Remaining useful life estimation by classification of predictions based on a neuro-fuzzy system and theory of belief functions

Emmanuel Ramasso, *Member, IEEE,* Rafael Gouriveau, *Member, IEEE*

***Abstract*—Various approaches for prognostics have been developed, and data-driven methods are increasingly applied. The training step of these methods generally requires huge datasets to build a model of the degradation signal, and estimate the limit under which the degradation signal should stay. Applicability and accuracy of these methods are thereby closely related to the amount of available data, and even sometimes requires the user to make assumptions on the dynamics of health states evolution. Following that, the aim of this paper is to propose a method for prognostics and remaining useful life estimation that starts from scratch, without any prior knowledge. Assuming that remaining useful life can be seen as the time between the current time and the instant where the degradation is above an acceptable limit, the proposition is based on a classification of prediction strategy (CPS) that relies on two factors. First, it relies on the use of an evolving real-time neuro-fuzzy system that forecasts observations in time. Secondly, it relies on the use of an evidential Markovian classifier based on Dempster-Shafer theory that enables classifying observations into the possible functioning modes. This approach has the advantage to cope with a lack of data using an evolving system, and theory of belief functions. Also, one of the main assets is the possibility to train the prognostic system without setting any threshold. The whole proposition is illustrated and assessed by using the CMAPPS turbofan dataset. RUL estimates are shown to be very close to actual values, and the approach appears to accurately estimate the failure instants, even with few learning data.**

***Index Terms*—Prognostics, Takagi-Sugeno systems, belief functions, classification of prediction.**

E. Ramasso and R. Gouriveau are with FEMTO-ST Institute, Automatic Control and Micro-Mechatronic Systems Department (AS2M), UMR CNRS 6174 - UFC / ENSMM / UTBM, 24 rue Alain Savary, Besançon, 25000 France. e-mails: firstname.familyname@femto-st.fr

ACRONYMS AND ABBREVIATIONS

BBA	Basic Belief Assignment
CBM	Condition-Based Maintenance
CMAPPS	Commercial Modular Aero-Propulsion System Simulation
CPS	Classification of Prediction Strategy
EvHMM	Evidential Hidden Markov Model
exTS	Evolving extended Takagi-Sugeno system
FN, FP	False negative, false positive
HMM	Hidden Markov Model
ITS	Iterative transition estimation algorithm
KL	Kullback-Leibler divergence
PHM	Prognostics and health management
RCGI	Regrouping components with geometric interaction algorithm
RLS	Recursive Least Squares
RUL	Remaining Useful Life

NOTATIONS

$\mathbf{X}$, and $\mathbf{Y}$	Input, and output data sets
$\hat{\mathbf{Y}}$	Estimation of $\mathbf{Y}$
$\mathbf{Z}$	joint input-output space
$\epsilon = \mathbf{Y} - \hat{\mathbf{Y}}$	Residual of estimates
msp	Multi-step ahead predictions
NL	Number of training data used to train exTS
NC	Number of training data to infer predictions in exTS and then used in EvHMM
F	Dimension of the feature vector
H	Horizon of prediction
k	Time instant
m^{Ω_k}	Basic belief mass defined on the frame of discernment Ω_k
q, pl	Commonality, Plausibility functions
M	Number of components in a state in EvHMM
N	Number of states in EvHMM
θ_k	Linear model parameters in exTS at k
C_k	Uncertainty of model parameters at k
$\mathcal{I}$	Interval of good prediction
$\mathcal{A}_{RUL}^{k_0}$	Accuracy of RUL estimates at critical time k_0
E	Difference between predicted and true RUL

I. INTRODUCTION

Prognostics is now recognized as a key process in maintenance strategies as the estimation of the remaining useful life (RUL) of equipment allows avoiding critical damage and expense. Various prognostics approaches have now

been developed, classified into three categories: model-based, data-driven, and experience-based approaches [1]–[4]. Data-driven approaches aim at transforming raw monitoring data into relevant information and behavior models (including the degradation) of the system. They take as inputs the current monitoring data, and return as outputs predictions or trends about the health state of the system. These approaches offer an alternative to other approaches, especially in cases where obtaining in-situ data is easier than constructing physical or analytical behavior models. Indeed, in many applications, measured input-output data is the major source of information for a deeper understanding of the system degradation. Following that approach, data-driven approaches are increasingly applied to machine prognostics (mainly techniques from Artificial Intelligence). However, data-driven approaches are highly statistically dependent on the quantity and quality of operational data that can be gathered from the system. This effect is the topic addressed in this paper: a method for prognostics is proposed to face the problem of lack of information and missing prior knowledge in prognostics applications.

The approach aims at predicting the failure mode early, while the system can switch between several functioning modes. The approach is based on a classification of predictions strategy (CPS), and consists thereby in two main phases. 1) An evolving neuro-fuzzy system (exTS) is used for on-line multi-step ahead prediction of observations (prediction step). This phase is able to start from scratch, and is thus well-suited for applications where only a small amount of data are available. 2) The predicted observations are then classified into functioning modes using an evidential Markovian classifier called Evidential Hidden Markov Model (EvHMM), and based on Dempster-Shafer theory (classification step). This classifier relies on a training procedure that adapts the number of parameters according to the data. The use of belief functions makes this classifier robust to a lack of information.

To our knowledge, the idea of using classifiers instead of manually-tuned thresholds in prognostics and health management (PHM) has been initially mentioned in [5] with Cumulative Shock Models, and in [6] where the authors presented the concept of post-prediction situation assessment. The use of the sequence of states method has then been introduced in [7]. In this paper, a method is proposed to automatically build the threshold from both a set of data and some labels representing possible functioning modes. Compared to previous work, the main advantage of using a classifier is the possibility to consider multidimensional health indices or sensor measurements. The method described in this paper is an enhancement of previous works published in [7], [8], and in two international conferences supported by the IEEE Reliability Society: [9], [10]. In particular, three main contributions can be pointed out.

1) RUL estimation is performed by a classification of predictions strategy. In the proposed scheme, there is no use of *a priori* failure thresholds. Instead, RUL estimates are performed by detecting transitions to faulty modes.
2) The approach combines two efficient tools for handling a lack of information: a neuro-fuzzy system (exTS), and an Evidential Hidden Markov Model (EvHMM).
3) A procedure is proposed to train the EvHMM classifier.
4) The proposed methodology is validated on a dataset generated from the Commercial Modular Aero-Propulsion System Simulation (CMAPPS) by studying the influence of the quantity of data in RUL estimation.

The paper is organized in three main parts. The global prognostics approach is first presented. Then, main theoretical backgrounds concerning prediction and classification steps are given. The whole proposition is finally illustrated on a real-world prognostics problem concerning the prediction of an engines health. This part enables deeply analyzing the effect of the size of the training dataset.

II. Prognostics architecture, a classification of prediction strategy

A. The approach as a specific case of CBM

According to the standard ISO 13381-1:2004, prognostics is the "estimation of time to failure and risk for one or more existing and future failure modes" [11]. It is thereby a process for predicting the RUL before a failure occurs. However, prognostics cannot be seen as a single task because all aspects of failure analysis and prediction have to be performed. This idea is highlighted within the Condition-Based Maintenance (CBM) concept. Usually, a CBM system is decomposed into seven layers, one of them being that of prognostics [12]. The main purpose of each layer is described in the following.

1) The sensor module provides the system with digitized sensor or transducer data.
2) The processing module performs signal transformations and feature extractions.
3) The condition monitoring module compares on-line data with expected values.
4) The health assessment module determines if the system has degraded.
5) The prognostics module predicts the future condition of the monitored system.
6) The decision support module provides recommended actions to fulfill the mission.
7) The presentation module can be built into a regular machine interface.

In this paper, only layers 3 through 5 are considered.

