E2GKpro: An evidential evolving multi-modeling approach for system behavior prediction with applications

Lisa Serir, Emmanuel Ramasso, Patrick Nectoux, Noureddine Zerhouni

FEMTO-ST Institute, UMR CNRS 6174 - UFC / ENSMM / UTBM, Automatic Control and Micro-Mechatronic Systems Dep., 25000, Besançon, France

Abstract

Nonlinear dynamical systems identification and behavior prediction are difficult problems encountered in many areas of industrial applications such as fault diagnosis and prognosis. In practice, the analytical description of a nonlinear system directly from observed data is a very challenging task because of the the too large number of the related parameters to be estimated. As a solution, multi-modeling approaches have lately been applied and consist in dividing the operating range of the system under study into different operating regions easier to describe by simpler functions to be combined. In order to take into consideration the uncertainty related to the available data as well as the uncertainty resulting from the nonlinearity of the system, evidence theory is of particular interest, because it permits the explicit modeling of doubt and ignorance. In the context of multi-modeling, information of doubt may be exploited to properly segment the data and take into account the uncertainty in the transitions between the operating regions. Recently, the Evidential Evolving Gustafson-Kessel algorithm (E2GK) has been proposed to ensure an online partitioning of the data into clusters that correspond to operating regions. Based on E2GK, a multi-modeling approach called E2GKpro is introduced in this paper, which dynamically performs the estimation of the local models by upgrading and modifying their parameters while data arrive. The proposed algorithm is tested on several datasets and compared to existing approaches. The results show that the use of virtual centroids in E2GKpro account for its robustness to noise and generating less operating regions while ensuring precise predictions.

Keywords: Online evidential clustering, Multi-modeling, Belief functions theory, Behavior modeling, Virtual centroids

1. Introduction

1.1. Nonlinear systems and multi-model approaches

Dealing with nonlinear systems behavior identification and prediction is a widely encountered problem in real world applications in engineering, industry, time series analysis, prediction and fault diagnosis [1]. Modeling their behavior from observed data is a difficult task to perform because the identification of nonlinear systems involves a large number of related parameters to be estimated. Usually, a model consists in a set of functional relationships between the elements of a set of variables. One way to overcome the complexity related to nonlinearity is to adopt multi-model approaches [2–5].

Using multi-model approaches is motivated by the difficulty, and sometimes the inability to analytically describe the system's behavior in its entire operating range. This problem can be considerably reduced by considering that the system's behavior gradually evolves along the operating range. Thus, the system could locally be described by simple functions corresponding to some operating regions. Such an approach can be seen as a weighted contribution of a set of models approximating the whole system's behavior, each of which is valid in a well defined interval or covers a part of the whole feature space of the problem to be solved. The description of the global system's behavior is then made by the combination of the local models. The contribution of each local model to the assessment of the multi-model's

Email address: lisa.serir@femto-st.fr (Lisa Serir)

output is quantified by an *activation degree*. So, the general goal is to determine the contribution rate of each local model in order to minimize the identification error.

The identification task involves two steps: a structural and a parametric identification. The structural identification consists in determining the number of models and the associated activation degrees. It is based on the partitioning of the whole system's feature space and permits the specification of the structure of the local models. The parametric identification is performed to evaluate the parameters of the local models. One can use either a static or a recursive methodology [6].

1.2. Uncertainty management in multi-model approaches

Complex system dynamics often generate significant uncertainty, since understanding the response of nonlinear systems is a very challenging task. In engineering applications, it is very common that the input information to perform the desired analysis is qualitatively and quantitatively limited. Uncertainty sources are numerous and may take the form of system variability, environmental and operational conditions, data acquisition errors, among other sources that vary depending on the application at hand.

This imperfection of the data must be taken into account in the modeling process. In cases where uncertainty cannot be fully attributed to intrinsic variability (aleatory uncertainty), the uncertainty is said to be epistemic and is due to lack of knowledge. Aleatory uncertainty refers to the inherent variation associated with the physical system under question and its environment and cannot be reduced, whereas epistemic uncertainty refers to the lack of knowledge or incomplete information regarding quantities or processes of the system or the environment. In any case, uncertainty quantification is required in order to understand the capabilities and limitations of the modeling process. While probability theory is well suited to deal with aleatory uncertainties (intrinsic variability), other formalisms exist that are more appropriate to manage epistemic uncertainty [7], among which, fuzzy sets or possibility theory and evidence theory, also known as belief functions theory [8] which are the most prominent ones.

In the context of multi-modeling, fuzzy set theory has been used to deal with imprecision within data [9, 10]. Recently, fuzzy rule-based models of Takagi-Sugeno (TSK) type [11] have been widely used in modeling applications of complex systems, due to their flexibility and computational efficiency. TSK models are multi-models with fuzzily defined regions of validity of the local models. The main advantage of the TSK models is that since the local regions are fuzzily defined, the resulting global model can be nonlinear (of high order) while the local models can be very simple. Usually linear (first order) sub-models are considered [11, 12].

A first order Takagi-Sugeno model can be seen as a multi-model structure consisting of linear models. It is based on a fuzzy decomposition of the input space. For each part of the state space, a fuzzy rule can be constructed to make a linear approximation of the input, and the global output is a combination of all rules. Then, the parameters of the models (nonlinear parameters of membership degrees and linear parameters for the consequent of each rule) are tuned in an appropriate learning procedure. Usually, the identification of the linear parameters is addressed by some gradient descent variant whereas nonlinear parameters are determined by a clustering of the input space. This kind of approach has been applied to build a neuro-fuzzy predictor in the context of prognosis application in [13]. It was based on the evolving extended Takagi-Sugeno system (exTS) proposed by Angelov [14].

1.3. On belief functions and their application in TS models

Ramdani *et al.* [15] exploited the theoretical framework of belief functions to deal with uncertainties in multi-modeling. The authors developed a multi-modeling strategy founded on a TSK fuzzy model. The basic idea was to consider a fuzzy rule-based system with a belief structure as output. The focal elements of each rule were formed by a subset of a collection of functional models, each of which was constructed based on a fuzzy model of Takagi-Sugeno type. The main advantage of this approach remains in the use of belief functions theory to determine the activation degrees of the local models because these functions have the particularity to enable the explicit modeling of doubt and ignorance. Their proposed methodology is an *offline* approach and requires the entire dataset to be available in advance for the modeling process.

In this paper, we propose to adapt the offline approach developed in [15] to make it online, in order to deal with sequential data, meaning that the data arrive gradually. In the sequel, we will use the word "evolving" to qualify an algorithm which is able to adapt its parameters online.

The proposed algorithm is called E2GKpro and relies on the Evidential Evolving Gustafson-Kessel algorithm (E2GK) initially developed in [16] to sequentially perform the clustering phase using the formalism of belief functions.

The clustering is then followed by the online identification of local linear models. Therefore, the estimation of the local models is dynamically performed by upgrading and modifying their parameters as data arrive. The activation degrees of each local model are directly provided by the E2GK algorithm in the form of belief masses, and the global model is a combination of all the local models.

The paper is organized as follows. First, the basics of belief functions theory are given and illustrated on a simple clustering problem (Section 2). Next, the proposed E2GKpro algorithm is detailed (Section 3). Finally, the algorithm is applied on several datasets and compared with [15] (Section 4).

2. Background on Evidence Theory

Dempster-Shafer theory of evidence, also called belief functions theory, is a theoretical framework for reasoning with partial and unreliable information. It was first introduced by A. P. Dempster (1968). Dempster's work on upper and lower probabilities led G. Shafer to establish the basis of the theory of evidence [8]. Shafer demonstrated the usefulness of belief functions for the modeling of uncertain knowledge. Later, Ph. Smets proposed a general framework, the *Transferable Belief Model* (TBM) [17], presenting belief functions as an alternative to subjective probabilities. The TBM is used for uncertainty representation and combination of various pieces of information without additional priors. In particular, it offers the possibility to explicitly represent doubt and conflict. We give here some of the basic notions of the theory and refer the reader to [17] for a more complete description.

The central notion of the theory of belief functions is the *basic belief assignment* (BBA), also called *belief mass assignment*, that represents the *belief* of an agent in subsets of a finite set Ω, called the frame of discernment. It is defined by:

$$\begin{array}{cccc} m: & 2^{\Omega} & \rightarrow & [0,1] \\ & A & \mapsto & m(A) \ , \end{array} \qquad (1)$$

with $\sum_{A\subseteq\Omega} m(A) = 1$. A belief mass can be assigned not only to a singleton ($|A| = 1$), but also to a *subset* ($|A| > 1$) without assumption concerning additivity of the measure m [7]. This property permits the explicit modeling of doubt and conflict which constitutes a fundamental difference with probability theory. The subsets A of Ω such that $m(A) > 0$ are called the *focal elements* of m. Each focal element A is a set of possible values of ω in Ω.

Complete ignorance corresponds to $m(\Omega) = 1$, whereas perfect knowledge of the value of $\omega \in \Omega$ is represented by the allocation of the whole mass of belief to a unique singleton of Ω, and m is then said to be *certain*. In the case of all focal elements being singletons, m boils down to a probability function and is said to be *Bayesian*.

