ELSEVIER

Computers and Chemical Engineering 26 (2002) 1735–1754

Computers & Chemical Engineering

www.elsevier.com/locate/compchemeng

Neural virtual sensor for the inferential prediction of product quality from process variables

R. Rallo [a], J. Ferre-Giné [a], A. Arenas [a], Francesc Giralt [b,*]

[a] *Departament d'Enginyeria Informàtica i Matemàtiques, Escola Tècnica Superior d'Enginyeria, Universitat Rovira i Virgili, Avinguda dels Països Catalans, 26, Campus Sescelades, 43007 Tarragona, Catalunya, Spain*

[b] *Departament d'Enginyeria Química, Escola Tècnica Superior d'Enginyeria Química (ETSEQ), Universitat Rovira i Virgili, Avinguda dels Països Catalans, 26, Campus Sescelades, 43007 Tarragona, Catalunya, Spain*

Received 20 April 2000; received in revised form 5 July 2002; accepted 5 July 2002

Abstract

A predictive Fuzzy ARTMAP neural system and two hybrid networks, each combining a dynamic unsupervised classifier with a different kind of supervised mechanism, were applied to develop virtual sensor systems capable of inferring the properties of manufactured products from real process variables. A new method to construct dynamically the unsupervised layer was developed. A sensitivity analysis was carried out by means of self-organizing maps to select the most relevant process features and to reduce the number of input variables into the model. The prediction of the melt index (MI) or quality of six different LDPE grades produced in a tubular reactor was taken as a case study. The MI inferred from the most relevant process variables measured at the beginning of the process cycle deviated 5% from on-line MI values for single grade neural sensors and 7% for composite neural models valid for all grades simultaneously.

Keywords: Inferential prediction; Virtual sensors; Selection of variables; Radial basis function; Kohonen neural networks; Fuzzy ARTMAP; Polyethylene

1. Introduction

Neural network systems have been widely used to model and control dynamic processes because of their extremely powerful adaptive capabilities in response to nonlinear behaviors (Barto, Sutton, & Anderson, 1983). This adaptability is the result of the architectures themselves (parallel and/or with classification capabilities) and of the learning algorithms used, which are biologically inspired and incorporate some aspects of the organization and functionality of brain neuron cells (Hertz, Krogh, & Palmer, 1991). As a result, neural systems can mimic high-level cognitive tasks present in human behavior, and operate in ill-defined and time-varying environments with a minimum amount of human intervention. For example, they are capable of (i) learning from the interaction with the environment, without restrictions to capture any kind of functional relationship between information patterns if enough training information is provided, (ii) generalizing the learned information to similar situations never seen before, and (iii) possessing a good degree of fault tolerance, mainly due to their intrinsic massive parallel layout. These properties make neural computing appealing to many fields of engineering.

The application of neural systems is especially interesting to control and to optimize chemical plants (Hunt, Sbarbaro, Zbikowski, & Gawthrop, 1992) since the kind of time-dependent problems dealt with in process engineering are highly non-linear and, thus, it is difficult to obtain detailed predictions from first principle models in real time. A specific area of intrinsic interest to chemical manufacturing processes is the prediction of the quality of final products. This is even more vital in cases where it is difficult to implement reliable and fast on-line analyzers to measure relevant product properties

* Corresponding author. Tel.: +34-977-559-638; fax: +34-977-558-205

E-mail address: fgiralt@etseq.urv.es (F. Giralt).

0098-1354/02/$ - see front matter
PII: S0098-1354(02)00148-5

and to establish appropriate control strategies for production. Such situations can lead to a significant production of off-grades, especially during the on-line operations involved to change product specifications. An alternative is to develop on-line estimators of product quality based on available process information. One of the most powerful and increasingly used methodologies is the inferential measurement (Martin, 1997). This method consists in the forecast of product quality or of difficult to measure process indicators from other more reliable or easily performed plant measurements, such as pressures, flow rates, concentrations or temperatures.

The purpose of the current study is to develop a virtual sensor to infer product quality from other more easily measured process variables using several adaptive neural network architectures as well as different techniques for data preprocessing, including the selection of variables and the construction of appropriate training and test sets. The networks that have been considered are a predictive Fuzzy ARTMAP neural classifier, and a hybrid network that combines a new strategy to construct dynamic unsupervised classifiers with a supervised predictor. The neural virtual sensors (soft-sensors) developed have been applied to infer the Melt Index (MI) of different low-density polyethylene (LDPE) grades measured on-line in operating plants. The paper is organized as follows. In Section 2 all aspects concerning the design and implementation of virtual sensors systems are described, highlighting several techniques for data preprocessing, their basic architectures and proposed modifications, learning algorithms and major drawbacks. The performance of sensors has been illustrated and evaluated with a case study of LDPE quality inference. The results obtained are then discussed and concluding remarks about the design and implementation of virtual sensors systems are finally presented.

2. Neural virtual sensor

A virtual sensor is a conceptual device whose output or inferred variable can be modeled in terms of other parameters relevant to the same process. The software (sensor) should be conceived at the highest cognitive level of abstraction so that a sufficiently accurate characterization of the total system behavior could be attained in terms of errors between the validated or measured data and the predicted outputs. Artificial neural networks are an adequate choice because, in addition to the above, they can improve performance with time, i.e. are capable of learning real cause–effect relations between sensor's stimulus and its response when historical databases of the whole process are used for training. Fig. 1 shows a generic virtual sensor implementation within a manufacturing process. It can receive real-time readings of several process variables as well as feedback signals of downstream on-line analyzers for the target property. Both sets of data are needed for training the virtual sensor. Once trained, this virtual device uses only real time measurements of the selected process variables obtained by process sensors at certain times to infer the value of the product target property. The output can be redirected as information to the plant operator or to the control system to maintain optimal plant operation for a given product quality.

In the current study the aim is the design of a generic model for a virtual sensor able to predict the quality of a product based in the state of the process plant when the production cycle begins. The virtual sensor thus acts as an inferential measurement system that anticipates the downstream target variable as a function of other process variables measured when production starts. The dynamic characteristics of chemical plants and the level of accuracy for the output usually impose certain requirements to the neural system: (i) adaptive in architecture so that more processing units could be added during the training process when needed; (ii) robust under uncertainty of the input data due to noise or product grade transitions; and (iii) re-trainable if a new mode of operation never seen before occurs.

3. Data preprocessing

The inferential measurement systems based on neural networks are mostly developed using "data-based" methodologies, i.e. the models used to infer the value of target variables are developed using 'live' plant data measured from the process plant. This implies that inferential systems are heavily influenced by the quality of the data used to develop their internal models. Consequently, the first step to build an inferential measurement system is the preprocessing of data. During this preprocessing stage two kinds of actions are performed, first data cleaning and conditioning, and second the selection of the most relevant information to develop the model.

It is important to consider that usually the data acquired from field sensors are noisy and contain erroneous or spurious values. To condition these data all the faulty values must be discarded using filtering techniques like low-pass filters. Moreover, as the values are heterogeneous, they must be scaled into a range suitable for data processing. In addition, in complex industrial processes, such as chemical processing plants, the number of plant variables that can be measured is very large and the sampling rates used for these measurements are high. This implies the generation of large datasets containing a large amount of features. In those situations it is very useful to have an "intelligent

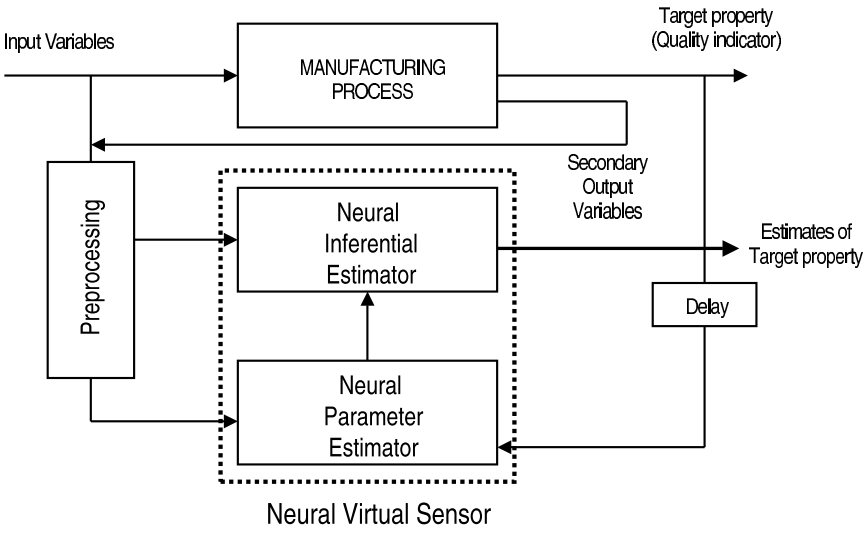

Fig. 1. Sketch of a generic virtual sensor implementation.

system'' capable of selecting the most relevant features and examples among all the available information with the purpose of optimizing the resources needed to build an accurate and reliable model for the process under consideration.

In the process of building a neural inferential measurement system a reduction in the dimension of the input space would simplify the input layer of the neural architecture and reduce the time needed for training. Additionally, if the class to which a certain input pattern belongs is known it should be possible to figure out the features that best discriminate between the different values of the target properties. The present study adapts the technique proposed by Espinosa, Arenas, and Giralt (2001b) to perform the selection of relevant variables to the case where the target variable (MI) fluctuates around a mean value. The discrimination of the training datasets from the complete set of plant measurements has been carried out following the work of Espinosa, Yaffe, Arenas, Cohen, and Giralt (2001a) to include all relevant information.

3.1. Selection of variables

A sensitivity analysis to reduce the dimension of the input space could be performed by using either projection techniques, which imply the definition on new combined variables, or by reducing the number of input variables while preserving the most relevant features of the complete input space. The first approach typically uses statistical analysis (descriptive statistics, cross-correlations, factorial analysis, PCA, etc.) to find relations among all variables and to select new representative prototypes from the subsets of related variable combinations. This process entails the projection of the input space into a lower dimension output space without the loss of significant information. The main drawback of this approach is that the newly created variables are difficult to interpret in terms of the primitive process variables. Also, orthogonal decomposition may not be the best approach to reduce the dimension of the input space, as has been found in pattern recognition problems related to structure identification in turbulent flows (Ferre and Giralt, 1993).

The method proposed here projects all subsets of the input space, with process variables (v_i) ordered according to relevance and each subset complemented with the target variable (P), onto the space generated by a self-organizing map (SOM) (Kohonen, 1990), which is a topology-preserving clustering process. The comparison of each map with the rest maps by means of a dissimilarity measure (see Appendix A and Kaski & Lagus, 1997) informs about the relevance of each combination of input variables in relation to the target variable. The key of this selection process resides in the similarity measure used and in the strategy to build the subsets of variables from the original input space.