B. Proposition of a data-driven classification of predictions strategy (CPS)

Consider a monitored system that can switch within various functioning modes. The proposed approach links multidimensional data to the RUL of the system (Fig. 1). Data are first processed (feature extraction, selection, and cleaning), and then used to feed a prediction engine which forecasts observations in time. These predictions are then analyzed by a classifier which provides the most probable state of the system. This action is the Classification of Predictions Strategy (CPS). The RUL is finally deduced thanks to the estimated time to reach the failure mode. The processing part is not considered in this paper, but the reader can refer to [9] for an example of

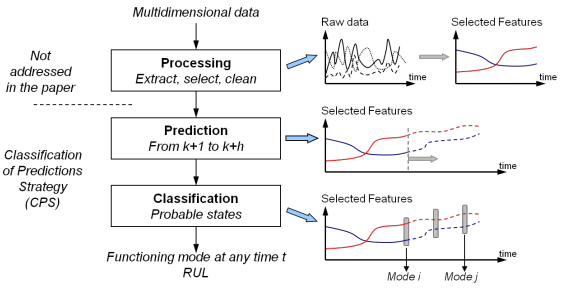


Figure 1. Prognostics architecture with CPS .

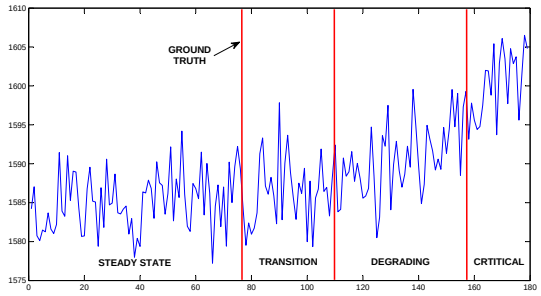


Figure 2. Segmentation of data.

variables selection based on Choquet Integral and information theory.

The classifier requires the data to be segmented into two or more functioning modes. It estimates at each time a confidence value that reflects how likely predictions are close to each functioning mode. This segmentation is a prior information that can be provided either by expert annotation (if available) [9], or by a clustering tool [13], [14]. For example, in Fig. 2, the data depicted concern the evolution of a health performance index segmented into four functioning modes: steady state, degrading state, transition state, and critical state. The set comprising the data and the ground truth concerning the modes is called the training dataset.

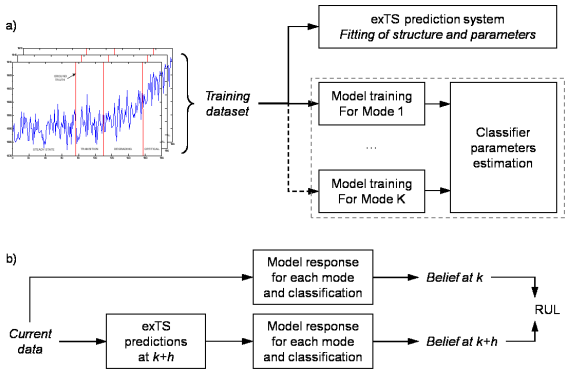


Figure 3. CPS Procedure: a) training step, b) testing step.

C. CPS procedure, and algorithm

In this paper, prediction and classification steps are performed by two different tools (detailed in the sequel) that are the exTS [15], and the (EvHMM) [10]. Both algorithms can be trained using a small amount of data, and were developed to cope with modeling time series when only a few data are available.

1) Algorithm exTS can start from a few data points to initialize the fuzzy rules, and then its structure (number of rules and parameters) is adapted recursively for each new data point.
2) Algorithm EvHMM adapts its parameters according to the amount of data available, and manages uncertainties using belief functions [16].

The different steps to estimate the RUL by CPS strategy are represented in Fig. 3. It requires 1) a training dataset composed of N_{exp} experiments, each of them being composed of F time-series features; and 2) the set of labels corresponding to the functioning mode at any time in each time series.

A part of the training dataset (NL experiments) is first used to learn a prediction model for each feature (F predictors are thus built). At this step, neuro-fuzzy approximation algorithms (such as exTS) are used to face the disparity of data in a simple manner, and without prior knowledge or human assumptions. The neuro-fuzzy system is then used to perform predictions on NC experiments. Those predictions, accompanied by the labels corresponding to the functioning modes, are finally used to train a classifier system that aims at assessing the health state at any time (current, and future functioning modes). The underlying idea of feeding the classifier with predictions is to build a classifier system that is able to compensate for the error of predictions.

Note that the proposed classification approach is not a discriminate one (learning a classifier for a class against another). We would rather use a system composed of various one-class classifiers, which is more relevant in the case where the amount of data is too small for some modes. Indeed, in real applications, subsets of modes are generally very unbalanced, with many more data points concerning normal modes rather than faulty ones [17].

The role of the whole classification system is to detect a transition from a normal state to a fault state within the predictions. Compared to other approaches for RUL estimation, the proposed CPS is a process that enables one to estimate the RUL without the need of thresholds. Moreover, thresholding is generally applied to one-dimensional degradation signals, while the proposed CPS can be applied to a multi-dimensional one. In the experimental tests, we study the influence of the amount of prior information on RUL estimates, and demonstrate that the proposed approach is well suited when priors are limited.

III. Temporal predictions with an evolving neuro-fuzzy system

A. Objectives

The aim of this part of the CPS strategy is to forecast observations in time. Obviously, this step of prognostics is critical, and must be dealt with in an appropriate manner to provide accurate predictions, and thereby better RUL estimates. Also, predictions must be sufficiently long to ensure usefulness of the full prognostics process. This section describes the approach used to perform long term multi-step ahead predictions.

Assuming that data are defined in a multidimensional space, i.e. $\mathbf{X}_k = [\mathbf{X}_k^1 \quad \mathbf{X}_k^2 \quad \dots \quad \mathbf{X}_k^F]$, the aim of the prediction module is to forecast in time the evolution of the data values, specifically

$$\mathbf{X}_{k+1 \to k+H} = [\mathbf{X}_{k+h}^1 \ \mathbf{X}_{k+h}^2 \ \dots \ \mathbf{X}_{k+h}^F] \tag{1}$$

where $h = [1, H]$. For each feature $i \in 1 \dots F$, the multi-step ahead prediction problem consists of estimating future values of the time series $\hat{\mathbf{X}}_{k+1 \to k+H}^i = \left[\hat{x}_{k+1}^i \ , \ \hat{x}_{k+2}^i \ , \ \hat{x}_{k+3}^i \ , \ \dots \ , \ \hat{x}_{k+H}^i\right]$. This approximation can be expressed as

$$\hat{\mathbf{X}}_{k+1 \to k+H}^i = \widehat{msp}(\mathbf{SX}_k^i) \tag{2}$$

where, msp is the multi-step ahead prediction model, and $\mathbf{SX}_k^i \in \mathbf{X}_k^i$ is known as the set of regressors (for example $\mathbf{SX}_k^i = [x_k^i \ , \ x_{k-1}^i \ , \ x_{k-2}^i]$).
Many approaches exist in literature to build each one of the prediction systems (for each dimension) [18]. According to previous works [19], recent papers focus on the interest of using hybrid systems for prediction. More precisely, first order Takagi-Sugeno (TS) fuzzy models have shown improved performance over conventional approaches [20]–[27]. In this paper, the evolving extended Takagi Sugeno system (exTS) introduced in [15] is considered.

B. First order Takagi-Sugeno systems

A first order TS model aims at approximating an input-output function. It can be seen as a multi-model structure consisting of linear models that are not necessarily statistically independent [15]. 1) The input space is fuzzily partitioned, 2) a fuzzy rule is assigned to each region of the input space and provides a local linear approximation of the output, and 3) the final output is a combination of the whole set of rules.