A BBA m is said to be *normal* if $m(\emptyset) = 0$. A normalized BBA m^* can be computed from a BBA m by:

$$m^*(A) = \begin{cases} \dfrac{m(A)}{1 - m(\emptyset)} \text{ if } A \neq \emptyset \\ 0 \text{ otherwise} \end{cases} \qquad (2)$$

This process is called Dempster *normalization*.

A positive value of $m(\emptyset)$ is considered if one accepts the *open-world assumption* stating that the set Ω might not be complete, and thus ω might take its value outside Ω. The conflict is then interpreted as a mass of belief given to the hypothesis that ω might not lie in Ω. This interpretation is useful in clustering for outliers detection [18].

Several functions - in *one-to-one correspondence* [17] - can be computed from a BBA. Among these functions, the *plausibility* function [19] is defined by:

$$pl(A) = \sum_{B\cap A\neq\emptyset} m(B), \ \forall A \subseteq \Omega \ , \qquad (3)$$

where $pl(A)$ represents the maximal degree of belief supporting the subset A. It is important to note that pl boils down to a probability measure when m is a Bayesian BBA and to a possibility measure when the focal elements are nested [20]. Probability and possibility measures are thus recovered as special cases of belief functions.

Decision making in the TBM framework consists in the choice of the best hypothesis using the *pignistic probability* distribution [17] defined as:

$$\mathbf{BetP}(\omega) = \sum_{A:\omega\in A} \frac{m(A)}{|A|}, \quad \forall \omega \in \Omega \ . \qquad (4)$$

where each mass of belief $m(A)$ is equally distributed among the elements of A. If the BBA is subnormal ($m(\emptyset) \neq 0$), a preliminary normalization step has to be performed (Eq. 2).

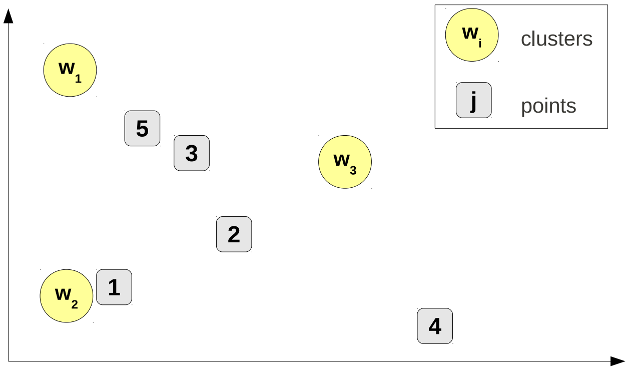


Figure 1: An example of basic belief assignments for a clustering problem.

Example 1. *Considering a set $\Omega = \{\omega_1, \omega_2, \omega_3\}$ of 3 clusters and $N = 5$ data points represented in figure 1, Tab. 1 gives an example of belief masses regarding the membership of each data point to the clusters. BBA's for each data in Tab. 1 illustrate various situations which shows that data 1 certainly belongs to cluster 2. The mass of belief m_2 represents an imprecise situation where data point 2 belongs to cluster ω_2 or ω_3. The class of data point 3 is completely unknown. As $m_4(\emptyset) = 1$, data point 4 is considered as an outlier, i.e., its class does not lie in Ω. Finally, the more general case of partial knowledge is represented for data 5.*

Table 1: Example of basic belief assignment in a clustering problem (see Figure 1 for their graphical representation).

A	$\emptyset$	ω_1	ω_2	$\{\omega_1, \omega_2\}$	ω_3	$\{\omega_1, \omega_3\}$	$\{\omega_2, \omega_3\}$	$\{\omega_1, \omega_2, \omega_3\}$
$m_1(A)$	0	0	1	0	0	0	0	0
$m_2(A)$	0	0	0	0	0	0	1	0
$m_3(A)$	0	0	0	0	0	0	0	1
$m_4(A)$	1	0	0	0	0	0	0	0
$m_5(A)$	0.02	0.3	0.1	0.2	0.1	0.05	0.03	0.2

For a clustering problem, dealing with doubt between clusters is of particular interest. Doubt is generally encountered in data transition and can be useful to limit the number of clusters in the final partition. In the context of multi-modeling, information of doubt may be used to segment properly the feature space taking transitions into account in the modeling process.

3. E2GKpro: Evidential Evolving Prediction

The physical system is supposed to evolve and go by different local linear models, unknown in advance. Each local model has thus to be identified online, as data arrive. The observed data at the current time-step is composed of a n-dimensional input feature vector $x_k = [x_1 \dots x_n]^\intercal \in \mathbb{R}^n$ and an output $y_k \in \mathbb{R}$, with $k \geq 1$ being the current time-instant.

Given (x_k, y_k), E2GKpro proceeds in three phases (after an initialization phase referred to as *Step 0*):

- *Step 1*: A clustering phase, which detects the current operating regions of the system. The number of clusters c can evolve, in particular when a new operating region is detected. At the current instant k, each cluster $i = 1 \dots c$ is identified by two parameters: a center $v_{i,k} \in \mathbb{R}^n$ and a covariance matrix $\Sigma_{i,k} \in \mathbb{R}^n \times \mathbb{R}^n$, both adapted according to (x_k, y_k). This phase relies on the evidential evolving clustering algorithm called E2GK [16].

- *Step 2*: A regression phase, which updates the local linear models corresponding to each cluster. A local model relates x_k to y_k such that, given only the input x_k, the system would be able to predict the output $\hat{y}_k$ as close as possible to the true value y_k. Therefore, in this phase, the goal is to optimize the vector of parameters of the i-th local model, denoted $\pi_i = [a_{i0} \;\; a_{i1} \;\; a_{i2} \ldots a_{in}]^{\mathrm{T}}$, given some observed couples (x_k, y_k).
- *Step 3*: A prediction phase, which estimates the value $\hat{y}_k$ taken by the output, given the input x_k. The prediction is computed by a weighted sum of local linear models, where the weights are provided directly by the clustering phase:

$$\hat{y}_k = \sum_{i=1}^{c} \lambda_{i,k} \cdot x_{e,k} \cdot \pi_i \tag{5}$$

  where $x_{e,k} = [1; x_k]$ is the extended input data vector and $\lambda_{i,k}$ is the normalized activation degree of the i-th cluster at time k.

This three-phases scheme is quite common in existing work in evolving systems modeling [1]. The originality of the proposed method lies in the use of the theory of belief functions to both represent the uncertainty concerning the characterization of local models and estimate their activation degrees.

In the sequel, the whole method is described step by step.

3.1. Step 0: Initialization

Let consider that a few data points are available at the beginning of the modeling, such that two operating regions (clusters, thus $c = 2$) can be identified, with a center, a covariance matrix and a local model. This initial configuration enables us to present in the sequel how the belief functions are generated.

As suggested in [14], the clustering phase can be improved if performed in the input-output space $z_k = (x_k, y_k)$. Given the initial input-output data z_k, a standard clustering algorithm such as the Gustafson-Kessel one, can be used to find the coordinates $v_{i,k}, i = 1, 2$ of the two first clusters and to initialize their covariance matrices $\Sigma_{i,k}, i = 1, 2$.

3.1.1. Step 0a: Computing the partition

Given the centers and the covariance matrices, the evidential partition matrix, which estimates the belonging degree of the k-th data point to each possible subset of clusters $A_i \in 2^\Omega \backslash \emptyset$, can be computed [16]:

$$m_{ik} = \frac{|A_i|^{-1} \cdot d_{ik}^{-2}}{\sum\limits_{A_l \neq \emptyset} |A_l|^{-1} \cdot d_{lk}^{-2} + \delta^{-2}} \ , \tag{6}$$

and the mass assigned to the empty set is equal to:

$$m_{\emptyset k} = 1 - \sum_{A_i \neq \emptyset} m_{ik} \ , \tag{7}$$

where d_{ik} denotes the distance between data point z_k and subset A_i, and $\delta \in \mathbb{R}^+$ controls the amount of data considered as outliers ($\delta = 10$ was chosen in [16]).

Only singleton focal elements (clusters $\omega_k \in \Omega$) are associated with centroids but the particularity of the mass computation in the evidential evolving clustering algorithm (E2GK) holds in the consideration of virtual centroids located at the barycenters of subsets of clusters [18]. This makes the proposed approach different from usual clustering approaches (possibilistic and probabilistic ones).

In Eq. 6, the distance d_{ik} between a data point at time k and any nonempty subset $A_i \subseteq \Omega, A_i \neq \emptyset$ is computed by first defining the center of each subset A_i as the barycenter $\bar{v}_{i,k}$ of clusters' centers composing A_i:

$$\bar{v}_{i,k} = \frac{1}{|A_i|} \sum_{l=1}^{c} b_{li} \cdot v_{i,k}, \tag{8}$$

with

$$b_{li} = \begin{cases} 1 & \text{if } \ \omega_l \in A_i, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The covariance matrix $\Sigma_{i,k}$ can be computed as:

$$\Sigma_{i,k} = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \sum_{A_j \ni \omega_i} (z_k - \overline{v}_{i,k}) \cdot (z_k - \overline{v}_{i,k})^{\mathrm{T}} \ , \tag{10}$$

which is an analog in the evidential framework of the fuzzy covariance matrix computed for any nonempty subset of clusters.

