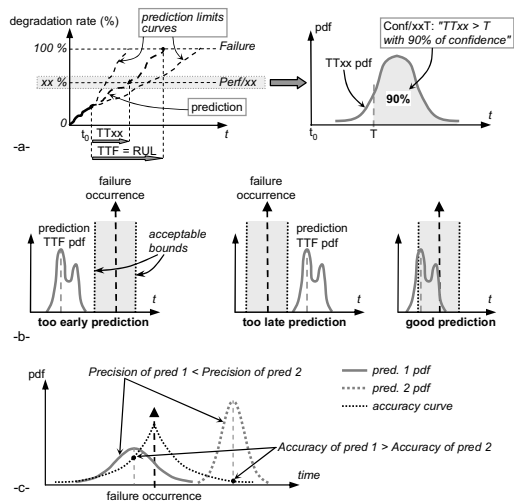


Figure 4: Some prognostic metrics

2.4. Perform good prediction: a critical issue

As stated before, prognostic is mostly assimilated to a prediction process, and all prognostic metrics follow from it. Thus, one has to pay a particular attention to this issue and practitioners are used to adopt a “trial and error” approach to build the most suitable prediction tool as possible. This aspect is critical and can even impede the development of real prognostic applications. Next sections aim at reducing the influence of human choices in order to improve the feasibility of prognostic.

3. The exTS: a suitable self-built prediction tool

3.1. First-order TS systems for prediction / forecasting

According to some authors, the methods presented in this section are sometimes labeled as “prognostic techniques”. However, most of them refer to what, in this paper, is called “prediction / forecasting”.

Prognostic approaches are used to be distinguished, in three main categories [13]: experience-based, evolutionary or estimation-based and model-based prognostics (Figure 5). Since this classification depends on the nature of data and knowledge used in practice, one uses to speak about experience-based, data-driven and model-based prognostics. In a few words, experience-based prognostic methods are used in statistical reliability applications to predict the probability of a failure at any time. Data-driven approaches rely on gauging the drift and the rate of change of the current state of the system with regard to a set of known performance degradations. Model-based methods suppose that the degradation process can be formalized in a mathematical and analytical form.

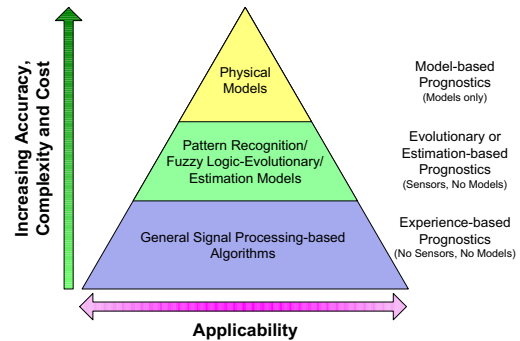


Figure 5: Hierarchy of prognostic approaches [13]

Real systems are complex and their behavior is often non linear, non stationary. This makes harder a modeling step, even impossible. Yet, a prediction tool must deal with it. Moreover, monitoring systems have evolved and it is now quite easy to online gather data. According to all this, data-driven approaches have been increasingly applied to prediction problems in general and to machine prognostics in particular, and works emphasize on the interest of using hybrid systems for prediction purpose. More precisely, first order Takagi-Sugeno (TS) fuzzy models have shown improved performances over conventional approaches [9, 11] and they appear to be adequate to perform the degradation modeling step of prognostics [17].

3.2. First-order TS systems: principle

A first-order TS model can be seen as a multi-model structure consisting of linear models that are not necessarily independent [26]. It is based on the fuzzy decomposition of the input space. For each part of the state space, a fuzzy rule is formed, the global output been a combination of the whole rules. A TS model is composed of 5 layers. In Figure 6, the model has two inputs variables. Two membership functions (antecedent fuzzy sets) are assigned to each one of them. The TS model is finally composed of two fuzzy rules. (That can be generalized to the case of n inputs and N rules). The rules perform a linear approximation of inputs:

$$\begin{aligned} R_i : & \text{ if } x_1 \text{ is } A_i^1 \text{ and } \dots \text{ and } x_n \text{ is } A_i^n \\ \text{THEN } & y_i = a_{i0} + a_{i1}.x_1 + \dots + a_{in}.x_n \end{aligned} \quad (1)$$

where R_i is the i^{th} fuzzy rule, N is the number of rules, $X_t = [x_1, \dots, x_n]^T$ is the input vector, A_i^j denotes the antecedent fuzzy sets, $j = [1, n]$, y_i is the output of the i^{th} linear subsystem with parameters a_{iq} , $q = [1, n]$.

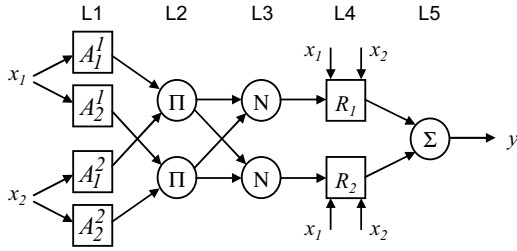


Figure 6: A First-order TS model

In layer 1, let assume Gaussian antecedent fuzzy sets to define the regions of fuzzy rules in which the local linear sub-models are valid:

$$\mu_i^j = \exp\left(-4 \cdot \left[\frac{\|x - x_i^*\|^j}{\sigma_i^j}\right]^2\right) \quad (2)$$

where σ_i^j is the spread of the membership function, and x_i^* is the focal point (center) of the i^{th} rule antecedent.