A TS model is depicted in Fig. 4 with two inputs variables, two membership functions (antecedent fuzzy sets) for each of them, and the output of the TS model is a linear combination of two fuzzy rules. The rules perform a linear combination of inputs, specifically

$$\begin{aligned} &R_i : \text{If } x_1 \ is \ A_i^1 \ , \dots \text{and } x_n \ is \ A_i^n, \\ &\text{then } y_i = a_{i0} + a_{i1}x_1 + \dots + a_{in}x_n. \end{aligned} \tag{3}$$

R_i is the i^{th} fuzzy rule, N is the number of rules, $\mathbf{X}_n = [x_1, ..., x_n]^T$ 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, and a_{il} are its parameters, $l = [0, n]$.
Due to their generalization capabilities, Gaussian antecedent fuzzy sets are generally assumed to define the regions of fuzzy rules in which the local linear sub-models are valid.

$$\mu_i^j = exp^{-(4\left\|x - x^{i*}\right\|_j)/(\sigma_i^j)^2} \tag{4}$$

with σ_i^j being the spread of the membership function, and x^{i*} being the 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

$$\tau_i = \mu_i^1(x_1) \times \dots \times \mu_i^n(x_n) \ , \ \lambda_i = \tau_i \Big/ \sum_{v=1}^N \tau_v. \tag{5}$$

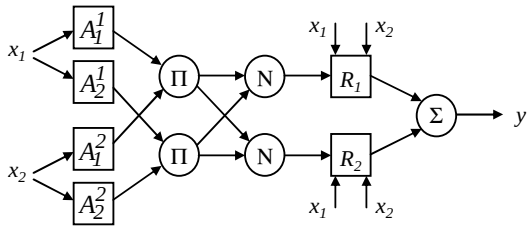


Figure 4. A First-order TS model with 2 inputs.

Let $\pi_i = [a_{i0}, \dots, a_{in}]$ be the parameters vector of the i^{th} sub-model, and $\mathbf{X}_e = [1 \ \mathbf{X}_n^T]^T$ be the expanded data vector. The output is expressed as

$$y = \sum_{i=1}^N \lambda_i y_i = \sum_{i=1}^N \lambda_i \mathbf{X}_e^T \pi_i \tag{6}$$

A TS model has two types of parameters. The non-linear parameters are those of the membership functions represented by Gaussians membership functions which have two parameters: the center, and the spread in (4). These parameters are referred to as premise or antecedent parameters. The linear parameters form the consequent part of each rule such as a_{il} in (3). All these parameters have to be tuned as described later.

C. Learning procedure of exTS

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.

These algorithms cannot be fully detailed in this paper, but are well described in [15], [28].
The exTS clustering phase is performed on the global input-output data space: $\mathbf{Z} = [\mathbf{X}_n^{\mathrm{T}}; \mathbf{Y}_m^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{Z} \in \Re^{n+m}$, where $n+m$ defines the dimension of the input-output data space ($m = 1$ in this paper). Each exTS sub-model operates in a sub-area of $\mathbf{Z}$. This clustering algorithm is based on the calculus of a potential which represents the capability of data to form a cluster (antecedent of a rule). The procedure starts from scratch; and, as more data are available, the model evolves by replacement or rules updates. This approach enables the adjustment of the non-linear antecedent parameters.
The RLS phase aims at updating the consequent parameters. At any learning step k, (6) can be expressed as

$$\hat{y}_{k+1} = \sum_{i=1}^N \lambda_i y_i = \sum_{i=1}^N \lambda_i \mathbf{X}_e^T \pi_i = \psi_k^T \hat{\theta}_k \tag{7}$$

where $\psi_k^T = [\lambda_1 x_1^T, ..., \lambda_n x_n^T]_k^T$ is the vector of the inputs weighted by normalized firing (λ) of the rules (updated thanks to the clustering phase). $\hat{\theta}_k = [\hat{\pi}_1^T, ..., \hat{\pi}_N^T]_k^T$ is an estimation of the linear parameters of the sub-models obtained by applying the RLS procedure

$$\hat{\theta}_k = \hat{\theta}_{k-1} + C_k \psi_k (y_k - \psi_k^T \hat{\theta}_{k-1}) \ ; \ k = 2, 3, ... \tag{8a}$$

$$C_k = C_{k-1} - \left[C_{k-1} \psi_k \psi_k^T C_{k-1}\right] \Big/ \left[1 + \psi_k^T C_{k-1} \psi_k\right] \tag{8b}$$

with C_k the $R(n+1) \times R(n+1)$ covariance matrix of parameters errors. Initial conditions are given by $\theta_1 = 0$,

$C_1 = \Omega I$ where Ω is a large positive number [15], [28].

The main advantage of the exTS results from the clustering phase for which no assumption is required about the structure (number of clusters and parameters initialization). Indeed, an exTS is able to update the parameters without the intervention of an expert. Moreover, it has a flexible structure that evolves as data are gradually collected, which enables one to form new rules or modifying existing ones. This characteristic is useful to cope with non-stationary signals.

D. Multi-step ahead predictions with the exTS

When using connexionist systems (such as exTS), the multi-step ahead prediction model msp can be obtained in different manners. [19] provides an overview of those approaches, and discusses their respective performances. According to this work, the approach they named the Iterative approach appears to be the most common one, and the simplest to implement. Also, this approach offers a compromise between accuracy and complexity. Last but not least, the Iterative approach is the only one to be able to predict at any horizon of prediction, whereas, in other approaches, the end-user has to set in advance the final horizon of prediction, which can be difficult because the time of failure is unknown. Thus, in this paper, multi-step ahead predictions are performed thanks to an exTS-based Iterative model that can be explained as follows.
Multi-step predictions are provided by using a single tool (exTS) that is tuned to perform a one-step ahead prediction $\hat{x}_{k+1}$. This estimated value is used as one of the regressors of the model to estimate the subsequent regressors, and the operation is repeated until the estimation of $\hat{x}_{k+H}$. Formally,

$$\hat{x}_{k+h} = \begin{cases} \text{if } h = 1, f^1\left(x_k, \ldots, x_{k+1-p}, [\theta^1]\right) \\ \text{elseif } h \in \{2, \ldots, p\}, \\ f^1\left(\hat{x}_{k+h-1}, \ldots, \hat{x}_{k+1}, x_k, \ldots, x_{k+h-p}, [\theta^1]\right) \\ \text{elseif } h \in \{p+1, \ldots, H\}, \\ f^1\left(\hat{x}_{k+h-1}, \ldots, \hat{x}_{k+h-p}, [\theta^1]\right) \end{cases} \tag{9}$$

where $\left\{f^1, [\theta^1]\right\}$ is the one-step ahead exTS-based prediction model with its parameters set calculated during the learning phase, and p is the number of regressors used, i.e. the number of past discrete values used for prediction. This type of architecture enables performing multi-step ahead predictions without building various predictors (thereby with a single learning phase). Note that, from the time $h > p$, predictions are made only on evaluated data, and not on observed data.
Fig. 5 shows the evolution of a performance index of an engine, and the prediction that can be obtained thanks to the exTS-based Iterative approach. Note that, in this figure, all predictions (from 51 to 231) where made at time $k = 50$.

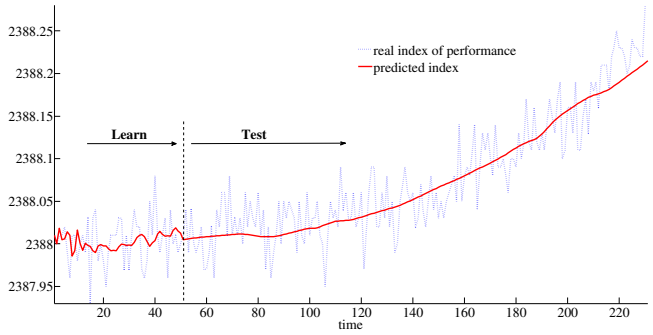


Figure 5. Example of multi-step ahead predictions of a performance index of an engine with an exTS-based Iterative model.

IV. Evidential Hidden Markov Model for Classification of Temporal Predictions

A. Objectives

The aim of this part of the CPS strategy is to classify the predictions made by the exTS into meaningful states. Because the problem deals with time series modeling, Hidden Markov Models (HMM) [29] appear to be a good option. In this paper, developments are focused on an extension of HMM to manage uncertainties based on Dempster-Shafers theory of belief functions [16], [31] described in [10], and called evidential HMM (EvHMM). EvHMM were first proposed to cope with statistical modeling of time series using sparse data. This condition is particularly the case in industrial applications where the cost of data acquisition and interpretation is high. Besides, because the exTS-based algorithm for prediction can be trained using few data, the classifier should also have the same capability. It also strengthens the use of belief functions for the classification step (CPS).
EvHMM are used for classification in both normal and faulty classes. One EvHMM is built using data from the normal class, and another one from data in the faulty class. For each EvHMM, one needs to set the number of states (which represent latent variables), and the set of components in each state. The set of states at time k is denoted by $\Omega_k = \{\omega_1, \ldots, \omega_K\}$, and the basic belief assignment (BBA) m^{Ω_k} is defined on the powerset 2^{Ω_k} to represent imprecision and uncertainty about the possible states at a given time k; specifically,

$$\begin{aligned} &m^{\Omega_k} : 2^{\Omega_k} \to [0,1], A \to m^{\Omega_k}(A) \\ &\textstyle\sum_{A \subseteq \Omega_k} m^{\Omega_k}(A) = 1. \end{aligned} \tag{10}$$

The estimation of BBAs from data is explained below.