The first step in the process of best feature selection was to define a strategy to populate the set of ordered variables, S from the set of all N input variables, $V = \{v_1, v_2, \ldots, v_N\}$. This procedure can be formalized either following the selection (ordering) criterion proposed by Espinosa et al. (2001b) to develop Quantitative Structure Property/Activity Relationships (QSPR/QSAR) models or by the (absolute) value of the correlation of each input variable with the target or quality property, $|\mathrm{corr}(v_i, P)|$. The latter has been adopted here since the current problem is simpler and does not require an exact classification of plant events according to a prefixed knowledge, as it is the case in QSPR/QSAR models where similar chemicals should be classified into similar classes. Also, P values and plant operating conditions should oscillate around their prefixed mean values in a properly operating chemical plant and correlation

information between these fluctuating signals alone should be sufficient to order input variables. Using this correlation approach all variables are ranked by correlation and the input set V were reordered so that, $S = \{s_1, \ldots, s_N\}$, where N is the cardinality of S and $\text{corr}(s_i, P) \geq \text{corr}(s_{i+1}, P)\ \forall\ i$: 1, ..., $N-1$.

Let us now consider the selection of the most adequate subset of input variables from S, i.e. the minimum set containing all relevant information. SOMs were determined for all N subsets that can be generated from S by successively adding to the first subset formed by the first element and the target variable, $S_1 = \{s_1, P\}$ the rest of ordered variables, until the last subset $S_N = \{s_1, \ldots, s_N, P\}$ was formed. The corresponding SOMs for all $S_i\ \forall\ i$: 1, ..., $N-1$ were trained and the dissimilarity D(L,M) between every pair of maps (L,M) was computed using Eq. (A3) in Appendix A. The average dissimilarity with all the rest of the possible variable configurations was computed and the S_i yielding the minimum average dissimilarity was selected as the set of minimum dimension containing the most relevant features of the input space This selected set can be considered as the subset of variables most similar, in terms of information contents about the target property, to all other possible ordered subsets formed from the input variables.

3.2. Training set

As some variables are more relevant than others, also some input patterns of data may be more unique than others and should be considered for training. The appropriate selection of patterns for the training is one of the most important tasks in machine learning. Different strategies can be used for the selection of the most suitable training set of data among all the available process information. One method to construct the training set consists in the selection of data from the time series of recorded plant data while keeping the remaining data sequence to test the performance of the sensor. This facilitates the representation of data and the evaluation as if the sensor was operating under real plant conditions, predicting the target property sequentially in time. This selection procedure does not assure, however, that all significant, singular or redundant, information was used for sensor development during training.

An alternative is to select the most suitable training and test sets from the complete pool of available patterns contained in the time-records, independently of their position in the time-sequences. This assures that all significant, singular or redundant, information is presented to the sensor during the learning stage and that testing is performed with data that, while not previously seen by the sensor, has similar characteristics to those belonging to the classes or categories used for training. This requires the application of pre-classification algorithms such as fuzzy ART, as suggested by Espinosa et al. (2001a), or the dynamic clustering procedure outlined in Section 4. Patterns located in regions of the process state space with low "population density" are labeled as candidates for training and those located in high density regions as candidates for testing. Test sets are finally built by randomly selecting a certain number of patterns labeled as test candidates.

The pre-screening option chosen in the present study is the application of the algorithmic method proposed by Tambourini and Davoli (1994) because with small training sets it assures good network generalization capabilities. The key idea is to include as learning examples those that exhibit classification difficulties during training. The hypothesis made is that these patterns contain most of the important classification information about the problem under study. They presumably are the patterns that best represent the different problem classes and have to be included in the training set to supplying the network with good examples. If the network learns successfully these training set patterns, it could be expected to perform well with unknown patterns, since it has already learned most of the characteristics that distinguish the problem classes.

4. Neural architectures

Three neural models have been developed and evaluated to build the virtual sensor. One is based on a predictive Fuzzy ARTMAP architecture that has been capable of learning the dynamics of large-scale structures in a turbulent flow (Giralt, Arenas, Ferre-Giné, Rallo, & Kopp, 2000) and to develop successful QSPR and QSAR models for the prediction of physicochemical properties and biological activities (Espinosa et al., 2001a; Espinosa et al., 2001b). The other two sensors are based on two supervised algorithms that connect a Radial Basis Function (RBF) layer with the required output. The RBF layer has been constructed following a new procedure that allows dynamic network growing during the training stage. Outputs are either the average target property value for all the input patterns belonging to the cluster that is activated or the target property value that results from the incorporation of RBFs at the cluster centers.

4.1. Fuzzy ARTMAP

The Fuzzy ARTMAP neural network is formed by a pair of fuzzy ART modules, Art_a and Art_b, linked by an associative memory and an internal controller (Carpenter, Grossberg, Marcuzon, Reinolds, & Rosen, 1992), as shown in Fig. 2. The Fuzzy ART architecture

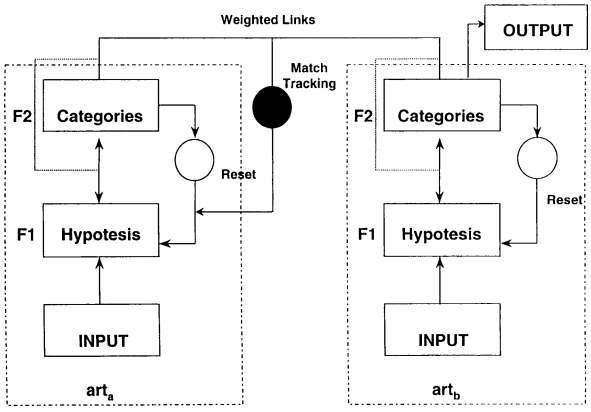

Fig. 2. Modified fuzzy ARTMAP neural network architecture.

was designed by Carpenter, Grossberg, and Rosen (1991) as a classifier for multidimensional data clustering based on a set of features. The elements of the set of n-dimensional data vectors $\{\xi^1, \ldots, \xi^p\}$, where p is the number of vectors to be classified, must be interpreted as a pattern of values showing the extent to which each feature is present. Every pattern must be normalized to satisfy the following conditions:

$$\xi^i \in [0,\ 1]^n$$
$$\sum_{j=1}^{n} \xi_j^i = k \qquad \forall \quad i = 1,\ \ldots,\ p \tag{1}$$

The classification procedure of fuzzy ART is based on *Fuzzy Set Theory* (Zadeh, 1965).

The similarity between two vectors can be established by the grade of the membership function, which for two sets (l, m) of generic vectors can be easily calculated as

$$\text{grade}(\xi^1 \subset \xi^m) = \frac{|\xi^l \wedge \xi^m|}{|\xi^l|} \tag{2}$$

In Eq. (3) the fuzzy AND operator $\wedge$ is defined by,

$$\wedge: \begin{array}{c} [0,\ 1]^n \times [0,\ 1]^n \to [0,\ 1]^n \\ (\xi^l,\ \xi^m) \to \xi^i \end{array} \tag{3}$$

The components of the image vector that results from this application (3) are

$$\xi_j^i = \min(\xi_j^l,\ \xi_j^m) \quad \forall\ j = 1,\ \ldots,\ n \tag{4}$$

The norm $|\cdot|$ in Eq. (3) is the sum of the components of the vector given by Eq. (4).

The classification algorithm clusters the data into groups or classes with a value for the grade of membership in Eq. (2) greater than the *vigilance parameter* ρ. The value of ρ controls the granularity of the classes and allows the implementation of a desired accuracy criterion in the classification procedure. A weight vector ω^μ represents each class μ. The procedure starts by creating the first class from the first pattern presented to the network,

$$\omega^1 = \xi^1 \tag{5}$$

The rest of input patterns ξ^i $(i = 2, \ldots, p)$ are then presented to the network and if the similarity of ξ^i with any established class μ is greater than ρ then ξ^i is classified into this class, and the representative of this class is updated according to

$$\omega_{\text{new}}^\mu = \omega_{\text{old}}^\mu \wedge \xi^i \tag{6}$$

Otherwise a new class represented by ξ^i is created. Eq. (6) is the learning rule of the net. The mechanisms to speed up the process and to conduct the classification properly can be found elsewhere (Carpenter et al., 1991).

The dynamics of Fuzzy ARTMAP are essentially the same as two separate Fuzzy ART networks, each one working with a part of the training information. The first part could be interpreted as the vector input pattern and the second one as the desired classification output (supervisor). The associative memory records the link between the classes corresponding to the input pattern and the desired classification. The internal controller supervises if a new link is in contradiction with any other previously recorded. If no contradiction is found, the link is recorded; otherwise the pattern is re-classified with a larger vigilance parameter. Once the network has been trained it can be used to classify input vectors without any additional information.

The Fuzzy ARTMAP architecture, which has been successfully applied to educe the different classes of large-scale events present in free turbulence (Ferre-Giné, Rallo, Arenas, & Giralt, 1996), was designed to classify data and, thus, cannot generate an output pattern after the training stage. To implement predictive capabilities, the categories educed by the system from the learned information are linked to the desired outputs, as depicted in Fig. 2. This is mathematically equivalent to defining an application from the space of categories to that of output patterns, the image of the application being defined by examples of patterns provided to the neural system in a supervised manner. The accuracy of the procedure increases asymptotically towards a constant value with the number of examples used for training, i.e. when the space of outputs is accurately mapped. In the predictive mode, only the category layer of Art_b in Fig. 2 is active and linked to Art _a to provide an output for each input vector presented to this module (Giralt et al., 2000).

4.2. Dynamic unsupervised layers

The success of predictive fuzzy ARTMAP in difficult forecasting problems, together with its limitations when interpreting information with underlying periodicity (Carpenter et al., 1992) and limited training, has motivated the search for other potentially suitable systems for sensor development, such as dynamic

unsupervised layers. One of the challenges related with the application of this neural system is the design of a node generation mechanism and of a clustering process that are simple enough and simultaneously compatible with different supervised output generation procedures.

4.2.1. Node generation

The current approach to construct the unsupervised RBF layer is performed in four steps: (i) set-up an initial configuration with the center of the first cluster formed by one pattern chosen randomly from the training dataset; (ii) determine the minimal mean distance between patterns; (iii) use this distance as the constant maximum attention radius $d_{\max}$ to control the generation of new nodes over the node-generation process; and (iv) present a new input pattern, ξ, to the network and compute the Euclidean distance between the pattern and all nodes, and adapt the structure of the network according to the following two rules. If the input pattern is located inside the region of influence of any node i, the pattern is classified and the center of this node is adapted using a winner takes-all approach based on Kohonen's learning rule (a similar approach using k-means was used by Hwang and Bang (1997) to speed up the training of RBF layers),

$$c_i(n+1) = c_i(n) + \alpha(n)[\xi - c_i(n)] \tag{7}$$

with n denoting the training epoch, $c_i(n)$ the center vector of the selected node, and $\alpha(n)$ a monotonically decreasing learning rate computed as $\alpha(n) = \alpha_0 1/\sqrt{n}$, which controls the adaptation or upgrading of the center of the cluster. Otherwise, if the input pattern is located outside the region of influence of all the nodes, a new node is created with the center located at the point that defines the input pattern. The procedure is repeated until the number of nodes stabilizes and either the classification or the number of iterations reaches a predetermined minimum or maximum value, respectively.

