

# Manufacturing quality control by means of a Fuzzy ART network trained on natural process data

Massimo Pacella<sup>a,\*</sup>, Quirico Semeraro<sup>b</sup>, Alfredo Anglani<sup>a</sup>

<sup>a</sup>*Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, Via per Monteroni, Lecce 73100, Italy*

<sup>b</sup>*Dipartimento di Meccanica, Politecnico di Milano, Via Bonardi, Milano 20133, Italy*

Received 25 October 2002; received in revised form 24 November 2003; accepted 24 November 2003

## Abstract

In order to produce products with constant quality, manufacturing systems need to be monitored for any unnatural deviations in the state of the process. Control charts have an important role in solving quality control problems; nevertheless, their effectiveness is strictly dependent on statistical assumptions that in real industrial applications are frequently violated. In contrast, neural networks can elaborate huge amounts of noisy data in real time, requiring no hypothesis on statistical distribution of monitored measurements. This important feature makes neural networks potential tools that can be used to improve data analysis in manufacturing quality control applications. In this paper, a neural network system, which is based on an unsupervised training phase, is presented for quality control. In particular, the adaptive resonance theory (ART) has been investigated in order to implement a model-free quality control system, which can be exploited for recognising changes in the state of a manufacturing process. The aim of this research is to analyse the performances of ART neural network under the assumption that predictable unnatural patterns are not available. To such aim, a simplified Fuzzy ART neural algorithm is firstly discussed, and then studied by means of extensive Monte Carlo simulation.

© 2004 Elsevier Ltd. All rights reserved.

*Keywords:* Manufacturing quality control; Fuzzy ART neural network; Monte Carlo simulation

## 1. Introduction

Statistical process control (SPC) is a methodology based on several techniques, which is aimed at monitoring manufacturing process output measurements. Control charts are the most widely applied SPC tools used to reveal unnatural variations of monitored measurements, as well as to locate their assignable causes. To use a control chart, samples of the output are collected during manufacturing process, and sample statistics are then plotted on the chart. If the process is in a natural state, the sample statistics are expected to plot within specific control limits. On the other hand, if a special cause of variation is present, the sample statistics are likely to plot outside the predefined control limits. When an unnatural variation is signalled by control chart, operators search for the special cause and make

necessary corrections and adjustments to bring the process back to the natural state.

Nowadays, with the widespread exploitation of automated production and inspection in several manufacturing environments, the tasks of SPC traditionally performed by quality practitioners have to be automated. Hence, computer-based algorithms need to be developed to implement, or at least help quality practitioners to carry out, the various quality control tasks automatically. Neural networks are promising and effective analysis tools and, in the last decade, they have been widely used in quality control (Zorriassantine and Tannock, 1998). What makes neural networks popular is their ability to learn from experience and to handle uncertain and complex information in a competitive and quality demanding environment. Notably, neural networks are suitable for quality control because of their capacity for handling noisy measurements requiring no assumption about statistical distribution of the monitored data.

Several researchers have investigated applications of neural networks for manufacturing quality control.

\*Corresponding author. Tel.: +39-0832-297-253; fax: +39-0832-297-279.

E-mail address: [massimo.pacella@unile.it](mailto:massimo.pacella@unile.it) (M. Pacella).

Pugh (1991) proposed the first reported neural network for quality control. The multi-layer perceptron (MLP) network, trained by means of the supervised back-propagation (BP) learning algorithm (Bishop, 1995), has been used to detect mean shifts. Guo and Dooley (1992), and Smith (1994) trained an MLP BP network for identifying positive shifts in both mean and variance. Cheng (1995) later, trained an MLP BP neural network for identifying positive/negative shifts and upward/downward trends of the process mean as well. Guh and Tannock (1999) developed an MLP BP neural network approach for concurrent unnatural pattern recognition. Cook et al. (2001) discussed the development of an MLP BP neural network to identify changes in the variance of serially correlated process parameters.

The MLP BP network has been exploited successfully in pattern recognition, but the slowness in training still poses some inconveniences for practical applications. Indeed, the convergence of the BP algorithm requires a huge number of iterations, as well as an adequate number of training examples. Therefore, other feed-forward neural networks for quality control have been proposed in the literature. For example, Cook and Chiu (1998), in order to recognise mean shifts in auto-correlated manufacturing process parameters, proposed a radial basis function (RBF) neural network system.

A shared feature of the most diffused neural methods for quality control is the exploitation of supervised training algorithms. The use of these techniques is based on the hypothesis that user knows in advance the group of unnatural patterns that must be discovered by neural network. A prior knowledge of pattern shapes is essential for generating training data that mimic the actual unnatural outcomes. However, in real industrial cases, unnatural process outcomes cannot be manifested by the appearance of predictable patterns, thus mathematical models are not readily available or they cannot be formulated.

The present paper proposes a different neural network approach for process monitoring, when no previous knowledge on the distribution of unnatural data is available. The proposed approach is based on the adaptive resonance theory (ART) neural network that is capable of fast, stable and cumulative learning.

The ART network is a neural algorithm, which is used to cluster arbitrary data into groups with similar features. Al-Ghanim (1997) presented an ART1 neural network (the binary version of the ART algorithm) as a means to distinguish natural from unnatural variations in the outcomes of a generic manufacturing process. The author proposed to train the ART1 network using a set of natural data patterns produced by the monitored process. During the training phase, the network clusters natural patterns of data into groups with similar features, and when it is confronted by a new input, it produces a response that indicates which cluster the

pattern belongs to (if any). Therefore, the neural network is not intended to indicate the type of unnatural pattern detected in process outputs. It provides an indication that a structural change in process outputs has occurred when input pattern does not fit to any of the learned natural categories.

The use of a neural system that monitors process outputs without a prior knowledge of unnatural patterns is appealing in real industrial applications. Indeed, only knowledge of the natural behaviour of the process is required in order to train the neural network. Furthermore, the neural network can operate in a plastic mode (i.e. a continuous and cumulative training mode) as long as new patterns are presented to it.

The remainder of this paper is structured as follows. The ART is presented in Section 2. The reference test case is illustrated in Section 3. The proposed Fuzzy ART neural system and the training/testing algorithms are discussed in Section 4 and 5, respectively. Then, simulation methodology and experimental results are both provided in Section 6. Finally, the last section gives conclusions and discusses some directions for further research.

## 2. The adaptive resonance theory

ART was introduced as a theory of human cognitive information processing. This theory has led to an evolving series of neural network models for unsupervised and supervised category learning. These models, including ART1, ART2, ARTMAP, Fuzzy ART, and Fuzzy ARTMAP, are capable of learning stable recognition categories in response to arbitrary input sequences (Pao, 1989; Hagan et al., 1996).

ART1 can stably learn to categorise binary inputs and ART2 can learn to categorise analog patterns presented in an arbitrary order. ARTMAP can rapidly self-organise stable categorical mappings between  $m$ -dimensional input vectors and  $n$ -dimensional output vectors. Fuzzy ART, which incorporate computations from fuzzy set theory into the ART1 neural network, is capable of fast stable learning of recognition categories in response to arbitrary sequences of either analog or binary input patterns (Huang et al., 1995; Georgiopoulos et al., 1996, 1999). Fuzzy ARTMAP, the combination of ARTMAP with Fuzzy ART, can rapidly learn stable categorical mappings between analog input and output vectors.

### 2.1. The ART algorithm

ART is composed of two major subsystems, the attentional and the orienting subsystem. While in the attentional subsystem familiar patterns are processed, the orienting subsystem resets the neural activity

whenever an unfamiliar pattern is presented as input. Two layers of nodes, namely F1 (called the comparison layer) and F2 (called the recognition layer) which are fully connected by both bottom-up and top-down weights, compose the attentional subsystem. The bottom-up and top-down weights between F1 and F2 can be updated adaptively in response to input patterns.

While the comparison layer (F1) acts as a feature detector that receives external input, the recognition layer (F2) acts as a category classifier that receives internal patterns. The application of a single input vector leads to a neural activity that produces a pattern in both layers F1 and F2. These patterns remain in the network only during the application of current input. The orienting subsystem is responsible for generating a reset signal to F2 when the bottom-up input pattern and the top-down template mismatch according to a vigilance criterion. This reset signal, if sent, will stop the neural activity of the recognition layer, and during training the network adapts its structure by immediately storing the novelty in additional nodes of the layer F2. If the reset signal is not sent, the formerly coded pattern associated with the category node that represents the best match to current input, will be modified to include the input features. The vigilance criterion depends on the vigilance parameter. The choice of high values for the vigilance parameter implies that only a slight mismatch will be tolerated before a reset signal is emitted. In contrast, small values of vigilance imply that large mismatches will be tolerated.

