Available online at www.sciencedirect.com
SCIENCE DIRECT®
Engineering Applications of Artificial Intelligence 18 (2005) 645–662
Engineering Applications of
ARTIFICIAL
INTELLIGENCE
www.elsevier.com/locate/engappai

Understanding ART-based neural algorithms as statistical tools for manufacturing process quality control

Massimo Pacella[a,*], Quirico Semeraro[b]

[a]Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, Via per Monteroni, Lecce 73100, Italy
[b]Dipartimento di Meccanica, Politecnico di Milano, Via Bonardi, Milano 20133, Italy

Received 16 September 2004; accepted 4 February 2005
Available online 1 April 2005

Abstract

Neural networks have recently received a great deal of attention in the field of manufacturing process quality control, where statistical techniques have traditionally been used. In this paper, a neural-based procedure for quality monitoring is discussed from a statistical perspective. The neural network is based on Fuzzy ART, which is exploited for recognising any unnatural change in the state of a manufacturing process. Initially, the neural algorithm is analysed by means of geometrical arguments. Then, in order to evaluate control performances in terms of errors of Types I and II, the effects of three tuneable parameters are examined through a statistical model. Upper bound limits for the error rates are analytically computed, and then numerically illustrated for different combinations of the tuneable parameters. Finally, a criterion for the neural network designing is proposed and validated in a specific test case through simulation. The results demonstrate the effectiveness of the proposed neural-based procedure for manufacturing quality monitoring.

Keywords: Statistical process control; Artificial intelligence; Adaptive resonance theory; Cluster analysis; Neural network design

1. Introduction

Neural networks have recently received a great deal of attention in a wide variety of applications where statistical methods have usually been employed. As an example, neural networks are used for classification and regression problems because of their ability to elaborate large amounts of data in real-time, and their capacity for handling noisy and uncertain data. Ripley (1994) provided a comparison study between statistical methods and neural networks for classification problems. Stern (1996) introduced the use of neural network models from the perspective of an applied statistician using a regression problem as an example. There is also an increasing emphasis on reviewing neural networks theory from a statistical perspective. Cheng and Titterington (1994) have shown that some statistical procedures can be given a neural network expression. Hwang and Ding (1997) and De Veaux et al. (1998) considered the problem of constructing confidence intervals for neural networks in nonlinear regression applications. A comprehensive reference on neural network theory is provided by Haykin (1999), while Bishop (1995) provided a general introduction of neural networks for statisticians.

Since neural networks are able to recall learned patterns from noisy or incomplete representations, they were also extensively exploited in statistical process control (SPC) applications where quality characteristics of a process are monitored in order to detect any unusual event that may occur (Zorriassantine and Tannock, 1998). The application of neural networks to SPC can be commonly classified into two categories, i.e. pattern recognition and unnatural behaviour detection.

*Corresponding author. Tel.: +39 083 229 7253; fax: +39 0832 297 279.
E-mail address: massimo.pacella@unile.it (M. Pacella).

0952-1976/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2005.02.001

The pattern recognition provides a mechanism for identifying different types of unnatural patterns in real time on the series of process quality measurements. The patterns identified then serve as the primary information for identifying the causes of unnatural process behaviour. Reports of using neural networks for pattern recognition can be found in Hwarng and Hubele (1993a,b), Hwarng and Chong (1995), Cheng (1995, 1997), Chang and Aw (1996), Cook and Chiu (1998), Guh and Hsieh (1999), Guh and Tannock (1999), Chang and Ho (1999), Perry et al. (2001), Cook et al. (2001).

In the other category, detecting unnatural process behaviours, one of the earliest applications can be found in Al-Ghanim (1997) which proposed a system that is capable of signalling any change in the structure of a manufacturing process. In particular, the binary implementation of adaptive resonance theory (ART) was trained on a set of natural data in order to cluster them into groups with similar features. After training, the neural network can provide an indication that a change in process outputs has occurred when the series of process data does not fit to any of the learned categories. Although the work of Al-Ghanim represented a remarkable new use of neural networks for quality control, the author found that his pioneering methodology did not have the same degree of sensitivity as is possible using other neural network approaches. This drawback can be mainly ascribed to the binary coding of the ART algorithm as it is a less flexible way of using process data than a method based on graded continuous number encoding.

Our recent researches in manufacturing process quality control extend Al-Ghanim's methodology and present outperforming ART-based approaches for unnatural behaviour detection (Pacella et al., 2004a,b). In particular, simplified ART algorithms (based on the Fuzzy ART), which do not require binary coding of input data, have been investigated. In Pacella et al. (2004a) the neural network was trained using a series of process natural output data in a similar manner to that of Al-Ghanim. In Pacella et al. (2004b) it was demonstrated that the training set can even be limited to a single vector whose components are equal to the process nominal value. In the post-training phase, Fuzzy ART compares input vectors to learned categories and produces a signal if the current input does not fit to any of the natural templates.

These approaches can achieve similar performances in signalling a sustained change of process mean with those of the cumulative sum (CUSUM) control chart (Montgomery, 2000), but at the same time, they are also capable to detect a wide set of potential unnatural changes that cannot be addressed by a sole CUSUM chart. Indeed, for transient or dynamic changes of process mean, Fuzzy ART can outperform traditional charting techniques, which are designed to detect these particular changes, as a Shewhart control chart with a set of run rules and sensitizing rules (Montgomery, 2000). Since it can model different control strategies simultaneously, the proposed approach can be exploited as the sole tool for signalling a generic modification in the state of the process, so it provides a powerful diagnostic tool for detecting assignable causes in actual industrial processes.

Fuzzy ART was mainly chosen because its responses to input stimulus can be easily explained, in contrast to other neural networks, where typically it is more difficult to realise why an input produces a specific output. Indeed, one of the major features of using the aforementioned neural networks is that they are "black box" models as the effects of their parameters are generally not interpretable. On one hand, this is not a problem for many applications in which the emphasis is on prediction rather than on model building or model understanding. On the other hand, the method of choosing the values of neural network parameters is not well implemented as it is based on an experimental process where different values are used and evaluated. The problem with this is that it can be very time consuming, especially because neural networks typically have slow convergence rates. In the SPC field, this leaves the user with empirically developing, for the process control case at hand, the relationship between neural network performances and its parameters.

The aim of this paper is to provide a detailed description of the proposed Fuzzy ART approach, as a tool for detecting unnatural process behaviours, to quality practitioners with a statistical background. We achieve this by deriving a statistical model of Fuzzy ART algorithm in a very specific case in order to understand the capabilities and potentials of neural networks for manufacturing quality control. Fuzzy ART is firstly described by means of geometrical concepts, and then a probabilistic model is applied to it in order to estimate the effect of three tuneable parameters on the performance of the control procedure. Statistical methods are then used in order to derive analytically bound limits for monitoring performances. A practical result, which is obtained from such a statistical model, is a criterion for deciding on the values of network parameters which should be used in order to obtain a predefined monitoring performance for the process control case at hand.