The current algorithm tends to create an appropriate number of clusters since it determines the attention radius based on the distribution of the training patterns. This allows the construction of the minimal clustering necessary to achieve a good classification in accordance with the attention radius chosen. The process of complying with a given classification accuracy has to be tested for generalization by trial and error, which is a customary practice when working with neural systems.

4.2.2. Output generation

Once the classifier is built it is necessary to effect an output from the unsupervised layer. This procedure could consist in a hybrid approach that combines the current dynamic unsupervised classifier with a supervised learning engine. In this work two techniques for producing this output are presented. The first is a *clustering average* based on the labeling of the unsupervised layer using the values of the target variable. One of the most common labeling processes consists in averaging the target value for each of the training patterns belonging to a given cluster, like in the k-means algorithm. This averaged value is subsequently the output of the network. This labeling algorithm can be summarized as follows: (i) Obtain a pattern from the training set; (ii) compute the winner node; (iii) add its value to the output value of the winner node; (iv) increase the pattern counter for this node; (v) repeat (i) until all patterns have been processed; (vi) compute the average output value for each node. Once the dynamic unsupervised layer is labeled the network is ready to infer the target property values, i.e. act as a virtual sensor.

The second technique used in the current study to obtain an output from the unsupervised layer consists in the placement of RBF over the cluster centers with supervised training to adjust the output. This neural network is herein after identified as *Dynamic Radial Basis Function network (DYNARBF)*. RBF neural networks allow the parametrization of any function $f(x)$ as a linear combination of non-linear basis functions (Powell, 1987, 1992; Broomhead & Lowe, 1988; Lee & Kil, 1988; Moody & Darken, 1989; Poggio & Girosi, 1990a,b; Musavi, Ahmed, Chan, Faris, & Hummels, 1992),

$$f(x) = p + \sum q_j G(\|x - x_j\|) \tag{8}$$

where j is the function index, the norm $|\cdot|$ is the Euclidean distance (Park & Sandberg, 1991), x_j are the centers of the proposed basis functions, p and q are adjustable parameters and G is a radial kernel function. In the current model a Gaussian activation function is used,

$$G(r_j) = \exp(-r_j^2/2\sigma_j) \tag{9}$$

where r_j is the Euclidean distance to the center of the j-class and σ_j is the dispersion of the Gaussian. For each activation function (9) the center position x_j and its width σ_j must be determined to define a receptive field around the node. Both can be determined by an unsupervised learning process, in which data are clustered into "classes" or nodes. The idea is to pave the input space (or the part of it where the input vectors lie) with the receptive field of these nodes (Gaussians in this case).

A map between the RBFs outputs and the desired process outputs is then constructed in a second supervised training stage. It should be noted that the RBF neural network needs some a-priori hypothesis concerning the number of nodes that will be used for any particular problem. This is a major drawback in the application of RBFs because the approximation error is

highly dependent on the number of nodes. Usually, more nodes imply more accuracy in the mapping of the predicted target values over the training dataset. Nevertheless, "over-fitting" during training could in some cases imply a loss of network generalization capabilities during testing. There is no a priori methodology to estimate rigorously the number of nodes for optimal generalization. These issues have been the subjects of research in the past (Platt, 1991; Fritzke, 1994a,b; Berthold & Diamond, 1995). These studies share the common strategy of extending on-line the structure of the neural network to reduce a given measure of the classification error. In this study the Growing Unsupervised Layer explained above is used to overcome this drawback.

4.2.3. Training procedure

The neural system is trained in two separate phases. First, the unsupervised layer that defines the number of radial functions in the hidden layer as well as their position in input space is constructed. The width of each radial function σ_j is usually calculated ad hoc as the mean Euclidean distance between the k-nearest neighbors of node j,

$$\sigma_j = \sigma_0 \sum_{i=1}^{K} d(x_j,\ x_i) \tag{10}$$

The center of the i-Gaussian is x_i, σ_0 is a constant for width scaling, K is the cardinality of the neighborhood and d the Euclidean distance.

The activation of the functions once placed over the hidden layer is accomplished by

$$\begin{cases} f_j(\xi) = \exp\left(\dfrac{-d^2(\xi,\ x_j)}{2\sigma_j^2}\right) \\ A_j(\xi) = \dfrac{f_j(\xi)}{\sum_{k=1}^{N} f_k(\xi)} \end{cases} \tag{11}$$

where ξ is the pattern presented to the network, A_j the activation of node j and N the total number of gaussians.

In a second supervised learning stage the activation of the RBF layer for a given input pattern ξ is related to the desired output $\tilde{\theta}$. First, the output of the network is computed as,

$$\theta_i = \sum_{j=1}^{N} w_{ij} A_j(\xi) \tag{12}$$

Then the weights are updated using the common Delta rule minimization error procedure,

$$\Delta w_{ij} = \beta(\theta_i - \tilde{\theta}_i) A_j(\xi) \tag{13}$$

Here, the activation of node j is the value of the Gaussian function when the input i is presented with a constant learning rate β.

The combination of Eqs. (11) and (12), jointly with the information contained at the centers (x_i) and in the widths (σ_i) of the RBFs, and with the weights connecting the activation of each radial function with the output node (w_i), yields a analytical model relating the target property with the parameters of the neural network:

$$P = \sum_{i=1}^{N} w_i \frac{\exp(-d^2(\xi,\ x_i)/2\sigma_i^2)}{\sum_{j=1}^{N} \exp(-d^2(\xi,\ x_j)/2\sigma_j^2)} \tag{14}$$

In this equation N is the number of RBF nodes, d the Euclidean distance and ξ is the input vector of the process variables presented to the network

5. Case study: virtual sensor for melt index in LDPE process plant

5.1. Problem statement

The polymerization of ethylene to produce LDPE is usually carried out in tubular reactors 800–1500 m long and 20–50 mm in diameter at pressures of 600–3000 atmospheres (Chan, Gloor, & Hamielec, 1993; Lines et al., 1993). The quality of the polymer produced is determined essentially by the MI, which is measured by the flow rate of polymer through a die. The on-line measurement of this quantity is difficult and requires close human intervention because the extrusion die often fools and blocks. As a result, in most plants the MI is evaluated off-line with an analytical procedure that takes between 2 and 4 h to complete in the laboratory, leaving the process without any real-time quality indicator during this period. Consequently, a model for estimating the MI on-line would be very useful both as an on-line sensor and as a forecasting system. In addition, it would allow the supervision of the overall process and to avoid any mismatch of product quality during product grade transitions. However, a model derived from first principles capable of continuously predicting accurate MI values for any LDPE process in real time is still non-existent. Instead, some production plants use data based, linear and non-linear correlation models to overcome this deficiency.

Fig. 3 summarizes the main characteristics of the LDPE plants studied. Several sets of real data corresponding to several production cycles are analyzed here. The MI values used here were determined every 10 min with on-line sensors that were calibrated by off-line determinations. The error associated with these on-line MI measurements is $\pm 2\%$. Changes in MI correspond to changes in the physical or chemical characteristics of the desired product (grade transitions). The six LDPE product grades cluster into three families according to

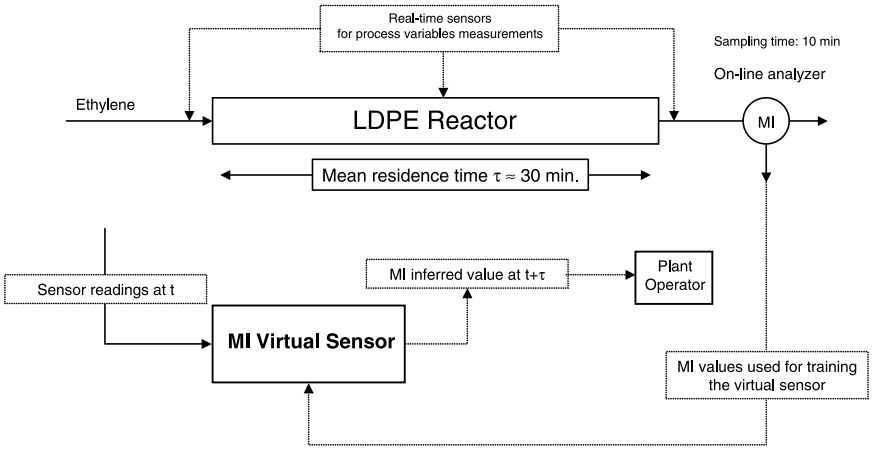

Fig. 3. LDPE plant diagram with typical time scales.

their average MI values and polymer densities, as shown in Fig. 4. Each of these families contains two grades. Also each cluster has a different population density, with the highest one corresponding to the lowest value of MI.

A pool of process information, formed by the 25 process variables (pressures, flow rates, temperatures of the cooling/heating streams of the reactor, etc.) listed in alphabetical order in Table 1, has been chosen to develop the virtual sensors. This table also includes the correlation of each variable with the MI for the six grades produced. The characterization and the prediction of the time-variation of MI (last variable listed in Table 1) has been approached with both the complete set of 25 variables and the reduced sets for all grades, following the procedure explained in a previous section. In the case of composite models applicable to all families simultaneously, these process variables have been complemented with a normalized label to identify grades and facilitate product transitions during production cycles. All the time records of the 25 variables have been acquired by field measurement instruments, and sent to the control computer of the plant for their processing. This computer receives voltage signals that are converted into fix point numeric data within a certain range.

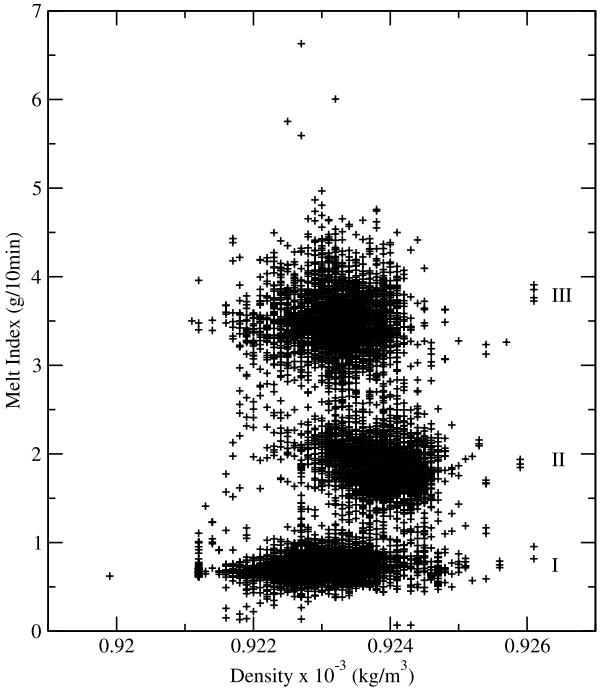

Fig. 4. Families of LDPE identified by the variation of MI with polymer density.