## 2.2. Fuzzy ART

Using one of the unsupervised ART networks rather than the simpler competitive learning system, important stability properties of the network can be exploited (Haykin, 1999). Indeed, unlike competitive learning, when new patterns are produced by the monitored process, ART networks can continue to learn (without forgetting past learning) and incorporate new information. ART1, ART2 and Fuzzy ART are examples of unsupervised ART methods, which are capable of learning in both off-line (batch) and on-line (incremental) training modes. Dissimilarities among input patterns are only considered in their measurement space for clustering (unsupervised training). After clustering this space, each of their clusters is given by a weight vector (the *template*).

ART1 only tolerates binary ('0' or '1' coded) numbers within an input vector. ART2 and Fuzzy ART can process any real number, scaled to the continuous range between 0 and 1. The differences between ART2 and ART1 reflect the modifications needed to accommodate patterns with continuous-valued components. The F1 field of ART2 is more complex because continuous-valued input vectors may be arbitrarily close together.

The F1 field in ART2 includes a combination of normalisation and noise suppression, in addition to the comparison of the bottom-up and top-down signals needed for the reset mechanism.

Fuzzy ART is the most recent adaptive resonance framework that provides a unified architecture for both binary and continuous value inputs. Fuzzy ART operations reduce to ART1 (which accepts only binary vectors) as a special case. The generalisation of learning both analog and binary input patterns is achieved by replacing the appearance of the logical AND intersection operator ( $\cap$ ) in ART1 by the MIN operator ( $\wedge$ ) of fuzzy set theory.

By incorporation from fuzzy set theory into ART1, Fuzzy ART does not require a binary representation of input patterns to be clustered; however, it possesses the same desirable properties as ART1 and a simpler architecture than that of ART2. There are two important differences between ART2 and Fuzzy ART.

- The first is in their measures of dissimilarity between input patterns and templates: Fuzzy ART uses the city-block distance metric (or *Manhattan* distance, which derives from the MIN operator of the fuzzy set theory) rather than the Euclidean norm distance as in the ART2. Each category of Fuzzy ART is represented by the simplest statistics about its data: the minimum and the maximum in each dimension, which are learned to conjointly minimise predictive error and maximise predictive generalization. A hyper-rectangle represents the range of acceptable category vectors. Multiplications are not required for weight adaptations and the algorithm can perform well with few digits of weight precision. On the other hand, the ART2 architecture requires highly complex reset and choice functions, which are based on Euclidean norm.
- The second is in the way they pre-process their data (normalisation of input patterns). For ART2 the normalisation of input patterns is obtained by dividing each vector by its Euclidean norm. Hence, ART2 is able to achieve good categorisation of input patterns only if they are all normalised to a constant common length. However, such normalisation can destroy valuable amplitude information that, instead, is necessary for quality monitoring. In order to save such information, Fuzzy ART uses *complement coding*, which transforms every  $M$ -dimensional input vector into a  $2M$ -dimensional vector, as normalisation pre-processing. With complement coding, Fuzzy ART is able to achieve good categorization of input patterns even if input vectors have not the same norm.

An additional desirable property of Fuzzy ART is that, due to the simple nature of its architecture, responses of the neural network to input patterns are

easily explained, in contrast to other models, where in general, it is more difficult to explain why an input pattern produces a specific output. Significant insight has been gained in the past by attributing a geometrical interpretation to the Fuzzy ART categories, and recently, novel geometric concepts have been introduced in the original framework. Detailed properties of learning for Fuzzy ART can be found in Huang et al. (1995), Georgiopoulos et al. (1996, 1999), Anagnostopoulos and Georgiopoulos (2002).

Because of its algorithmic simplicity, and of the several properties that facilitate the implementation of the neural network, Fuzzy ART neural network has been exploited in this paper for analog pattern clustering in quality monitoring applications.

### 3. The reference manufacturing process model

In order to investigate Fuzzy ART performances for quality control applications, a generic manufacturing process has been synthetically reproduced by means of a computer program. The code is based on the pseudo-random number generator provided by the MATLAB<sup>®</sup> software environment (Vattulainen et al., 1995).

The running of a control chart, as well as of a neural network algorithm, can be considered a periodically repetitive statistical test. At each time  $t$ , a specific subset of the past output data is used to evaluate the state of the process. The null hypothesis  $H_0$  and the alternative hypothesis  $H_1$  of the test can be formulated as follows:

- $H_0$ : the process is in a natural state.
- $H_1$ : the process is not in a natural state.

As with every statistical test, errors of Type I and Type II can occur. They can be formulated as follows:

- *Error of Type I*: Some action is taken although the process is under control (*false alarm*).
- *Error of Type II*: No action is taken although the process is out of control.

The focus of this research is on processes with a single quality parameter of interest. Let  $\{Y_t\}$  be the random sequence of the observed quality characteristic, where  $t = 1, 2, \dots$  denotes a discrete time index or part number. The random time series  $\{Y_t\}$  has been simulated by means of a probabilistic model. A process in a natural state is realistically modelled by a system in which the output is the sum of a constant nominal mean (i.e. the process target  $\mu$ ), added to a random natural variation component. This random component, which models the natural process variability, is a time series of *normally, independently and identically distributed* (NIID) values with mean zero and common standard deviation  $\sigma$ . Without loss of generality, it is assumed that  $\mu = 0$  and

$\sigma = 1$  (a standardisation of the monitored measurements can be implemented otherwise). This model gives a close approximation to many types of practical manufacturing processes. In situations where these assumptions are violated, a power transformation technique can be implemented in order to reduce anomalies such as non-normality and heteroscedasticity of the monitored measurements.

On the other hand, when the process starts drifting from the natural state, a form of a special disturbance signal overlaps the series of process output measurements. This special disturbance signal is usually referred to as the *unnatural pattern*.

Thus, let  $\{Z_t\}$  be the time series of the natural process data, and let  $\{S_t\}$  be the time series of the special disturbance signal. The statistical test can be formulated as follows:

$$\begin{aligned} Z_t &\sim \text{NIID}(0, 1), \\ H_0 &: Y_t = Z_t, \\ H_1 &: Y_t = Z_t + S_t. \end{aligned} \quad (1)$$

In order to simulate the process in an unnatural state, the mean *shift* will be used as *unnatural pattern* in the reference test case (Montgomery, 2000). The primary possible causes for the mean shift may result from the introduction of new machines, workers, or methods. Other possible reasons include the minor failures of a machine part or changes in the skill level of the operators. Assuming  $\varphi$  the amplitude of the shift and  $\tau$  the instant of shifting, then a shift *pattern* can be modelled as follows:

$$S_t = \begin{cases} 0 & t < \tau \\ \varphi & t \geq \tau \end{cases} \quad \forall t = 1, 2, \dots \quad (2)$$

A quality control system is designed in order to perform binary discrimination between natural and unnatural classes of data. Two performance measures are used.

- The first is the ability to model the common causes of variation without creating Type I errors (i.e. false alarms), which indicate that the process is out of control when it is in fact not. This property is experimentally measured by reporting the mean of Type I errors (i.e. the sample mean of the alarm signals) occurring in process data only having natural sources of variation. This value is a consistent point estimator of the parameter  $\alpha = P\{H_1|H_0\}$  i.e. the expected probability that the control system signals an alarm when the process is in the natural state.
- The second is the ability to detect unnatural patterns in process output data. This property is measured experimentally by reporting the mean of Type II errors (i.e. the non-alarm signals) when a special disturbance has been introduced in process data. This value is a consistent point estimator of the parameter



$\beta = P\{H_0|H_1\}$ , i.e. the expected probability that the control system signals no alarm although the process is actually out of control. Generally, the objective of any quality control system is to detect changes of the process parameters as fast as possible (a small Type II error rate), without too many false alarms (a small Type I error rate).