B. Classification in EvHMM

The exTS estimates the future values taken by each feature, i.e. $\hat{\mathbf{X}}^i_{k+1 \to k+H}, i = 1 \ldots F$. Predictions are then gathered in the vector $\mathbf{X}_{k+1 \to k+H} = [\mathbf{X}^1_{k+h} \quad \mathbf{X}^2_{k+h} \quad \ldots \quad \mathbf{X}^F_{k+h}]$, which becomes the input of the EvHMM classifier. Given a training dataset, a set of predictions can be generated and labeled as normal class ($\mathbf{X}^{Norm}_{k+1 \to k+H}$), or faulty class ($\mathbf{X}^{Fault}_{k+1 \to k+H}$), from which two respective classifiers λ_{Norm}, and λ_{Fault} can be built. Note that sequences of data $\mathbf{X}^{Norm}_{k+1 \to k+H}$ or $\mathbf{X}^{Fault}_{k+1 \to k+H}$ are generally called observations in the HMM community, and denoted O_k at time k, or $\mathbf{O}_{1:H}$ for the whole sequence, where H represents the number of observations (for a given sequence).
The parameters $\lambda_r, r \in \{Norm, Fault\}$ of a EvHMM are composed as follows.

- The BBA representing transitions between states at two consecutive time instants are denoted as $m_a^{\Omega_k}(\cdot|S_i)$. It is a conditional BBA defined on Ω_k conditionally to subsets $S_i \subseteq \Omega_{k-1}$.
- The BBA on states given observations is $m_b^{\Omega_k}(S_i|O_k)$.

Given EvHMMs λ_{Norm}, and λ_{Fault}, the goal of the classification process (Algorithm 1) is to choose the EvHMM that best fits observations. The classification criterion is given by

$$\mathcal{L}_e(\lambda_r) = \frac{1}{H}\sum_{k=1}^{H} \log pl_\alpha^{\Omega_k}(\Omega_k|\lambda_r) \tag{11}$$

with

$$\lambda^* = \underset{r}{\text{argmax}}\ \mathcal{L}_e(\lambda_r) \tag{12}$$

The prediction of a subset S_j is computed using the law of total plausibility, and combined with observations to update belief on states.

$$q_\alpha^{\Omega_k}(S_j) = \sum_{S_i \subseteq \Omega_{k-1}} m_\alpha^{\Omega_{k-1}}(S_i) \cdot q_a^{\Omega_k|\Omega_{k-1}}(S_j|S_i) \cdot q_b^{\Omega_k}(S_j|O_k) \tag{13}$$

In (13), q is the *communality function* obtained from a BBA using

$$q^{\Omega_k}(B) = \sum_{C \supseteq B} m^{\Omega_k}(C)\ . \tag{14}$$

Commonalities are in *one-to-one correspondance* with BBA [16], and make the combination rules easier to compute. In the same way, a plausibility is given by

$$pl^{\Omega_k}(B) = \sum_{C \cap B \neq \emptyset} m^{\Omega_k}(C)\ . \tag{15}$$

In (13), the BBA at $k = 1$ can be defined as $m_\alpha^{\Omega_1}(\Omega_1) = 1$, reflecting full ignorance about the first state. Moreover, commonalities q_a conditional to subsets with cardinality greater than 1 are computed using the disjunctive rule of combination [10], reducing the number of parameters to be estimated. Besides, as in probabilistic HMM, the conflict resulting from the conjunctive combination between observations and prediction has to be canceled out by normalisation at each iteration of the forward propagation [10]. The normalisation process consists in redistributing uniformly $1 - \sum_j \alpha_k(j)$ to each state at k. Similarly, as in standard HMM, backward and smoothing variables can be defined [10].

Algorithm 1 EvHMM Classification

Require: model λ_r with q_b at each k and q_a {Belief on transitions and on states given observations. }
Ensure: Evidential likelihood $\mathcal{L}_e$
Ensure: Evidential filtered estimate α
1: **for all** instants $k = 1$ to H **do**
2: α = Forward propagation {(13)}
3: α_* = Normalise α
4: **end for**
5: Compute $\mathcal{L}_e$ {eq. 11 and 12}

C. Learning procedure of EvHMM

Training the EvHMM consists of estimating q_a (transitions), as well as the parameters of the models that generate belief functions conditional to observations O_k. As underlined in [10], applying an iterative procedure such as Expectation-Maximization often used in HMM is not relevant because successive forward and backward propagations imply conjunctive combinations, which gradually generates specific BBAs focused on singletons, therefore loosing the interest of using belief functions. We rather propose two separate processes: one for observation models (called the regrouping components with geometric interaction algorithm (RCGI)), and one for transitions (called the iterative transition estimation algorithm (ITS)), described below.

1) RCGI, and Observations models training: The proposed training process of observation models is decomposed in two steps:

- clustering data into M clusters (called components), and
- regrouping the M components into N states.

The main features of this algorithm (Alg. 3) are depicted in Fig. 6.

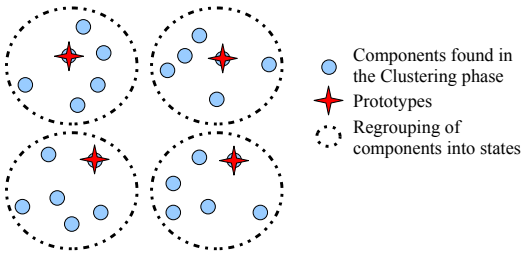


Figure 6. RCGI steps with $N = 4$, and $M = 6$.

Step 1 - Clustering. The first step consists of paving the feature space by first finding $M \times N$ components in the data (see filled circles in Fig. 6):

$$\Lambda_0 \leftarrow \text{ find } M \times N \text{ components using a clusterer.} \tag{16}$$

This phase can be performed by any clustering approach. In this paper, we considered that only a small amount of data are available. Therefore, we use an adaptive method that can find an optimal number of components according to the data distribution [30].

Step 2 - Regrouping. In probabilistic HMM, a set of states N and a number of components for each state M has to be chosen. Then a Baum-Welch algorithm finds the parameters of each component in each state [29]. The regrouping of components into states is done automatically by maximizing likelihood. In [10], we adapted this algorithm for EvHMM as follows. Let $M \times N$ components found by the Clustering phase (16). We then need to find N states, each one composed of M components. For that, we developed the RCGI procedure described in Alg. 3. RCGI assumes that EvHMM is used for time series modeling, and therefore the relative position of components is important.