Table 1
Process variables and correlations with the MI for all the LDPE grades produced.

Variable name	Units	$\|R\|$ with MI
Compressor throughput	Tm/h	0.044
Concentration 1	%	0.527
Concentration 2	%	0.249
Concentration 3	%	0.168
Concentration 4	%	0.018
Density	g/cm^3	0.183
Extruder power consumption	A	0.583
Extruder speed	rpm	0.056
Flow rate 1	kg/h	0.021
Flow rate 2	kg/h	0.042
Flow rate 3	kg/h	0.595
Flow rate 4	kg/h	0.626
Level	%	0.126
MI	g/10min	1.000
Pressure	Kg/cm^2	0.052
Temperature 1	°C	0.305
Temperature 2	°C	0.023
Temperature 3	°C	0.522
Temperature 4	°C	0.115
Temperature 5	°C	0.122
Temperature 6	°C	0.136
Temperature 7	°C	0.428
Temperature 8	°C	0.446
Temperature 9	°C	0.112
Volumetric flow rate 1	l/h	0.324
Volumetric flow rate 2	l/h	0.518

The data presented in the following analysis correspond to time intervals of 10 min. The virtual sensors anticipate the MI values from process variables measured when the production cycle begins. The duration of cycles is determined by the mean residence time inside the reactor ($\tau \approx 30$ min). This choice is independently confirmed by the spectral analysis of plant data for all grades, which show an underlying periodicity around 1.1 residence time τ units.

5.2. Virtual sensor implementation

Fig. 3 depicts the current sensor implementation within the LDPE process flow sheet. It can receive real-time readings of process variables as well as feedback signals of downstream on-line analyzers; both sets of data were used for training (and later adapting) the virtual sensor. Once trained, this virtual device uses only real time measurements of the selected process variables made by process sensors at any time to infer the value of the product target property when leaving the reactor. The output can be redirected as information to the plant operator or to the control system to maintain optimal plant operation for a given product quality.

The aim of the current study is to implement a virtual sensor to predict the quality of LDPE, i.e. the target variable $P = \mathrm{MI}$, based in the state of the plant. The virtual sensor behaves as a black-box model that relates the MI of produced LDPE to other process variables measured when the corresponding production cycle began. It is important to remark that the current approach is not a time-series analysis, since it is time independent. The functionality between the output and the input variables is of the form $\mathrm{MI}(t+\tau) = \Phi[v_1(t), \ldots, v_n(t)]$, where each $v_i(t)$ is any of the process variables listed in Table 1, excluding MI, measured at time t. Nevertheless, since field measurements are available as a time series the current neural sensors have also been developed and operated according to production time-sequences and cycles.

The three types of neural architectures introduced in the previous sections have been used to develop the virtual sensor. In addition a fourth linear model has been used as a reference for evaluation purposes. This linear model is formulated in terms of normalized variables as,

$$\mathrm{MI} = a_0 + a_1 v_1 + \ldots + a_N v_N + a_{N+1} v_{N+1} \quad (15)$$

In this equation $a_j \ \forall \ j$: 0, …, $N+1$ are adjustable coefficients and $v_i \ \forall \ i$: 1, …, N are each of the $N = 25$ normalized process variables listed in Table 1 and measured simultaneously at the beginning of the production cycle. The last variable v_{N+1} in Eq. (15) correspond to a label identifying the product grades, which is used only for the composite model and its normalized value identifies product quality and grade. The values of the coefficients a_j in Eq. (15) are not reported here for brevity. Results showed that high or low values of these coefficients did not necessarily correspond with high or low correlations of variables with MI.

Two types of virtual sensors were developed for each neural system considered: single models for every product grade and composite models for all LDPE grades jointly. Each kind of model was trained and tested with datasets defined according to the sequential and pre-classification procedures described in previous sections, so that the performance of each sensor could be assessed as independently as possible of the limited number of training patterns available. Furthermore, the original (complete) and reduced sets of variables have been used in order to investigate the sensitivity of the input pattern dimension in sensor's performance. On the other hand, the composite sensors, applicable to all grades simultaneously, should provide a good estimate of the benefits derived from the classification capabilities of all the neural systems considered. Table 2 includes the most relevant information concerning the training and test sets used in the current study to develop sensors from the complete input dataset of 25 process variables (upper part of Table 2) and from the reduced sets selected by the sensitivity analysis using SOMs (lower part of Table 2). From the point of view of the MI, the two sets of data used for training and testing each kind of neural sensor, characterized by the different input dimension, had similar characteristics, i.e. equivalent average MI values, and similar number of different and of single MI values in the data records. The parameters used to develop (train and test) the three current neural sensor models are summarized in Table 3.

5.3. Preprocessing of variables

The data used for training and testing the virtual sensor have been acquired from the historical logs recorded in a real LDPE plant for the 25 process variables and MI listed in Table 1. These variables are sufficient to capture the dynamics of the LDPE plant for the six different quality grades (MI) considered, grouped into three families of final product. This table also includes the absolute value of their correlation with MI. Data were filtered to discard abnormal situations and to improve the quality of the inference system. The input and output variables were normalized with respect to their maximum operation values for all the LDPE grades considered, so that $\forall$ MI, $v_i \in [0, 1]$.

Data from the time records of the process variables and MI were first separated into training and test sets as indicated in Table 2. The first selection procedure used was aimed at preserving the time-series structure of the recorded plant data for each type of LDPE analyzed. This sequential procedure consisted in separating from

Table 2
Characteristics of the datasets used for training and testing the four virtual sensors with the complete and the reduced set of input variables

Product families	Grade	# Patterns (test/train%)	MI avg		# Different MI values (%)		# Single MI values (%)	
			Sequential	Pre-classification	Sequential	Pre-classification	Sequential	Pre-classification
Training set for the complete process input information								
I	A	3043	0.74	0.74	321 (11)	321 (11)	99 (3)	99 (3)
	B	3823	0.73	0.72	284 (7)	283 (7)	82 (2)	81 (2)
II	C	640	1.76	1.75	250 (39)	277 (43)	104 (16)	115 (18)
	D	957	1.74	1.73	261 (27)	273 (29)	113 (12)	115 (12)
III	E	1445	2.00	2.00	371 (26)	383 (27)	150 (10)	152 (11)
	F	4295	3.46	3.45	721 (17)	730 (17)	243 (6)	241 (6)
I+II+III	A–F	14 203	1.80	1.80	1389 (10)	1406 (10)	292 (2)	287 (2)
Test set for the complete process input information								
I	A	100 (3.3)	0.73	0.72	69	68	46	50
	B	100 (2.6)	0.70	0.75	70	65	50	43
II	C	100 (15.6)	1.68	1.73	76	88	57	78
	D	100 (10.4)	1.73	1.72	80	74	62	53
III	E	100 (6.9)	1.94	1.99	84	84	68	71
	F	100 (2.3)	3.46	3.49	71	82	46	67
I+II+III	A–F	600 (4.2)	1.70	1.43	367 (61)	228 (38)	221 (37)	102 (17)
Training set for the reduced process input information								
I	A	3043	0.74	0.72	321 (11)	305 (10)	99 (3)	85 (3)
	B	3823	0.73	0.72	284 (7)	280 (7)	82 (2)	79 (2)
II	C	640	1.76	1.72	250 (39)	266 (41)	104 (16)	113 (18)
	D	957	1.74	1.72	261 (27)	258 (27)	113 (12)	104 (11)
III	E	1445	2.00	2.00	371 (26)	378 (26)	150 (10)	154 (11)
	F	4295	3.46	3.50	721 (17)	558 (13)	243 (6)	111 (2)
I+II+III	A–F	14 203	1.80	1.80	1389 (10)	1358 (9)	292 (2)	284 (2)
Test set for the reduced process input information								
I	A	100 (3.3)	0.73	0.72	69	52	46	34
	B	100 (2.6)	0.70	0.72	70	65	50	43
II	C	100 (15.6)	1.68	1.77	76	69	57	47
	D	100 (10.4)	1.73	1.72	80	46	62	23
III	E	100 (6.9)	1.94	2.00	84	77	68	58
	F	100 (2.3)	3.46	3.50	71	63	46	38
I+II+III	A–F	600 (4.2)	1.70	1.25	367 (61)	271 (45)	221 (37)	160 (27)

the time records of all variables the last 100 patterns for testing. This facilitated the representation of data and the evaluation as if the sensors were operating under real plant conditions, predicting MI sequentially in time. The pre-classification procedure outlined in the previous section was secondly applied to generate optimized training sets. The main characteristics of these training and test sets are also summarized in Table 2. Finally, the method described before for the selection of the most relevant features or input variables has been applied in order to reduce the size of the training set, thus reducing the CPU time needed to adapt the virtual sensor. The sets of ordered features for each model are summarized in Table 4. Note that the 25 input process variables ordered in Table 4 by correlation do not include the product grade label, even for the composite model situation, since this information is added only in this case after the best set of reduced variables is selected from the ordered complete set.

Table 3
Training parameters for the three neural models

	Fuzzy ARTMAP	Clustering average	DynaRBF
Single grade model All variables	$\rho = 0.995$		$d_{max} = 0.05$
Composite model All variables		$d_{max} = 0.05$	$\alpha_0 = 0.1$ (in Eq. (7))
Single grade model Reduced variables	$\rho = 0.9995$	$\alpha_0 = 0.1$ (in Eq. (7))	$\beta = 0.1$ (in Eq. (13))
Composite model Reduced variables			$\sigma_0 = 1.0$ (in Eq. (10))

6. Results and discussion

Table 2 summarizes the characteristics of the datasets used to build and test the different virtual sensors

Table 4
Ordered input process variables according to correlation with MI for all product grades