#### 4. Outline of the proposed Fuzzy ART system

Fig. 1 shows the proposed neural system for quality control and the manufacturing process model. Let  $\{Y_t\}$  be the random sequence of the observed quality characteristic ( $t = 1, 2, \dots$ ). The control system accepts as input the process output  $Y_t$ , and it produces the output binary signal  $b_{nn,t}$ , which will be set to  $b_{nn,t} = 1$  if the process is considered in a natural state,  $b_{nn,t} = 0$  otherwise.

As showed by Fig. 1, some data pre-processing stages take place before they are presented to the Fuzzy ART neural network. The first stage is called the *Window Forming*. Through it, the most recent  $M$  observations are collected, to form the network  $M$ -dimensional input vector. Denoting as  $\underline{Y}_t$  the output of the *Window Forming* stage, at each time of index  $t \geq M$  it can be defined as

$$\underline{Y}_t = [Y_{t-M+1}, Y_{t-M+2}, \dots, Y_{t-1}, Y_t] \quad t \geq M. \quad (3)$$

In the most diffused literature, the parameter  $M$  is referred to as the *Window Size* of the quality control neural system (Cheng, 1995, 1997; Cheng and Cheng, 2001).

The second pre-processing stage (*Coding*) takes an  $M$ -dimensional input pattern  $\underline{Y}_t$  and transforms it into the corresponding  $M$ -dimensional output vector (say  $\underline{I}_t$ ) whose components fall into the interval  $[0,1]$ . This pre-

processing stage consists of a linear re-scaling of the input variable into the range  $[0,1]$ . Denoting with  $l > 0$  a proper limit for the absolute variation of the process output values  $\{Y_t\}$  from the nominal mean (in the reference test case, the parameter  $l$  is equal to  $l = 3$ ), the *Coding* pre-processing is described by the following Eq. (4):

$$\underline{I}_t = [I_{t-M+1}, I_{t-M+2}, \dots, I_{t-1}, I_t] \quad t \geq M, \quad (4)$$

$$\begin{cases} I_\tau = 0, & Y_\tau < -l; \\ I_\tau = \frac{1}{2} \left( 1 + \frac{Y_\tau}{l} \right) & -l \leq Y_\tau \leq l; \\ I_\tau = 1, & l < Y_\tau; \end{cases} \quad t - M + 1 \leq \tau \leq t,$$

Besides comparison (F1) and recognition (F2) layers, a complementary field (F0), has been implemented to form the attentional subsystem. In the layer F0, complement coding on the incoming input vectors  $\underline{I}_t = [I_{t-M+1}, I_{t-M+2}, \dots, I_{t-1}, I_t]$  is implemented. At each time of index  $t \geq M$ , the F0 field accepts an  $M$ -dimensional input vector  $\underline{I}_t$ , and it produces a  $2M$ -dimensional output vector  $\underline{I}_t^c$  towards the F1 layer. Denoting  $\underline{1}$  as the  $M$ -dimensional all-one vector, the following schema can be used for *complement coding*:

$$\underline{I}_t^c = (\underline{I}_t, \underline{1} - \underline{I}_t) = [I_{t-M+1}, I_{t-M+2}, \dots, I_t, 1 - I_{t-M+1}, 1 - I_{t-M+2}, \dots, 1 - I_t]. \quad (5)$$

The F1 and F2 fields are fully connected by weighed links. The  $2M$ -dimensional vector, whose components are the weights that connect node  $j$  of F2 field to each node of F1, is designated by  $\underline{w}_j^c$ . The orienting subsystem consists of a single node called the *reset node*. The reset node output, which depends on the vigilance parameter  $\rho$ , affects the nodes of the F2 layer.

In quality control applications, the operating phases of the Fuzzy ART neural network can be carried out as follows.

For training, it is assumed that process output patterns (the training list) are available. Commonly, such data are the measurements of a quality parameter of interest when only natural causes of variation are in effect. The number of patterns in the training list, hereafter called the *learning period* (Al-Ghanim, 1997), depends on the time length of the process output series that is used for neural network training. The underlying assumption is that in the learning period the process under inspection produces only natural outputs that are clustered into categories by means of the Fuzzy ART network. Obviously, the neural network will cluster patterns that are similar to each other in the same category. The ART algorithm allows user to control the degree of similarity of patterns placed in the same cluster. The meaning of similarity depends on the

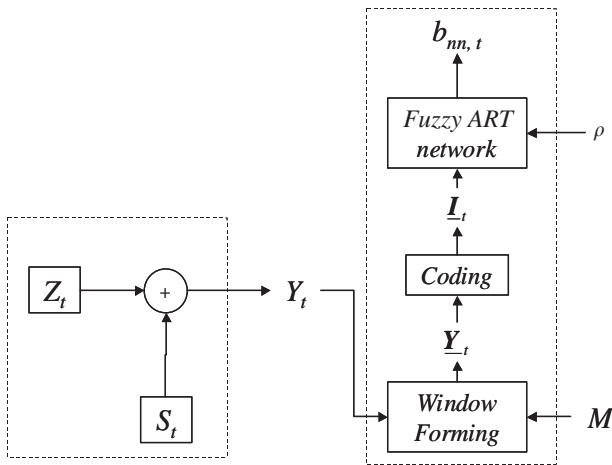


Fig. 1. A general manufacturing system model and the proposed neural system for quality control.

vigilance parameter  $\rho$  that is used in the training phase. Higher vigilance imposes a stricter matching criterion that separates input patterns into finer categories. In contrast, lower vigilance tolerates greater mismatches, and it produces coarse categories. During training, the maximal vigilance parameter enables Fuzzy ART to classify input patterns into the highest recognition rate. In particular, if the vigilance assumes the maximum value, then the neural network forms categories that duplicate the specific patterns used for training (*Perfectly Learned Patterns—PLP*). A cluster that has been formed during the PLP training phase cannot change in the subsequent list presentation; consequently, the sufficient number of list presentations to assure the convergence of training phase, can be reduced to one (Huang et al., 1995).

In the testing phase, the learning process is disengaged (i.e. no more weights adaptations or cluster creations are allowed) and patterns from a test list are presented in order to evaluate neural network performances. A pattern from the test list is presented to the neural network, and through the Fuzzy ART algorithm, a category is chosen if it is found appropriate to represent the input. The criterion to determinate how well a category represents the cluster, to which the input pattern belongs, depends on the vigilance value chosen for the testing phase (that can be different from the value used in the training one). In particular, higher vigilance imposes a stricter matching criterion, while lower vigilance tolerates greater mismatches between input patterns and the ART categories.

## 5. Training and testing of the proposed Fuzzy ART system

Before discussing the Fuzzy ART algorithm in more detail, some preliminary notations must be introduced. From now on, the size of vector  $\mathbf{x}$  will be denoted as:  $|\mathbf{x}| = \sum_i |x_i|$ . The minimum between vectors  $\mathbf{x}$  and  $\mathbf{y}$  is defined as  $\mathbf{x} \wedge \mathbf{y} = [\min(x_1, y_1), \dots, \min(x_i, y_i), \dots]$ , while the maximum as  $\mathbf{x} \vee \mathbf{y} = [\max(x_1, y_1), \dots, \max(x_i, y_i), \dots]$ . The distance between  $\mathbf{x}$  and  $\mathbf{y}$  vectors is defined as  $\text{dis}(\mathbf{x}, \mathbf{y}) = |\mathbf{x} \vee \mathbf{y}| - |\mathbf{x} \wedge \mathbf{y}|$ . Using the definition of min and max operators, the distance function can be rewritten as  $\text{dis}(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$ .

### 5.1. Training phase

During training, the maximal vigilance is used (PLP). In Appendix A, a detailed description of the PLP training algorithm has been reported. It can be noticed that the PLP learning approach causes that each input pattern is stored in a distinct category during training. A very important and convenient characteristic of the proposed learning approach is the short training time.

Indeed, a cluster that has been formed during the PLP training phase cannot change in the subsequent list presentation and, consequently, the number of list presentation can be reduced to one. Furthermore, the sequence of training presentation has not influence on the neural network training, as each pattern is stored in a specific category.