An overview of this paper is as follows. In this section, a brief outline of the paper, have been provided. In Section 2, the characteristics of Fuzzy ART neural network are summarised through geometrical arguments, while the reference manufacturing process model is illustrated in Section 3. The proposed neural system and the training procedure are both presented in Section 4. In Section 5, the neural network is analysed by means of statistical arguments. In Section 6, upper bound limits

for the errors of Types I and II are obtained analytically as a function of the configuration parameters. An analytical criterion for neural network designing is proposed in Section 7, and validated for a specific test case through simulation. The paper concludes with a summary and two appendixes which illustrate analytical proofs and simulation comparisons.

2. Fuzzy ART

Fuzzy ART is based on the adaptive resonance theory, thus it operates by summarising similar data into *categories*, and is based on fuzzy set theory operations, and hence input values, as well as the weights of network links, can range only between zero and one.

This network is composed of two major subsystems, the attentional and orienting subsystem. Three fields of nodes, namely $F0$ (the input layer), $F1$ (called the comparison layer) and $F2$ (the competitive layer of the network, also called the recognition layer), compose the attentional subsystem. The bottom-up and top–down weights, which fully connect $F1$ and $F2$ layer nodes, can be updated adaptively in response to input patterns. Properties of learning for Fuzzy ART can be found in Huang et al. (1995); Georgiopoulos et al. (1996, 1999), Anagnostopoulos and Georgiopouolos (2002).

2.1. Training of the neural network for quality control

When applied in quality control applications, Fuzzy ART may be trained under two different scenarios (Pacella et al., 2004a,b). In the first of these scenarios, a collection of input patterns, the so-called *training list*, is available. Such patterns are formed using a series of process output data. The time length of the process output series, which is used to train the network, is referred to as the *learning period* (Al-Ghanim, 1997). The underlying assumption is that in such a period, the process under inspection produces only natural outputs, which are clustered into categories by the ART network. Obviously, patterns that are similar to each other will be clustered in the same category. The clustering process performed in the training phase depends on the *vigilance parameter*. Higher vigilance imposes a stricter matching criterion that separates input patterns into finer categories. On the contrary, lower vigilance tolerates greater mismatches, and it produces coarse categories.

In the second scenario of training, a list of natural patterns is not available (e.g. in the case of new installed process). In this case, only the process *target* is known in advance, and the process is monitored to verify a constant *nominal mean* equal to the target. In such a scenario one vector, whose components are equal to the nominal value, forms the training list and since no clustering is performed by the network, the vigilance parameter has no influence during training. Consequently, learning produces only a category that is identical to the single training vector.

2.2. The Fuzzy ART algorithm

Before discussing the Fuzzy ART algorithm in more detail, some notation must be introduced. Let $U = [0, 1]$ be the interval of real numbers included between zero and one, and let U^M be the unit M-dimensional hyper-cube, this comprises M-dimensional vectors whose components fall in U. The unit hyper-cube U^M serves as an input space for a Fuzzy ART that comprises M nodes in the input layer $F0$. In the sequel of paper the operators of Table 1 will be employed.

Merging the expressions of min and max fuzzy operators allows for deriving an alternative definition of the *city-block distance* (also known as *Manhattan distance*) as follows:

$$\mathrm{dis}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \sum_{i=1}^{M} |x_i - y_i|.$$

Let t denote a discrete time index at which the state of the process is evaluated by means of the neural-based

Table 1
Some nomenclature of Fuzzy ART algorithm

Norm operator	$\lvert\cdot\rvert : U^M \to [0, M]$	$\underline{\boldsymbol{x}} \in U^M$ $\lvert \underline{\boldsymbol{x}} \rvert = \sum_{i=1}^{M} \lvert x_i \rvert$
Fuzzy min operator	$\cdot \wedge \cdot : U^M \times U^M \to U^M$	$\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \in U^M$ $\underline{\boldsymbol{x}} \wedge \underline{\boldsymbol{y}} = [\min(x_1, y_1), \ldots \min(x_i, y_i), \ldots]$
Fuzzy max operator	$\cdot \vee \cdot : U^M \times U^M \to U^M$	$\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \in U^M$ $\underline{\boldsymbol{x}} \vee \underline{\boldsymbol{y}} = [\max(x_1, y_1), \ldots \max(x_i, y_i), \ldots]$
City-block distance	$\mathrm{dis}(\cdot, \cdot) : U^M \times U^M \to [0, M]$	$\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \in U^M$ $\mathrm{dis}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = \lvert \underline{\boldsymbol{x}} \vee \underline{\boldsymbol{y}} \rvert - \lvert \underline{\boldsymbol{x}} \wedge \underline{\boldsymbol{y}} \rvert$

control system. From now on, a M-dimensional vector serving at time of index t as input to the neural network will be denoted as $\underline{\boldsymbol{I}}_t \in U^M$, while the notation $\underline{\boldsymbol{I}}_t^c = (\underline{\boldsymbol{I}}_t, \underline{\mathbf{1}} - \underline{\boldsymbol{I}}_t) \in U^{2M}$ (where $\underline{\mathbf{1}}$ denotes the M-dimensional all-ones vector) will be used to designate its *complement-coded form*. The integer M is called the *window size*: it results that $|\underline{\boldsymbol{I}}_t^c| = M$. Complement coding occurs in the field $F0$, and the resulting $2M$-dimensional vector $\underline{\boldsymbol{I}}_t^c$ is the input pattern to the layer $F1$.

The layer $F2$ consists of two kinds of nodes: *committed* and *uncommitted*. Each committed node is associated to a specific category. The information describing each Fuzzy ART category j is stored in a *template*. The template of category j, which is indicated by the notation $\underline{\boldsymbol{w}}_j^c \in U^{2M}$, is the $2M$-dimensional top–down vector of weights related to the connections from the jth node in $F2$, to all nodes in $F1$ layer. On the other hand, uncommitted nodes, which feature a template of all-ones components, do not correspond to real categories and they represent the "blank" memory of the system.

An important quantity related to the Fuzzy ART algorithm is the *category match function*. With respect to a specific input pattern $\underline{\boldsymbol{I}}_t^c$ at the layer $F1$, the category match function-value of the jth committed node in the $F2$ layer is given by

$$\rho(\underline{\boldsymbol{I}}_t^c, \underline{\boldsymbol{w}}_j^c) = \frac{|\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_j^c|}{M}. \tag{1}$$

The category match function-value of the jth node, with respect to an input pattern $\underline{\boldsymbol{I}}_t$ is the quantity used in comparison to the *vigilance parameter* $\rho \in [0, 1]$. The comparison of the category match function to the vigilance parameter ρ constitutes the *vigilance test*. The vigilance test is performed by the orienting subsystem, and it can be considered as a tuneable, novelty detection method that points out a typical pattern with respect to existing categories. More specifically, a committed node of index j failing the vigilance test with respect to a pattern $\underline{\boldsymbol{I}}_t$ can be interpreted as follows: $\underline{\boldsymbol{I}}_t$ does not fit the characteristics of category j and therefore, the node is being reset via the reset node. The vigilance test can be expressed as follows:

$$\rho(\underline{\boldsymbol{I}}_t^c, \underline{\boldsymbol{w}}_j^c) \geqslant \rho. \tag{2}$$