Grade A	Grade B	Grade C	Grade D	Grade E	Grade F	All grades
Temperature 3	Temperature 3	Temperature 2	Flow rate 1	Level	Extruder power	Flow rate 4
Density	Density	Temperature 9	Volumetric flow rate 2	Temperature 6	Volumetric flow Rate 1	Flow rate 3
Temperature 9	Concentration 1	Volumetric flow rate 2	Temperature 2	Flow rate 4	Temperature 9	Extruder power
Concentration 1	Volumetric flow Rate 1	Extruder power	Concentration 3	Temperature 7	Temperature 5	Concentration 1
Temperature 8	Temperature 9	Temperature 1	Temperature 9	Temperature 8	Temperature 4	Temperature 3
Extruder speed	Extruder power	Concentration 1	Temperature 1	Volumetric flow rate 1	Compressor throughput	Volumetric flow rate 2
Temperature 7	Temperature 8	Extruder speed	Volumetric flow rate 1	Concentration 1	Temperature 7	Temperature 8
Concentration 2	Temperature 7	Temperature 3	Concentration 2	Extruder speed	Level	Temperature 7
Concentration 3	Concentration 4	Density	Extruder speed	Volumetric flow rate 2	Temperature 8	Volumetric flow rate 1
Volumetric flow rate 1	Volumetric flow rate 2	Concentration 3	Flow rate 4	Temperature 1	Temperature 3	Temperature 1
Temperature 4	Extruder speed	Flow rate 3	Flow rate 2	Compressor throughput	Concentration 2	Concentration 2
Compressor throughput	Flow rate 2	Temperature 6	Density	Flow rate 3	Density	Density
Extruder power	Temperature 1	Concentration 2	Concentration 1	Temperature 9	Flow rate 3	Concentration 3
Level	Flow rate 1	Level	Concentration 4	Concentration 2	Flow rate 2	Temperature 6
Pressure	Flow rate 3	Concentration 4	Pressure	Temperature 4	Temperature 6	Level
Flow rate 2	Temperature 5	Flow rate 4	Flow rate 3	Density	Concentration 1	Temperature 5
Temperature 6	Level	Flow rate 2	Level	Extruder power	Flow rate 1	Temperature 4
Temperature 1	Compressor throughput	Flow rate 1	Temperature 8	Concentration 4	Flow rate 4	Temperature 9
Temperature 5	Temperature 2	Temperature 4	Temperature 6	Concentration 3	Concentration 4	Extruder speed
Flow rate 1	Concentration 3	Volumetric flow rate 1	Temperature 3	Temperature 5	Extruder speed	Pressure
Temperature 2	Concentration 2	Pressure	Extruder power	Temperature 2	Temperature 1	Compressor throughput
Concentration 4	Temperature 4	Temperature 8	Temperature 7	Flow rate 2	Temperature 2	flowrate2
Volumetric flow rate 2	Pressure	Temperature 7	Temperature 5	Temperature 3	Pressure	Temperature2
Flow rate 3	Flow rate 4	Compressor throughput	Compressor throughput	Flow rate 1	Concentration 3	Flow rate 1
Flow rate 4	Temperature 6	Temperature 5	Temperature 4	Pressure	Volumetric flow rate 2	Concentration 4

The horizontal lines distinguish the sets of reduced process variables (minimum dissimilarity) for each product grade and composite grades.

developed for three different families of LDPE identified in Fig. 3 according to their different MI values. Each of these families includes two product grades that are produced under different process conditions (dynamics). Single models for each product grade as well as composite ones valid for the ensemble of all LDPE families have been developed to test and compare the performance of the different sensor currently under consideration. The abovementioned data preprocessing techniques aimed at reducing the number of variables and at optimizing the composition of the training set, have been applied in all cases. The results obtained for all virtual sensors considered are summarized in Tables 5 and 6.

It should be noted that the average characteristics of the training and test sets listed in Table 2 differ slightly for single grade models and significantly for composite ones, as illustrated by the average MI values listed for each case. Differences are even more noticeable when the pre-classification technique for data selection is applied, and between models built with either all input variables or with the most relevant input features only. This should be kept in mind when comparing relative errors for different models, particularly for composite

Table 5
Absolute and relative mean errors, and standard deviations for the test sets predicted by all sensor models built with all process variables after training with both the sequential and the pre-classified set of patterns

Family	Grade	Linear model Abs (%) Std. dev.	Clustering average Abs (%) Std. dev.	Fuzzy ARTMAP Abs (%) Std. dev.	DynaRBF Abs (%) Std. dev.
Sequential					
I	A	0.100	0.072	0.069	0.066
		(13.7)	(9.9)	(9.4)	(9.0)
		0.099	0.066	0.052	0.053
	B	0.080	0.063	0.073	0.057
		(11.4)	(9.0)	(10.4)	(8.1)
		0.063	0.065	0.061	0.050
II	C	0.510	0.438	0.460	0.456
		(30.3)	(26.1)	(27.4)	(27.1)
		0.411	0.378	0.385	0.384
	D	0.148	0.190	0.163	0.152
		(8.5)	(10.9)	(9.4)	(8.8)
		0.126	0.132	0.118	0.106
III	E	0.323	0.295	0.302	0.286
		(16.6)	(15.2)	(15.6)	(14.7)
		0.303	0.281	0.276	0.291
	F	0.393	0.383	0.358	0.339
		(11.3)	(11.1)	(10.3)	(9.8)
		0.272	0.262	0.260	0.234
I+II+III	A–F	0.295	0.157	0.163	0.151
		0.301	0.160	0.152	0.148
Pre-classified					
I	A	0.107	0.057	0.049	0.048
		(14.9)	(7.9)	(6.8)	(6.7)
		0.165	0.050	0.045	0.067
	B	0.113	0.051	0.038	0.041
		(15.1)	(6.8)	(5.1)	(5.5)
		0.243	0.053	0.027	0.065
II	C	0.195	0.147	0.131	0.149
		(11.3)	(8.5)	(7.6)	(8.6)
		0.202	0.113	0.116	0.166
	D	0.132	0.063	0.058	0.059
		(7.8)	(3.7)	(3.4)	(3.4)
		0.159	0.065	0.060	0.064
III	E	0.164	0.064	0.076	0.056
		(8.2)	(3.2)	(3.8)	(2.8)
		0.165	0.066	0.077	0.049
	F	0.213	0.149	0.165	0.146
		(6.1)	(4.3)	(4.7)	(4.2)
		0.165	0.151	0.174	0.151
I+II+III	A–F	0.182	0.121	0.118	0.125
		0.178	0.114	0.120	0.115

models. In this case the average MI values of the sparse test sets will neither coincide with that of the training set nor with that of any of the six grades considered individually, and will differ significantly when both pre-classification and variable reduction techniques are applied to develop the composite models. For example, the average MI values for both the sequential and pre-classification training sets used to develop composite models with the reduced set of input variables is equal to 1.80 in Table 2 while that for the corresponding test sets are 1.7 and 1.25, respectively. Thus, only absolute errors are reported in Tables 5 and 6 for the composite models.

6.1. Models with the complete set of variables

Table 5 summarizes the performance of all neural sensor models developed with the complete set of input process variables after training with both the sequential and the pre-classified sets of data given in the upper part of Table 2, which also includes the test sets used for these models. The absolute and relative mean errors listed in Table 5 for single grade models and sequential training (upper part) indicate that all neural sensors, especially DynaRBF, function better than the linear model on the overall, with absolute mean errors for the

Table 6
Absolute and relative mean errors, and standard deviations for the test sets predicted by all sensor models built with the reduced set of process variables after training with both the sequential and the pre-classified set of patterns

Family	Grade	Linear model Abs (%) Std. dev.	Clustering average Abs (%) Std. dev.	Fuzzy ARTMAP Abs (%) Std. dev.	DynaRBF Abs (%) Std. dev.
Sequential					
I	A	0.083	0.064	0.066	0.059
		(11.4)	(8.8)	(9.0)	(8.1)
		0.066	0.051	0.061	0.055
	B	0.070	0.058	0.070	0.057
		(10.0)	(8.3)	(10.0)	(8.1)
		0.055	0.052	0.057	0.049
II	C	0.432	0.380	0.368	0.338
		(25.7)	(22.6)	(21.9)	(20.1)
		0.378	0.322	0.296	0.275
	D	0.153	0.166	0.185	0.175
		(8.8)	(9.6)	(10.7)	(10.1)
		0.135	0.126	0.147	0.135
III	E	0.291	0.224	0.217	0.219
		(15.0)	(11.5)	(11.2)	(11.3)
		0.294	0.224	0.166	0.202
	F	0.376	0.374	0.397	0.360
		(10.9)	(10.8)	(11.5)	(10.4)
		0.493	0.537	0.320	0.507
I+II+III	A–F	0.281	0.247	0.235	0.237
		0.325	0.338	0.318	0.343
Pre-classified					
I	A	0.055	0.037	0.038	0.031
		(7.6)	(5.1)	(5.3)	(4.3)
		0.047	0.034	0.037	0.034
	B	0.068	0.042	0.048	0.039
		(9.4)	(5.8)	(6.7)	(5.4)
		0.061	0.038	0.050	0.035
II	C	0.113	0.073	0.088	0.048
		(6.4)	(4.1)	(5.0)	(2.7)
		0.132	0.082	0.098	0.051
	D	0.043	0.038	0.046	0.020
		(2.5)	(2.2)	(2.7)	(1.2)
		0.062	0.048	0.046	0.022
III	E	0.113	0.097	0.083	0.097
		(5.6)	(4.8)	(4.1)	(4.8)
		0.168	0.200	0.120	0.143
	F	0.055	0.066	0.091	0.050
		(1.6)	(1.9)	(2.6)	(1.4)
		0.064	0.069	0.132	0.041
I+II+III	A–F	0.232	0.078	0.095	0.081
		0.286	0.166	0.167	0.162

linear model ranging from 0.080 to 0.510, respectively, for I(B) and II(C), and those for the neural sensors from 0.057 to 0.460. The standard deviations listed also in Table 5 are of the same order of magnitude as the absolute mean errors for all models with sequential training.

The better performance of neural sensors is clearer in family I, where all models yield consistently good predictions. Increases of approximately 40 and 20% in prediction accuracy are obtained for grades I(A) and I(B), respectively, with the neural models compared to the linear model. Deficient training is the cause for the anomalous high errors of grades II(C) and III(E) and for the slightly better performance of the linear model for grade II(D) since they disappear when the pre-classified set of data was used for training, as shown in the lower part of Table 5. Note that the small number of 640 training patterns available for training grade II(C) (see Table 2) aggravates this situation. In addition, the training set has the largest test to train ratio (15.6%) and is the most heterogeneous one, as indicated by its high percentage of different and single MI values in Table 2. The fact that an increase of 50% in the number of training patterns between II(C) and II(D) reduces errors

by approximately a factor of three, i.e. to the average levels in Table 5, reinforces the above arguments but also indicates that this family of LDPE was produced under distinct non-linear process dynamics.

To illustrate the genesis of the average errors given in Table 5, i.e. the detailed performance of the sensors built with sequential training, Fig. 5 shows the measured and predicted time-records for grade I(A). The linear model is unable to follow the time-sequence of the measured MI, while clustering average sensor and particularly the DynaRBF model perform reasonably well, consistently with the average errors in Table 5. The performance of fuzzy ARTMAP, while being reasonable on the average (see Table 5), yields a step-like response around the average MI. This is again a clear indication of the insufficient training information provided by the sequential set. The fact that this effect is more evident for the usually highly performing fuzzy ARTMAP algorithm in difficult classification problems arises from its need for increased training for periodic signals, as illustrated in the benchmark reported by Carpenter et al. (1992) and in tests carried out during the course of the current study. The plant data for all grades have an underlying periodicity around 1.1 residence time τ units.