The Mahalanobis-like distance d_{ik} is then used to cope with clusters with arbitrary shape. For any nonempty subset of clusters A_i represented by its center $\overline{v}_{i,k}$ and its covariance matrix $\Sigma_{i,k}$, the distance $d_{i,k}$ is given by:

$$d_{ik}^2 = \left\| z_k - \overline{v}_{i,k} \right\|_{S_{i,k}}^2 = (z_k - \overline{v}_{i,k}) \cdot S_{i,k} \cdot (z_k - \overline{v}_{i,k})^{\mathrm{T}} \ , \tag{11a}$$

$$S_{i,k} = \left[\det(\Sigma_{i,k})\right]^{1/n} \cdot \Sigma_{i,k}^{-1} \ , \tag{11b}$$

Given the definition of the initial clusters, the local models can be initialized.

3.1.2. Step 0b: Initializing the local models

Let θ_i be the parameters of the local linear model for the i-th cluster. Let X_i denote the set of input points belonging to the i-th cluster and Y_i the corresponding outputs. The local linear model can be optimized by a standard least squared approach:

$$\theta_i = \left(X_i^{\mathrm{T}} \cdot X_i \right)^{-1} \cdot X_i^{\mathrm{T}} \cdot Y \tag{12}$$

At this stage, the initialization is completed. The E2GKpro algorithm can now be run online: if the observed data is an input-output couple z_k then the clustering and the local models can be updated, whereas if only the input data x_k is observed, a prediction $\hat{y}_k$ of the output can be estimated.

3.2. Step 1: Updating the clusters when an input-output datum is observed

When a new input-output datum z_k is observed, the boundary of each cluster is first estimated (Step 1a) in order to know whether the new data point is close or far from the closest cluster. Then the clustering structure is modified (Step 1b) by either adapting the existing clusters or by adding a new cluster. The consistency of the structure is then checked before updating the local models (Step 2).

3.2.1. Step 1a: Find the clusters' radii

The radius r_i of the i-th cluster is computed by:

$$r_i = \underset{\forall z_k \in \, i\text{-th cluster and } \ \lambda_{ik} > c^{-1}}{\text{median}} \left\| z_k - v_{i,k} \right\|_{S_{i,k}} \ , \tag{13}$$

where λ_{ik} is the confidence degree that point x_k belongs to a singleton cluster ($\omega_i \in \Omega$). The confidence degrees can be estimated by several processes as proposed in [16] such as the pignistic transformation (Eq. 4) which transforms the belief mass distribution $m_{\cdot,k}$ into a probability distribution suitable for decision-making:

$$\lambda_{i,k} = \sum_{A_j, \omega_i \in A_j} \frac{m_{j,k}}{|A_j|} \tag{14}$$

The closest cluster to z_k is then found by:

$$p = \operatorname*{argmin}_{i=1}^{c} d_{ik}^2 \tag{15}$$

and according to the Mahalanobis-like distance d_{pk}^2 between both the center of the closest cluster and z_k, two possible cases are considered:

- Case 1: $d_{pk}^2 \leq r_p$, i.e. z_k belongs to an existing cluster, inducing a clusters' update;
- Case 2: $d_{pk}^2 > r_p$, i.e. z_k is too far from existing clusters, involving the creation and the validation of a new cluster.

3.2.2. Step 1b: Adapting the structure

Case 1: if $d^2_{pk} \leq r_p$, the updating of the p-th cluster is performed by modifying the closest cluster's center:

$$v_{p,k+1} = v_{p,k} + \theta \cdot \Delta \ , \tag{16}$$

where

$$\Delta = z_k - v_{p,k} \ , \tag{17}$$

and $\theta \in [0, 1]$ is the updating rate.

The inverse of the covariance matrix as well as its determinant, which are used to estimate the belief masses, can be recursively adapted by [21]:

$$\Sigma^{-1}_{p,k+1} = \left(I - \frac{\theta \cdot \Sigma^{-1}_{p,k} \cdot \Delta^{\mathrm{T}} \cdot \Delta}{\xi} \right) \cdot \Sigma^{-1}_{p,k} \cdot \frac{1}{1-\theta}, \tag{18}$$

where I is the identity matrix and

$$\xi = 1 - \theta + \theta \cdot \Delta \cdot \Sigma^{-1}_{p,k} \cdot \Delta^{\mathrm{T}} \tag{19}$$

and

$$\det\left(\Sigma_{p,k+1}\right) = (1-\theta)^{n-1} \cdot \det\left(\Sigma_{p,k}\right) \cdot \xi. \tag{20}$$

The partition matrix can then be computed by first estimating the Mahalanobis-like distance (Eq. 11b) and then the belief masses (Eq. 6, 7).

Case 2: if $d_{pk} > r_p$, the number of clusters is incremented ($c \leftarrow c + 1$). The initialization of the parameters of the new cluster can be performed by accepting the incoming data z_k as the new center ($v_{c,k+1}$) and the covariance matrix ($\Sigma_{c,k+1}$, more precisely both its inverse and determinant) can be chosen equal to the covariance matrix of the closest cluster, i.e. $\Sigma_{p,k}$. Another strategy, which appeared practically much more efficient in different applications, is to consider the $n \cdot (n+1)/2$ last data points as being relevant for the new cluster and to use them to estimate both its center and covariance matrix.

The creation of a cluster makes the partition evolve with the possibility to drastically reduce the number of points in some clusters. It was proposed in [16] to validate the partition by ensuring that each cluster in the new partition has a minimum of points denoted P_{tol}. If a cluster does not satisfy this constraint, it is simply removed. Therefore, the number of clusters can increase and decrease accounting for a better modeling of the data structure as demonstrated in [16].

The parameter P_{tol} controls the minimum number of points in each cluster and also enables the user to guarantee the validity of both the covariance matrices and the local models (as detailed below). This parameter is context-determined and its sensitivity was studied in [16].

3.3. Step 2: Online estimation of the local models' parameters

For online applications, the estimates of the parameters of local linear models at a given instant k, i.e. $\pi_{i,k} = [a_{i0,k} \ \ a_{i1,k} \ \ a_{i2,k} \ldots a_{in,k}]^{\mathrm{T}}$ should be computed given their previous estimates $\pi_{i,k-1}$ and the new input vector z_k.

Since the models are linear, it can be performed by a recursive least squared approach:

$$C_{i,k} = C_{i,k-1} - \frac{\lambda_i(x_{k-1}) \cdot C_{i,k-1} \cdot x^{\mathrm{T}}_{e,k-1} \cdot x_{e,k-1} \cdot C_{i,k-1}}{1 + \lambda_i(x_{k-1}) \cdot x^{\mathrm{T}}_{e,k-1} \cdot C_{i,k-1} \cdot x_{e,k-1}} \tag{21a}$$

$$\hat{\pi}_{i,k} = \hat{\pi}_{i,k-1} + C_{i,k} \cdot x_{e,k-1} \cdot \lambda_i(x_{k-1}) \cdot \left(y_k - x^{\mathrm{T}}_{e,k-1} \cdot \hat{\pi}_{i,k-1} \right) \tag{21b}$$

with $k > 2$ (time-step), $i = [1, \ c]$ (c is the number of clusters which equals the number of local models), $x_{e,k} = [1; x_k]$ is the extended input data vector, and $C_{i,k} \in \mathbb{R}^{(n+1)\times(n+1)}$ is the covariance matrix of the i-th local model at time k, which is updated by the Ricatti equation as in the Kalman filter. The initial conditions ($k = 1$) are set to $\hat{\pi}_1 = 0$ and $c_{i,1} = \alpha \cdot I$, where I is the identity matrix and α a large value (for example 100).

3.4. *Step 3: Prediction*

Given an input datum x_k, the goal of the prediction phase is to estimate the value $\hat{y}_k$ taken by the output. A simple weighted sum of local linear models can be applied:

$$\hat{y}_k = \Psi^{\mathrm{T}} \cdot \theta \tag{22}$$

where $\Psi = [\lambda_{1,k} \cdot x_{ek}; \ldots; \lambda_{i,k} \cdot x_{ek}; \ldots; \lambda_{i,c} \cdot x_{ek}]$ is the vector of the inputs weighted by the normalized activation degrees of the local models, and $\theta = [\pi_{1,k}; \ldots; \pi_{i,k}; \pi_{c,k}]$ is the vector of the parameters of the linear local models.

3.5. *Complexity issues*

Belief functions are defined on the power set of the set of clusters, therefore, given z_k, 2^c belief masses have to be estimated. For some practical applications, where the processing time has to be as short as possible, this drawback of E2GKpro needs to be dealt with. In [16], a solution that relies on the concept of k-additive belief masses was proposed. It consists in computing the belief masses for subsets with a cardinality less or equal to a given number [22].