Using the maximum vigilance during training, the others network parameters (the choice parameter and learning rate) have no influence on learning (Huang et al., 1995), as once a category is formed, it cannot be modified during subsequent training iterations (the maximum vigilance causes a reset signal if a committed node wins the competition in the F2 layer during training). Each new training pattern is clustered into an uncommitted node, and the neural network forms a number of categories equal to the number of different training patterns. Consequently, each top-down vector (or category) can be also expressed in the following form  $\mathbf{w}_j^c = (\mathbf{w}_j, \mathbf{1} - \mathbf{w}_j)$ , where  $\mathbf{w}_j$  is an  $M$ -dimensional pattern and  $\mathbf{1}$  is the  $M$ -dimensional “all-one” vector. An important consequence of the latter result is that the size of vectors  $\mathbf{w}_j^c$  is equal to  $M$ , indeed  $|\mathbf{w}_j^c| = |(\mathbf{w}_j, \mathbf{1} - \mathbf{w}_j)| = |\mathbf{w}_j| + M - |\mathbf{w}_j| = M$ .

### 5.2. Testing phase

Fig. 2 depicts the basic architecture of Fuzzy ART neural network, which has been implemented by means of the NeuralWorks Professional II Plus<sup>®</sup> software environment (NeuralWare, 1997).

Assume that at time of index  $t \geq M$  an  $M$ -dimensional input pattern  $\mathbf{I}_t$  is presented at the F0 field. The appearance of the  $2M$ -dimensional pattern  $\mathbf{I}_t^c$  across the F1 field produces bottom-up inputs that affect the nodes in the F2 layer. These bottom-up inputs are given by the following Eq. (6) for indexes  $1 \leq j \leq N$  and  $\alpha$  arbitrarily chosen in the range  $(0, +\infty)$ .

$$T_j(\mathbf{I}_t^c) = \frac{|\mathbf{I}_t^c \wedge \mathbf{w}_j^c|}{\alpha + M}. \quad (6)$$

The bottom-up inputs activate a competition process among the F2 nodes, which eventually leads to the activation of a single node in F2, namely the node that receives the maximum bottom-up input form F1. Let  $J$  be the index of the node in the F2 layer that maximises the choice function  $T_j(\mathbf{I}_t^c)$ , i.e. the vector  $\mathbf{w}_J^c$  is assumed as the representative category for the input pattern  $\mathbf{I}_t^c$ . The appropriateness of the natural category to represent the input pattern is checked by comparing the ratio of Eq. (7), to the vigilance parameter  $\rho$  of the testing phase, which assumes values in the range  $(0, 1)$ .

$$\frac{|\mathbf{I}_t^c \wedge \mathbf{w}_J^c|}{M}. \quad (7)$$

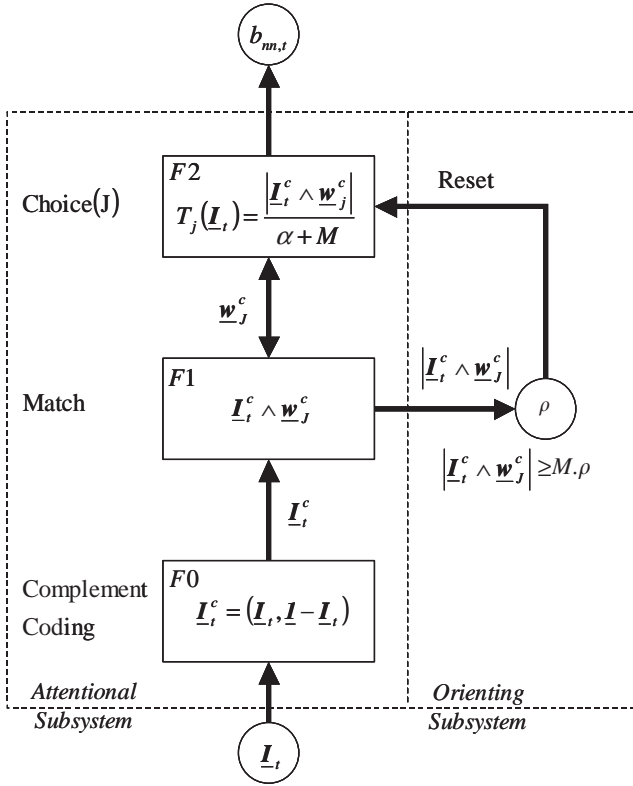


Fig. 2. The proposed Fuzzy ART neural network for manufacturing quality control.

If such ratio is not less than the vigilance parameter  $\rho$ , then the output is set to  $b_{nn,t} = 1$  (natural input), otherwise, the algorithm produces the output  $b_{nn,t} = 0$  (unnatural input). Since it results that

$$\begin{aligned}
 |I_t^c \wedge w_J^c| &= |(I_t, \mathbf{1} - I_t) \wedge (w_J, \mathbf{1} - w_J)| \\
 &= |[I_t \wedge w_J, (\mathbf{1} - I_t) \wedge (\mathbf{1} - w_J)]|, \\
 |I_t^c \wedge w_J^c| &= |[I_t \wedge w_J, \mathbf{1} - (I_t \vee w_J)]| \\
 &= |I_t \wedge w_J| + M - |I_t \vee w_J|, \\
 |I_t^c \wedge w_J^c| &= M - \text{dis}(I_t, w_J).
 \end{aligned}$$

It may be noted that the node  $J$ , whose top-down weight vector  $w_J^c = (w_J, \mathbf{1} - w_J)$  maximises the choice function (6), is also that node which minimises the distance  $\text{dis}(I_t, w_J)$ . Thus, an input pattern is classified as natural if the following condition (8) is passed:

$$\frac{|I_t^c \wedge w_J^c|}{M} \geq \rho \Leftrightarrow \text{dis}(I_t, w_J) \leq M(1 - \rho). \quad (8)$$

## 6. Experiment and analysis

Three parameters can affect the performances of the proposed neural system for quality control.

1. The vigilance parameter  $\rho$  of testing phase, it can take values in the range (0,1).

2. The window size  $M$ , it can take values in  $\{1, 2, 3, \dots\}$ , i.e. any non-zero integer.
3. The training period  $d$ , i.e. the number of training patterns.

In order to estimate their effects on the system performances, which are measured in terms of both Type I error and Type II error rates, a complete experimental design was used (Montgomery, 1997). For simplicity, each of the three factors was evaluated by means two proper levels (high and low), thus eight experimental scenarios was analysed.

Both a number of 15 training data sets and of 15 different testing data sets were produced by means of Monte Carlo simulation. To make this random generation different for all data sets, a different seed of the MATLAB<sup>®</sup> pseudo-random generator was specified for each distinct training or testing series. Simulation replications were obtained combining each training data set to all the testing sets. Thus, a number of  $15 \times 15 = 225$  different replications, for each of the eight experimental scenarios, were obtained.

Each training data set included a number of  $d$   $M$ -dimensional vectors, while testing data sets were formed by means of 10000  $M$ -dimensional vectors. Training data sets, as well as testing data sets, used to estimate Type I errors, were simulated by means of the normal distribution function (mean  $\mu = 0$  and standard deviation  $\sigma = 1$ ). On the other hand, the testing data sets used to estimate Type II errors, were simulated through a shift pattern (Eq. (2)) of 1.5 units of standard deviation ( $\varphi = 1.5$ ), and starting point in the fifth observation ( $\tau = 5$ ).