The firing level τ_i and the normalized firing level λ_i of each rule are obtained as follows (layers 2 and 3):

$$\tau_i = \mu_i^1(x_1) \times \dots \times \mu_i^n(x_n), \quad \lambda_i = \tau_i / \sum_{k=1}^N \tau_k \quad (3)$$

The model output is the weighted averaging of individual rules' contributions (layers 4 and 5):

$$y = \sum_{i=1}^N \lambda_i y_i = \sum_{i=1}^N \lambda_i x_e^T \pi_i \quad (4)$$

where $\pi_i = [a_{i0}, a_{i1}, \dots, a_{in}]$ is the vector parameter of the i^{th} sub-model, and $x_e = [1 \ X^T]^T$ (expanded vector).

A TS model has two types of parameters. The non-linear parameters are those of the membership functions (the center and the spread deviation for a Gaussian membership). These kinds of parameter are referred to as premise or antecedent parameters. The second types of parameters are the linear ones that form the consequent part of each rule (a_{iq} in equation 1).

3.3. Fitting a neuro-fuzzy systems: critical steps

A TS model can approximate an input-output function. In practice, this kind of model must be tuned to fit to the studied problem (prediction in our case). This "identification problem" is defined as follows.

The output of a fuzzy inference system (FIS) at time t is $y(t)$, and its input $X(t)$. Let note Z^t the "data set" as:

$$Z^t = \{y(1), X(1), \dots, y(t), X(t)\} \quad (5)$$

A model (the FIS) can be constructed as a mapping from past data Z^{t-1} to the next output $y(t)$. This model is known as the predictor model:

$$\hat{y}(t) = f(Z^{t-1}) \quad (6)$$

where $\hat{y}(t)$ represents the estimated output. This problem is a parameterizable mapping:

$$\hat{y}(t, \theta) = f(Z^{t-1}, \theta) \quad (7)$$

where θ is the vector of parameters. Since Z^{t-1} is continuously increasing (with new data), the identification problem is reformulated with a vector of fixed dimension (considering some past values of each variable):

$$\hat{y}(t, \theta) = f(\varphi(t), \theta) \quad (8)$$

where φ is known as the regression vector.

Using this notations, the problem of "model identification" of a FIS for prediction purpose implies at least four complementary tasks to be performed.

1. *Choice of input signals.* This step aims at determining which variables are relevant to model the desired input-output relation, i.e., to identify the exogenous variables of the phenomenon. It can be expert-made or result from processing techniques (feature extraction and selection). This problem is not addressed in this paper, but for subsequent explanations, let note $X = [x_1, x_2, \dots, x_n]$ the vector of exogenous variables that enables to estimate y , the endogenous variable.

2. *Inputs definition.* In order to make a prediction, one has to select the set of regressors for modeling (φ) of both the exogenous and endogenous variables. According to equation 8, and assuming h the horizon of prediction, the prediction at time $t+h$ is expressed as:

$$\hat{y}(t+h, \theta) = f(\varphi(t+h), \theta) \quad (9)$$

$$\varphi(t+h) = \begin{bmatrix} y(t-1), \dots, y(t-q_y), \\ x_1(t-1), \dots, x_1(t-p_{x_1}), \\ \dots \\ x_n(t-1), \dots, x_n(t-p_{x_n}) \end{bmatrix} \quad (10)$$

3. *Design of the structure.* This step aims at defining the form of the FIS $f(\cdot, \cdot)$. One has to choose the number and type of membership functions for each input (layer 1) and the number of rules which depends on the numbers of connections in between layers 2 and 3. The design of a TS model is thereby quite flexible. Nevertheless, choices made at this step directly influence the accuracy performance of the model as well as its complexity (as more membership functions and rules are chosen, as more parameters have to be tuned).

4. *Fitting the parameters.* The parameters of a TS model must be tuned to fit to the studied problem. This is the aim of the learning procedure and different approaches can be used. Most of the methods are based on the optimization of a cost function to minimize the “distance” between the predictions of the FIS and the real output data. Given k input-output past data, the estimated vector of parameters $\hat{\theta}$ is obtained as follows:

$$\hat{\theta}_k = \operatorname{argmin}_{\theta} [V_k(Z^k, \theta)] \quad (11)$$

$$V_k(Z^k, \theta) = \frac{1}{k} \sum_{t=1}^k d(\varepsilon(t, \theta)) \quad (12)$$

where, V_k is a cost (or loss) function, $\varepsilon(t, \theta)$ is the error of prediction, and $d(\cdot)$ is a distance measure. The classical quadratic cost function is mainly used since it allows to apply the least square procedure (see section 3.5):

$$V_k(Z^k, \theta) = \frac{1}{2k} \sum_{t=1}^k [y(t) - \hat{y}(t, \theta)]^2 \quad (13)$$

3.4. Reducing human influence with the evolving exTS

Whereas the first of these four steps is not studied in this paper, the second one is addressed in section 4. Following paragraphs enable to discuss the above points 3 and 4: choice of a structure and fit of parameters.

The simplest method to construct a TS fuzzy system is the “mosaic scheme”: the user defines the architecture of the model and the antecedent’s parameters values [27]. A prediction technique for prognostic purpose should not be tuned by an expert as it can be too difficult to catch the behavior of the monitored equipment. Thereby, this approach must be leaved aside to reduce the influence of practitioner.

Gradient descent algorithms have been adapted to the TS structure in order to calculate the antecedent parameters by the standard back-propagation procedure (see

for example the ANFIS system [28]). In the same way, genetic algorithms can be used to compute the fitting of antecedent parameters [29]. These approaches allow updating parameters by a learning process but are based on a fixed structure of the model: they require the user to choose the number of membership’s functions, and to initialize various algorithms parameters. Unfortunately, the accuracy of predictions is fully dependent on this and this type of approaches should also not be retained.