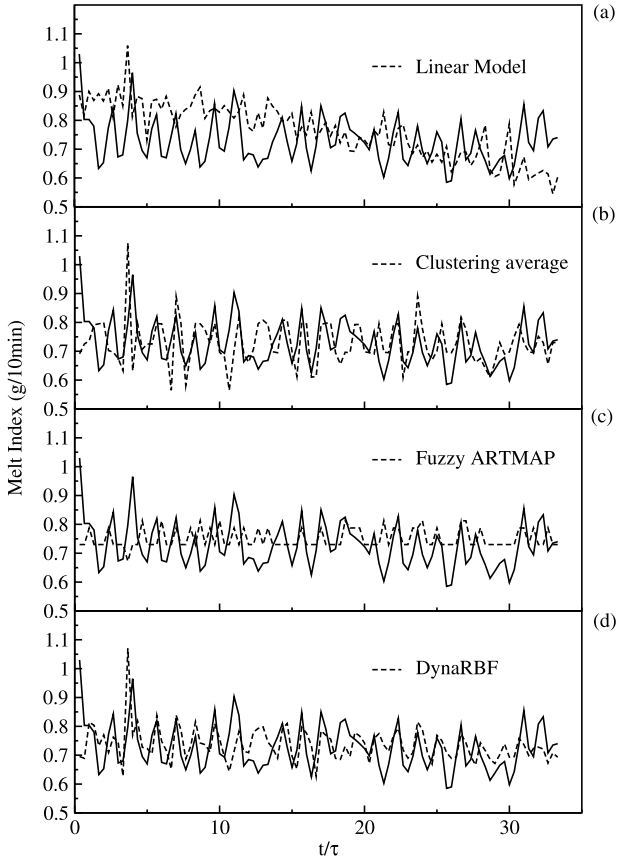

Fig. 5. Comparison of measured with predicted MI for grade I(A) using single neural models with sequential training and all available variables. (a) Linear; (b) clustering average; (c) fuzzy ARTMAP; (d) DynaRBF.

The performance of the single grade models developed with all variables and trained with the pre-classified set of MI values is summarized in the lower part of Table 5. The average performance of all virtual sensors for single grades trained using this pattern pre-classification selection procedure is significantly better in terms of both mean errors and standard deviations for all product grades than that reported in the upper part of Table 5 for sequential training. This effect is most significant for the grades with less number of patterns in Table 2, i.e. grades II(C), II(D) and III(E), where errors in the predictions decrease by more than a factor of three with the change of training sets. It is also more noticeable in the fuzzy ARTMAP sensor for its especial sensitivity to training in systems with periodical behavior, as explained before. For instance, the absolute mean error of predictions obtained for grade II(C) with the DynaRBF model drops from 0.456 (27.1%) to 0.149 (8.6%) and from 0.460 (27.4%) to 0.131 (7.6%) for fuzzy ARTMAP. The same applies to grade III(E) and for the other grades with errors decreasing between 30 and 60%.

The better performance of the single grade neural sensors with respect to the linear model is more evident and consistent when training is carried out with the pre-classified sets of data. The average relative errors for the former models range from 2.8 to 8.6% while for the latter they range from 6.1 to 15.1%. This tendency is even clearer in terms of standard deviations. The average performance of DynaRBF and fuzzy ARTMAP are similar, with an overall relative error of 5.2%, followed by clustering average with 5.7%, and the linear model with 10.6%. The classification capabilities of fuzzy ARTMAP can be inferred by the lower standard deviations of predictions that are obtained when the more appropriate pre-classified set of data is used to train the networks. The results for the three neural sensors are remarkable considering the $\pm 2\%$ error associated with the on-line MI measurements.

A detailed comparison of performances between the linear model and the DynaRBF sensor is shown in Fig. 6 for the three product families. Under the training scheme with pre-classified set of patterns detailed results can only be evaluated in terms of deviations of predicted MI with respect to measured values since training patterns are not presented to the neural models according to the time-sequence of plant measurements. Fig. 6 confirms the improved performance obtained with the DynaRBF model during testing. This figure also shows that there is an overlap in the MI distribution between product grades II(C) and III(E). This could be one of the reasons why product grade II(C) exhibits the maximum errors in Table 5.

The potential of the currently proposed neural sensing technology should become more evident when attempting to predict the behavior of the ensemble of LDPE grades simultaneously with only one sensor, i.e. with a

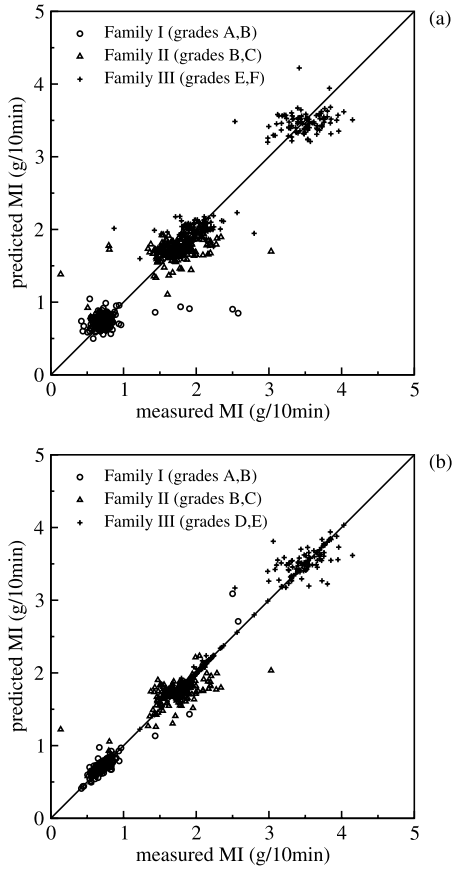

Fig. 6. Comparison of measured MI time-records with predictions obtained using the (a) single grade linear model and (b) single grade DynaRBF model trained with the pre-classified dataset of data for the three families of LDPE grades studied.

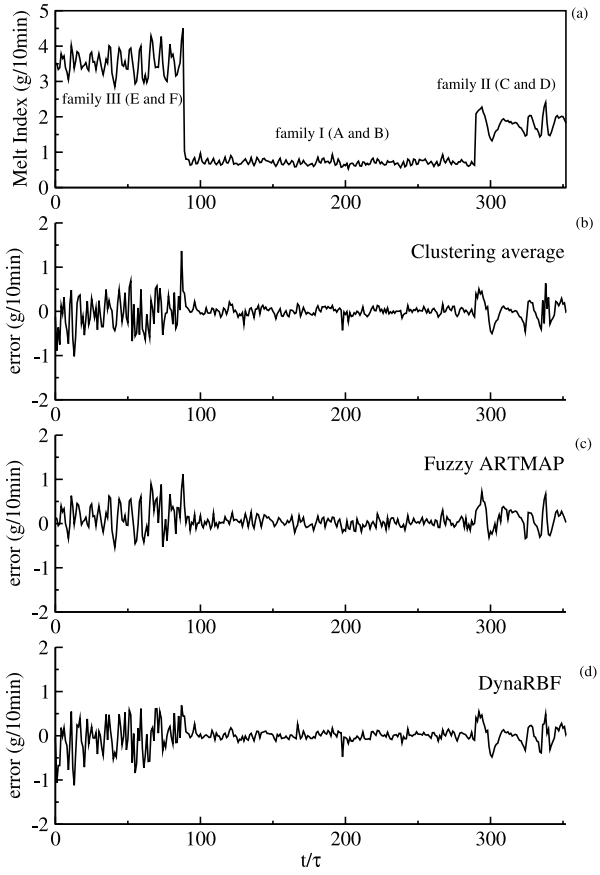

Fig. 7. Time-records of measured MI and for the errors of predictions obtained with composite neural sensor models applicable simultaneously to all the families of LDPE considered and trained with the sequential set of data. (a) measured MI; (b) clustering average; (c) fuzzy ARTMAP; (d) DynaRBF.

composite sensor model. As a consequence, the four models have also been tested with the more difficult problem of forecasting the quality of the three LDPE families (I+II+III) simultaneously, i.e. forecasting grade transitions. Table 5 also summarizes the composite model results obtained with both training sets formed by the 14 203 patterns indicated in Table 2 and using the 25 process variables listed in Table 1 complemented by the normalized label to identify the six grades. The composite models were tested with the remaining 600 patterns, so that the test to train ratio was kept comparable to that for single grade models.

Table 5 indicates that the absolute mean errors for the three composite neural sensor models with sequential training are approximately equal to 0.157 compared to 0.295 for the composite linear correlation model. This expected good performance of the neural systems, also reflected by the respective standard deviations of 0.153 and 0.301, is examined in Fig. 7 for a production cycle including the three families. The behavior of the three neural sensors is similar over time according to the time sequence of the plots of the errors (Fig. 7b–d) with respect to the measured MI (Fig. 7a). Predictions made over patterns corresponding to family I (grades A and B) show the lowest fluctuations in error while those for family III are the highest, as was also the case for the single grade sensors in sequential training in Table 5. This behavior is partially due to deficient training as discussed previously. Note that the MI variation shown in Fig. 7a also illustrates the differences in process dynamics between product families. The use of the most suitable training set obtained by pre-classification in the four composite models reduces absolute errors and standard deviations of predictions from 0.157 and 0.153 to 0.121 and 0.116, respectively, for all virtual sensor models and from 0.295 and 0.301 to 0.182 and 0.178 for the linear correlation model, as shown also in Table 5. In this case of composite models the performance of neural sensors is also better.

6.2. Models using the reduced set of variables

All single and the composite sensor models developed for the reduced set of process variables selected by dissimilarity of SOMs have been trained and tested by

using the datasets given in the lower part of Table 2. The reduction in the number of input variables has the advantages of (i) dealing with a lower dimensional problem, (ii) cutting-back any noise that could contaminate the measurements of the discarded variables, and (iii) avoiding variables that could provide conflicting information with respect to more correlated variables in relation to the target MI. Fig. 8 shows the variation of the average cumulative dissimilarity between SOMs that has been calculated for each grade by successive insertion of the variables listed in Table 4, according to Appendix A. The variables located beyond the minimum dissimilarity point in the plots of Fig. 8 or below the separation line in Table 4 do not contribute with any additional relevant information to explain the qualitative and quantitative behavior of MI and can be discarded from the input process plant information.