Before presenting simulation results, let us elaborate on the experiment model in more detail. The relationship between the Type I error and the network parameters can be formulated as follows.

$$\alpha(\rho, M, d) = f(\rho, M, d) + e(\rho, M, d), \quad (9)$$

where the function  $f(\rho, M, d)$  is the expected Type I error level for the parameter combination  $(\rho, M, d)$ , and  $\alpha(\rho, M, d)$  is the actual simulation measure. The actual measure is affected by an error  $e(\rho, M, d)$  that can be considered as an occurrence of a random variable with mean zero. Three sources of variability can affect the random error  $e(\rho, M, d)$ : the training set, the testing set and the interaction between them. The effect of training data set on the actual measure  $\alpha(\rho, M, d)$  is designated as  $e_t(\rho, M, d)$ , while the effect of testing data set is designated as  $e_r(\rho, M, d)$ . Therefore, denoting  $e_{tr}(\rho, M, d)$  the effect of the interaction between the two factors, Eq. (9) can be rewritten as follows:

$$\begin{aligned}
 \alpha(\rho, M, d) &= f(\rho, M, d) + e_t(\rho, M, d) \\
 &\quad + e_r(\rho, M, d) + e_{tr}(\rho, M, d). \quad (10)
 \end{aligned}$$

Denoting with symbols  $\sigma_i^2(\rho, M, d)$ ,  $\sigma_{ii}^2(\rho, M, d)$ ,  $\sigma_{li}^2(\rho, M, d)$  the variance values of random components  $e_i(\rho, M, d)$ ,  $e_l(\rho, M, d)$  and  $e_{li}(\rho, M, d)$ , respectively, then the variance of  $\alpha(\rho, M, d)$  can be rewritten as  $\sigma_\alpha^2(\rho, M, d) = \sigma_i^2(\rho, M, d) + \sigma_l^2(\rho, M, d) + \sigma_{li}^2(\rho, M, d)$ . A similar model can be formulated for the Type II error, so the following Eq. (11) can be used in this case:

$$\beta(\rho, M, d) = g(\rho, M, d) + e_L(\rho, M, d) + e_T(\rho, M, d) + e_{LT}(\rho, M, d). \quad (11)$$

That implies  $\sigma_\beta^2(\rho, M, d) = \sigma_L^2(\rho, M, d) + \sigma_T^2(\rho, M, d) + \sigma_{LT}^2(\rho, M, d)$ .

6.1. Analysis

As already mentioned, to find out how the three parameters ( $\rho, M, d$ ) influence network performances, both the Type I and Type II error rates have been estimated in eight scenarios. Two window sizes ( $M = 10-25$ ), two levels of the vigilance parameter ( $\rho = 0.85-0.90$ ) and two learning periods ( $d = 100-1000$ ), have been used. The resulting Type I error rates have been presented in Table 1.

Following Fig. 3 shows the main-effect plots for each experimental factor.

It results that as the vigilance parameter increases, the Type I error rate increases too. Similarly, a higher window size causes the Type I error to increase. In contrast, a long training period produces a smaller Type I error. It can be noticed that variations of the vigilance parameter and of the window size can have considerable effects on Type I errors, while changes of the training period have only slight influence on the false alarm rate. Moreover, the Analysis of Variance (ANOVA—Montgomery, 1997) of the simulation results, underlines that there is considerable interaction between the vigilance parameter  $\rho$  and the window size  $M$ , in terms of their effects on Type I errors.

The eight standard-deviation point estimators  $\hat{\sigma}_\alpha(\rho, M, d)$  of Table 1 measure the spread of performances around the false alarm point-estimators  $\hat{f}(\rho, M, d)$ . It results that changes in the training data set, as well as in the testing data set, have not important

effects on Type I errors. Nevertheless, it can be noticed that short learning period ( $d = 100$ ) can cause higher influence of the training data set on the false alarm rate presented by the neural system.

Type II errors are reported in following Table 2, while the main effect plots are depicted in Fig. 4.

In this case, it can be deduced that as the vigilance parameter increases, the Type II error decreases. Similarly, a larger window size causes the decrease of Type II error, while a longer training period produces slightly higher Type II error. By comparing plots of Figs. 3 and 4, it appears that the effects of the three parameters on the Type II error are smaller than those presented on Type I error. As for Type I error measures, the Analysis of Variance shows that there is significant interaction between the vigilance parameter and window size in terms of Type II error as well.

Results of Table 2 reveal that changes of training/testing data set can have valuable effects on Type II errors. Moreover, it can be observed that the variability, which is due to changes of training set, is higher than that obtained varying the testing set. In other words, it appears that the spread of the neural network performances in terms of Type II error mainly depends on the training data set rather than on the choice of the testing data set.

To analyse in more detail the experimental results, and how the vigilance parameter  $\rho$  affects network

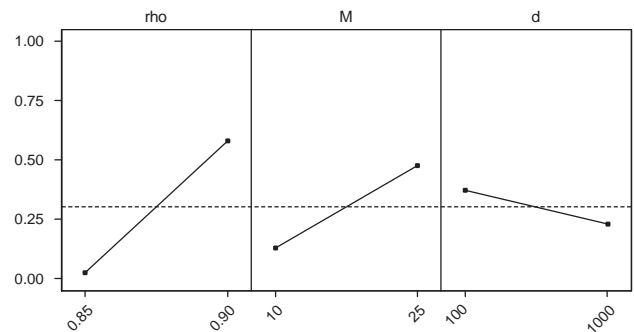


Fig. 3. Type I main effect plot. Simulation results based on 225 replications of 10,000 trials for each of the eight scenarios. The horizontal line represents the mean value of the Type I error.

Table 1  
Type I errors and standard deviations. Simulation results based on 225 replications of 10,000 trials for each of the eight scenarios

$\rho$	$M$	$d$	$\hat{f}(\rho, M, d)$ (%)	$\hat{\sigma}_i(\rho, M, d)$ (%)	$\hat{\sigma}_l(\rho, M, d)$ (%)	$\hat{\sigma}_{li}(\rho, M, d)$ (%)	$\hat{\sigma}_\alpha(\rho, M, d)$ (%)
0.85	10	100	1.100	0.197	0.187	0.136	0.304
0.85	10	1000	0.027	0.004	0.017	0.018	0.025
0.85	25	100	7.656	2.293	0.898	0.509	2.515
0.85	25	1000	0.337	0.031	0.134	0.084	0.161
0.90	10	100	42.276	3.773	1.191	0.671	4.013
0.90	10	1000	7.661	0.189	0.662	0.292	0.748
0.90	25	100	97.906	1.090	0.187	0.319	1.151
0.90	25	1000	84.036	1.781	1.018	0.604	2.138



Table 2

Type II errors and standard deviations. Simulation results based on 225 replications of 10,000 trials for each of the eight scenarios

$\rho$	$M$	$d$	$\hat{g}(\rho, M, d)$ (%)	$\hat{\sigma}_L(\rho, M, d)$ (%)	$\hat{\sigma}_T(\rho, M, d)$ (%)	$\hat{\sigma}_{LT}(\rho, M, d)$ (%)	$\hat{\sigma}_\beta(\rho, M, d)$ (%)
0.85	10	100	26.970	20.174	0.829	0.547	20.198
0.85	10	1000	57.938	11.517	1.042	0.632	11.581
0.85	25	100	1.101	3.533	0.089	0.301	3.548
0.85	25	1000	4.298	3.998	0.531	0.387	4.052
0.90	10	100	2.238	3.486	0.139	0.237	3.497
0.90	10	1000	8.114	3.405	0.477	0.406	3.461
0.90	25	100	0.000	0.000	0.000	0.001	0.001
0.90	25	1000	0.001	0.001	0.000	0.004	0.004

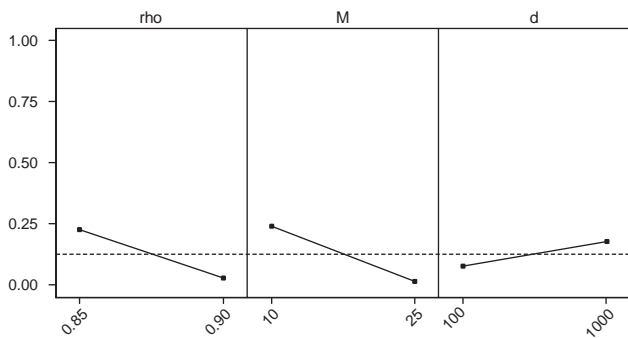


Fig. 4. Type II main effect plot. Simulation results based on 225 replications of 10,000 trials for each of the eight scenarios. The horizontal line represents the mean value of the Type II error.

performances, consider the plots shown in Fig. 5. Each graph depicts the distributional characteristics of experimental results. The box-plots (also called box-and-whisker plots) of either Type I or Type II error rates with each combination of the parameters  $M = 10-25$ ,  $\rho = 0.85-0.90$  and  $d = 100-1000$ .

Simulation results demonstrate that the most important effects on the Type I error and Type II error rates derive from the vigilance parameter, the window size and their interaction as well. In contrast, larger training period can have only slight effects on the Type I error and Type II error rates. Nevertheless, large values of  $d$  can reduce the variability of neural network performances, especially when a large window size is used. In order to confirm such result, Fuzzy ART neural network has further been tested using both a larger window size and a higher training period. In the following Table 3 the Type I error and Type II error estimators have been reported in the case of  $M = 75$  and  $d = 7500$ , over a number of 225 replications of 10,000 trials produced by means of Monte Carlo simulation. As already mentioned, replications have been obtained using combinations of 15 training data sets and 15 different testing data sets.