In opposition, clustering approaches (and some genetic algorithms) require less *a priori* information: they automatically generate the adequate structure of the model (number of membership functions, and of rules) [30, chap. 17]. However, in practical applications, the learning process is effective only if sufficient data are available and, when trained, such a TS model is fixed. Thereby, if the behavior of input and/or output changes significantly with regards to the learning phase (like in a degradation process), predictions can suffer from the lack of representative learning data.

In order to continuously integrate the dynamic of signals, evolving algorithms have finally been developed [26, 31, 32]. These algorithms are based on clustering methods and therefore, do not require the user to define the structure of the TS model. In opposition to all previous approaches, they do not need a complete learning data set to start the identification process of the TS model. The structure is flexible and evolves with the data gathered from the system (modification or addition of rules). This kind of self constructing predictors are thereby very interesting for prognostics applications where it is very difficult, even impossible, to formalize the behavior of the system. A particular evolving TS model is that one proposed by [26, 33]: the evolving extended TS system (exTS).

3.5. Theoretical exTS backgrounds

The learning procedure of exTS is composed of two phases: (1) an unsupervised data clustering technique is used to adjust the antecedent parameters, (2) the supervised Recursive Least Squares (RLS) learning method is used to update the consequent parameters.

a) Clustering phase: update of antecedent parameters

The exTS clustering phase processes on the global input-output data space: $z = [x^T, y^T]^T$, $z \in R^{n+m}$, $n+m$ defines the dimensionality of the input/output data space. Each one of the sub-model of exTS operates in a sub-area of z . The procedure is based on the calculus of a “potential” which is the capability of a data to form a cluster (antecedent of a rule). The clustering procedure starts from scratch and, as more data are available, the model evolves by replacement or upgrade of rules.

The coordinates of the first cluster center are those of the first data point ($z_1^* \leftarrow z_1$). The potential of the first data point is set to the ideal value ($P_1(z_1) \leftarrow 1$). Four steps are then performed for each new data gathered.

Step 1. Starting from $k = 2$, the potential P_k of the data point z_k is recursively calculated at time k :

$$P_k(z_k) = \frac{k-1}{k-1 + \sum_{j=1}^{k-1} \sum_{i=1}^{n+m} \|z_i - z_k\|_2^2} \quad (14)$$

Step 2. The potential of the cluster/rule centers is recursively updated:

$$P_k(z^*) = \frac{(k-1)P_{k-1}(z^*)}{k-2 + P_k(z^*) + P_k(z^*) \sum_{j=1}^{k-1} \|z^* - z_{k-1}\|_2^2} \quad (15)$$

Step 3. The potential of the data point (step 1) is compared to boundaries issued from the potential of the cluster centers (step 2):

$$(\underline{P} \leq P_k(z_k) \leq \bar{P}) \quad (16)$$

where $\bar{P} = \max_{i=1}^N \{P_i(z^*)\}$ is the highest density/potential, $\underline{P} = \min_{i=1}^N \{P_i(z^*)\}$ is the lowest density/potential and N is number of centers clusters (x_i^* , $i = [1, N]$) formed at time k .

Step 4. If, the new data point has a potential in between the boundaries any modification of the rules is necessary. Else, they are two possibilities:

- if the new data point is closed to an old center ($\min_{i=1}^N \|x_k - x_i^*\|_j < \frac{\sigma_i^j}{2}$), then the new data point (z_k) replaces this center ($z_i^* \leftarrow z_k$),
- else, the new data point is added as a new center and a new rule is formed ($N = N + 1$; x_N^*).

Note that, the exTS learning algorithm presents an adaptive calculation of the radius of the clusters (σ_i^j). See [33] for more details.

b) RLS phase: update of consequent parameters

At step k , equation 4 can be expressed as follows:

$$\hat{y}_{k+1} = \sum_{i=1}^N \lambda_i y_i = \sum_{i=1}^N \lambda_i x_e^T \pi_i = \psi_k^T \hat{\theta}_k \quad (17)$$

where $\psi_k^T = [\lambda_1 x_e^T, \dots, \lambda_n x_n^T]^T$ is a vector of the inputs, weighted by normalized firing (λ) of the rules, and $\hat{\theta}_k = [\hat{\pi}_1^T, \dots, \hat{\pi}_N^T]^T$ are parameters of the sub-models.

The estimated parameters based on k data samples are obtained by applying the following RLS procedure:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k (y_{k+1} - \psi_k^T \hat{\theta}_{k-1}); \quad k = 2, 3, \dots \quad (18)$$

$$C_k = C_{k-1} - C_{k-1} \psi_k \psi_k^T C_{k-1} / (1 + \psi_k^T C_{k-1} \psi_k) \quad (19)$$

with initial conditions $\theta_1 = [\pi_1^T, \dots, \pi_N^T]^T = 0$, $C_1 = \Omega I$, where Ω is a large positive number, ΩI a $R(n+1) \times R(n+1)$ co-variance matrix.

3.6. Brief synthesis

Thanks to its evolving capability, the exTS system does not requires the user to define the structure of the predictor and to initialize the parameters (critical steps 3 and 4 defined in section 3.3). It is thereby a useful tool to tempt to reduce human influence when building a prediction model.