For time-series analysis, if the processing time is an important criterion, we propose to consider a 2-additive belief mass, which is practically relevant because of the following reasons:

- The belief masses on singletons (subset with cardinality equal to one) represent the activation degrees of clusters and reflects how close is located a data point to the centers.
- The belief masses on unions of two clusters (subset with cardinality equal to two) represent the cases where the system is possibly making a transition between operating regions that could mean a change from one functioning mode to another.

In E2GKpro, the complexity can be further reduced if the masses are not used as such. Indeed, the algorithm is based on the activation degrees of singletons computed from the belief masses. Therefore, we propose to compute directly the normalized activation degrees using the pignistic transformation (Eq. 14), without storing the belief masses. For a given point z_k:

- Compute the belief mass $m_{j,k}$ for a given subset A_j with $|A_j| \leq 2$,
- Update $\lambda_{i,k} \leftarrow \lambda_{i,k} + m_{j,k}/|A_j|$ if $w_i \in A_j$ (initial values of λ being set to 0).

In case the plausibility transform [19] is used instead of the pignistic one for decision-making, one can apply a similar scheme by updating $\lambda_{i,k}$ as follows:

$$\lambda_{i,k} \leftarrow \lambda_{i,k} + m_{j,k}/|A_j| \text{ if } w_i \cap A_j \neq \emptyset. \tag{23}$$

4. Experiments

4.1. *Non linear static function approximation [15]*

To illustrate the proposed methodology, the following univariate function was chosen:

$$y(x) = 3 \cdot \exp(-x^2) \cdot \sin(\pi \cdot x) + \eta \tag{24}$$

This function was taken from [15] for comparison purpose. In this definition, η is a Gaussian noise with zero mean and $\sigma^2 = 0.15$. By using random inputs x uniformly distributed in $[-3, 3]$, 300 samples of $y(x)$ were obtained.

In [15], the input-output data $z_k = (x_k, y_k)$ were clustered using the Gustafson-Kessel algorithm (GK) with $K = 7$ clusters. Then, the approximation using a TS model was estimated: GK+LS (resp. GK+WLS) means that a least squared algorithm (resp. weighted LS) was applied on the clustering results, whereas GK+FUSION stands for the combination (made offline) of GK+LS with GK+WLS using belief functions. The performance of these approximations was assessed by the mean squared error (MSE) recalled for comparison purpose in Table 2. The MSE is defined as:

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} (\hat{y}_k - y(k))^2 \tag{25}$$

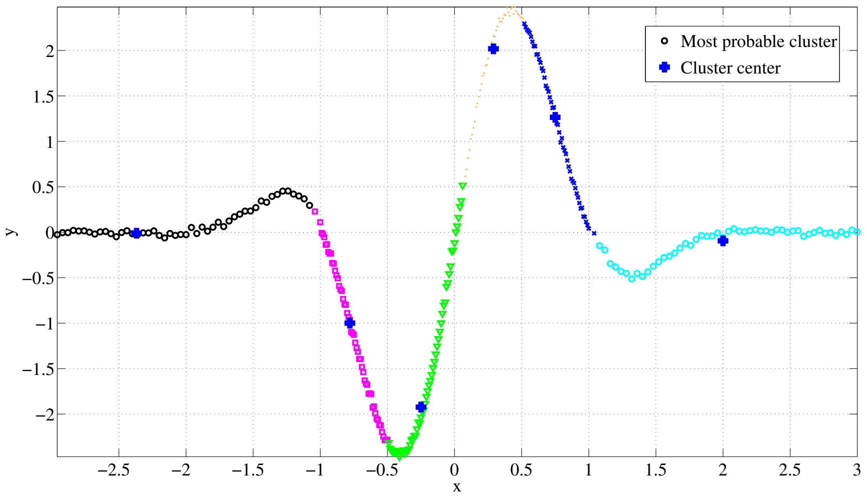


Figure 2: Most activated clusters and their centers.

and the NDEI, which is often used in process identification [23], is also depicted (for E2GKpro). It is defined by:

$$\text{NDEI} = \frac{\sqrt{MSE}}{\sigma(y)} \tag{26}$$

where $\sigma(y)$ is the standard deviation of the target (true value of y).

The results provided by the proposed E2GKpro algorithm are provided in Table 2. “E2GKpro + KF” means that E2GKpro was used with the Kalman filter as proposed in Section 3.3 with $P_{\text{tol}} = 40$, $\theta = 0.1$, and using $z_k = (x_k, y_{k-1}, y_k)$. Compared to [15], the input-output data vector was extended with the previous output due to the online filtering process (Section 3.3).

Model	Parameters	MSE	NDEI
GK+LS, offline [15]	$K = 7$, $z_k = (x_k, y_k)$	0.0058	-
GK+WLS, offline [15]	$K = 7$, $z_k = (x_k, y_k)$	0.0196	-
GK+FUSION, offline [15]	$K = 7$, $z_k = (x_k, y_k)$	0.0181	-
E2GKpro+KF, online	$P_{\text{tol}} = 40$, $\delta = 10$, $z_k = (x_k, y_{k-1}, y_k)$	0.0020	0.0332

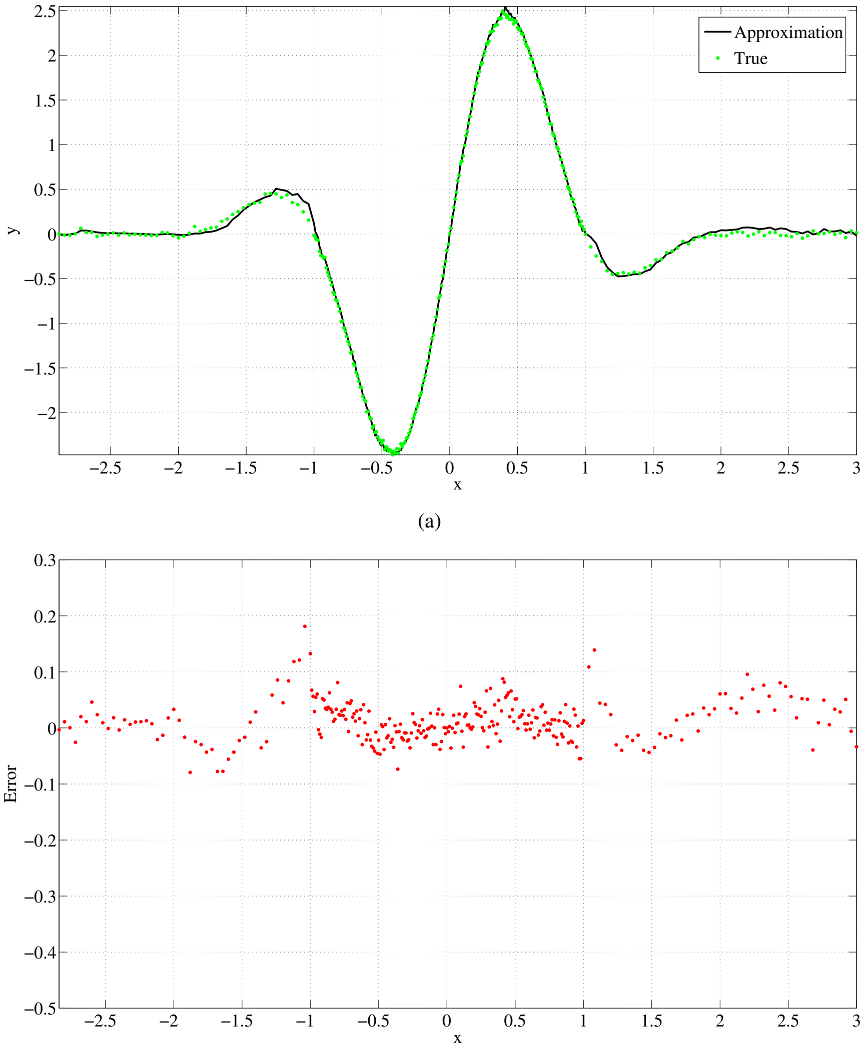
Table 2: Error on the first dataset. The variance on the results of E2GKpro was lower than 10^{-5}.

The results of E2GKpro were obtained by first generating 30 datasets (since η is a random variable), then running E2GKpro and finally averaging. E2GKpro (online) algorithm with the Kalman filter reduced the error made by the “GK+FUSION” (offline) algorithm proposed in [15] by approximately 90 percents. Note that this improvement was obtained with 6 clusters instead of 7 as in [15]. The result provided by “GK+LS” (offline clustering followed by least squared) is better than the online strategy because the models are optimised using all the data and using more clusters. This approach is however not well-suited for data streams.

As an illustration of E2GKpro results on this first application, let us first consider the clustering provided in Figure 2. It depicts the most probable cluster for each data point as well as the cluster centers. The segmentation presents a sequence of well-separated clusters.

Figure 3 is a pictorial representation of the prediction along time provided by the Kalman filter integrated in E2GKpro algorithm to estimate the parameters of the local models.