In this case, it can be observed that both training and testing data sets have no influence on the neural network performances in terms of Type II errors. Indeed, the Type II error is equal to its lower limit  $\hat{\beta} = 0\%$

constantly. On the other hand, the influence of the training set on the Type I error results either small ( $\rho = 0.85$ ), or null ( $\rho = 0.90$ ). In this case, the huge training period  $d$ , and the large windows size  $M$  make the network not easily influenced by changes of training data set.

## 6.2. Selection of neural network parameters

Even if the optimal combination of the network parameters depends on economic factors related to Type I error and Type II error costs, a reasonable strategy to use the proposed system can be obtained using following remarks:

First, the *window forming* pre-processing stage implies that the neural network cannot release a signal on the process state before the first  $M$  parts have been produced. Thus, if a process were in an unnatural state due an incorrect set up operation, at least  $M - 1$  out-of-control outputs could be produced before a signal is emitted by the neural network. Consequently, large values of *window size* can increase scrap and rework costs. On the other hand, simulation results have demonstrated that larger window sizes are recommended in order to reduce Type II errors. Thus, the choice of  $M$  should be based on the minimum change in the mean that is important to detect, in the sense that lower disturbances in the process requires higher window sizes, in order to be detected efficiently.

The duration of training period depends on the availability of process natural outcomes. About 1000 data have shown to be appropriate to train the neural network in the reference manufacturing process. However, when the number of natural samples is limited to form an adequate training data set, bootstrap sampling can be used (Wehrens et al., 2000). Bootstrapping sampling scheme can be adopted to generate many samples from limited observations by pursuing the fact that these samples could represent an underlying process distribution.

Finally, it has been observed that the vigilance parameter has a great influence on Type I error; hence, it can be used to control the false alarm level. For

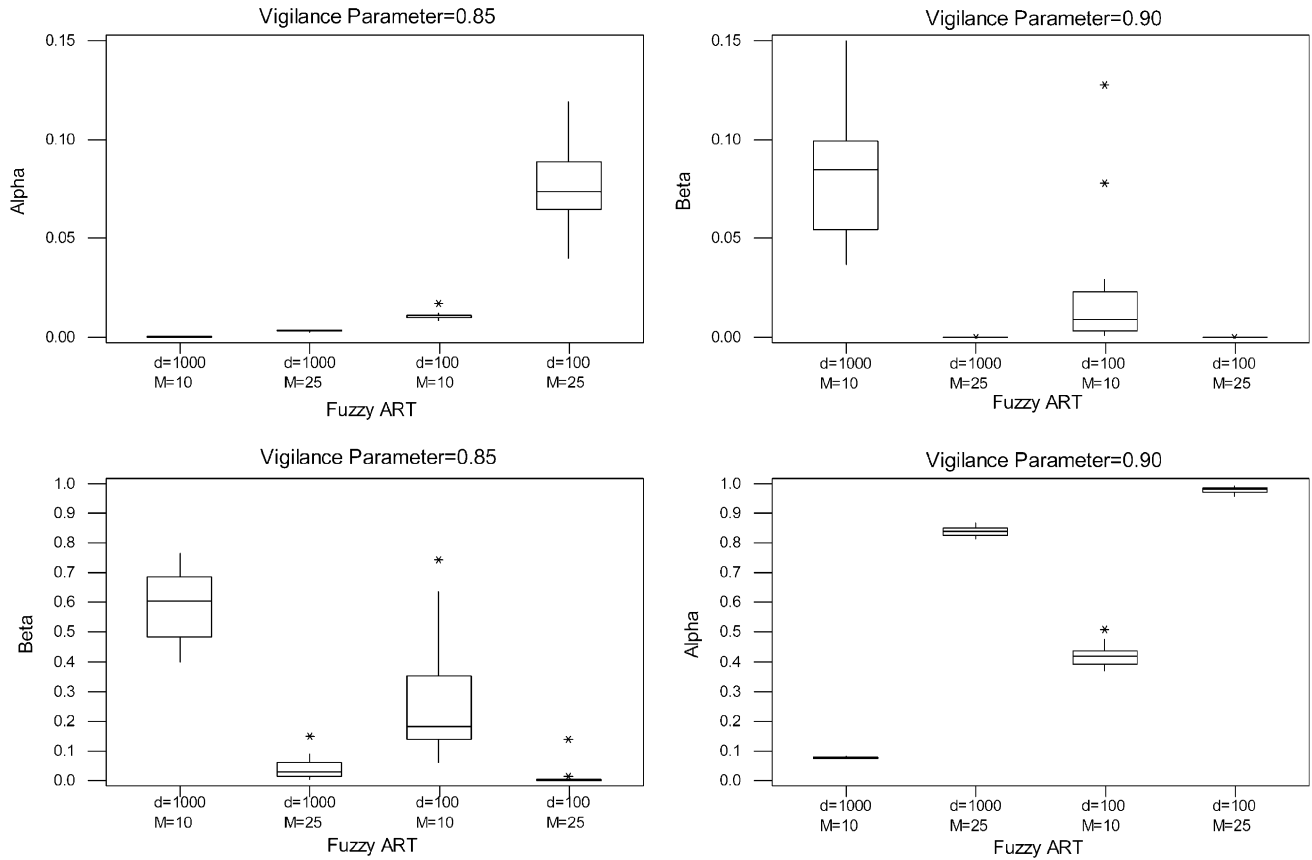


Fig. 5. Box-plots of experimental results. Simulation results based on 25 replications of 10,000 trials for each value of the vigilance parameter. The bottom of each box is at the first quartile, and the top is at the third quartile value. The central line is at the median value.

Table 3

Type I and Type II errors and standard deviations, for  $M = 75$  and  $d = 7500$ . Simulation results based on 225 replications of 10,000 trials for each value of the vigilance parameter

$\rho$	$\hat{f}(\rho, M, d)$ (%)	$\hat{\sigma}_I(\rho, M, d)$ (%)	$\hat{\sigma}_T(\rho, M, d)$ (%)	$\hat{\sigma}_{II}(\rho, M, d)$ (%)	$\hat{\sigma}_z(\rho, M, d)$ (%)
0.85	3.472	0.267	0.974	0.405	1.088
0.90	100.00	0.000	0.000	0.000	0.000
$\rho$	$\hat{g}(\rho, M, d)$ (%)	$\hat{\sigma}_I(\rho, M, d)$ (%)	$\hat{\sigma}_T(\rho, M, d)$ (%)	$\hat{\sigma}_{LT}(\rho, M, d)$ (%)	$\hat{\sigma}_\beta(\rho, M, d)$ (%)
0.85	0.000	0.000	0.000	0.000	0.000
0.90	0.000	0.000	0.000	0.000	0.000

example, once the window size has been fixed (depending on the Type II error), the vigilance parameter can be adjusted in order to obtain the appropriate Type I error rate from the neural system.

Therefore, a strategy for applying the proposed Fuzzy ART control system in real applications can be obtained by means of the following steps:

- Identify the minimum variation of the process-mean, which it is important to detect quickly (say  $\varphi$  in terms of units of standard deviations).
- Choice the window size  $M$  as a function of the smaller standardises change in the mean considered important to detect quickly (say  $\varphi$ ). Usually, smaller

$\varphi$  requires higher window size, thus a rule of thumb is to choice  $M = m/\varphi$  ( $m = 75$  is the recommended value).

- Use a set of representative process natural data for network training.
- Adjust the value of the vigilance parameter in order to obtain a predefined Type I error rate (*tuning phase*), indeed higher vigilance imposes higher Type I errors, lower vigilance impose lower Type II errors.

### 6.3. Over and under training

The correct training level of the proposed ART-based approach is only influenced by the number of training

patterns. The avoidance of both under-training and over-training is an important point for every neural network application (including quality monitoring). Over-training can result when, due to excessive exposure to the training data set, the network loses its ability to generalise and can only recognise the training data. Under-training can occur when training set is of such small size that the number of training patterns is inadequate to cover the complete state-space of the process natural behaviour. The performance of the proposed ART neural network does not suffer from over-training phenomena, but only from under-training.