4. Applying parsimony principle on an exTS

4.1. Problem statement

A TS model is a universal approximator, i.e., a model that has the ability to approximate any function to an arbitrary degree of accuracy. Following that, one can be tempted to use the model that better fits the data. However, there is a point at which increasing the complexity of the model marginally increases the fitting. More over, a model that excessively fit to the learning data set can have low generalization capability since it will only be able to approximate data near from this set. Thus, it is also useful to keep the model as simple as possible as long as it fits reasonably well the data. Following that, practitioners are faced up with the problem of balancing the accuracy of the model built and its complexity.

Whereas the number of rules is automatically computed in the clustering phase of the exTS learning procedure, the number of inputs must be set by the practitioner: which is the right set of regressors (see the second critical steps defined in 3.3)? This implies a more or less complex fuzzy structure and the prediction performance follows from it. Thereby, the purpose of this section is to propose a way to automatically generate an accurate exTS prediction system that reaches a compromise between complexity and generalization capability. The approach is based on the use of parsimony criteria, that implicate the complexity and the accuracy to be defined.

4.2. Defining the complexity of an exTS system

The architecture of an exTS system is singular in that the number of rules is equal to the number of membership's functions by input. Thereby and assuming that exTS is a first-order TS model, the number of linear parameters can be simply determined by the relation $nb_l = N \times (n + 1)$, where, N is the number of rules and n , the number of inputs. As being Gaussian, each membership has 2 parameters to be tuned and the total amount of non-linear parameters is expressed by $nb_{nl} = 2 \times n \times N$. The total number of parameters of an exTS can be interpreted as an indicator of the complexity of the model:

$$p = \dim(\theta) = nb_l + nb_{nl} = N(3 \times n + 1) \quad (20)$$

4.3. Defining parsimony criteria

Parsimony criteria provide a mathematical formulation of the compromise in between complexity and quality of predictions of a model [34]. These criteria are thereby based on two aspects: the number of adjusted parameters (p), and a loss function that traduces the accuracy of past estimations (likelihood function L). Assuming S the number of data samples used when fitting the model, the most commonly used likelihood function is the mean square error:

$$L = \frac{1}{S} \sum_{i=1}^S (y - \hat{y})^2 \quad (21)$$

Various parsimony criteria are proposed in literature: the Akaike Information Criterion (AIC), the Rissanen's Minimum Description Length (MDL), the Final Prediction Error (FPE) and the Bayesian Information Criterion (BIC). These criteria are defined as follows.

$$\begin{aligned} AIC &= S \times \log(L) + 2 \times p \\ MDL &= (1 + [p \times \log(L)]/S) \times L \\ FPE &= [(S + p)/(S - p)] \times L \\ BIC &= -2 \times \log(L) + p \times \log(S) \end{aligned} \quad (22)$$

These criteria are designed in order to decrease as long as the model significantly fits the data better, and to increase when improvements are marginal. Following that, various identification / prediction models can be ranked according to their parsimony criteria, the more suitable been that of the lowest ones (suitable in the sense of "compromise in between accuracy and complexity"). Consider Figure 7 for an illustration. Note also that AIC , MDL , FPE and BIC criteria can be implemented without the aid of subjective judgment.

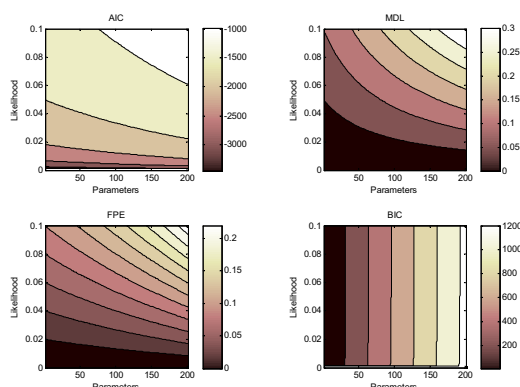


Figure 7: Evolution of parsimony criteria

4.4. Balancing complexity and accuracy of an exTS

As stated in equations 9 and 10, in order to build a prediction model, one has to define the set of regressors of both the exogenous and endogenous variables. Let note $O = \{p_{x_1}, p_{x_2}, \dots, q_y\}$ the order of the model containing the number of past values used for prediction for each input. Depending on this choice, many forecasting systems can be constructed. In addition, and thanks to equations 20 and 21, parsimony criteria (AIC , MDL , FPE and BIC) of all these potential models can be easily computed during the learning phase. Following that, one is able to determine which model presents the lowest one and is thereby providing a compromise between complexity and accuracy. This procedure is depicted in Figure 8. In this paper, various parsimony criteria are proposed for discussion purpose but a single one is sufficient to select the number of inputs.

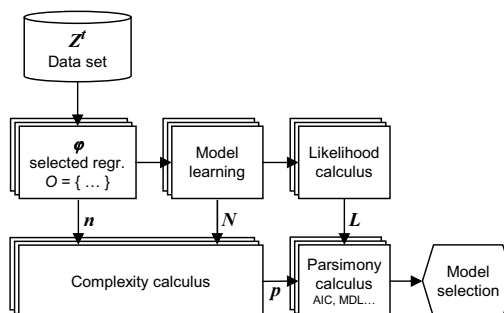


Figure 8: Procedure to build a "suitable" exTS

4.5. Experiments and discussion

The procedure as been applied on a data set issued from an hair dryer. It has been contributed by W. Favoreel from the Kuleuven Unversity [35]. The aim of predictions is to approximate the air temperature of a dryer by learning real gathered data. That can be assimilated to the prediction step of the prognostics process. Data have inside one exogenous variable (x_1) and one endogenous variable (y):

- x_1 : voltage of the heating device,
- y : air temperature.