Given Λ_0, the set of $M \times N$ components provided by the clustering phase, the N sets of states are denoted $\Lambda_i, i = 1 \ldots N$, such that $\cap_i \Lambda_i = \emptyset$ and $\cup_i \Lambda_i = \Lambda_0$. The cardinality $|\Lambda_i|$ can

be different for each state, but for the sake of simplicity we consider here the same cardinality. RCGI thus fills an $M \times N$ association matrix $\mathbb{A}$ with

$$\mathbb{A}(i,j) = \begin{cases} 1 & \text{if component } j \text{ is assigned to state } i \\ 0 & \text{otherwise.} \end{cases} \quad (17)$$

a) Initialisation: RCGI first requires one component for each state, which are determined in four steps (Alg. 2). First, we compute pairwise distances (Euclidean) between all components. The result is an $N \times M$ matrix $[\mathcal{D}(i,j)]$ where elements are the distances between components i and j:

$$\mathcal{D}(i,j) \leftarrow \text{Distance between comp. } i \text{ and } j. \quad (18)$$

Then, we find the farthest component from all others, as

$$c_1 = \underset{j}{\text{argmax}} \sum_i \mathcal{D}(i,j). \quad (19)$$

In the third step, the farthest component from c_1 is estimated as

$$c_2 = \underset{j, j \neq c_1}{\text{argmax}} \ \mathcal{D}(\text{comp. } c_1, j). \quad (20)$$

At this stage, we have two states, each with one component. To find the first component for the remaining $N-2$ states, we consider the distance between c_1 and c_2, and divide it into $N-1$ segments of equal-length. Denote $\hat{c}_i$ as the estimated component for state $i = 3 \dots N$. Therefore, c_i is given by the closest component to $\hat{c}_i$:

$$c_i = \underset{j, j \neq c_l, l > i}{\text{argmin}} \ \mathcal{D}(\text{comp. } \hat{c}_i, j), i = 3 \dots N. \quad (21)$$

In Fig. 6, the result of the initialization step is represented by the stars on the chosen components.

Example 1: Consider the data in Fig. 7. The figure represents a set of $N = 4$ states, each one being corrupted by $M = 3$ components' additive noise (different for each state). Ideally, there are 12 components. Assume that the components are characterized by the center means μ = [4.2 3.2 2.2 1.2 1.6 2.7 0.7 3.4 3.7 0.8 3.6 2.3]. Thus, criterion (18) gives the values $\mathcal{D}(i,j)$ = [51.68 22.08 16.48 34.88 24.64 16.28 53.08 26.08 33.88 48.96 31.04 15.96]. Therefore, $c_1 = 7$ ($\mu_7 = 0.7$), and $c_2 = 1$ ($\mu_1 = 4.2$). Then the segment length is $(4.2 - 0.7)/3 = 1.1667$; thus, $\hat{c}_3 = 3.033$, and $\hat{c}_4 = 1.8667$, leading to $c_3 = 2$ (with $\mu_2 = 3.2$), and $c_4 = 5$ (with $\mu_5 = 1.6$). Finally, the first components of each state are $7, 1, 2$, and 5.

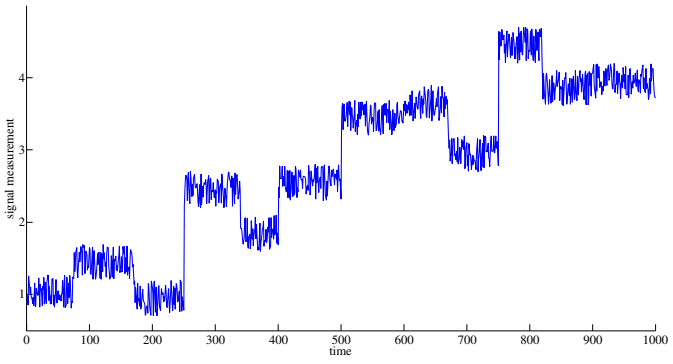


Figure 7. Signal to be segmented.

b) Association: A component j in Ω_0 is associated to a state i if the latter is the closest state to j:

$$\begin{aligned} j^* &= \underset{j}{\text{argmin}} \ \mathcal{D}'(\text{component } j, \text{state } i) \\ \mathbb{A}(i, j^*) &= 1. \end{aligned} \quad (22)$$

A representation of this assignment is depicted by dotted circles in Fig. 6.

A state can be composed of several components; therefore it is necessary to adapt the distance measure $\mathcal{D}'$ to compare a single component (j) to a set of components (composing state i). For distribution-based clusterers (such as Gaussian mixtures models as considered in experiments), we use the Kullback-Leibler (KL) divergence between both the distribution $p_j \equiv p(y|j)$ of data points y in component j and the distribution $p_i \equiv p(y|i)$ of data points y in the mixture of components composing state i:

$$\mathcal{D}'(j,i) = \text{KL}\Big(p_i \,||\, p_j \Big) \quad (23)$$

For mixtures of continuous densities, the KL divergence does not have a closed-form, but can be estimated by Monte-Carlo sampling. Samples are thus drawn from the mixture associated to p_i; and given a set of i.i.d. sampled points $y_1 \dots y_n \dots y_{N_s}$, we can approximate the KL by its Monte-Carlo estimate as

$$\hat{\text{KL}} = \frac{1}{N_s} \sum_n \log\left(\frac{p(y_n|i)}{p(y_n|j)}\right) \xrightarrow[N_s \to \infty]{} \text{KL}(p_i || p_j). \quad (24)$$

As for tests, we used $N_s = 1e^5$ samples.

Algorithm 2 ONE_STATE_RCGI

Require: Set of components Ω_0
Require: Number of states N {assume the same number of components for each state}
Ensure: Find N prototypes: $\mathbb{A}(j) = 1, j = 1 \dots |\Omega_0|$ if component j is a prototype
1: Compute distances between all components ($[\mathcal{D}(i,j)]$)
2: Find the farthest component: $c_1 \Rightarrow A(C_1) = 1$
3: Find the farthest component from c_1: $C_2 \Rightarrow A(c_2) = 1$
4: Find $N-2$ components between c_1 and c_2 as described in the text: assign $A(c_i) = 1, i = 3 \dots N$

Example 2: RCGI is applied on the data described in the previous example. It finds a set of $N = 4$ states, with $M = 3$ components each. The resulting association is [7 10 4] *for state* 1, [1 9 11] *for state* 2, [2 8 6] *for state* 3, *and* [5 3 12] *for state* 4. *The obtained segmentation is given in Fig. 8, in which the states were renumbered* (1, 2, 3, 4) *according to the order of appearance.*

2) ITS, transition estimation: After RCGI is performed, transitions are estimated as

$$m_{\hat{a}_0}^{\Omega_k \times \Omega_{k+1}} \propto \sum_{k=1}^{H-1} \left(m_b^{\Omega_k \uparrow \Omega_k \times \Omega_{k+1}} \textcircled{\cap} \, m_b^{\Omega_{k+1} \uparrow \Omega_k \times \Omega_{k+1}} \right) \quad (25)$$

up to a constant $1/(H-1)$, and where $m_b^{\Omega_k \uparrow \Omega_k \times \Omega_{k+1}}$ is the *vacuous extension* [31] of the belief mass $m_b^{\Omega_k}(\cdot|O_k)$

Algorithm 3 RCGI

Require: Set of components Ω_0 {characterized by some parameters}
Require: Number of states N {$M = |\Omega_0|/N$ since we assume the same number of components for each state}
Ensure: Association matrix $\mathbb{A}(i,j) = 1$ if component j is assigned to state i
1: $\mathbb{A}(:,1) \leftarrow$ ONE_STATE_RCGI(Ω_0, N) (Alg. 2) {Initialisation, then remove the prototypes from Ω_0.}
2: **for** states $i = 1$ To N **do**
3: **while** $\sum_j \mathbb{A}(i,j) < M$ **do**
4: **for all** remaining components j in Ω_0 **do**
5: Compute the distance $\mathcal{D}^{'}(i,j)$ between state i and component j {See comments in text}
6: **end for**
7: $\mathbb{A}(i,j^*) = 1$ with $j^* = \text{argmin}_j \mathcal{D}^{'}(i,j)$ {Assign a component to state i}
8: $\Omega_0 \leftarrow \Omega_0 - \{j^*\}$ {Update remaining components}
9: **end while**
10: **end for**

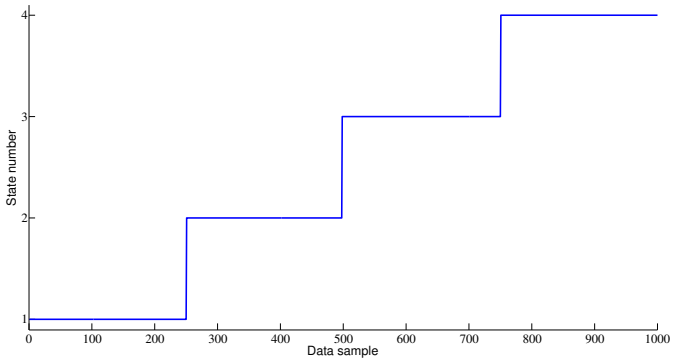


Figure 8. Segmentation after RCGI.