2.3. Training phase

In this work, the focus is on a Fuzzy ART trained by means of the process nominal value μ. Let $\underline{\boldsymbol{I}}_\mu$ be the single training M-dimensional vector, whose components code the nominal value, and let $\underline{\boldsymbol{I}}_\mu^c$ be the $2M$-dimensional training vector that results from complement coding in the $F0$ layer. Firstly, all the top–down weights of the neural network are initialised to one, and all nodes of the $F2$ layer are uncommitted. One of the uncommitted nodes of the $F2$ layer is then selected to represent the training pattern $\underline{\boldsymbol{I}}_\mu^c$, and the corresponding $2M$-dimensional top–down weight vector $\underline{\boldsymbol{w}}_\mu^c$ is set equal to $\underline{\boldsymbol{w}}_\mu^c = \underline{\boldsymbol{I}}_\mu^c$. The result of this training procedure is that a single committed node is produced in the $F2$ layer; therefore, the template vector $\underline{\boldsymbol{w}}_\mu^c$ can be rewritten as $\underline{\boldsymbol{w}}_\mu^c = (\underline{\boldsymbol{w}}_\mu, \underline{\mathbf{1}} - \underline{\boldsymbol{w}}_\mu)$ where $\underline{\boldsymbol{w}}_\mu = \underline{\boldsymbol{I}}_\mu$.

2.4. Post-training phase

Let us assume that at time of index t an M-dimensional input pattern $\underline{\boldsymbol{I}}_t$ is presented at the $F0$ field. The appearance of the $2M$-dimensional pattern $\underline{\boldsymbol{I}}_t^c$ across the $F1$ field produces the activation of the single node that has been committed in the $F2$ layer of the network (whose top–down weight vector is indicated by $\underline{\boldsymbol{w}}_\mu^c$). The appropriateness of the natural category to represent the input pattern is checked in the Fuzzy ART algorithm by the vigilance test of Eq. (2). In this case

$$\begin{aligned} |\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_\mu^c| &= |(\underline{\boldsymbol{I}}_t, \underline{\mathbf{1}} - \underline{\boldsymbol{I}}_t) \wedge (\underline{\boldsymbol{w}}_\mu, \underline{\mathbf{1}} - \underline{\boldsymbol{w}}_\mu)| \\ &= |\underline{\boldsymbol{I}}_t \wedge \underline{\boldsymbol{w}}_\mu| + M - |\underline{\boldsymbol{I}}_t \vee \underline{\boldsymbol{w}}_\mu| \\ &= M - \mathrm{dis}(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu). \end{aligned}$$

Therefore, the category match function can be expressed via a geometry-based quantity as follows:

$$\rho(\underline{\boldsymbol{I}}_t^c, \underline{\boldsymbol{w}}_\mu^c) = \frac{M - \mathrm{dis}(\underline{\boldsymbol{I}}_t \wedge \underline{\boldsymbol{w}}_\mu)}{M}, \tag{3}$$

where $\mathrm{dis}(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu)$ is the aforementioned *city-block* distance between the M-dimensional input vector $\underline{\boldsymbol{I}}_t$ and the nominal value template $\underline{\boldsymbol{w}}_\mu$ of the hyper-cube U^M. The *match region* $V_\mu(\rho)$, with respect to the template $\underline{\boldsymbol{w}}_\mu$, and for a particular value of ρ, is defined as the following subset of U^M:

$$\begin{aligned} V_\mu(\rho) &= \{\underline{\boldsymbol{x}} \in U^M : \rho(\underline{\boldsymbol{x}}^c, \underline{\boldsymbol{w}}_\mu^c) \geqslant \rho\} \\ &= \{\underline{\boldsymbol{x}} \in U^M : \mathrm{dis}(\underline{\boldsymbol{x}}, \underline{\boldsymbol{w}}_\mu) \leqslant M(1-\rho)\}. \end{aligned}$$

The quantity $M(1-\rho)$ can be considered as the size of the match region $V_\mu(\rho)$. It stands for the maximum distance that a pattern $\underline{\boldsymbol{x}}$ can have from the nominal value template $\underline{\boldsymbol{w}}_\mu$, so that it can be considered as a natural one. A geometrical interpretation of the Fuzzy ART algorithm can be formulated as follows: the input pattern $\underline{\boldsymbol{I}}_t \in U^M$ is considered as a natural one if and only if it belongs to the match region $V_\mu(\rho)$

$$\underline{\boldsymbol{I}}_t \in V_\mu(\rho). \tag{4}$$

As it can be noticed, the match region size is a monotonically decreasing function of ρ: the larger the vigilance parameter is, the smaller the size of the match region. The nominal template can be geometrically represented as a point of the Fuzzy ART input space U^M.

3. The manufacturing process model

The goal of SPC is to monitor the quality characteristic of a process in order to detect any unusual event that may occur. Two sources of variability are considered by SPC, unassignable cause variability, variability that is *natural* to the process and that can only be modified by changing the existing production system, and assignable cause variability, observed *unnatural* variability that can be traced to a particular problem. SPC is based on the use of statistical techniques, and graphical displays such as control charts, in order to detect assignable cause of variability (Montgomery, 2000).

Process monitoring denotes the use of a control system that can cyclically check the desired stable state of the process. The use of a control system can lead to the elimination of assignable causes pointed to by unnatural behaviour. Its properties can be described in terms of probabilities; in fact, process monitoring parallels statistical hypothesis testing (H_0: the process is in a natural state; H_1: the process is in an unnatural state). As with every statistical test, errors of Types I and II can occur with probabilities $\alpha = P\{H_1|H_0\}$ and $\beta = P\{H_0|H_1\}$, respectively.

Assume to sample the product from a process and to measure a single quality characteristic of interest. Denote by $\{Y_n\}$ the time series of the quality characteristic measurements obeying the following model:

$$Z_n \sim \mathrm{NID}(0,1),$$
$$H_0 : Y_n = \mu + Z_n,$$
$$H_1 : Y_n = \mu + Z_n + S_n,$$

where

- $n = 1, 2, \ldots$ denotes a discrete time index, or part number, in which the process quality characteristic is measured;
- $\{Z_n\}$ is a time series of random deviations with zero mean, standard deviation equal to 1, normally and identically distributed: $Z_n \sim \mathrm{NID}(0,1)$. They model the natural variation of process output data due to unassignable causes and the variance is assumed fixed to one as it does not cause loss of generality.
- $\{S_n\}$ is an arbitrary disturbance time series that models the unnatural variation due to assignable causes.

In this work, the special signal is formulated as a random *mixture* of positive and negative deviations from the process nominal value. Essentially, $S_n = \pm\varphi$ where $\varphi > 0$ is the magnitude of the mixture measured in terms of unit of standard deviation. The model of Eq. (5) has been exploited, where m_n denotes the status of a binary $\{-1, 1\}$ Markov's chain at time of index n.

$$S_n = \varphi \cdot m_n; \quad m_n \in \{-1, 1\}. \tag{5}$$

In this model, the status of the binary Markov's chain only changes with probability $p_c = P\{m_{n+1} \neq m_n | m_n\}$; thus $1 - p_c = P\{m_{n+1} = m_n | m_n\}$. Special cases of this model are: (1) the constant shift of the mean (when $p_c = 0$) and (2) the systematic variation of the mean (when $p_c = 1$).