The next figure (4) illustrates the evolution of the three parameters in the six local models, i.e. $\pi_{i,k} = [a_{i0,k} \;\; a_{i1,k} \;\; a_{i2,k}]^{\mathsf{T}}$, $i = 1 \ldots 6$ (see Section 3.3), along time. The evolution is made smooth by the Kalman filter, which also explains the most important errors in the nonlinearities. When a new cluster is added in E2GKpro, a new local model is created



(a)

(b)

Figure 3: Prediction and error provided by the scheme “E2GKpro + KF”.

Model	Parameters	# rules	NDEI
E2GKpro+KF	$P_{\text{tol}} = 40, \theta = 0.1, \delta = 10$	35	0.3263
eTS [14]	θ	113	0.0954
DENFIS [23]	θ, D_{thr}	58	0.276
DENFIS [23]	–	883	0.033
ESOM [25]	ϵ, α, σ, γ, T_p	114	0.32
ESOM [25]	–	1000	0.044
EFuNN [26]	$\theta, S_{\text{thr}}, E_{\text{thr}}, \text{lr}_1, \text{lr}_2$, WLSE	193	0.401
EFuNN [26]	–	1125	0.094
RAN [27]	τ, ϵ, θ, κ, δ_{max}, δ_{min}	113	0.373
NEURAL gas [27]	–	1000	0.062

Table 3: Error on the second dataset for the proposed approach (E2GKpro) using the Kalman filter (KF), and comparison with other approaches in terms of NDEI and number of parameters (the meaning of parameters for each method can be found in the reference given in the first column).

Model	Parameters	# rules	NDEI
E2GKpro+KF	$P_{\text{tol}} = 40, \theta = 0.1, \delta = 10$	43	0.3621
eTS [14]	θ	124	0.3096

Table 4: Error on the second dataset with noise.

and initialized. The discontinuities are due to the initial values of local model parameters, in particular the covariance matrix.

Finally, for a given data point at instant k, the opposite of the maximum degree of belief regarding the membership of this point to each cluster:

$$\text{Conf}_k = 1 - \max_{i=1}^{c} \lambda_{ik}, \tag{27}$$

reflects the remainder of the mass not assigned to the most probable cluster. This value can be interpreted as a confidence degree in the fact that the current partition (Eq. 6, 14) does *not* well fit the newest data. An illustration of this confidence degree is depicted in Figure 6 which emphasizes that the highest values of the confidence degree are obtained for the nonlinear parts generally located between two existing clusters.

4.2. Prediction on a chaotic time-series [14]

The proposed E2GKpro algorithm was tested on a benchmark problem: the Mackey-Glass chaotic time-series prediction. The results are compared with several existing approaches.

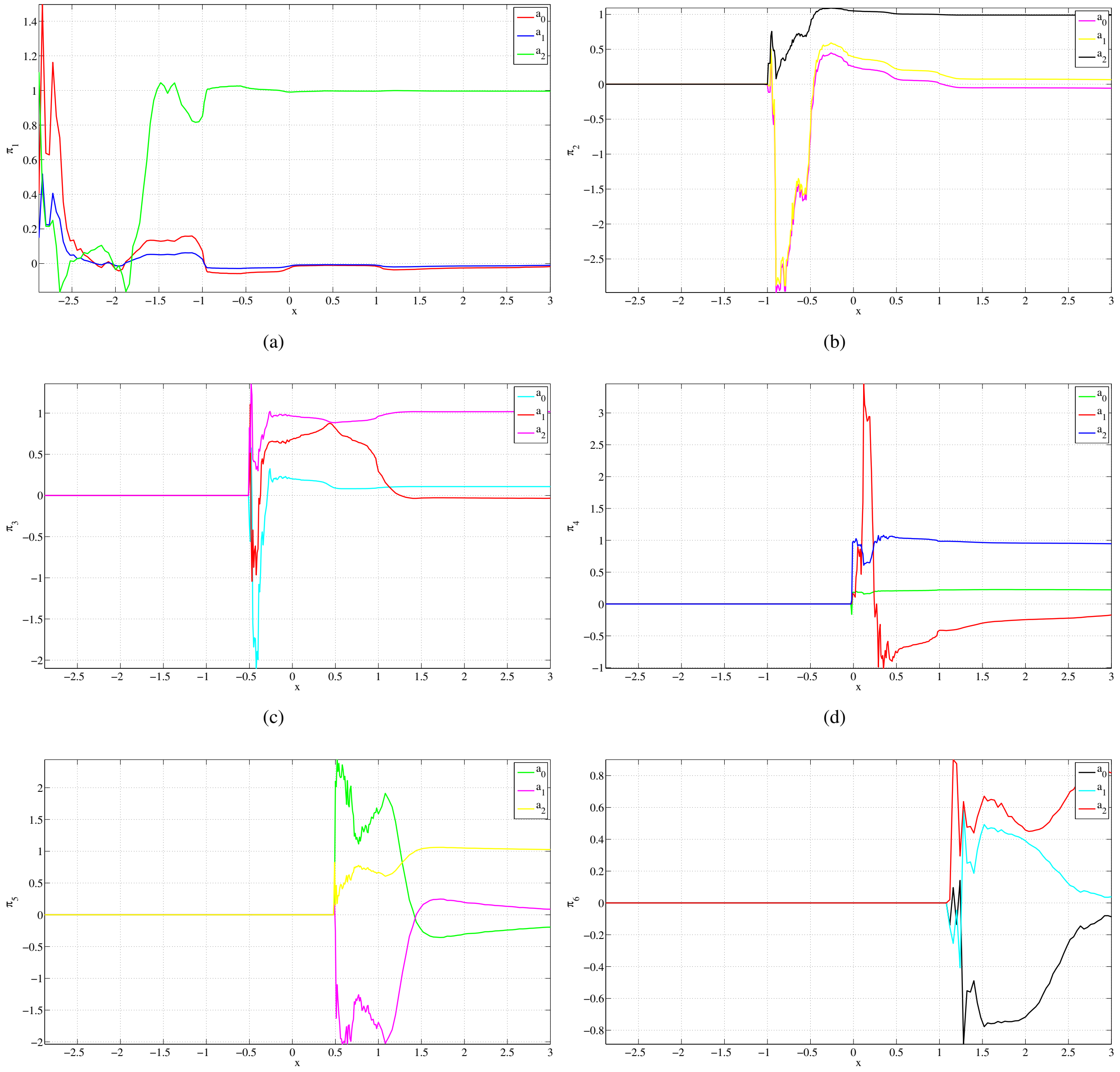
The time-series is generated by:

$$x(t) = \frac{a \cdot x(t-\tau)}{1 + x^{10}(t-\tau)} - b \cdot x(t) \ , \tag{28}$$

with the same parameters as in [14, 23]: $a = 0.3, b = 0.1, \tau = 20$ and $x_0 = 1.2$. Two sets of data were built as proposed in [23]: 3000 data points from $t = 201$ to $t = 3200$ for training, and 500 data points from $t = 5001$ to $t = 5500$ for validation. In [14], the time-step was chosen equal to 1 (as done in this paper), while 0.1 was chosen in [23].

The aim of the task is to predict the value taken by the signal 85 steps ahead, i.e. $x(t + 85)$, while using as inputs the vector $[x(t-18) \ ; \ x(t-12) \ ; \ x(t-6) \ ; \ x(t)]$. As suggested in [14], the performance is assessed by the NDEI criterion. The results are reported in Table 3 with a comparison with some well-known algorithms of the literature (see [24] for a complete review). The most important parameters are also mentioned in the table in order to quantify the amount of supervision required to tune the algorithms. Note that the parameter P_{tol} in E2GKpro was set to 40, as in the previous application.

This table shows that E2GKpro generates a model with only 35 rules, which is the smallest number of rules compared to the present literature. The error rate, represented by the NDEI, makes it quite competitive, with sometimes better results than existing approaches. The obtained model can thus be practically useful because the less the number of rules is (while ensuring low error), the easier is the interpretation. Table 4 gives the NDEI for the case where the



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Evolution of the parameters for each rule in the first application.

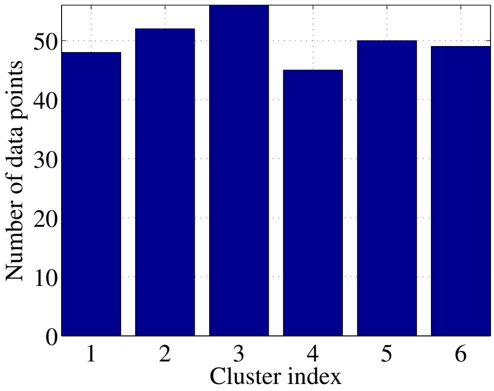


Figure 5: Number of points in each cluster for the first application (P_{tol}).