Experiments aim, on one side, to evaluate the ability of an exTS system to predict a non-linear function, and on the other side, in order to evaluate the ability of the input selection procedure based on parsimony criteria to reach a compromise between generalization and complexity. The Auto Regressive exogenous approach

(ARX) has been used for comparisons: this model is extensively employed in modeling and prediction of times series. Given a set of inputs x_i , an ARX prediction can be expressed as follows.

$$\begin{aligned}
 A(w) \cdot \hat{y}(t) &= \sum_{i=1}^n B_i(w) \cdot x_i(t) \\
 A(w) &= 1 + \alpha_1 w^{-1} + \dots + \alpha_{q_y} w^{-q_y} \\
 B_i(w) &= \beta_1^i w^{-1} + \dots + \beta_{p_{x_i}}^i w^{-p_{x_i}} \\
 w^{-\tau} \cdot x(t) &= x(t - \tau)
 \end{aligned} \tag{23}$$

As for the exTS, let note $O = \{p_{x_1}, p_{x_2}, \dots, q_y\}$ the order of an ARX model, with total amount of parameters:

$$p = q_y + p_{x_1} + \dots + p_{x_n} \tag{24}$$

The data set contains 1000 samples. The learning phase was stopped after 500 data samples, and the remaining samples served to test the models at one step ahead. The accuracy of learning was estimated with the mean square error (MSE_{train}) which is the most popular error measure, and prediction performance in test was assessed by using the coefficient of determination ($R2_{test}$) which is a measure of how well future outcomes are likely to be predicted by the model.

For both models (ARX and exTS), the procedure of input selection based on parsimony criteria has been applied in order to identify the *a priori* more suitable structures, i.e., that ones that balance complexity and accuracy (4 structures per type of model, let note them ARX_{AIC} , $exTS_{MDL}$...). In order to extract more solid conclusions from the tests and expand the comparison to a same model (and not from one to another), the most simple structures (ARX_{pmin} , $exTS_{pmin}$) as well as the most accurate models in training ($ARX_{MSE_{train}}$, $exTS_{MSE_{train}}$) and in test ($ARX_{R2_{test}}$, $exTS_{R2_{test}}$) have also been searched by scanning all the set of possible inputs (thanks to the set of potential regressors φ). Results are shown in Table 1.

exTS versus ARX. According to accuracy performance indicators, whatever the type of structure is considered (the most simple, the most accurate in learning...), the exTS system always overpass the ARX model in term of prediction. An example is given in Figure 9 that depicts the prediction results for the two most simple models ARX_{pmin} and $exTS_{pmin}$. Considering our final applicative objective (prognostics of failures), this is of good omen since the exTS not only appears to be a suitable prediction tool, but also is able to evolve to take into account the dynamic of real systems without prior knowledge and human intervention.

Accuracy and complexity. Results enable to notice that, whatever the model of prediction is, the accuracy

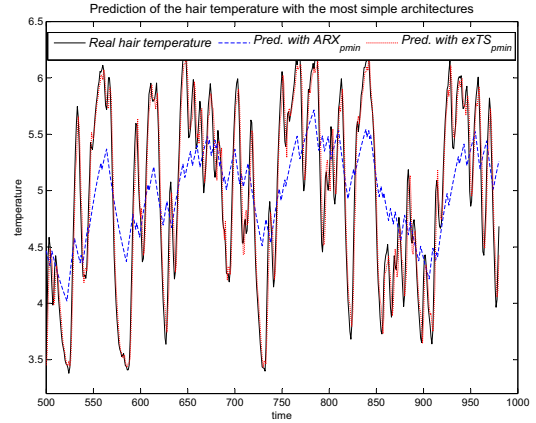


Figure 9: Prediction results with the simplest models

globally grows as the complexity increases (Figure 10): the more $R2_{test}$ is near from 1, the more parameters the model has. This confirms that accuracy and complexity are actually correlated and that there is no way to maximize the satisfaction on both criteria. More over, note also that the most accurate model in training is not the most accurate in test, and that the most accurate in test is not necessarily the most complex model. That also reinforces the importance of looking for a compromise.

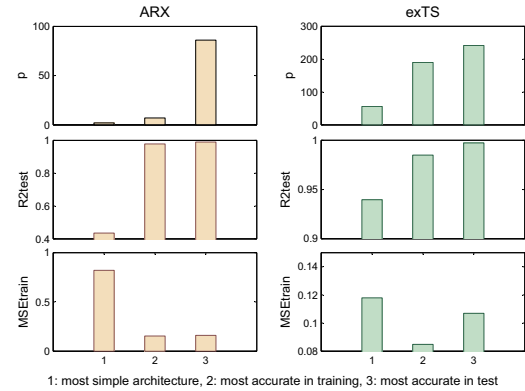


Figure 10: Some accuracy and complexity results

Interest of the procedure. The interest of using the automatic input selection based on parsimony criteria can be examined. Let take for example the results obtained with the AIC criterion. Whereas the most accurate prediction model for both systems ($ARX_{R2_{test}}$, $exTS_{R2_{test}}$) has much more parameters than this one obtained by the input selection procedure (ARX_{AIC} and