4. The proposed control system

Let n be the aforementioned time index in which the process quality characteristic is measured, and denote by t the time index in which a signal (either the process is in a natural state or in an unnatural state) is emitted by the neural network. t is equal to $t = [M + (n-1)P]$ where $M \in \{1, 2, \ldots\}$ is the dimension of the neural network input vector (i.e. the aforementioned *window size*) and the parameter $P \in \{1, 2, \ldots, M\}$ is the number of non-overlapping measures between two consecutive neural network input vectors. On one hand, $P = 1$ implies that the neural network emits a signal whenever a new single measure for the process quality characteristic is collected. On the other hand, $P = M$ implies that M new measures must be collected before the neural network can emit next signal. In any case, the need to arrange the series of quality measurements as M-dimensional vectors implies that the system provides no indication on the process state during the first $M - 1$ time intervals. Fig. 1 shows the proposed neural system for quality control and the manufacturing process model.

As depicted in Fig. 1, some pre-processing stages of input data take place before they are presented to the

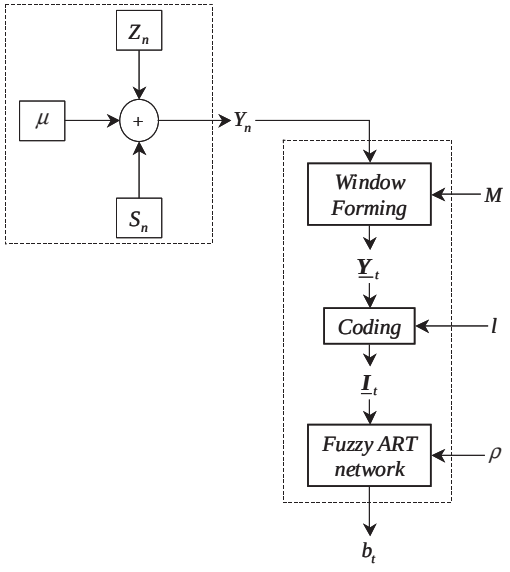

Fig. 1. The proposed neural system for quality control.

Fuzzy ART neural network. The first stage is called the *Window Forming* and it depends on the window size. Through this stage, the most recent M observations are collected to form the network M-dimensional input vector. Denoting as $\underline{\boldsymbol{Y}}_t$ the output of this stage, at each time of index t, we get

$$\underline{\boldsymbol{Y}}_t = [Y_{t-M+1}, Y_{t-M+2}, \ldots, Y_{t-1}, Y_t]. \tag{6}$$

The second pre-processing stage takes as input an M-dimensional input pattern $\underline{\boldsymbol{Y}}_t$ and transforms it into the output vector $\underline{\boldsymbol{I}}_t \in U^M$. This pre-processing stage is called *Coding* and it consists of a re-scaling of the input variable into the range $[0, 1]$. The Coding schema depends on the tuneable parameter $l > 0$, which is an appropriate saturation level for the absolute deviations of process output values $\{Y_n\}$ from the process nominal value. Assuming a symmetric distribution of deviation, the procedure of Eq. (7) is adopted for *Coding*, at each time of index t.

$$\underline{\boldsymbol{I}}_t = [I_{t-M+1}, I_{t-M+2}, \ldots, I_{t-1}, I_t]$$
$$\begin{cases} I_\tau = 0, & Y_\tau \leqslant \mu - l, \\ I_\tau = \frac{1}{2}\left(1 + \frac{Y_\tau - \mu}{l}\right) & \mu - l < Y_\tau < \mu + l, \\ & t - M + 1 \leqslant \tau \leqslant t, \\ I_\tau = 1, & \mu + l \leqslant Y_\tau. \end{cases} \tag{7}$$

At each time of index t, the binary output b_t is produced by the neural network. Consider that such signal is set to $b_t = 1$ when the process is supposed staying in a natural state, $b_t = 0$ otherwise. Basing on the neural network description provided in Section 2, we can formulate

$$\forall t; b_t = 1 \;\Leftrightarrow\; \underline{\boldsymbol{I}}_t \in V_\mu(\rho) \;\Leftrightarrow\; \mathrm{dis}(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu) \leqslant M(1-\rho). \tag{8}$$

Thus, by using Eq. (7) the criterion of Eq. (8) can be rewritten as follows:

$$\forall t; b_t = 1 \;\Leftrightarrow\; \sum_{r=1}^{M} \min(l, |Y_{t-M+r} - \mu) \leqslant 2lM(1-\rho). \tag{9}$$

Let MAD_t be the *Mean Absolute Deviation* at time of index t, i.e. the mean of absolute deviations from μ of the most recent M consecutive process output values. The absolute deviations are constrained to be not greater than the limit l. In formula

$$\mathrm{MAD}_t = \frac{\sum_{r=1}^{M} \min(l, |Y_{t-M+r} - \mu|)}{M}.$$

Therefore, the criterion expressed by former Eq. (9) can be reformulated as follows:

$$\forall t; b_t = 1 \;\Leftrightarrow\; \mathrm{MAD}_t \leqslant 2l(1-\rho). \tag{10}$$

In other words, at each time of index t the neural Fuzzy ART network emits an alarm ($b_t = 0$) if and only if the random variable MAD_t exceeds the limit $2l(1-\rho)$.

5. A statistical analysis

In this section, a statistical analysis of the random variable MAD_t is provided based on the probabilistic model described in Section 3. Suppose that the mono-dimensional quality characteristic is measured with reference to a nominal mean value μ, which is assumed in the sequel, without loss of generality, equal to $\mu = 0$. In addition, assume $P = M$, i.e. there are non-overlapping measures between two consecutive input vectors (in this case the neural network can emit a signal whenever M new measures are collected and $t = M, 2M, 3M, \ldots$).

The goal of our statistical analysis is to calculate the expected mean and variance values of the random variable MAD_t. To such a goal, seven specific functions of parameters l and S_τ, which in the sequel are referenced to as the *psi-functions*, are introduced. Next, we will proceed with the formulation of the expected mean and variance values of the random variable MAD_t by means of the *psi-functions*. Proofs of the properties presented in the sequel are supplied in Appendix.

The proposed methodology is focused on small changes of process mean and thus we assume $S_\tau < l$. The psi-functions are reported in Table 2, where $\Phi(\cdot)$ is the cumulative standard normal distribution function.

For l large enough ($l \to \infty$) and/or for a in-control process ($S_\tau = 0$), the psi-functions can be rewritten as reported in Table 3.