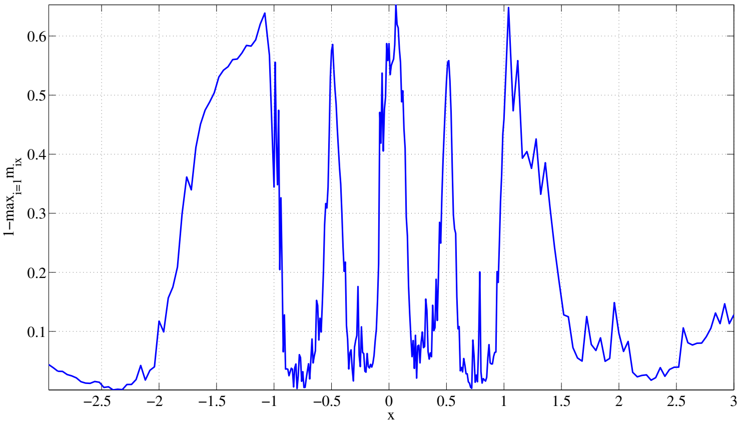


Figure 6: Confidence degree in the fact that the current partition does *not* well fit the newest data.

Mackey-Glass time-series is corrupted by 5% uniform random noise as proposed in [14]. This result shows that the E2GKpro algorithm is robust to noise since the NDEI is quite stable. This conclusion was also drawn in [16] where the clustering based on E2GK (without prediction) outperforms [21] in the presence of noise. The use of virtual clusters in belief mass computation accounts for the low number of rules and for the stability of the results against noise.

As shown in [16], the most critical parameter in E2GKpro is the parameter P_{tol}, which was shown to be much less sensitive than in [21] because of the use of belief functions. In both applications presented in this paper, P_{tol} was set to the same value (40). Concerning the parameter δ, it sets the amount of mass assigned to the empty set, which is then spread out over λ after decision-making. In all E2GK applications, this parameter was set to $\delta = 10$. Concerning θ, it is a common threshold found in all evolving algorithms, which makes them more or less sensitive to the influence of a new data point. In both applications presented in this paper, $\theta = 0.1$.

Finally, Figure 7 illustrates the evolution of the error made by E2GKpro with the Kalman filter. As in many filtering approaches, the error is generally high when the variation is important, for example between 5200 and 5300.

4.3. Application on a real-world dataset from the PRONOSTIA platform

4.3.1. Description of PRONOSTIA

As a second multi-dimensional case, we chose to implement the proposed method on a dataset provided by the PRONOSTIA platform. PRONOSTIA is an experimentation platform (Figure 8) dedicated to the test and validation of the machinery prognosis approaches, focusing on bearing prognostics. It was developed in the Department of Automatic Control and Micro-Mechatronic Systems (AS2M) of FEMTO-ST[1] institute. The main objective of PRONOSTIA is to provide real experimental data [28] that characterize the degradation of a ball bearing along its whole operational life (until fault/failure). The collected data are vibration and temperature measurements of the rolling bearing during its functioning mode.

The internal bearing ring is put in rotation, while the external bearing ring is maintained fixed. A radial load is applied on the external bearing ring in order to simulate its functioning. To speed up the degradation, the load exceeds the maximal load recommended by the supplier. The originality of this experimental platform lies not only in the conjunction of the characterization of both the bearing functioning (speed, torque and radial force) and its degradation (vibrations and temperature), but also in the possibilities, offered by the platform, to make the operating conditions of the bearing vary during its useful life. Figure 8(c) depicts a bearing before and after the experiment.

The bearing operating conditions are determined by instantaneous measures of the radial force applied on the bearing, the rotation speed of the shaft handling the bearing, and of the torque inflicted on the bearing. During a test, the rolling bearing starts from its nominal mode until the fault state. The bearing behavior is measured using different types of sensors (Figure 8(b)) such as miniaturized acceleration sensors and temperature probe.

The raw signals provided by the sensors are processed in order to extract relevant information concerning bearings states. Several techniques have been implemented and gathered in a signal processing toolbox with Matlab (Fig. 8(d)): time-domain methods (RMS, skewness and kurtosis, crest factor, K-factor, Peak-to-Peak), frequency-domain methods (spectral and cepstrum analysis, envelope detection), time-frequency domain (short-time Fourier transform) and wavelets (discrete transform).

4.3.2. Prognostic on the PRONOSTIA platform

The data used here are available on the PRONOSTIA website[2]. More precisely, the data related to the first functioning condition are used.

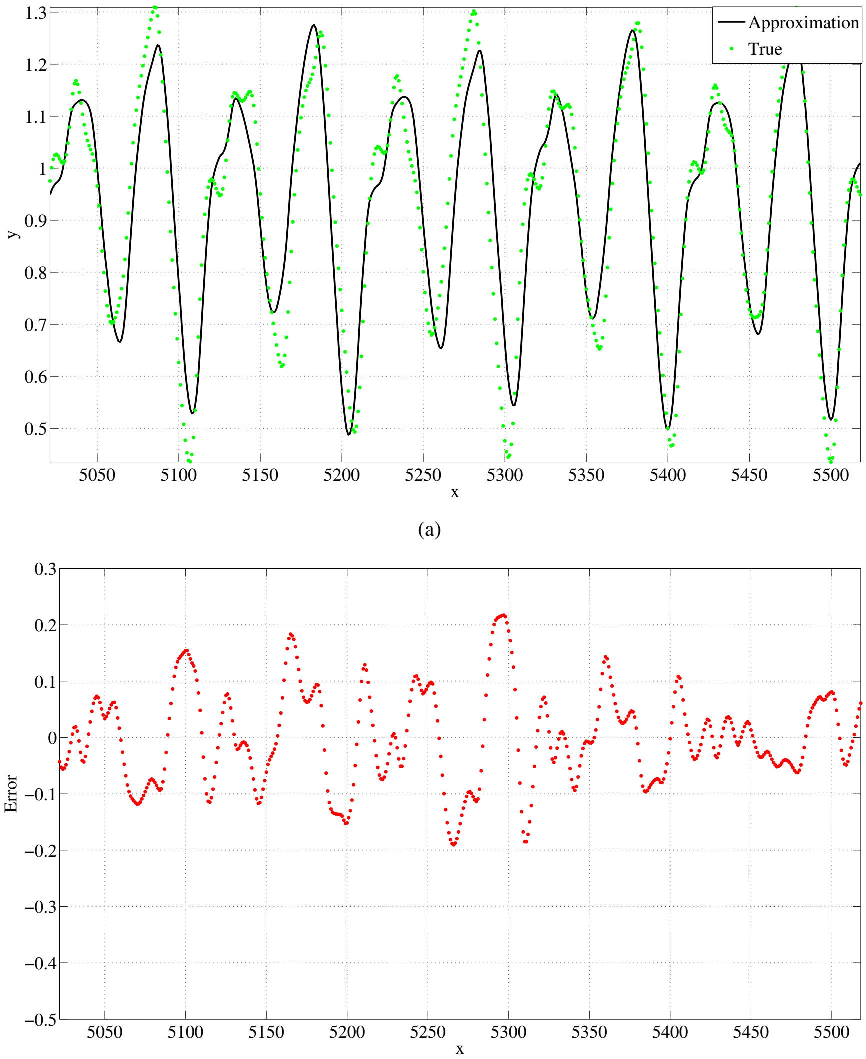
From the horizontal accelerometer data, two features are generated:

- The RMS which is computed at a given time k in a window of size 50. Let S be the signal of the accelerometer and $\mu(S_W)$ its average value, then:

$$RMS(k) = \sqrt{\frac{1}{W}\sum_{i=k-W+1}^{k}\left(S(i) - \mu(S_W)\right)^2} \tag{29}$$

[1] Franche-Comte, Electronics, Mechanics, Thermal Processing, Optics - Science and Technology

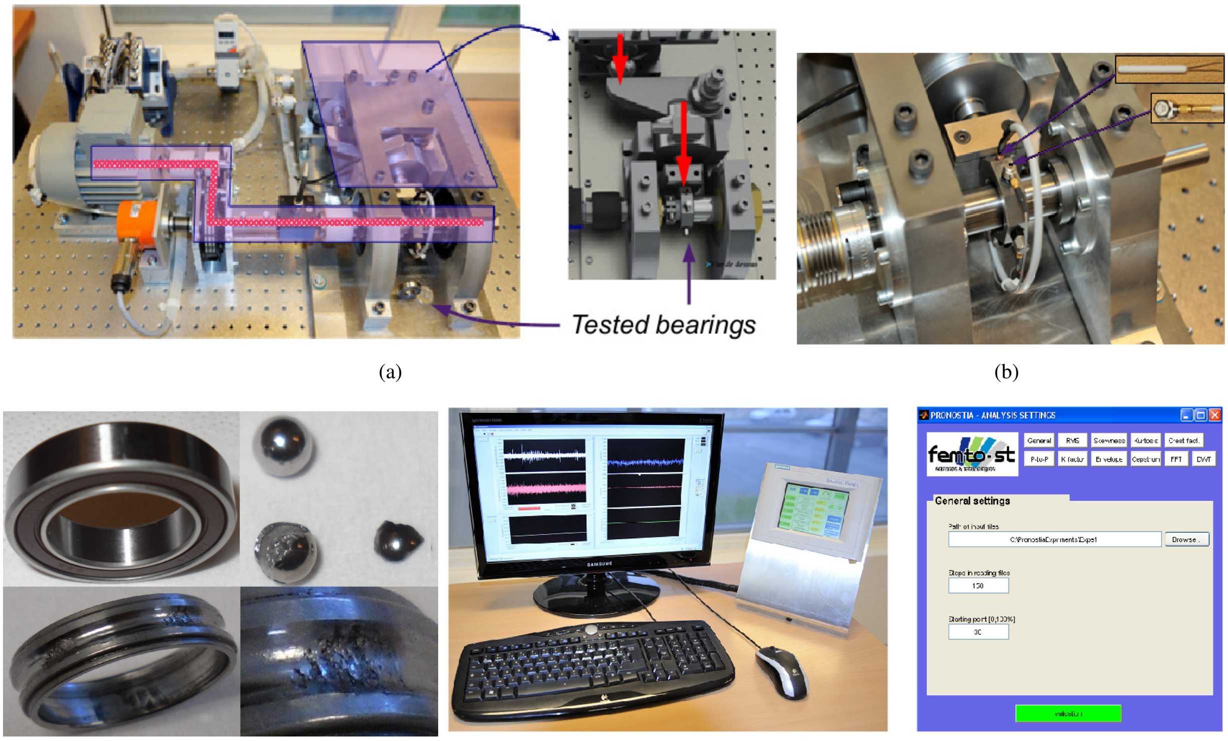
[2] See `http://www.femto-st.fr/ieee-PHM2012-data-challenge`.