The choice of both vigilance parameter and training period can be derived from the maximum false alarm rate that can be accepted in the actual application (i.e. Type I error rate). This section explains the method that determines the vigilance parameter  $\rho$  and the training period  $d$  for a given window size  $M$ . The goal is to obtain a predefined false alarm rate and, at the same time, to maximise the recognition rate of unnatural process data.

To find out appropriate values of vigilance and learning period for a given window size, the false alarm rate of Fuzzy ART is evaluated on batches of quality measurements produces by the process in a natural state. As previously observed, the false alarm rate is a strictly increasing function on the vigilance parameter, and it is a decreasing function on the training period. Therefore, a suitable combination of the two parameters can be found by iteratively increasing the learning period and decreasing the vigilance parameter in order to obtain the required false alarm rate. The method for finding the values of learning period and vigilance of the network is as follows:

1. Initialise: define the batch dimension of natural patterns, which will form an initial training set (say  $d$ ), an upper bound of vigilance parameter (say  $0 < \rho \leq 1$ ), a vigilance step-variation (say  $h > 0$ ), and an upper bound for false alarm rate (say  $\alpha_{\max}$ ).
2. Collect from the process in a state of natural behaviour a set of data of length  $d + M$ . Use this set for network training by PLP approach (vigilance equal to 1).
3. Collect from the process in a state of natural behaviour a new set of data of length  $d + M$ . Evaluate the false alarm rate on this new set of data (say  $\alpha$ ) setting the vigilance parameter equal to  $\rho$ .
4. Test terminal criteria: if  $\alpha \leq \alpha_{\max}$  then set the vigilance parameter to  $\rho$  and the training period to  $d$ .
5. Upgrade the neural network training (incremental learning mode of ART algorithm) by using the testing data set of step 3, and the PLP approach. After training, set a new (smaller) value of vigilance upper bound by setting  $\rho = \rho - h$ . Go to step 3.

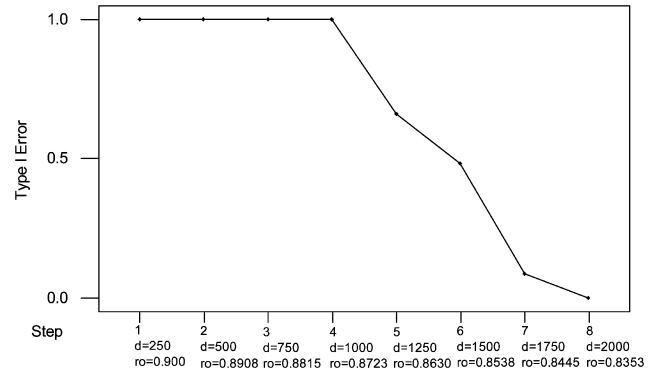


Fig. 6. Type I error vs. step number for the configuration method that determines the vigilance parameter  $\rho$  and the training period  $d$  for a given window size  $M$ .

Table 4

Type I and Type II errors and standard deviations.  $M = 75$ ,  $\rho = 0.8353$  and  $d = 2000$

$\rho$	$M$	$d$	$\phi$	$\hat{f}(\rho, M, d)$ (%)	$\hat{\sigma}_z(\rho, M, d)$ (%)
0.8353	75	2000	0.0	0.277	0.270
$\rho$	$M$	$d$	$\phi$	$\hat{g}(\rho, M, d)$ (%)	$\hat{\sigma}_\beta(\rho, M, d)$ (%)
0.8353	75	2000	0.5	92.965	3.151
0.8353	75	2000	1.0	8.291	5.241
0.8353	75	2000	1.5	0.000	0.000

This algorithm has been applied for finding  $\rho$  and  $d$  values of a neural network with window size  $M = 75$ . The reference manufacturing process has been simulated by using the model of Section 4. The initial setting parameters has been chosen as follows:  $d = 250$ ,  $\rho = 0.9$ ,  $\alpha_{\max} = 0.27\%$ ,  $h = 0.00925$ . The procedure stopped after nine iterations and the following configuration of the neural network has been obtained:  $d = 2000$ ,  $\rho = 0.8353$ . Fig. 6 depicts the Type I error rate produced by the neural network in each step of the algorithm.

In Table 4 the Type I error and Type II error estimators have been reported in the case of  $M = 75$ ,  $\rho = 0.8353$  and  $d = 2000$ . As already mentioned, replications have been obtained using combinations of 15 training data sets and 15 different testing data sets. Fig. 7 depicts the box-plots of the experimental results.

## 7. Concluding remarks

With the advanced data-collection systems (e.g. coordinate measure machine, machine vision system and scanning laser system), and the movements towards computer-integrated manufacturing (CIM), a large amount of measurements on each product can be obtained at the point of manufacture. In this new manufacturing scenario, a demand exists for the automation of SPC implementation. New methods

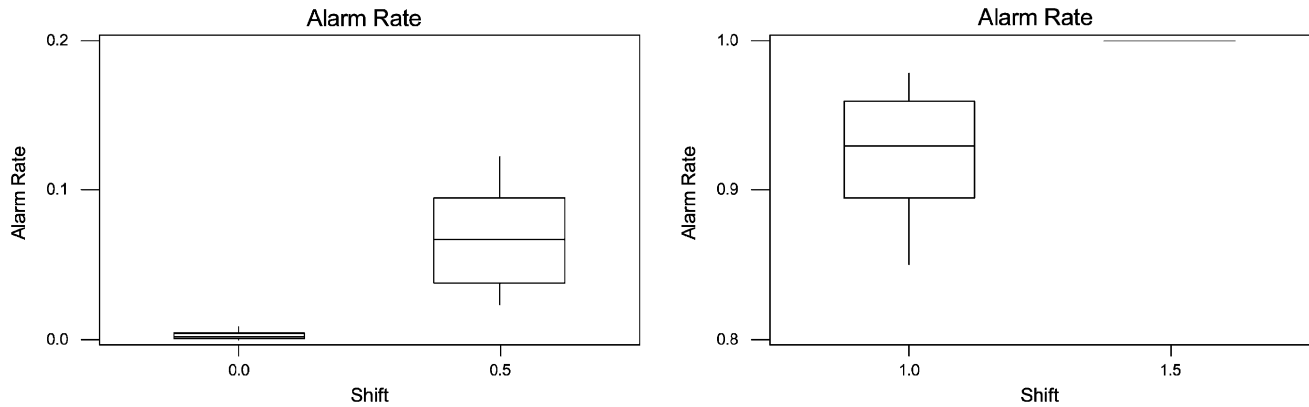


Fig. 7. Box-plots of experimental results.  $M = 75$ ,  $\rho = 0.8353$  and  $d = 2000$ . The bottom of each box is at the first quartile, and the top is at the third quartile value. The central line is at the median value.

should be developed in order to accommodate rapid data inputs rates, to utilise data from each product manufactured and, in general, to take advantage of this newly available data.

In this work, a new neural technique has been investigated as promising tool for the above-mentioned purposes. In particular, a Fuzzy ART neural system has been presented for manufacturing quality monitoring. The proposed neural system is mainly intended for identifying unnatural process behaviour by detecting changes in the state of the process.

Several characteristics of the Fuzzy ART neural network make it a practical tool for quality control applications over supervised ones. The most evident is the short learning time required by the proposed approach. Since the maximum vigilance parameter has been exploited for network learning, the number of list presentations required to achieve a training of the network can be reduced to only one. Once a pattern is presented to the neural network, it is stored into a specific category and no more iteration is required to learn it. As an example of computational time, by means the proposed PLP learning approach, the neural network requires around 2 min to perform training for 7500 patterns of 75 points on a Pentium III 800 MHz.

Potential industrial applications of the proposed approach are continuous product manufacturing operations, including the manufacture of paper and wood products, chemicals, and cold-rolled steel products. In these cases, with the widespread exploitation of automated production and inspection, the tasks of SPC traditionally performed by quality practitioners have to be automated. Computer-based algorithms need to be developed to implement the various quality control tasks automatically. In addition, data collected at regular time intervals, by automatic sensors, can be serially correlated. Consequently, the strength of traditional SPC control charts to identify the presence of assignable causes is weakened or invalidated.