Property 1. *Assuming that* $Y_n = Z_n + S_n$, *where* $Z_n \sim \mathrm{NID}(0, 1)$, *since* $t = M, 2M, 3M, \ldots$, *it follows that* $\mathrm{MAD}_t(M, l)$ *is a time series of independent and identical distributed* (*i.i.d.*) *random variables with mean and*

Table 2
The psi-functions

$$\Psi_1(l, S_\tau) = \frac{1}{\sqrt{2\pi}}\left[2\exp\left(-\frac{S_\tau^2}{2}\right) - \exp\left(-\frac{(l-S_\tau)^2}{2}\right) - \exp\left(-\frac{(l+S_\tau)^2}{2}\right)\right]$$
$$\Psi_2(l, S_\tau) = S_\tau[\Phi(l - S_\tau) + \Phi(-l - S_\tau) - 2\Phi(-S_\tau)]$$
$$\Psi_3(l, S_\tau) = l[\Phi(-l - S_\tau) + 1 - \Phi(l - S_\tau)]$$
$$\Psi_4(l, S_\tau) = -\frac{1}{\sqrt{2\pi}}\left[(l - S_\tau)\exp\left(-\frac{(l-S_\tau)^2}{2}\right) - (-l - S_\tau)\exp\left(-\frac{(-l-S_\tau)^2}{2}\right)\right]$$
$$\Psi_5(l, S_\tau) = (1 + S_\tau^2)\,[\Phi(l - S_\tau) - \Phi(-l - S_\tau)]$$
$$\Psi_6(l, S_\tau) = S_\tau\sqrt{\frac{2}{\pi}}\left[-\exp\left(-\frac{(l-S_\tau)^2}{2}\right) + \exp\left(-\frac{(-l-S_\tau)^2}{2}\right)\right]$$
$$\Psi_7(l, S_\tau) = l^2[\Phi(-l - S_\tau) + 1 - \Phi(l - S_\tau)]$$

Table 3
Special cases of the psi-functions

	$l \to \infty$	$S_\tau = 0$	$l \to \infty; S_\tau = 0$
$\Psi_1(l, S_\tau) =$	$\sqrt{\frac{2}{\pi}}\exp\left(-\frac{S_\tau^2}{2}\right)$	$\sqrt{\frac{2}{\pi}}\left[1-\exp\left(-\frac{l^2}{2}\right)\right]$	$\sqrt{\frac{2}{\pi}}$
$\Psi_2(l, S_\tau) =$	$S_\tau[1-2\Phi(-S_\tau)]$	0	0
$\Psi_3(l, S_\tau) =$	0	$2l\Phi(-l)$	0
$\Psi_4(l, S_\tau) =$	0	$-\sqrt{\frac{2}{\pi}}l\exp\left(-\frac{l^2}{2}\right)$	0
$\Psi_5(l, S_\tau) =$	$1+S_\tau^2$	$1-2\Phi(-l)$	1
$\Psi_6(l, S_\tau) =$	0	0	0
$\Psi_7(l, S_\tau) =$	0	$2l^2\Phi(-l)$	0

variance as follows:

$$E[\mathrm{MAD}_t] = \frac{1}{M}\sum_{r=1}^{M} E[\min(l, |Y_{t-M+r}|)], \tag{11}$$

$$\mathrm{Var}[\mathrm{MAD}_t] = \frac{1}{M^2}\sum_{r=1}^{M} \mathrm{Var}[\min(l, |Y_{t-M+r}|)], \tag{12}$$

where

$$E[\min(l, |Y_\tau|)] = \Psi_1(l, S_\tau) + \Psi_2(l, S_\tau) + \Psi_3(l, S_\tau), \tag{13}$$

$$\begin{aligned}\mathrm{Var}[\min(l, |Y_\tau|)] &= -E^2[\min(l, |Y_\tau|)] + \Psi_4(l, S_\tau) \\ &\quad + \Psi_5(l, S_\tau) + \Psi_6(l, S_\tau) + \Psi_7(l, S_\tau)\end{aligned} \tag{14}$$

In particular, for l large enough (*e.g.* $l > 5$), *the mean* $E[\min(l, |Y_\tau|)]$ *and the variance* $\mathrm{Var}[\min(l, |Y_\tau|)]$ *are, respectively, equal to*

$$E[|Y_\tau|] = \sqrt{\frac{2}{\pi}}\exp\left(-\frac{S_\tau^2}{2}\right) + S_\tau[1-2\Phi(-S_\tau)],$$

$$\mathrm{Var}[|Y_\tau|] = -\left\{\sqrt{\frac{2}{\pi}}\exp\left(-\frac{S_\tau^2}{2}\right) + S_\tau[1-2\Phi(-S_\tau)]\right\}^2 + 1 + S_\tau^2.$$

Property 2. *If the process is in a natural state*: $Y_n = Z_n$, *where* $Z_n \sim \mathrm{NID}(0,1)$, *then* MAD_t *is a series of i.i.d. random variables with the expected mean value of Eq.* (15), *which is constant over time, and which does not depend on the window size*

$$E[\mathrm{MAD}_t] = \sqrt{\frac{2}{\pi}}\left[1-\exp\left(-\frac{l^2}{2}\right)\right] + 2l\Phi(-l). \tag{15}$$

In addition, the variance value of $\mathrm{MAD}_t(M, l)$ *is constant over time and is equal to*

$$\begin{aligned}\mathrm{Var}[\mathrm{MAD}_t] &= -\frac{1}{M}\left\{\sqrt{\frac{2}{\pi}}\left[1-\exp\left(-\frac{l^2}{2}\right)\right] + 2l\Phi(-l)\right\}^2 - \frac{1}{M} \\ &\quad \times\left[\sqrt{\frac{2}{\pi}}l\exp\left(-\frac{l^2}{2}\right) + 1 - 2\Phi(-l) + 2l^2\Phi(-l)\right].\end{aligned} \tag{16}$$

For parameter l large enough (*practically* $l \geqslant 5$), *the exponential terms of Eqs.* (15) *and* (16) *may be neglected and we get*

$$E[\mathrm{MAD}_t] = \sqrt{\frac{2}{\pi}},$$

$$\mathrm{Var}[\mathrm{MAD}_t] = \frac{\pi-2}{M\pi}.$$

Property 3. *If the process is in an unnatural state, and a mixture disturbance is adopted to model such state* (*i.e.* $Y_n = Z_n \pm \varphi$, *where* $Z_n \sim \mathrm{NID}(0,1)$), *then the i.i.d. random variables of the time series* MAD_t *have a constant mean value that does not depend on the window size*

$$E[\mathrm{MAD}_t] = \Psi_1(l, \varphi) + \Psi_2(l, \varphi) + \Psi_3(l, \varphi). \tag{17}$$

The variance value is constant over time as well, and it results inversely proportional to the window size value M.

$$\begin{aligned}\mathrm{Var}[\mathrm{MAD}_t] = \frac{1}{M}\{&-E^2[\mathrm{MAD}_t] + \Psi_4(l, \varphi) + \Psi_5(l, \varphi) \\ &+ \Psi_6(l, \varphi) + \Psi_7(l, \varphi)\}.\end{aligned} \tag{18}$$

For l large enough, Eq. (17) *can be rewritten as follows*:

$$E[\mathrm{MAD}_t] = \sqrt{\frac{2}{\pi}}\exp\left(-\frac{\varphi^2}{2}\right) + \varphi[1-2\Phi(-\varphi)]$$

and Eq. (18) *can be rewritten as follows*:

$$\mathrm{Var}[\mathrm{MAD}_t] = -\frac{1}{M}\left\{\sqrt{\frac{2}{\pi}}\exp\left(-\frac{\varphi^2}{2}\right) + \varphi[1-2\Phi(-\varphi)]\right\}^2 + \frac{1+\varphi^2}{M}.$$

Fig. 2 presents the expected mean values of random variable MAD_t as a function of the mixture magnitude φ, as well as the standard deviation values for four window sizes $M = 1; 2; 10; 50$, assuming the coding parameter equal to $l = 3$.