(a)

(b)

Figure 7: Prediction (top) and error (bottom) for E2GKpro+KF in the second application.



Path of input files: C:\PronostiaExpriments\Expe1

Steps in reading files: 150

Starting point [0,100%]: 30

(a)

(b)

(c)

(d)

Figure 8: 8(a) The PRONOSTIA platform, 8(b) close-up view on sensors for degradation measurement, 8(c) example of degraded bearings and 8(d) the software for degradation analysis.

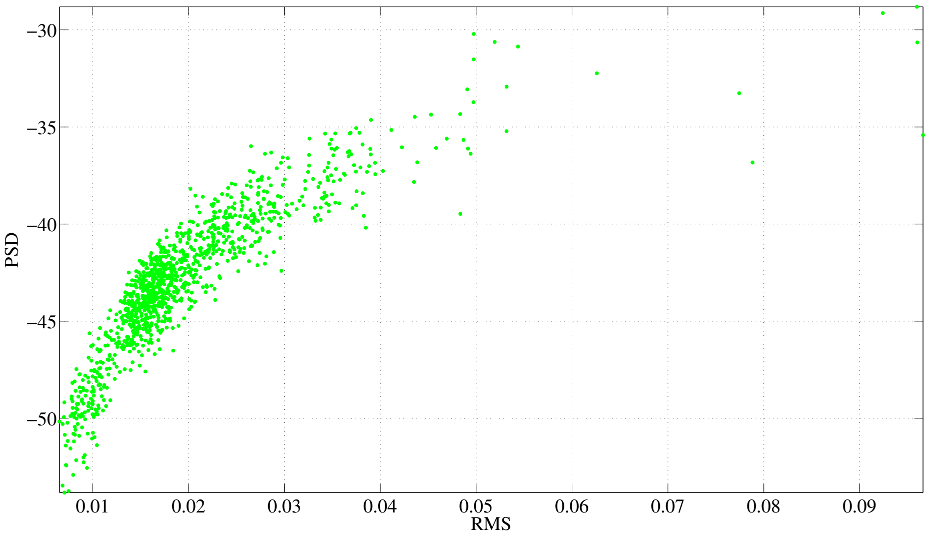


Figure 9: A set of data obtained from the PRONOSTIA platform.

- The mean value of the power spectral density in the same window as the RMS, denoted $PSD(k)$.

An illustration of these features is given in Figure 9 for the testing data used in the sequel. The data depicts a typical behavior of wear with a small amount of data with high variance close to the end of the experiment (top, right-hand side), while the normal behavior is represented by a larger amount of data points gathered in a smaller space (bottom, left-hand side). The variance is partly due to the high level of noise generated by the vibrations during the experiment.

Let $x(k) = [RMS(k) \quad PSD(k)]^{\mathsf{T}}$ the feature vector at instant k. The goal is to predict $x_{RMS}(k+100)$ and $x_{PSD}(k+100)$ given $x_{in} = [x(k-18) \quad x(k-12) \quad x(k-6) \quad x(k)]^{\mathsf{T}}$. For each data point, x_{in} is a vector with 4×2 elements and the output vector is $x_{out} = [x_{RMS}(k+100) \quad x_{PSD}(k+100)]^{\mathsf{T}}$. The clustering is performed in the input-output space made of 10 elements. Given an input data vector (with 8 elements), the prediction is estimated by projecting the centers and the covariance matrix onto the input space, then computing the degree of membership to each cluster, followed by the regression.

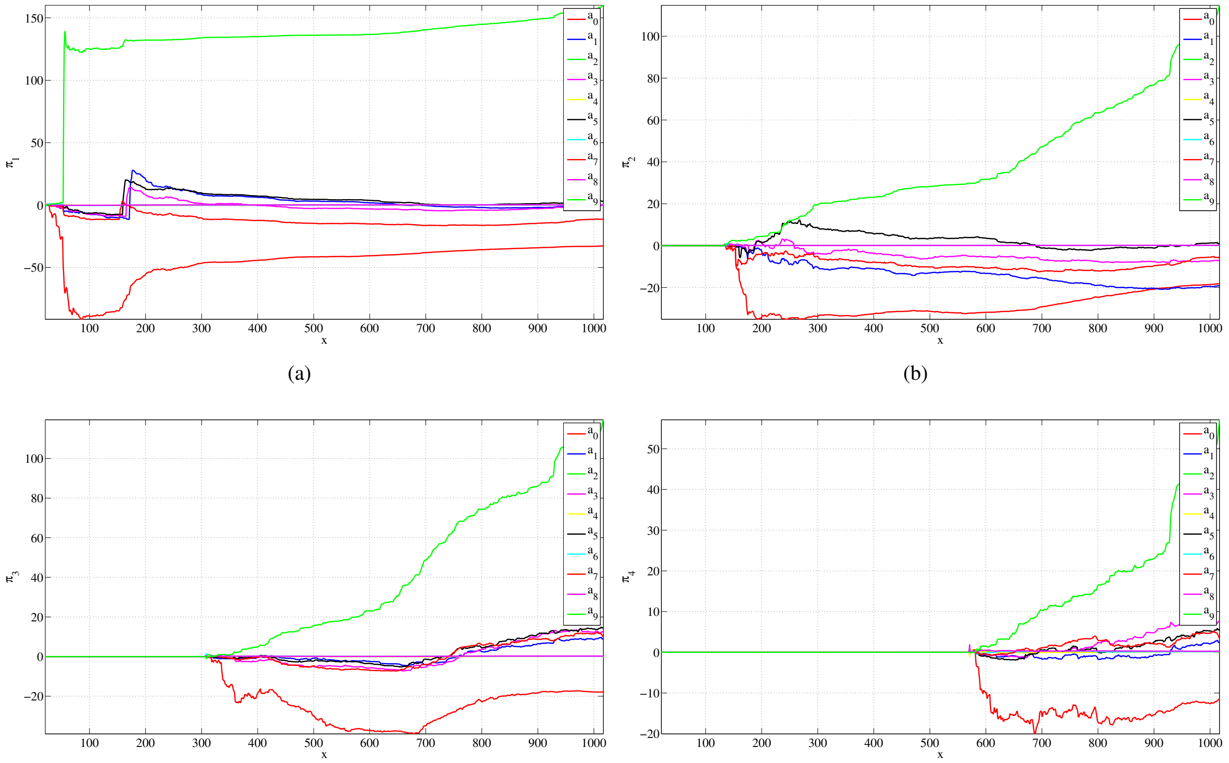
The engine is launched and the bearing is gradually degraded. The data are processed along time by E2GKpro, so that clusters representing operating regions are created and the local models estimated. Only one experiment is used to tune the parameters of E2GKpro. The data related to the training dataset is given in Figure 9. The local models are updated as displayed in Figure 10 for the four first rules. The local model being initialized, the predictions 100 steps ahead can be estimated. The result of the prediction on the training data is depicted on Figure 11 (NDEI = 0.4986).

The second experiment is then used as a testing dataset. An interesting characteristic of the PRONOSTIA's data holds in the fact that all experiments depict different degradation trends, although bearings with similar mechanical properties were used. An illustration of this variability is represented in Figure 12. In this kind of application, the use of online methods is thus well justified. This figure also shows the position of the clusters found during both training and testing. In the latter case, E2GKpro starts with the local models estimated during the former but E2GKpro still adapts the models accounting for new clusters in the new operating regions encountered in the testing dataset.

The result of the prediction on the second experiment (with features depicted in Figure 9 and on the left-hand side of Figure 12) is shown in Figure 11. In this example, E2GKpro generates smooth predictions, which appear useful in this real-world application because it gives the global trend of the functioning behavior. Another interesting aspect is the updated set of operating regions found during testing (Figure 12), which can then be used for another experiment.

5. Conclusion

E2GKpro is an evidential approach proposed for detecting, adapting and combining local models in order to analyze complex systems behavior based on sequential data. The approach relies on three main processes performed



(a)

(b)

(c)

(d)

Figure 10: Evolution of the parameters for rules 1 to 4 in the third application.