(provided by observations) on the cartesian product defined by

$$m_b^{\Omega_k \uparrow \Omega_k \times \Omega_{k+1}}(B|O_k) = m_b^{\Omega_k}(C|O_k) \text{ if } C \times \Omega_{k+1} = B \quad (26)$$

and 0 otherwise. Equation (25) is a generalization of the HMM transition estimate to belief functions when there is no prior information on transitions.

D. RUL estimation

Following the proposed architecture (Section II), an EvHMM λ_{Fault} is built corresponding to some data related to a faulty state ω_{Fault}, and one EvHMM λ_{Norm} for the normal state ω_{Norm}. Given a new experiment where the RUL has to be estimated, we first run the exTS algorithm to estimate the predictions at $t+h, h = 1 \dots H$. Inference procedures of both EvHMM models are then performed, and provide the likelihood of each model at each time-step of the predictions. The RUL is then defined as the time-instant where the likelihood of λ_{Fault} (faulty state model) becomes higher than the likelihood of λ_{Norm} (normal state model).

V. APPLICATION TO THE TURBOFAN DATASET

The aim of this part is to illustrate the capability of the proposed architecture to provide reliable estimates of the RUL.

A. Data sets

We considered the first CMAPPS dataset introduced during the first Int. Conf. on Prognostics and Health Management [32]. The dataset is a multiple multivariate time series with sensor noise. Each time series was from a different engine of the same fleet, and each engine started with different degrees of initial wear and manufacturing variation unknown to the user but considered normal. The engine was operating normally at the start, and developed a fault at some point. The fault grew in magnitude until system failure. The variability of the true RULs was studied in [33].

B. Feature selection

In [9], we proposed a feature selection approach based on the Kullback-Leibler divergence to select 8 complementary features among the 26 features found in the dataset (corresponding to columns $7, 8, 9, 11, 13, 15, 17, 18$). These 8 features were then used to train the prediction system. Among these 8 features, only 4 were kept by maximizing

$$\underset{\substack{\text{over all training data} \\ t \in \text{current training data}}}{\text{median}} U\left(\frac{\hat{X}_t(j)}{X_t(j)} > 0.95\right), j = 1 \dots 8 \quad (27)$$

where $U(x) = 1$ if x is true, 0 otherwise. This criterion enforces the predictions to be statistically close or above the real values in the training dataset.

C. Prediction and classification settings

1) Temporal predictions settings: As for the prediction step, each feature was estimated with an exTS-based iterative model for multi-step ahead prediction (as explained in Section III-D). Table I recalls the set of input variables used for that purpose, which can be automatically estimated, for example using a parsimony criteria [22].

Table I
SETS OF REGRESSORS FOR FEATURES PREDICTIONS

Feature	Inputs
1	x1(k), x1(k-1), x1(k-2)
2	x2(k), x2(k-1), x2(k-2)
3	x3(k), x3(k-1), x3(k-2)
4	x4(k), x4(k-1)
5	x5(k)
6	x6(k)
7	x7(k), x7(k-1)
8	x8(k), x8(k-1)

2) Classification settings: One EvHMM classifier was trained for the faulty state, and one for the normal state. Data concerning the faulty state correspond to the last 12 data of each time series (the remainder corresponding to the normal state). In this paper, only the data located after the transition from state 3 to 4 (last 12 data) were considered to train the EvHMM classifier. This figure shows that the RULs are spread

on a large range (from 50 to 350 time units).
The number of Gaussian components M was set automatically by an Expectation-Maximization (EM) algorithm using a minimum description length criterion (MDL) as proposed in [30]. The number of states N was set to the first prime number such as the modulus of M over the latter equals 0. The EM algorithm which estimates the parameters of the distributions requires initial values. We thus proceed as follows.

- Select random initial values of the parameters.
- Estimate the parameters (wait for convergence).
- Compute the model likelihood given the training data.

This process was repeated 10 times for both models, and the one with the highest likelihood was selected. Practically, the best models were obtained by considering the likelihood estimated by the Viterbi-like decoder proposed in [10].

D. Evaluation process

To improve the analysis of the results, and to get a more objective discussion on the interest of the proposed approach, the exTS-based Iterative model was trained and run with varying critical times, and different amounts of training data.

- Critical time (beginning time instant of the prediction): $k_0 = [50\ \ 90\ \ 130\ \ 150]$ time units.
- Number of training data: $NL = [2\ \ 5\ \ 10\ \ 20\ \ 30]$.

This condition enables us to discuss the influence, on the one hand, of the starting point of predictions, and, on the other hand, of the amount of available data to fit both the predictions and the classification models.
Still to remain statistically independent on the parameterization, a leave-one-out evaluation was performed to train the classifier before assessing the RUL estimates: 14 predicted time series were used to train the classifier ($NC = 14$ in section II-C), and 1 for testing; and this process was repeated 15 times, and the RULs averaged.
Fig. 9a depicts the actual RULs to be estimated on the 15 experiments as a function of the critical instant of prediction. One can note that the horizon length considered in the tests are challenging because the greatest one is 207 time-units (with $k_0 = 50$), while the shortest one is still 24 time-units (with $k_0 = 150$).
To assess the predictions, define the prediction error at a given time k by

$$E(k) = \text{true RUL} - \text{predicted RUL}. \tag{28}$$

We can then report prediction errors by histograms. To assess more precisely the errors made by the proposed system, we considered false negative and false positive rates [34], [35].

- False Negative (FN) cases correspond to late predictions such as $E(k) < -k_{FN}$ where k_{FN} is a user-defined FN threshold

$$FN(k) = \begin{cases} 1 & \text{if } E(k) < -k_{FN} \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

- False Positive (FP) cases correspond to early predictions such as $E(k) > k_{FP}$ where k_{FP} is a user-defined FP threshold

$$FP(k) = \begin{cases} 1 & \text{if } E(k) > k_{FP} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

The meaning of thresholds is represented in Fig. 10 where $\mathcal{I} = [-k_{FN}, k_{FP}]$.

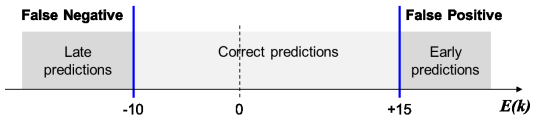


Figure 10. Metric of performance assessment, here $\mathcal{I} = [-10, +15]$.

E. Results

An example of results is given in Fig. 9.b that depicts the RUL estimates obtained for experiment #1 according to the critical instant of prediction k_0, and the size of the prediction learning set NL. As expected, the worst results are obtained with $NL = 1$. Also, as NL increases, the results' accuracy is enhanced, and RUL estimates are quite close together. This result serves to strengthen the interest of the proposed approach because few learning data are required to obtain good results. However, one should consider results on the whole set of experiments to avoid concluding falsely from a singular case.
Consider Fig. 11 that shows the distributions of the error (28) for all experiments. One can point out that, even for a small number of training data (less than 10), the proposed approach leads to accurate RUL estimates. For example, for the largest horizon of prediction, i.e. the most difficult case with $k_0 = 50$, less than 5 training data can be sufficient to estimate the RUL with a spread of the error less than 10 time units. A stable result (for any k_0) is obtained with NL $= 20$ training data.
As expected, the best RUL estimates are obtained for the largest number of training data (here NL $= 30$), and for the smallest horizon ($k_0 = 150$), even though competitive results are obtained with NL $= 20$, and $k_0 = [50\ \ 130]$.
The small amount of data can provide unexpected results such as those obtained with $k_0 = 50$, and NL $= 10$, where the system made more errors than for NL $= 5$ or NL $= 2$. This behavior is explained by the fact that the number of data is too small to pave the feature space properly in the clustering phase of both exTS and EvHMM. As expected, this effect decreases as the number of training data increases.