Similar graphs are reported by Fig. 3 in the case in which it results $l = 6$.

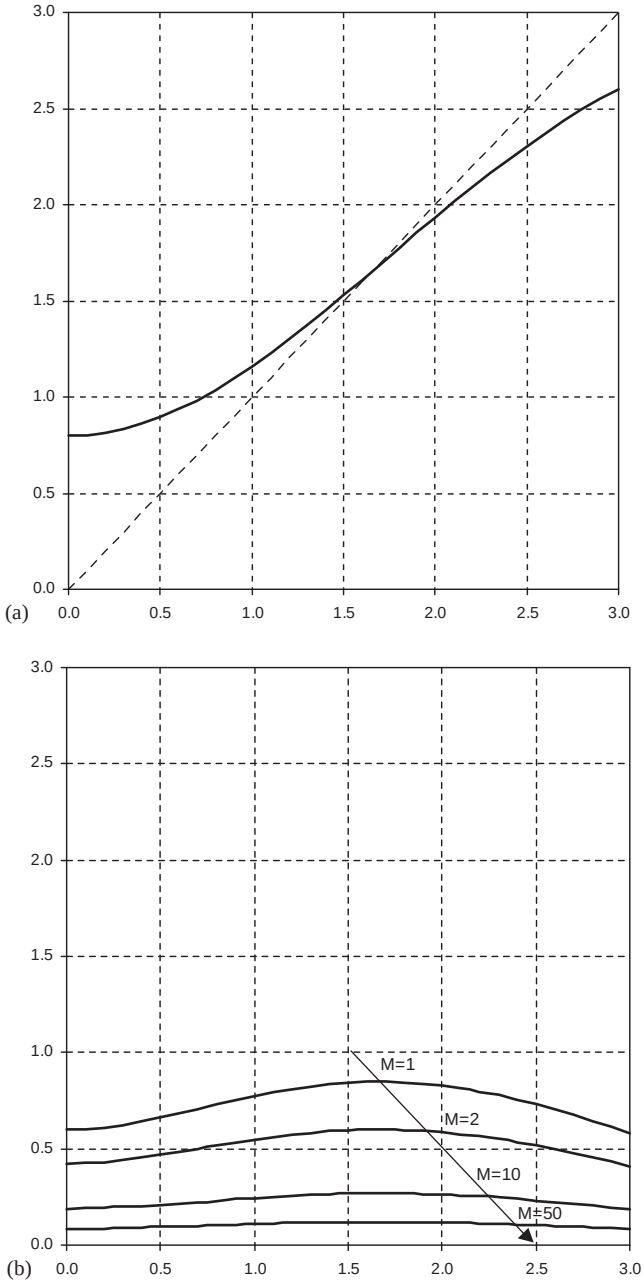

Fig. 2. Expected mean value (a) of the mean absolute deviation (MAD) (ordinate) versus absolute magnitude of a mixture special disturbance φ (abscissa). Standard deviation (b) of MAD (ordinate) versus φ (abscissa) for four value of the window size $M = 1; 2; 10;$ and 50. The parameter l has been fixed equal to $l = 3$.

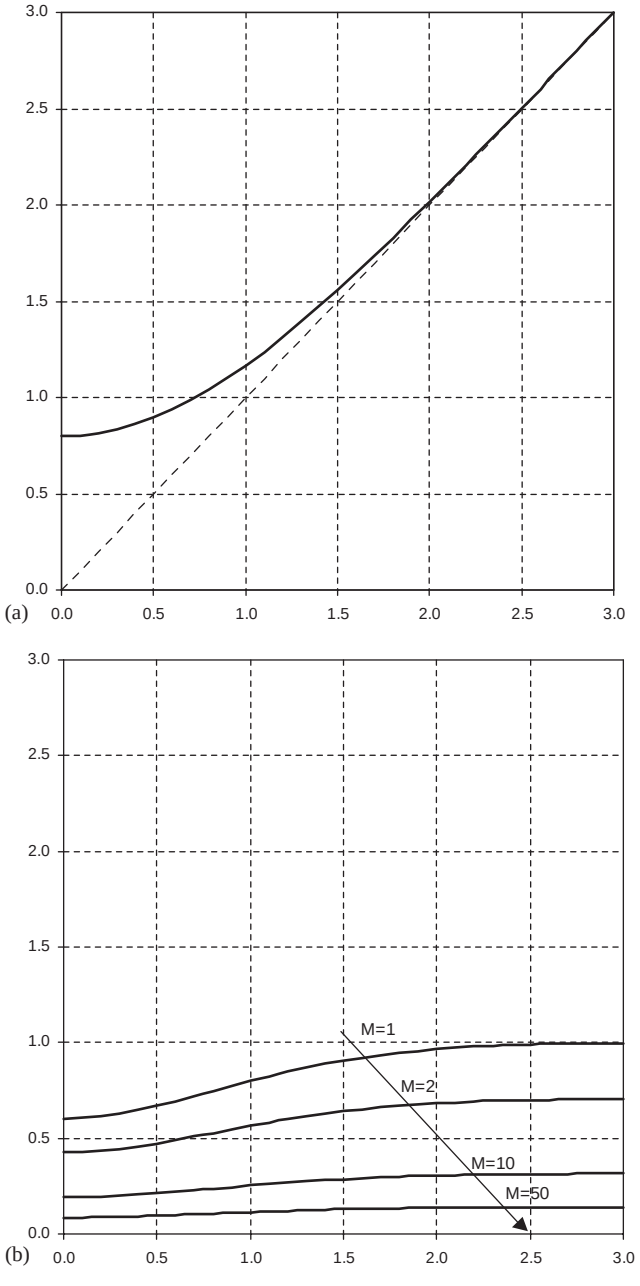

Fig. 3. Expected mean value (a) of the mean absolute deviation (MAD) (ordinate) versus absolute magnitude of a mixture special disturbance φ (abscissa). Standard deviation (b) of MAD (ordinate) versus φ (abscissa) for four value of the window size $M = 1; 2; 10;$ and 50. The parameter l has been fixed equal to $l = 6$.

6. Performance prediction theory for the neural network

The neural network performance for quality monitoring can be fully described in terms of Type I errors (the rate of alarms occurring in process data having only natural sources of variation) and of Type II errors (the rate of non-alarms occurring in process data having unnatural sources of variation). By Eq. (10), the probabilities of Types I and II errors, $\alpha = P\{H_1|H_0\}$ and $\beta = P\{H_0|H_1\}$, can be formulated as $\alpha = P\{\mathrm{MAD}_t > 2l(1-\rho)|H_0\}$ and $\beta = P\{\mathrm{MAD}_t \leqslant 2l(1-\rho)|H_1\}$, respectively.