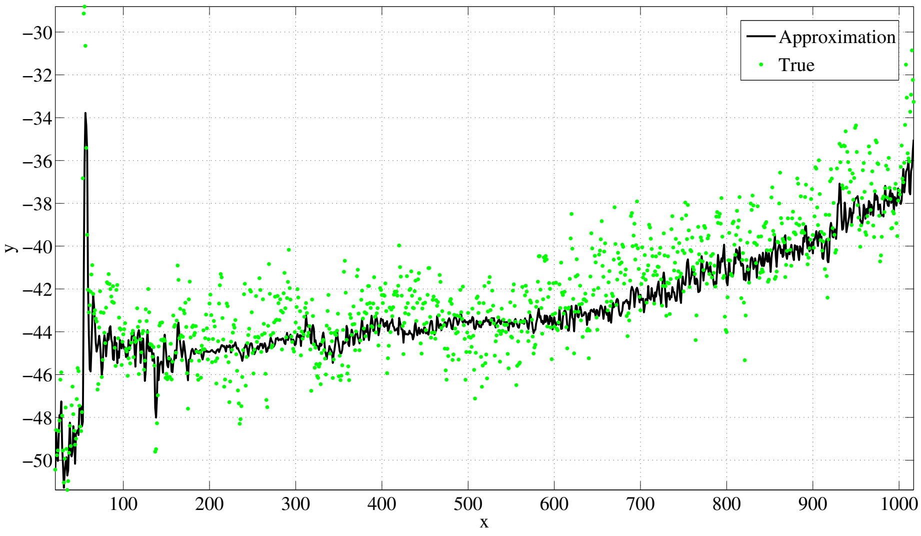


Figure 11: Real data and prediction on the training dataset.

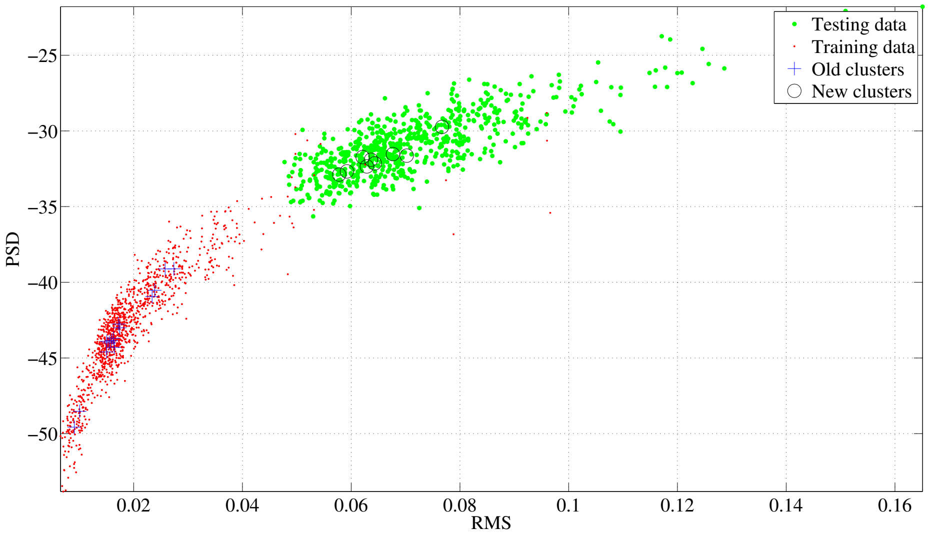


Figure 12: Illustration of high variability: training and testing data.

online: 1) the on-line clustering of the data describing the system to determine the different operating regions, 2) the creation, adaptation or removing of models locally computed for each cluster, and 3) the prediction of the future evolution.

Experiments were conducted on two benchmark datasets and one real-world problem on the PRONOSTIA platform. The results demonstrated the ability of the proposed method for an online segmentation of multi-dimensional time-series and reliable predictions. A comparison with other approaches was also made and emphasized the great interest of using belief functions. In particular, the number of rules is decreased compared to usual approaches, while ensuring limited error by using the concept of virtual centroids to represent transitions between operating regions.

Acknowledgement

This work is partially supported by a PEPS-INSIS-2011-2012 grant from the French National Center for Scientific Research (CNRS) under the administrative authority of the French Ministry of Research. We also thank Dr. Rafael Gouriveau (from FEMTO-ST) for his precious help on evolving systems, as well as the anonymous reviewers for their helpful comments that greatly improved the paper.

References

[1] P. Angelov, D. Filev, N. Kasabov, Evolving intelligent systems: Methodology and applications, IEEE Press Series on Computational Intelligence, John Wiley, New york, 2010.

[2] K. Madani, M. Rybnik, A. Chebira, Non linear process identification using a neural network based multiple models generator, LNCS series (2003) 647 – 654.

[3] A. Boukhris, G. Mourot, J. Ragot, Nonlinear dynamic system identification: a multiple-model approach, International Journal of Control 72 (2000) 591 – 604.

[4] R. Murray-Smith, Multiple model approaches to modeling and control, Taylor & Francis Publishers, 1997.

[5] O. Nelles, On the identification with neural networs as series-parallel and parallel models, in: International Conference on ANN, Paris, France, 1995.

[6] K. Madani, L. Thiaw, Self-organizing multi-modeling: A different way to design intelligent predictors, Neurocomputing 70 (2007) 2836 – 2852.

[7] G. J.Klir, Uncertainty and information. Foundations of generalized information theory, Wiley, New York, 2006.

[8] G. Shafer, A mathematical theory of evidence, Princeton University Press, 1976.

[9] M. Chandrashekhar, R. Ganguli, Uncertainty handling in structural damage detection using fuzzy logic and probabilistic simulation, Mechanical Systems and Signal Processing 23 (2) (2009) 384 – 404.

[10] T. Haag, J. Herrmann, M. Hanss, Identification procedure for epistemic uncertainties using inverse fuzzy arithmetic, Mechanical Systems and Signal Processing 24 (7) (2010) 2021 – 2034.

[11] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Trans. on Systems, Man, and Cybernetics 15 (1985) 116 – 132.

[12] P. Angelov, E. Lughofer, X. Zhou, Evolving fuzzy classifiers using different model architectures, Fuzzy Sets and Systems (2008) 3160 – 3182.

[13] M. El-Koujok, R. Gouriveau, N. Zerhouni, Reducing arbitrary choices in model building for prognostics: An approach by applying parsimony principle on an evolving neuro-fuzzy system, Microelectronics Reliability 51 (2011) 310 – 330.

[14] P. Angelov, D. Filev, An approach to online identification of Takagi-Sugeno fuzzy models, IEEE Trans. on Systems, Man, and Cybernetics 34 (2004) 484 – 98.

[15] M. Ramdani, G. Mourot, J. Ragot, A multi-modeling strategy based on belief function theory, in: CDC-ECC 05, 2005.

[16] L. Serir, E. Ramasso, N. Zerhouni, Evidential evolving Gustafson-Kessel algorithm for online data streams partitioning using belief function theory, International Journal of Approximate Reasoning 53 (2012) 747 – 768.

[17] P. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (1994) 191 – 234.

[18] M.-H. Masson, T. Denoeux, ECM: An evidential version of the fuzzy c-means algorithm, Pattern Recognition 41(4) (2008) 1384 – 1397.

[19] B. R. Cobb, P. P. Shenoy, On the plausibility transformation method for translating belief function models to probability models, International Journal of Approximate Reasoning 41 (2006) 314 – 330.

[20] D. Dubois, H. Prade, P. Smets, New semantics for quantitative possibility theory, in: 2nd International Symposium on Imprecise Probabilities and Their Applications, Ithaca, New York, 2001.

[21] O. Georgieva, D. Filev, Gustafson-Kessel algorithm for evolving data stream clustering, in: International Conference on Computer Systems and Technologies - CompSysTech 09, 2009.

[22] M. Grabisch, K-order additive discrete fuzzy measures and their representation, Fuzzy Sets and Systems 92 (1997) 167 – 189.

[23] N. Kasabov, Q. Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Trans. on Fuzzy Systems 10 (2002) 144 – 154.

[24] M. Watts, A decade of Kasabovs evolving connectionist systems: A review, IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews 39 (2009) 253 – 269.

[25] D. Deng, N. Kasabov, ESOM: An algorithm to evolve self-organizing maps from on-line data streams, in: International Joint Conference on Neural Networks, Vol. 6, Piscataway, NJ, 2000, pp. 3 – 8.

[26] N. Kasabov, Evolving fuzzy neural networks - algorithms, applications and biological motivation, Neural Networks (1998) 271–274.

[27] J. Platt, A resource allocation network for function interpolation, Neural Computation 3 (1991) 213 – 225.

[28] L. Serir, E. Ramasso, P. Nectoux, O. Bauer, N. Zerhouni, Evidential Evolving Gustafson-Kessel algorithm (E2GK) and its application to PRONOSTIA's data streams partitioning, in: IEEE Int. Conf. on Decision and Control, 2011, pp. 8273–8278.