The main advantage of the proposed system over others neural techniques is that it does not require previous information about unnatural pattern appearances, related mathematical models, or probability distribution functions. The proposed system can be potentially adopted to signal any kinds of unnatural pattern, so it provides a powerful diagnostic tool for detecting assignable causes in real processes. Therefore, the proposed method can be potentially used for identifying changes in the process parameters in the presence of correlation and it allows improved process control in continuous manufacturing operations.

There are three main possible directions for future research. First, the effect of departures from normality and independence for the reference manufacturing process must be thoroughly investigated. Second, the neural network system can be improved in order to recognise not just generic unnatural data, but also specific unnatural patterns. Third, the applicability of the proposed method can be extended and analysed in the case of multivariate processes.

## Acknowledgements

This research was partially funded by the Ministry for Instruction, University and Research (MIUR) of Italy, and the National Research Council of Italy (CNR). The authors would like to acknowledge the helpful comments of the reviewer.

## Appendix. A

Given a sequence of training process natural output designated as  $\{l_t\}$ , let

$$\{\underline{L}_1, \underline{L}_2, \dots, \underline{L}_d\} = \{[l_1, l_2, \dots, l_M], [l_2, l_3, \dots, l_{1+M}], \dots, [l_d, l_{d+1}, \dots, l_{d+M}]\} \quad (\text{A.1})$$



be the sequence of the  $M$ -dimensional training vectors, which result from the *window coding* stage. Denoting by  $d$  (the *learning period*) the number of the  $M$ -dimensional training patterns, let  $\{\underline{I}_1, \underline{I}_2, \dots, \underline{I}_d\}$  be the series of  $M$ -dimensional training vectors that result from the *Coding* stage, and let  $\{\underline{I}_1^c, \underline{I}_2^c, \dots, \underline{I}_d^c\}$  be the  $2M$ -dimensional training vectors that result from the *complement coding* stage at the F0 layer. The following algorithm describes the systematic implementation of the PLP learning approach for Fuzzy ART training.

1. Initialise the vigilance parameter to  $\rho = 1$  and the number of committed nodes in F2 to  $N = 0$  (a committed node in F2 is a node that has coded at least one input pattern; during the training phase, Fuzzy ART operates over all of the committed nodes along with a single uncommitted node).
2. Choose the  $n$ th input pattern ( $\underline{I}_n^c$ ) from the training list ( $1 \leq n \leq d$ ).
3. Calculate the bottom-up inputs to the  $N + 1$  nodes in F2 due to the presentation of the  $n$ th input pattern. When calculating bottom-up inputs consider all the committed nodes and the uncommitted node. These bottom-up inputs are calculated according to the following Eq. (A.2) where  $1 \leq j \leq N + 1$  and  $\alpha$  (the *choice parameter*) is arbitrarily chosen in the range  $(0, +\infty)$ .

$$T_j(\underline{I}_n^c) = \begin{cases} \frac{M}{\alpha + 2M} & \text{if } j \text{ is the uncommitted node,} \\ \frac{|\underline{I}_n^c \wedge \underline{w}_j^c|}{\alpha + |\underline{w}_j^c|} & \text{if } j \text{ is a committed node.} \end{cases} \quad (\text{A.2})$$

4. Chose the node in F2 that receives the maximum bottom-up input from F1. Assume that this node has index  $J$ . Check to see whether this node satisfies the vigilance criterion of Eq. (A.3).

$$|\underline{I}_n^c \wedge \underline{w}_J^c| = M \quad (\text{A.3})$$

Three cases are now distinguished.

- If node  $J$  is the uncommitted node, it satisfies the vigilance criterion since  $\underline{w}_J^c = [1, 1, \dots, 1] \Rightarrow |\underline{I}_n^c \wedge \underline{w}_J^c| = |\underline{I}_n^c| = M$ . In this case, increase the parameter  $N$  by one: this way a new uncommitted node in F2 is introduced, and its initial weight vector is chosen to be the “all-ones” vector. Go to step 5.
- If node  $J$  is a committed node, and the top-down weight vector is equal to  $\underline{w}_J^c = \underline{I}_n^c$ , then the vigilance criterion is satisfied since  $\underline{w}_J^c = \underline{I}_n^c \Rightarrow |\underline{I}_n^c \wedge \underline{I}_n^c| = |\underline{I}_n^c| = M$ . Go to step 5.
- Otherwise, exclude the node  $J$  by setting  $T_j(\underline{I}_n^c) = -1$ , and go to the beginning of Step 3.

5. The top-down weight vector of node  $J$  is set equal to  $\underline{w}_J^c = \underline{I}_n^c$ . If  $\underline{I}_n^c$  is the last input pattern in the training list ( $n = d$ ) then the learning process is considered completed. Otherwise, go to Step 2 to present the next in sequence input pattern by increasing the index  $n$  by one.

## References

- Al-Ghanim, A., 1997. An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. *Computers and Industrial Engineering* 32 (3), 627–639.
- Anagnostopoulos, G.C., Georgiopoulos, M., 2002. Category regions as a new geometrical concept in Fuzzy-ART and Fuzzy-ARTMAP. *Neural Networks* 15, 1205–1221.
- Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., Oxford.
- Cheng, C.S., 1995. A multi-layer neural network model for detecting changes in the process mean. *Computers and Industrial Engineering* 28 (1), 51–61.
- Cheng, C.S., 1997. A neural network approach for the analysis of control chart patterns. *International Journal of Production Research* 35 (3), 667–697.
- Cheng, C.S., Cheng, S.S., 2001. A neural network-based procedure for the monitoring of the exponential mean. *Computers and Industrial Engineering* 40, 309–321.
- Cook, D.F., Chiu, C.C., 1998. Using radial basis function neural networks to recognize shifts in correlated manufacturing process parameters. *IIE Transactions* 30, 227–234.
- Cook, D.F., Zobel, C.W., Nottingham, Q.J., 2001. Utilization of neural networks for the recognition of variance shifts in correlated manufacturing process parameters. *International Journal of Production Research* 39 (17), 3881–3887.
- Georgiopoulos, M., Fernlund, H., Bebis, G., Heileman, G.L., 1996. Order of search in Fuzzy ART and Fuzzy ARTMAP: effect of the choice parameter. *Neural Networks* 9 (9), 1541–1559.
- Georgiopoulos, M., Dagher, I., Heileman, G.L., Bebis, G., 1999. Properties of learning of a fuzzy ART variant. *Neural Networks* 12, 837–850.
- Guo, Y., Dooley, K.J., 1992. Identification of changes structure in statistical process control. *International Journal of Production Research* 30 (7), 1655–1669.
- Guh, R.S., Tannock, J.D.T., 1999. Recognition of control chart concurrent patterns using a neural network approach. *International Journal of Production Research* 37 (8), 1743–1765.
- Hagan, M.T., Demuth, H.B., Beale, M., 1996. *Neural Network Design*. PWS, Boston, MA.
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*, third ed. Prentice-Hall, Englewood Cliffs, NJ.
- Huang, J., Georgiopoulos, M., Heileman, J.L., 1995. Fuzzy ART properties. *Neural Networks* 8 (2), 203–213.
- Montgomery, D.C., 1997. *Design and Analysis of Experiments*, fifth ed. Wiley, New York.
- Montgomery, D.C., 2000. *Introduction to Statistical Quality Control*, fourth ed. Wiley, New York.
- NeuralWare Inc., 1997. *NeuralWorks Professional II Plus Reference Guide*. Pittsburgh, PA, USA.
- Pao, Y.H., 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, Reading, MA.
- Pugh, G.A., 1991. A comparison of neural networks to SPC charts. *Computers and Industrial Engineering* 21, 253–255.

- Smith, A.E., 1994. X-bar and R control chart interpretation using neural computing. *International Journal of Production Research* 32 (2), 309–320.
- Vattulainen, I., Kankaala, K., Saarinen, J., Ala-Nissila, T., 1995. A comparative study of some pseudorandom number generators. *Computer Physics Communications* 86, 209–226.
- Wehrens, R., Putter, H., Buydens, L.M.C., 2000. The bootstrap: a tutorial. *Chemometrics and intelligent laboratory systems* 54, 35–42.
- Zorriassantine, F., Tannock, J.D.T., 1998. A review of neural networks for statistical process control. *Journal of Intelligent Manufacturing* 9, 209–224.