It should be noted that the same variables but in a different order would have been selected if an ordering criteria based on both topological and correlation information (Espinosa et al., 2001b) would have been

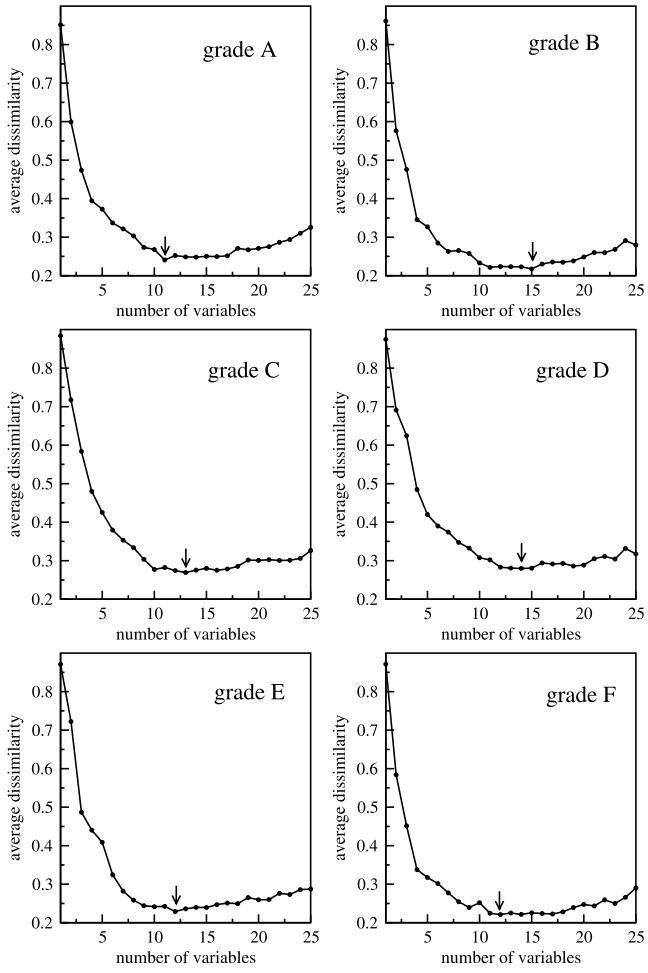

Fig. 8. Average dissimilarity between SOMs for the selection of a reduced and representative set of input variables for each and for the ensemble of product grades.

adopted. The application of this selection procedure allowed an approximate 50% reduction in the number of input data needed to develop all single sensor models. Table 6 summarizes the performance of all single grade models using the reduced set of input variables and training with sequential and pre-classified data.

The effects of variable reduction when the single grade sensor models were trained with the set of patterns selected sequentially are summarized in the upper part of Table 6. Comparison with the corresponding results in Table 5 shows that the performance of all single grade models, linear and neural, is maintained or improves slightly despite the reduction in the information provided to the virtual sensor. For example, the single models for grade I(B), all built with the first 15 variables listed in Table 4 according to the minimum dissimilarity in Fig. 8, yield predictions with an average absolute error and standard deviation of 0.062 and 0.053, respectively, for sequential training in Table 6, which is slightly better than the corresponding 0.064 and 0.059 for all 25 variables in Table 5. The same holds for the more difficult to predict grade II(C), where the 26.9% average relative error obtained for all variables with sequential sets (Table 5) drops to about 21.5% (Table 6) for the reduced set of 13 input variable selected from Fig. 8 and Table 4. In the case of grade III(E) the reduction in relative error is from 15.2% for all variables to 11.3% using 12 variables. Similar behaviors are observed for the other grades, both in terms of mean errors and standard deviations.

The effects of variable reduction when the single grade sensor models were trained with the best set of patterns selected by the pre-classification procedure can be observed in the lower part of Table 6. The errors of predictions drop on the average from approximately 5% in Table 5 to 4% in Table 6 for the neural sensors, and from 10 to 5.5% for the linear model. The reduction in input variables also causes a decrease in standard deviations. The same tendency of improvement or comparable performance is observed for each individual sensor and grade. DynaRBF is confirmed in Table 6 as the best single grade neural sensor with an overall absolute error and standard deviation of 0.048 (3.3%) and 0.054 for the six grades A–F, followed by clustering average with 0.059 (4%) and 0.079, and fuzzy ARTMAP with 0.066 (4.4%) and 0.081. In all cases, the mean errors of predictions are comparable to the $\pm 2\%$ error associated with MI on-line measurements. As discussed before, the unexpected poorer performance of the powerful ARTMAP classifier is due to insufficient training considering the underlying periodicity of the inferred target MI variable. The analyses of detailed input pattern classification and MI forecasting during testing support the average results in Table 6, but have not been included here for brevity. Comparison between the upper and lower parts of Table 6 confirms again the

adequacy of the pre-classification approach to select data for training all sensors.

This significant improvement in the performance of properly trained single grade virtual sensors caused by the adequate reduction in input information indicates that redundancy of information may be disadvantageous when dealing with field variables contaminated by measurement errors. Also, the inclusion of variables with conflicting or contradicting information in relation to that contributed by other variables with higher correlations with the target MI could result in a detrimental effect. Note that beyond the minimum points in the plots of Fig. 8 dissimilarity changes very slowly with the addition of more variables indicating that their inclusion or exclusion will not affect information but could contribute to the addition or subtraction of noise and/or of conflicting information with respect to their effect on MI.

The effects of the reduction of variables in the composite models are also included in Table 6 and the most relevant results highlighted in Fig. 9. The arrow in Fig. 9a indicates that the minimum dissimilarity is reached in this case when the first 17 variables are considered from the ordered list in the last column of Table 4. It is also clear from this figure that noise and experimental errors make the choice of the minimum dissimilarity more difficult. All composite models have been developed with an input formed by these 17 variables plus the normalized product grade label to facilitate product identification.

The performance of the linear composite model improves slightly with variable reduction when sequentially trained, with absolute errors decreasing from 0.295 in Table 5 to 0.281 in Table 6, but worsens when the best pre-classification set of data is used at the learning stage, increasing from 0.182 in Table 5 to 0.232 in Table 6. It should be noted that the simple additive nature of this model (Eq. (15)) could more easily cancel the noise effects and/or handle conflicting information and, thus, gain from the redundant information brought by any extra variable when appropriately trained; the lowest error of 0.182 and standard deviation of 0.178 correspond to the composite linear correlation built with all variables and trained with the pre-classified set of patterns (Table 5).

The functioning of the composite neural sensor models trained sequentially worsens significantly with the reduction of input variables; relative errors increase on the overall from 9% in Table 5 to 14% in Table 6, while standard deviations approximately double. For the more appropriate training set selected by pre-classification performance remains unchanged or improves slightly with variable reduction in terms of errors but the dispersion of predictions increase. These results indicate the difficulties encountered by the sensors to properly classify certain input patterns during testing

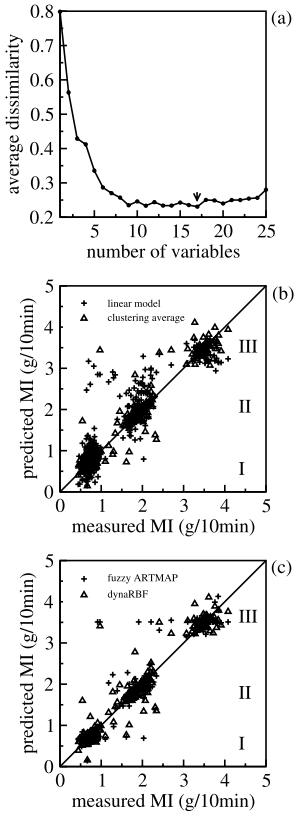

Fig. 9. Results obtained for the composite models with all sensors trained using the best set of pre-classified data and the reduced set of input variables. (a) Variation of average dissimilarity between SOMs with the inclusion of ordered variables; (b) measured vs. predicted MI for the linear and clustering average models; (c) measured vs. predicted MI for the fuzzy ARTMAP and DynaRBF models.

and to generate adequate outputs for MI for the large variety of process dynamics that occur in the LDPE plant. The best composite neural sensors are dynaRBF and clustering average, which yield the lowest absolute errors of about 0.080 for the case of pre-classified training (lower part of Table 6). The corresponding error for fuzzy ARTMAP is an acceptable 0.095 considering the periodicity effects discussed before. When the inadequate sequential training is applied in this case of less input information (reduced variables) the mean absolute error of predictions triples (upper part of Table 6), which is the highest in Tables 5 and 6 for neural composite models. This is consistent with the fact that any model deficiency should become more evident when dealing with the more difficult problem of predicting simultaneously all grades with reduced input information. These results are illustrated and corroborated in detail in Fig. 9b and c where the measured and predicted MI values corresponding to the test set of patterns are compared for the four composite models trained with pre-classified data. The superior performance of the neural models compared to the linear model is clear in these figures. The origin of the observed

deviations is similar to that illustrated in Fig. 7 for the complete set of variables and sequential training. Again, the lowest deviations correspond to family I.

7. Concluding remarks

A neural network-based methodology to design and build virtual sensors to infer product quality from process variables has been developed and tested. Three neural systems, together with a linear model, have been used to build different virtual sensor models. A predictive fuzzy ARTMAP algorithm is one of the architectures considered. The other two architectures, identified as clustering average and DynaRBF, have been built by the combination of a dynamic unsupervised clustering layer with two different supervised mapping procedures to implement the desired outputs. This hybrid approach facilitates the use of more elaborate learning algorithms for the supervised layer without affecting the underlying infrastructure based on the dynamic unsupervised clustering.

Self organizing maps (SOM) have been introduced and proven effective to estimate the relevance of certain features and patterns by means of dissimilarity measures and to select from this quantitative information the minimum number of process variables needed as input to the sensors. The neural sensors have been trained using data selected either by sequential order from the time records of variables or by pre-classification of plant data as a function of the difficulty to classify events within the pool of available plant information.

As a proof-of-concept of the generic virtual sensor model, three types of neural models have been developed to infer the MI of LDPE so that the accuracy of on-line correlation based techniques that are commonly used in industry is increased. Both single sensors for each LDPE grade and a composite model for all grades simultaneously have been implemented and tested. The three neural models for single grades, with all process variables measured at the beginning of the production cycle considered as input, predict MI with relative mean errors and standard deviations of approximately 5% when appropriately trained with pre-classified patterns, compared with the average errors and standard deviations of approximately 10 and 15%, respectively, obtained for the corresponding linear correlation models. A reduction in the number of variables up to 50% by dissimilarity measures of SOM decreases these errors for single grade neural and linear models to approximately 4 and 5.5%, respectively, with comparable decreases in standard deviations. This reduction, which sets the accuracy standards of virtual sensors close to the $\pm 2\%$ experimental error for on-line MI measurements, could be explained in terms of noise reduction and elimination of conflicting input information with respect to the target variable MI.

DynaRBF yields slightly more accurate predictions of MI than fuzzy ARTMAP and clustering average for single grade sensors. It is well known that fuzzy ARTMAP needs an extra amount of training information when the problem under study possesses some underlying periodicity, as is the case of the current on-line time-variation of MI. The results obtained both for single grade and composite models indicate that a neural implementation provides prediction reliability and accuracy. The proposed virtual sensors are capable of learning the relationships between process variables measured at the beginning of the production cycle and the quality of the final product. Their superior performance does not require any readjustment of parameters during production cycles.