Table II presents the accuracy of the RUL estimates for different intervals ($\mathcal{I} = [-10, 10];\ [-10, 20];\ [-20, 10];\ [-20, 20]$) with report to the critical time $k_0 = 50, 90, 130, 150$. According to these tables, the proposed architecture performs well on this dataset with accurate RUL estimates. Indeed, whatever the interval $\mathcal{I}$, at least 74.4% of RUL estimates appear to be correct predictions (as defined in Fig. 10). Regarding the interval size, the system demonstrates robust results for $[-20\ \ 10]$, and $[-20\ \ 20]$, where accuracies of predictions are very high, and similar whatever k_0 (from 85.6% to 94.4%). For small sizes such as $[-10\ \ 10]$ (where predictions have to be close to the ground truth), the proposed system reaches high accuracy, from 74.4% to 82.2% according to the value of k_0.

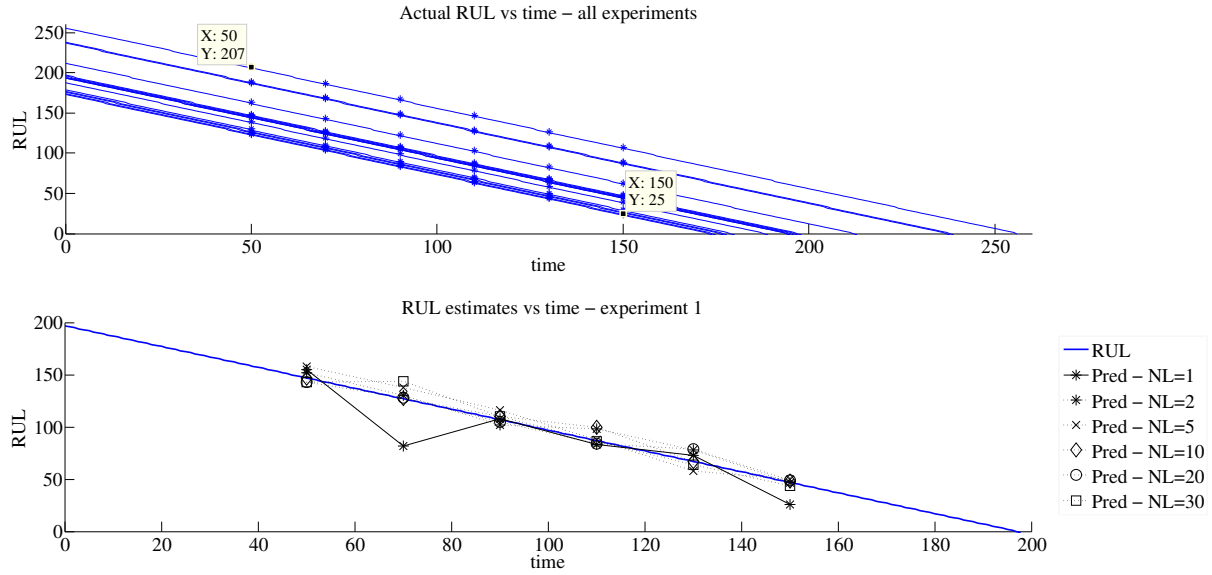


Figure 9. RUL of experiments: a) top, actual RUL accordingly to the instant of prediction; b) bottom, RUL estimates for experiment #1.

Table II
RUL ESTIMATES ACCURACY FOR CRITICAL TIMES $k_0 = 50, 90, 130,$ AND 150 (FROM SHORT TO LONG-TERM PREDICTIONS)

Interval $\mathcal{I}$	$\mathcal{A}_{RUL}^{k_0=50}$	$\mathcal{A}_{RUL}^{k_0=90}$	$\mathcal{A}_{RUL}^{k_0=130}$	$\mathcal{A}_{RUL}^{k_0=150}$
$[-10\ \ 10]$	74.4	75.6	81.1	82.2
$[-10\ \ 20]$	80.0	78.9	87.8	88.9
$[-20\ \ 10]$	86.7	86.7	86.7	87.8
$[-20\ \ 20]$	92.2	92.2	92.2	94.4

VI. CONCLUSION

An original, efficient architecture is proposed for health state assessment and prognostics. Leaving aside the features extraction and selection step, this architecture is composed of two modules: an evolving neuro-fuzzy system (exTS) for reliable multi-step ahead predictions, and an evidence theoretic Markovian classifier (EvHMM) for classification. The RUL is estimated by a classification of predictions strategy: predictions are first computed by exTS, and the instant of transition from the normal state to the faulty one is detected by the EvHMM to finally providing a RUL estimate.

The efficiency of the proposed architecture is demonstrated on NASA's turbofan dataset. The impact of the size of the training dataset is discussed, as well as the stability of RUL estimates performance according to the actual remaining time to failure (instant of prediction). The overall accuracy of RUL estimates is between 74.4% and 92.2% for very long-term prediction ($130, 150$ time units), and between 82.2% and 94.4% for short-term predictions ($50, 90$ time units). Also, the approach appears to be suitable even if few learning data are available.

ACKNOWLEDGEMENT

This work was carried out within the Laboratory of Excellence ACTION funded by the French Government through the program "Investments for the future" managed by the National Agency for Research (ANR-11-LABX-01-01). We thank the anonymous referees for their helpful comments.

REFERENCES

[1] C. Byington, M. Roemer, G. Kacprzynski, and T. Galie, "Prognostic enhancements to diagnostic systems for improved condition-based maintenance," *Proc. IEEE Int. Conf. on Aerospace*, vol. 6, 2002, pp. 2815–2824.

[2] A. Heng, S. Zhang, A. Tan, and J. Matwew, "Rotating machinery prognostic: State of the art, challenges and opportunities," *Mechanical Systems and Signal Processing*, vol. 23, pp. 724–739, 2009.

[3] A. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, pp. 1483–1510, 2006.

[4] G. Vachtsevanos, F. L. Lewis, M. Roeme, A. Hess, and B. Wug, *Intelligent Fault Diagnostic and Prognosis for Engineering Systems*. John Wiley & Sons, 2006.

[5] A. Usynin, "A generic prognostic framework for remaining useful life prediction of complex engineering systems," Ph.D. dissertation, The University of Tennessee, Knoxville, 2007.

[6] O. E. Dragomir, R. Gouriveau, N. Zerhouni, and R. Dragomir, "Framework for a distributed and hybrid prognostic system," *4th IFAC Conf. on Management and Control of Production and Logistics*, 2007.

[7] E. Ramasso, "Contribution of belief functions to Hidden Markov Models," *IEEE Workshop on Machine Learning and Signal Processing*, Grenoble, France, 2009, pp. 1–6.

[8] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of observations and states in time-series based on belief functions," *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 43, pp. 37–50, 2013.

[9] E. Ramasso and R. Gouriveau, "Prognostics in switching systems: Evidential markovian classification of real-time neuro-fuzzy predictions," *IEEE Int. Conf. on Prognostics and System Health Management*, Macau, China, 2010, pp. 1–10.