Computer simulation shows that the probability density distribution of the random variable MAD_t is unimodal. Thus, to find out how the parameters M, l and ρ influence the neural network performance, upper bound limits for the errors of Type I α, and of Type II β

can be obtained through the Vysochanskiĭ–Petunin inequality. This is based on the Gauss inequality and a proof of it is provided by Pukelsheim (1994).

Denoting by $\mu_0 = E[\mathrm{MAD}_t|H_0, M, l]$ and $\sigma_0^2 = \mathrm{Var}[\mathrm{MAD}_t|H_0, M, l]$ the mean and variance of MAD_t, respectively, under the hypothesis H_0 and given the values M and l, consider the Vysochanskiĭ–Petunin inequality, $\forall \xi > \sqrt{8/3}\sigma_0$ we get

$$P\{\mathrm{MAD}_t < \mu_0 + \xi|H_0\} \geqslant P\{|\mathrm{MAD}_t - \mu_0| < \xi|H_0\} \geqslant 1 - \frac{4}{9}\frac{\sigma_0^2}{\xi^2}$$

assuming $\xi = 2l(1-\rho) - \mu_0$ and $\rho < 1 - (\mu_0 + \sqrt{8/3}\sigma_0/2l)$, we have

$$1 - \alpha = P\{\mathrm{MAD}_t \leqslant 2l(1-\rho)|H_0\} \geqslant 1 - \frac{4\sigma_0^2}{9[2l(1-\rho) - \mu_0]^2}$$

hence, an upper bound limit for the Type I error is given by the following inequality:

$$\alpha \leqslant \frac{4\sigma_0^2}{9[2l(1-\rho) - \mu_0]^2}. \quad (19)$$

In a similar way, assuming $\mu_1 = E[\mathrm{MAD}_t|H_1, M, l]$ and $\sigma_1^2 = \mathrm{Var}[\mathrm{MAD}_t|H_1, M, l]$ the mean and variance of MAD_t, respectively, under the hypothesis H_1 and given the values M and l, consider the Vysochanskiĭ–Petunin inequality, $\forall \xi > \sqrt{8/3}\sigma_1$ we have

$$P\{\mathrm{MAD}_t > \mu_1 - \xi|H_1\} \geqslant P\{|\mathrm{MAD}_t - \mu_1| < \xi|H_1\} \geqslant 1 - \frac{4}{9}\frac{\sigma_1^2}{\xi^2}$$

assuming $\xi = \mu_1 - 2l(1-\rho)$ and $\rho > 1 - (\mu_1 - \sqrt{8/3}\sigma_1/2l)$, we get

$$1 - \beta = P\{\mathrm{MAD}_t > 2l(1-\rho)|H_1\} \geqslant 1 - \frac{4\sigma_1^2}{9[\mu_1 - 2l(1-\rho)]^2}$$

hence, an upper bound limit for the Type II error is given by the following inequality:

$$\beta \leqslant \frac{4\sigma_1^2}{9[\mu_1 - 2l(1-\rho)]^2}. \quad (20)$$

The effects of the parameters ρ, M and l on the monitoring system performance, are graphically reported in Figs. 4 and 5. In particular, the upper limits of Eqs. (19) and (20) are depicted as a function of the vigilance parameter ρ (which ranges between 0.5 and 1), at four values of the window size ($M = 1; 2; 10; 50$), and two levels of the limit l ($l = 3; 6$).

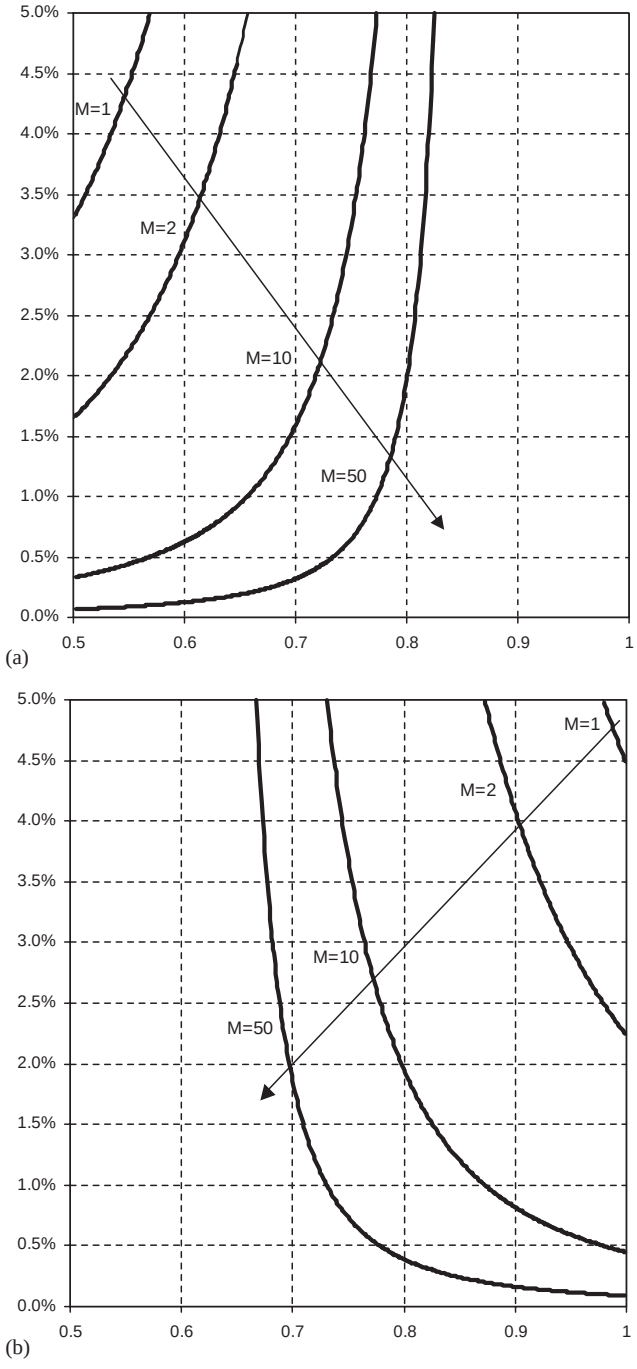

Fig. 4. Upper bound limits for errors of Types I and II given by the Vysochanskiĭ–Petunin inequality. Limit $l = 3$; Window Size $M = 1, 2, 10, 50$. (a) Upper bound limits for error of Type I (ordinate) vs. vigilance parameter (abscissa). (b) Upper bound limits for error of Type II (ordinate) vs. vigilance parameter (abscissa). Disturbance magnitude $\varphi = 2.5$.