	ARX	exTS
MOST SIMPLE STRUCTURES ($ARX_{pmin}, exTS_{pmin}$)		
$O = \{p_{x1}, q_y\}$; nb param.	{1, 1} ; $p = 2$	{1, 1} ; 8 rules ; $p = 56$
<i>MSEtrain</i>	0,8200	0,1179
<i>R2test</i>	0,4354	0,9397
MOST ACCURATE IN TRAINING ($ARX_{MSEtrain}, exTS_{MSEtrain}$)		
$O = \{p_{x1}, q_y\}$; nb param.	{4, 3} ; $p = 7$	{1, 5} ; 10 rules ; $p = 190$
<i>MSEtrain</i>	0,1540	0,0848
<i>R2test</i>	0,9782	0,9850
MOST ACCURATE IN TEST ($ARX_{R2test}, exTS_{R2test}$)		
$O = \{p_{x1}, q_y\}$; nb param.	{85, 1} ; $p = 86$	{4, 3} ; 11 rules ; $p = 242$
<i>MSEtrain</i>	0,1603	0,1069
<i>R2test</i>	0,9894	0,9974
STRUCTURES BALANCING COMPLEXITY AND ACCURACY		
<i>AIC</i>	-918,6320	-972,5249
$O = \{p_{x1}, q_y\}$; nb param.	{4, 4} ; $p = 8$	{1, 2} ; 8 rules ; $p = 80$
<i>MSEtrain</i>	0,1542	0,1038
<i>R2test</i>	0,9784	0,9827
<i>MDL</i>	0,1669	0,2368
$O = \{p_{x1}, q_y\}$; nb param.	{4, 2} ; $p = 6$	{1, 4} ; 9 rules ; $p = 144$
<i>MSEtrain</i>	0,1553	0,0849
<i>R2test</i>	0,9717	0,9853
<i>FPE</i>	0,1584	-20,1606
$O = \{p_{x1}, q_y\}$; nb param.	{4, 3} ; $p = 7$	{2, 9} ; 15 rules ; $p = 504$
<i>MSEtrain</i>	0,1540	0,1996
<i>R2test</i>	0,9782	0,9795
<i>BIC</i>	12,8261	624,0720
$O = \{p_{x1}, q_y\}$; nb param.	{1, 1} ; $p = 2$	{2, 1} ; 10 rules ; $p = 100$
<i>MSEtrain</i>	0,8200	0,2710
<i>R2test</i>	0,4354	0,9495

Table 1: Air temperature - simulation results

$exTS_{AIC}$), the accuracy of prediction are quite the same. As for an example, consider Figure 11 that shows the probability density function (pdf) of the percentage of error of prediction for $exTS_{R2test}$ and $exTS_{AIC}$ models. Both pdf curves are much closed together, the main difference being that the spread deviation of the “compromise structure” is slightly bigger than this one of the “most accurate structure”. In other words, the proposed approach really enables to balance complexity and generalization capability and allows to directly choose an adequate prediction structure.

5. Illustration: prognostic of a real engine health

5.1. Object

The proposed data-driven procedure is illustrated by using the challenge dataset of diagnostic and prognostics of machine faults from the first International Con-

ference on Prognostics and Health Management (2008) [36]. The dataset consists of multiple multivariate time series (26 variables) with sensor noise. Each time series is from a different engine of the same fleet and each engine starts with different degrees of initial wear and manufacturing variation unknown to the user and considered normal. The engine is operating normally at the start and develops a fault at some point. The fault grows in magnitude until system failure.

The whole prognostic procedure applied is depicted in Figure 12. It consists in three phases similar to that one of layers 2 to 5 of CBM architecture (see section 2.1). Data were first processed (feature extraction, selection and cleaning). It enabled to feed a prediction engine which forecasted observations in time. These predictions were then analyzed by a classifier which provided the most probable state of the system. The remaining useful life was finally deduced thanks to the

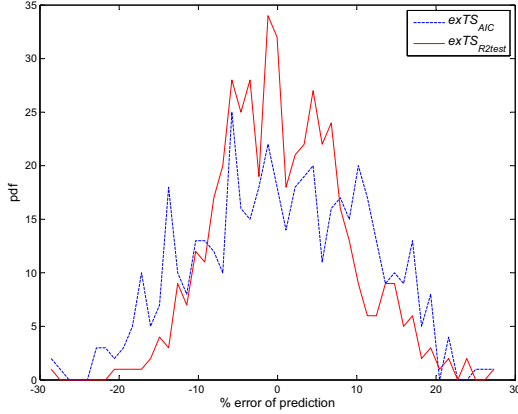


Figure 11: Pdf of the error of prediction (%): $exTS_{R2test}$ and $exTS_{AIC}$

estimated time to reach the failure mode. The processing, prediction and classification steps were supported by three different tools that are respectively based on information theory, on the self-built exTS system and on Dempster-Shafer theory. In this paper, only the prediction aspect is illustrated. For more details on selection and classification steps, refer to [37].

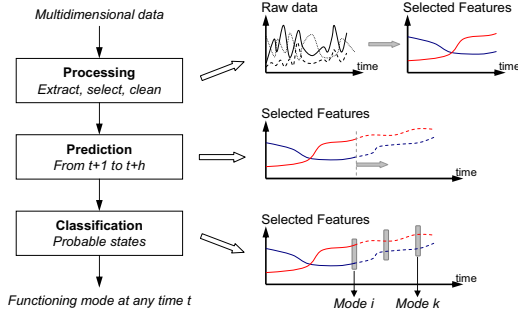


Figure 12: Whole procedure applied to the real engine health

5.2. Prediction results and discussion

The prediction step has been deployed in a reduced set of features: from the 26 variables of the row data set, 8 of them were first selected and cleaned. Following that, each feature has been estimated and predicted thanks to the approach proposed in this paper (with *AIC* criterion): with a self-built exTS prediction system at $t + 1$ and by applying the input selection procedure. In order to perform predictions at any time, a “cascade model” of each system has been built: previous predictions can be used as for the inputs for next predictions

(Figure 13). This type of architecture enables to perform multi-step ahead predictions (at any $t + h$) without building various systems (and thereby with a single learning phase). All predictions were made until time $t = 50$, so that, for each data test set, the prediction module provided the expected values of the considered performance index from time $t = 51$ to the end of the test series (around 250). For tests, 10 multivariate time series were used for training, and 10 for testing.