The three neural sensors perform similarly for composite models. Sensors built with the reduced set of input process variables and trained by pre-classification yield the best predictions. The out-performance of neural sensors with respect to linear correlations is even more evident in this case of composite models since neural sensors are capable of quickly adapting to new operating conditions of the plant, including grade transitions. Nevertheless, the effect of the reduction of input variables increases the standard deviation indicating that better training is needed.

Acknowledgements

This research was supported by DGICYT (Spain), projects PB96-1011 and PPQ2000-1339, and Comissionat per a Universitats i Recerca (Catalunya), projects 1998SGR00102 and 2000SGR00103.

Appendix A: The Self Organizing Map

The SOM algorithm (Kohonen, 1990) performs a topology-preserving mapping from a high-dimensional input space onto a low dimensional output space formed by a regular grid of map units. The lattice of the grid can be either rectangular or hexagonal. This neural model is biologically plausible (Kohonen, 1993) and is present in various brains' structures. From a functional point-of-view, the SOM resembles vector quantization (VQ) algorithms. These algorithms approximate, in an unsupervised way, the probability density functions of vectors by finite sets of codebook or reference vectors. The only purpose of these methods is to describe class borders using a nearest-neighbor rule (see for example the K-means algorithm in MacQueen (1967)). In contrast, SOM's units are organized over the space spanned

by the regular grid and the whole neighborhood is adapted in addition to the winner unit or neuron.

Each unit of the map is represented by a weight vector, $m_i = [m_{i1}, \ldots, m_{in}]^T$, where n is equal to the dimension of the input space. As in VQ, every weight vector describing a class is called a codebook. Each unit has a topological neighborhood N_i determined by the form of the grid's lattice, either rectangular or hexagonal. The number of units, as well as their topological relations, is defined (and fixed) at the beginning of the training process. The granularity (size) of the map will determine its subsequent accuracy and generalization capabilities. During its training process, the SOM forms an elastic net that folds onto the cloud formed by the input data, trying to approximate the probability density function of the original data by placing more codebook vectors where the data are dense and fewer units where they are sparse.

Training of the SOM proceeds as follows. At each training step, one sample pattern **x** is randomly chosen from the training data. Similarities (distances) between **x** and the codebook vectors are computed (the Euclidean distance is usually adopted), looking for the *BMU* or neuron. This similarity matching can be expressed as:

$$|x - m_{\text{bmu}}| = \min_i \left\{ \left| x - m_i \right| \right\} \tag{A1}$$

After finding the BMU and its topological neighbor cells, their degree of matching is increased by moving their codebook vectors in the proper direction in the input space. This competitive learning process follows a winner-takes-all approach, which can be described by the following rules:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)[x(t) - m_i(t)]; & i \in N_{\text{bmu}}(t) \\ m_i(t); & i \notin N_{\text{bmu}}(t) \end{cases} \tag{A2}$$

In this equation t denotes time (position), $N_{\text{bmu}}(t)$ is a decreasing neighborhood function around the best matching unit (BMU), and $\alpha(t)$ is a monotonically decreasing learning rate. Two modes of operation for the SOM can be considered in relation to the variation of the learning rate during training. An initial ordering phase in which the map is formed and tuning is coarse, and a later convergence phase, where the fine-tuning of the codebook vectors is performed. This adaptive approach corresponds to Hebbian learning.

A.1. Measures of dissimilarity for Kohonen maps

To select the most relevant input features the similarity between the maps for each variable in relation to MI must be measured. Several measures have been proposed to compare the positions of the reference vectors in different map structures (Bauer and Pawelzik, 1992; Kiviluoto, 1996; Willmann et al., 1994; Kraaijveld et al., 1992). In the present work the dissimilarity measure proposed by Kaski and Lagus (1997) is used. This measure is based on a goodness measure proposed by the same authors that combines an index of the continuity of the mapping from the dataset to the map grid with a measure of accuracy of the map. The dissimilarity between two maps L and M is defined by the average difference of their *goodness*:

$$D(L, M) = E\left[\frac{|d_L(x) - d_M(x)|}{d_L(x) + d_M(x)}\right] \tag{A3}$$

In this equation E is the average expectation, and $d(x)$ the distance from x to the second BMU, denoted by $m_{\text{bmu}'(x)}$, beginning at the first BMU or winner neuron, denoted by $m_{\text{bmu}(x)}$. Of all possible paths between $m_{\text{bmu}(x)}$ and $m_{\text{bmu}'(x)}$ the shortest path passing continuously between neighbor units is selected,

$$d(x) = \|x - m_{\text{bmu}}(x)\| + \min_i \sum_{k=0}^{K_{\text{bmu}'(x)}-1} \|m_{I_i(k)} - m_{I_i(k+1)}\| \tag{A4}$$

References

Barto, A. G., Sutton, R. S. & Anderson, C. H. (1983). Neuron-like adaptive elements that can solve difficult learning control problem. *IEEE Transactions on Systems, Man and Cybernetics 13* (5), 834.

Bauer, H. U. & Pawelzik, K. R. (1992). Quantifying the neighbourhood preservation of self-organizing maps. *IEEE Transactions on Neural Networks 3*, 570.

Berthold, M. R. & Diamond, J. (1995). Boosting the performance of RBF networks with dynamic decay adjustments. In G. Tesauro, D. S. Touretzky & T. K. Leen (Eds.), *Advances in neural information processing systems*, vol. 7, p. 521).

Broomhead, D. S. & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems 2*, 321.

Carpenter, G. A., Grossberg, S. & Rosen, D. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks 4*, 759.

Carpenter, G. A., Grossberg, S., Marcuzon, N., Reinolds, J. H. & Rosen, D. B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks 3*, 698.

Chan, W. M., Gloor, P. E. & Hamielec, A. E. (1993). A kinetic model for Olefin polymerization in high-pressure autoclave reactors. *American Institute of Chemical Engineering Journal 1* (39), 111.

Espinosa, G., Yaffe, D., Arenas, A., Cohen, Y. & Giralt, F. (2001a). A fuzzy ARTMAP based quantitative structure-property relationships (QSPRs) for predicting physical properties of organic compounds. *Industrial and Engineering Chemistry Research 40* (12), 2757.

Espinosa, G., Arenas, A. & Giralt, F. (2001b). An integrated SOM-fuzzy ARTMAP neural system for the evaluation of toxicity. *Journal of Chemical Information and Computer Sciences 41* (5), 1150.

Ferre, J. A. & Giralt, F. (1993). A pattern recognition approach to the analysis of turbulent signals. In J. P. Bonnet & M. N. Glauser

(Eds.), *Eddy structure identification in free turbulent shear flows*, p. 181). Kluwer Academic Publishers.

Ferre-Giné, J., Rallo, R., Arenas, A. & Giralt, F. (1996). Identification of coherent structures in turbulent shear flows with a fuzzy ARTMAP neural network. *International Journal of Neural Systems 7*, 559.

Fritzke, B. (1994). Fast learning with incremental RBF networks. *Neural Processing Letters 1* (1), 2.

Fritzke, B. (1994). Growing cell structure-a self-organizing networks for unsupervised and supervised learning. *Neural Networks 7* (9), 1441.

Giralt, F., Arenas, A., Ferre-Giné, J., Rallo, R. & Kopp, G. A. (2000). The simulation and interpretation of free turbulence with a cognitive neural system. *Physics of Fluids 12*, 1826.

Hertz, J. A., Krogh, A. & Palmer, R. G. (1991). *Introduction to the theory of neural computation. Santa Fe Institute Studies in the science of complexity*. Addison-Wesley.

Hunt, K. J., Sbarbaro, D., Zbikowski, R. & Gawthrop, P. J. (1992). Neural networks for control systems—a survey. *Automatica 28* (6), 1083.

Hwang, Y. S. & Bang, S. Y. (1997). An efficient method to construct a radial basis function neural network classifier. *Neural Networks 10* (8), 1495.

Kaski, S., & Lagus, K. (1997). Comparing self-organizing maps. In *Proceedings of ICANN'96* (p. 809).

Kiviluoto, K. (1996). Topology preservation in self-organizing maps. In *Proceedings of ICNN'96, IEEE International Conference on Neural Networks, 1*, p. 294).

Kohonen, T. (1990). The self-organizing map. *Proceedings IEEE 78* (9), 1464.

Kohonen, T. (1993). Physiological interpretation of the self-organizing map algorithm. *Neural Networks 6* (7), 895.

Kraaijveld, M. A., Mao, J. & Jain, A. K. (1992). A non-linear projection method based on Kohonen's topology preserving maps. In *Proceedings of 11 ICPR, 11th International Conference on Pattern Recognition*, p. 41).

Lee, S., & Kil, R. M. (1988). Multi-layer feedforward potential function network. In *Proceedings of the IEEE International Conference of Neural Networks* (p. 161). San Diego, I.

Lines, B., Hartlen, D., Paquin, F. D., Treiber, S., Tremblay, M. & Bell, M. (1993). Polyethylene reactor modelling and control design. *Hydrocarbon Processing June*, 119.

MacQueen, J. (1967). Some methods of classification and analysis of multivariate observations. In L. M. LeCam, J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. I, p. 281), University of California Press.

Martin, G. (1997). Consider soft sensors. *Chemical Engineering Progress July*, 66.

Moody, J. & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation 1*, 281.

Musavi, M. T., Ahmed, W., Chan, K. H., Faris, K. B. & Hummels, D. M. (1992). On the training of radial basis function classifiers. *Neural Networks 5*, 595.

Park, J. & Sandberg, I. W. (1991). Universal approximation using radial basis function networks. *Neural Computation 3*, 246.

Platt, J. C. (1991). A resource-allocating network for function interpolation. *Neural Computation 3* (2), 213.

Poggio, T. & Girosi, F. (1990a). Regularization algorithms for learning that are equivalent to multilayer networks. *Science 247*, 978.

Poggio, T. & Girosi, F. (1990b). Networks for approximation and learning. *Proceedings of the IEEE 78*, 1481.

Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. In J. C. Mason & M. G. Cox (Eds.), *Algorithms for approximation*, p. 143). Oxford: Clarendon Press.

Powell, M. J. D. (1992). The theory of radial basis functions in 1990. In W. Light (Ed.), *Advances in numerical analysis*, vol. II, p. 105). Oxford: Clarendon Press.

Tambourini, F., & Davoli, R. (1994). An algorithmic method to build good training sets for neural-network classifiers. Technical Report UBLCS-94-18. This report is available at: ftp://cs.unibo.it/pub/UBLCS/9418.ps.gz

Willmann, T., Der, R. & Martinetz, T. (1994). A new quantitative measure of topology preservation in Kohonen's feature maps. In *Proceedings of ICNN'94, IEEE International Conference on Neural Networks*, p. 645).

Zadeh, L. (1965). Fuzzy sets. *Information and Control 8*, 338.