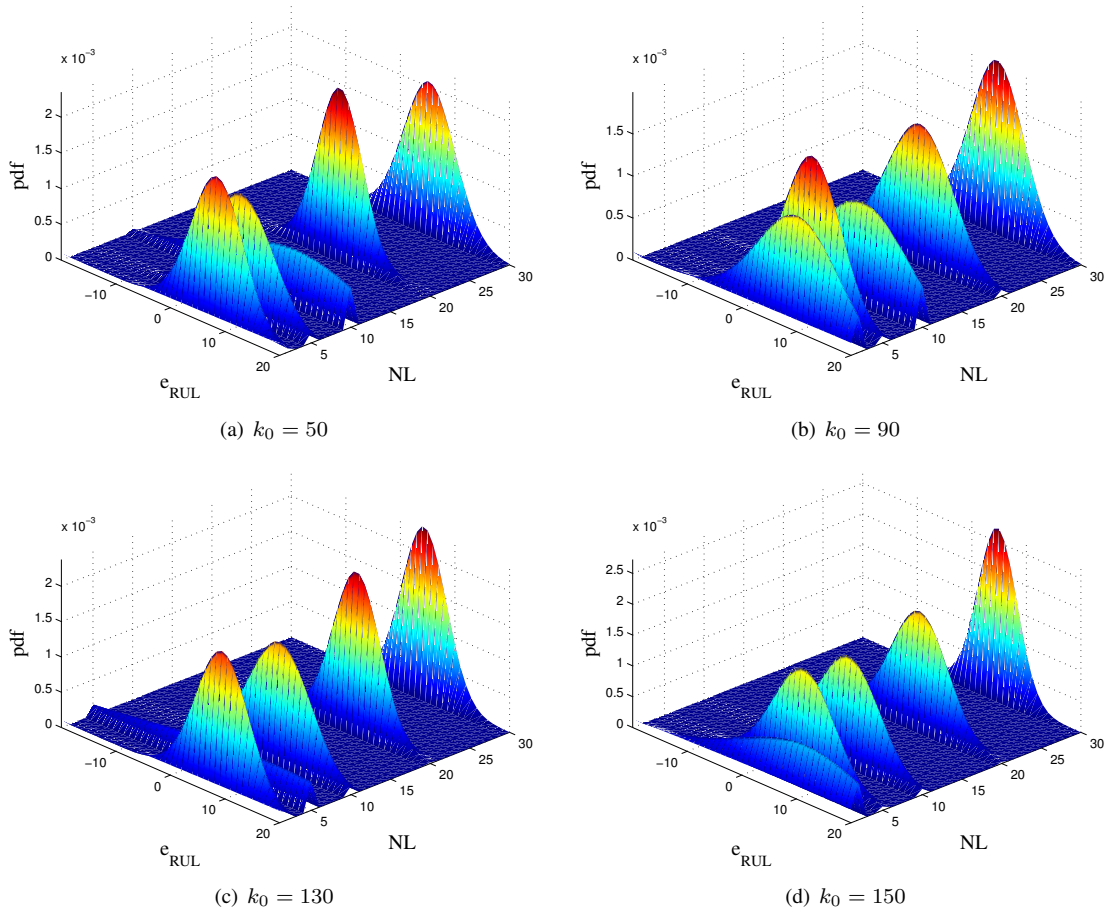
[10] L. Serir, E. Ramasso, and N. Zerhouni, "Time-sliced temporal evidential networks: the case of evidential HMM with application to dynamical system analysis," *IEEE International Conference on Prognostics and Health Management*, Denver, CO, USA, June 2011.

[11] ISO, *Condition monitoring and diagnostics of machines, prognostics, Part1: General guidelines*, International Standard, ISO13381-1, 2004.

[12] M. Lebold and M. Thurston, "Open standards for condition-based maintenance and prognostics systems," *Proc. of 5th Annual Maintenance and Reliability Conference*, 2001.

[13] K. Javed, R. Gouriveau, R. Zemouri, and N. Zerhouni, "Improving data-driven prognostics by assessing predictability of features," *Annual Conference of the PHM Society*, Montreal, Canada, September 2011.

[14] L. Serir, E. Ramasso, P. Nectoux, O. Bauer, and N. Zerhouni, "Evidential evolving Gustafsson-Kessel algorithm (E2GK) and its application to PRONOSTIA's data streams partitioning," *IEEE Int. Conf. on Decision and Control*, December 2011.



(a) $k_0 = 50$ (b) $k_0 = 90$

(c) $k_0 = 130$ (d) $k_0 = 150$

Figure 11. Distribution of errors with report to the size of the training dataset, for different horizons of prediction $k_0 = [50 \;\; 90 \;\; 130 \;\; 150]$.

[15] P. Angelov and D. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Trans. Syst. Man Cybern. - Part B: Cybernetics*, vol. 34, pp. 484–498, 2004.

[16] P. Smets and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, 1994.

[17] R. Gouriveau, E. Ramasso, and N. Zerhouni, "Strategies to face imbalanced and unlabelled data in phm applications," *Int. Conference on Prognostics and Systems Health Management*. Chemical Engineering Transactions, 2013.

[18] J. D. Gooijer and R. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, vol. 22, pp. 443–473, 2006.

[19] R. Gouriveau and N. Zerhouni, "Connexionist-systems-based long term prediction approaches for prognostics," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 909 – 920, 2012.

[20] Y.-L. Dong, Y.-J. Gu, K. Yang, and W.-K. Zhang, "A combining condition prediction model and its application in power plant," *Int. Conf. on Mach.ine Learning and Cyber.*, vol. 6, 2004, pp. 474–3478.

[21] M. El-Koujok, R. Gouriveau, and N. Zerhouni, "Towards a neuro-fuzzy system for time series forecasting in maintenance applications," *IFAC World Congress, Seoul, Korea*, 2008.

[22] M. El-Koujok, R. Gouriveau, and N. Zerhouni, "Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system," *Microelectronics Reliability*, vol. 51, pp. 310–320, 2011.

[23] V.-T. Tran, B.-S. Yang, and 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*, vol. 36, pp. 378–387, 2009.

[24] W. Wang, M. Golnaraghi, and F. Ismail, "Prognosis of machine health condition using neuro-fuzzy systems," *Mech. Syst. and Sign. Proc.*, 2004.

[25] W.-Q. Wang, F. Ismail, and M.-F. Goldnaraghi, "A neuro-fuzzy approach to gear system monitoring," *IEEE Transaction Fuzzy Systems*, vol. 12, pp. 710–723, 2004.

[26] W.-Q. Wang, "An adaptive predictor for dynamic system forecasting," *Mechanical Systems and Signal Processing*, vol. 21, pp. 809–823, 2007.

[27] R. Yam, P. Tse, L. Li, and P. Tu, "Intelligent predictive decision support system for condition-based maintenance," *International Journal of Advanced Manufacturing Technology*, vol. 17, pp. 383–391, 2001.

[28] P. Angelov and X. Zhou, "Evolving fuzzy systems from data streams in real-time," *Proc. Int. Symp. On Evolving Fuzzy Systems*, 2006, pp. 26–32.

[29] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, pp. 257–285, 1989.

[30] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[31] P. Smets, "Advances in the Dempster-Shafer theory of Evidence - what is Dempster-Shafer's model ?" 1994, pp. 5–34.

[32] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," *IEEE Int. Conf. on Prognostics and Health Management*, 2008.

[33] E. Ramasso, M. Rombaut and N. Zerhouni, " Joint Prediction of Continuous and Discrete States in Time-Series Based on Belief Functions," *IEEE Transactions on Cybernetics*, vol. 43, no. 1, pp. 37–50, 2013.

[34] K. Goebel and P. Bonissone, "Prognostic information fusion for constant load systems," *7th Annual Conference on Information Fusion*, 2003, pp. 1247–1255.

[35] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, and M. Schwabacher, "Metrics for evaluating performance of prognostic techniques," *International Conference on Prognostics and Health Management*, 2008, pp. 1–17.

Dr. Emmanuel Ramasso received both B.Sc. and M.Sc. degrees in Automation Science and Engineering from the University of Savoie in 2004, and earned his Ph.D. from the University of Grenoble in 2007. He pursued with a postdoc at the Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA) in 2008. Since 2009, he has been working as an associate professor at the National Engineering Institute in Mechanics and Microtechnologies (ENSMM) at Besançon (France). His research is carried out at FEMTO-ST institute, and focused on pattern recognition under uncertainties with applications to Prognostics and Structural Health Management.

Dr. Rafael Gouriveau received his engineering degree from National Engineering School of Tarbes (ENIT) in 1999, and his M.Sc. (2000) and his Ph.D. in Industrial Systems in 2003, both from the Toulouse National Polytechnic Institute (INPT). During his PhD, he worked on risk management and dependability analysis. In Sept. 2005, he joined the National Engineering Institute in Mechanics and Microtechnologies of Besançon (ENSMM) as Associate Professor. His main teaching activities are concerned with production, maintenance, manufacturing, and informatics domains. He is currently at the head of the PHM team in the Automatic Control and Micro-Mechatronic Systems department of FEMTO-ST. His research interests concern industrial prognostics systems using connexionist approaches like neuro-fuzzy methods, and the investigation of reliability modeling using possibility theory. He is also the scientific coordinator of PHM research axes at FCLAB (Fuel Cell Lab) Research Federation (CNRS).