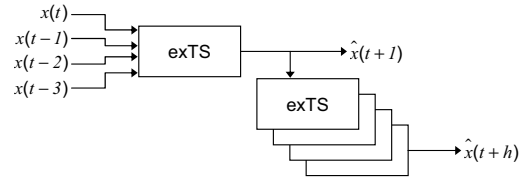


Figure 13: Cascade model for multi-step ahead prediction

Table 2 resumes the set of inputs variables that have been obtained thanks to the input selection procedure (the set of regressors). This result would have been prohibitively long to obtain if human-made.

Feature	Inputs
F1	$t, x_1(t), x_1(t-1)$
F2	$t, x_2(t)$
F3	$t, x_3(t)$
F4	$t, x_4(t), x_4(t-1), x_4(t-2), x_4(t-3)$
F5	$t, x_5(t), x_5(t-1), x_5(t-2)$
F6	$t, x_6(t), x_6(t-1), x_6(t-2), x_6(t-3)$
F7	$t, x_7(t), x_7(t-1)$
F8	$t, x_8(t), x_8(t-1)$

Table 2: Engine health - input selection

Figure 14 depicts the parsimony principle on the first feature. Results are similar as those of the hair temperature data set used in section 4.5: since the complexity increases with the accuracy of prediction, one has to look for a compromise.

Figure 15 shows an example of the evolution of the performance index for the first feature, and the prediction that can be obtained thanks to the prediction procedure. It is representative of the results that can be obtained with the approach proposed in this paper: the prediction system is suitable, even if self-built and without prior knowledge on its structure and parameters.

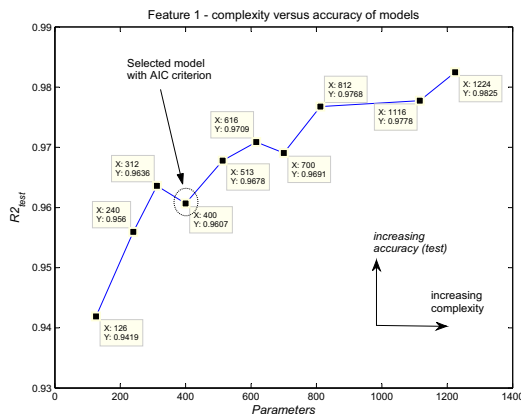


Figure 14: Complexity and accuracy of prediction models - feature 1

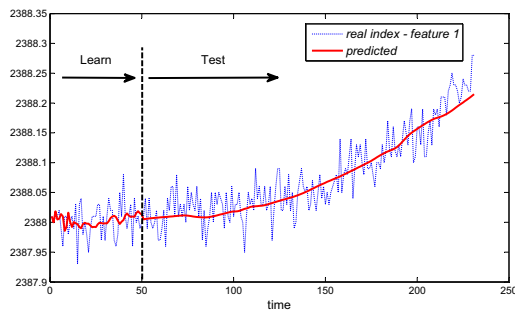


Figure 15: Example of prediction results

6. Conclusions

In maintenance field, prognostic is recognized as a key feature as the prediction of the remaining useful life of a system allows avoiding inopportune maintenance spending. However, it can be a non trivial task to develop and implement effective prognostics models including the inherent uncertainty of prognostics.

The aim of this paper is to point out a suitable prediction technique to perform the approximation and prediction of the degradation of an equipment, and for which any assumption or intervention of the practitioner is necessary, i.e., to propose a way to go through self built systems. That can be reached thanks to three aspects.

Firstly, the evolving exTS system is an accurate prediction tool that outperforms classical approaches. Indeed, this model has a high level of adaptation to the environment and to the changing data, and is thereby useful for complex modeling and prediction.

Secondly, the evolving capability of the exTS provides practitioners with a tool that does not need the

user to make assumptions on the structure or on initial condition for model building. The learning phase of the exTS starts from scratch and the predictor evolves as data are gathered, with no human intervention.

Thirdly, the procedure for input selection derived from the parsimony criteria enables to automatically generate a suitable architecture that balance accuracy and complexity, without human choices.

References

- [1] A. Muller, M.C. Suhner, B. Iung, Formalisation of a new prognostic model for supporting proactive maintenance implementation on industrial system, *Reliability Engineering and System Safety* 93 (2008) 234–253.
- [2] T. Brotherton, G. Jahns, J. Jacobs, D. Wroblewski, Prognosis of faults in gas turbine engines, in: 2000 IEEE International Conference on Aerospace, Vol. 6, 2000, pp. 163–171.
- [3] F.E. Ciarapica, G. Giacchetta, Managing the condition-based maintenance of a combined-cycle power plant: an approach using soft computing techniques, *Journal of Loss Prevention in the Process Industries* 19 (2006) 316–325.
- [4] A. Heng, S. Zhang, A. Tan, J. Matwev, Rotating machinery prognostic: State of the art, challenges and opportunities, *Mechanical Systems and Signal Processing* 23 (2009) 724–739.
- [5] A. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mech. Syst. and Sign. Proc.* 20 (2006) 1483–1510.
- [6] C. Byington, M. Roemer, G. Kacprzynski, T. Galie, Prognostic enhancements to diagnostic systems for improved condition-based maintenance, in: 2002 IEEE Aerospace Conf., Big Sky, USA, 2002.
- [7] J.G. De Gooijer, R.J. Hyndman, 25 years of time series forecasting, *International Journal of Forecasting* 22 (2006) 443–473.
- [8] G. Vachtsevanos, F.L. Lewis, M. Roemer, A. Hess, B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, New Jersey, Hoboken: Wiley & Sons, 2006.
- [9] W.Q. Wang, M.F. Goldnaraghi, F. Ismail, Prognosis of machine health condition using neuro-fuzzy systems, *Mechanical Systems and Signal Processing* 18 (2004) 813–831.
- [10] W.Q. Wang, F. Ismail, M.F. Goldnaraghi, A neuro-fuzzy approach to gear system monitoring, *IEEE Transaction Fuzzy Systems* 12 (2004) 710–723.
- [11] R.C.M. Yam, P.W. Tse, L. Li, P. Tu, Intelligent predictive decision support system for condition-based maintenance, *International Journal of Advanced Manufacturing Technology* 17 (2001) 383–391.
- [12] V.T. Tran, B-S. Yang, A.C.C. Tan, Multi-step ahead direct prediction for the machine condition prognosis using regression trees and neuro-fuzzy systems, *Expert Systems with Applications* 36 (2009) 378–387.
- [13] M. Lebold, M. Thurston, Open standards for condition-based maintenance and prognostics systems, in: 5th Annual Maintenance and Reliability Conference, 2001.
- [14] ISO, 13374-2, Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 2: Data processing, International Standard, ISO, 2006.
- [15] C.R. Farrar, F. Hemez, G. Park, A.N. Robertson, H. Sohn, T.O. Williams, A coupled approach to developing damage prognosis solutions, in: DAMAS 2003, Fifth International Conference on Damage Assessment of Structures, 2003.

- [16] D. Lin, V. Makis, Recursive filters for a partially observable system subject to random failure, *Advances in Applied Probability* 35 (2003) 207–227.
- [17] M. El-Koujok, R. Gouriveau, N. Zerhouni, Error estimation of a neuro-fuzzy predictor for prognostic purpose, in: *SAFEPRO-CESS 09, 7th IFAC International Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2009.
- [18] R. Chinnam, B. Pundarikaksha, A neuro-fuzzy approach for estimating mean residual life in condition-based maintenance systems, *International Journal of Materials and Product Technology* 20 (2004) 166–179.
- [19] ISO, 13381-1, Condition monitoring and diagnostics of machines - prognostics - Part1: General guidelines, International Standard, ISO, 2004.
- [20] O. Dragomir, R. Gouriveau, N. Zerhouni, F. Dragomir, Framework for a distributed and hybrid prognostics system, in: *MCPL 07, 4th IFAC Conference on Management and Control of Production and Logistics*, 2007.
- [21] O. Dragomir, R. Gouriveau, N. Zerhouni, Adaptive neuro-fuzzy inference system for mid term prognostics error stabilization, *International Journal of Computers, Communications and Control* 3 (2008) 271–276.
- [22] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, Metrics for evaluating performance of prognostic techniques, in: *IEEE International Conference on Prognostics and Health Management, PHM 2008*, 2008.
- [23] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, On applying the prognostic performance metrics, in: *Annual Conference of the Prognostics and Health Management Society*, 2009.
- [24] K. Goebel, P. Bonissone, Prognostic information fusion for constant load systems, in: *7th annual Conference on Fusion*, Vol. 2, 2005, pp. 1247–1255.
- [25] S.J. Engel, B.J. Gilmartin, K. Bongort, A. Hess, Prognostic, the real issues involved with predicting life remaining, in: *2000 IEEE International Conference on Aerospace*, Vol. 6, 2000, pp. 457–469.
- [26] P. Angelov, D. Filev, An approach to online identification of takagi-sugeno fuzzy models, *IEEE Transaction Syst. Man Cybern. - Part B: Cybernetics* 34 (2004) 484–498.
- [27] J. Espinosa, J. Vandewalle, V. Wertz, *Fuzzy Logic, Identification and Predictive Control*, Springer-Verlag, 2004.
- [28] J.S.R. Jang, C.T. Sun, Neuro-fuzzy modeling and control, *IEEE Proceedings* 83 (1995) 378–406.
- [29] P. Angelov, C. Xydeas, Fuzzy systems design: direct and indirect approaches, *Soft Computing* 10 (2006) 836–849.
- [30] J.V. de Oliveira, W. Pedrycz, *Advances in fuzzy clustering and its applications*, England, Chichester: Wiley & Sons, 2007.
- [31] N. Kasabov, Q. Song, Denfis: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Transaction on Fuzzy Systems* 10:2 (2002) 144–154.
- [32] N. Kasabov, *Evolving connectionist systems - the knowledge engineering approach*, 2nd edition, England, London: Springer Verlag, 2007.
- [33] P. Angelov, X. Zhou, Evolving fuzzy systems from data streams in real-time, in: *International Symposium On Evolving Fuzzy Systems*, 2006, pp. 26–32.
- [34] H. Akaike, A new look at the statistical model identification, *IEEE Transaction Automatic Control* 19 (1974) 716–723.
- [35] <ftp://ftp.esat.kuleuven.ac.be/sista/data/mechanical>.
- [36] A. Saxena, K. Goebel, D. Simon, N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in: *IEEE Prognostics and Health Management conference*, 2008.
- [37] E. Ramasso, R. Gouriveau, Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions damage propagation modeling for aircraft engine run-to-failure simulation, in: *IEEE Frontiers of prognostics and Health Management conference*, 2010.