6.1. Discussion

As it can be observed from Figs. 4 and 5, considering fixed the values of the window size M and coding limit l, when the vigilance ρ decreases, then the upper bound

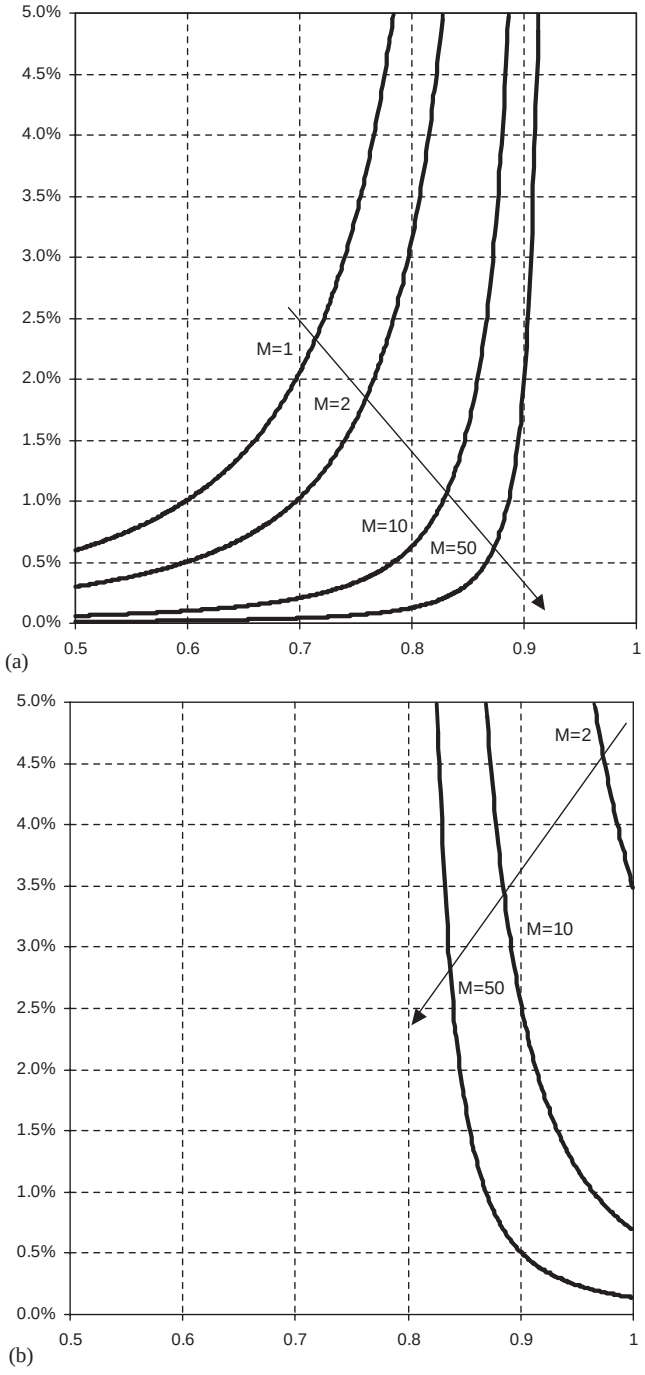

Fig. 5. Upper bound limits for errors of Types I and II given by the Vysochanskiĭ–Petunin inequality. Limit $l = 6$; Window Size $M = 1, 2, 10, 50$. (a) Upper bound limits for error of Type I (ordinate) vs. vigilance parameter (abscissa). (b) Upper bound limits for error of Type II (ordinate) vs. vigilance parameter (abscissa). Disturbance magnitude $\varphi = 2.5$.

limit of the Type I error decrease too. On the other hand, the upper bound limit for errors of Type II is a monotonically decreasing function of the vigilance parameter: the larger the value of ρ, the smaller the limit for errors of Type II. Besides, it results that as the vigilance parameter approaches the upper limit $\rho = 1$, then the upper bound limit for the errors of Type II approaches zero. A reasonable strategy is to adjust the value of ρ based on the Type I error rate that is considered acceptable in the actual quality monitoring application (higher vigilance imposes higher errors of Type I, lower vigilance impose lower errors of Type I).

As it can be also noticed from Figs. 4 and 5, the increase (decrease) rate presented by the upper bound limit of Type I (Type II) errors as function of the vigilance parameter, grows as the window size M increases. That is why a greater window size implies a smaller variance of the monitored variable MAD_t as it results also from Eq. (18). With a fixed value of the vigilance parameter and of the coding limit l it can be noticed that as the window size M increases then the errors of both Types I and II decrease. Thus, the use of a larger window size can improve the performance of neural system. In particular, larger window sizes are recommended in order to reduce errors of Type II and 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.

Finally, it should be pointed out that previous analytical results were obtained under the strict hypothesis that there are non-overlapping measures between two consecutive input vectors ($P = M \Rightarrow t = M, 2M, 3M, \ldots$). In this case, $\mathrm{MAD}_t(M, l)$ is a time series of i.i.d. random variables. In the reverse case, when a single-step moving window is used ($P = 1 \Rightarrow t = M, M + 1, M + 2, \ldots$), once the first M data are collected a new M-dimensional input vector for the neural network can be implemented whenever a new quality measurement becomes available. As a consequence, $\mathrm{MAD}_t(M, l)$ is not more i.i.d., instead it is positively autocorrelated over time. Nevertheless, the upper bound limits of Eqs. (19) and (20) are still appropriate as the actual variance of $\mathrm{MAD}_t(M, l)$ is always less than that analytically estimated.

7. Operating phases of the neural network

Basing on the previous results, a strategy can be derived in order to apply the proposed neural-based approach in quality monitoring applications. It is summarised in three main steps:

- *Configuration phase* (*choice of the parameters M and l*): In this phase, the parameters of the *Window Forming* (M) and *Coding* (l) pre-processing stages are both chosen. As previously observed, while the coding limit l has no relevant effects, the window size M can greatly influence the neural network performance. According to Eqs. (19) and (20), the larger window

size is, the lower the Types I and II error rates are. A strategy is to select the window size M according to the minimum magnitude that is important to recognise (lower magnitude requires higher window sizes).

- *Training phase*: The ART neural network is trained on the process nominal value μ, i.e. the training list is composed by the single vector $\underline{\boldsymbol{Y}}_{\mu}$ with M components equal to zero. Such training method represents the least supervised approach practically possible by using a single training vector.
- *Tuning phase* (*choice of the parameter* ρ): In the *tuning phase*, no more weight adaptations or cluster creations are allowed, and vectors from a *tuning list* are presented to the neural network in order to check the performance of different settings of the vigilance parameter ρ. The tuning vectors are examples of *natural data patterns*, which are obtained using either a series of real process data (measurements of the quality parameter of interest when only unassignable causes of variation are in effect), or a series of simulated data. A strategy is to choice ρ in order to maintain the false alarm rate (Type I error) about equal to a predefined value. By Eqs. (8) and (19) it results that higher vigilance imposes a stricter matching criterion to the natural template learned in training phase (this results in higher false alarm rates); on the contrary, lower vigilance tolerates greater mismatches (this results in lower false alarm rates).

7.1. An analytical criterion for the tuning phase

As it can be noticed, the process of determining the appropriate value of the vigilance parameter is an experimental process where trial values are exploited. The problem with this is that it can be time consuming, especially when considering several neural network configurations, i.e. for different combinations of the parameters M and l.

This section provides an analytical method for deciding on the vigilance parameter value that should be used in order to obtain a predefined Type I error rate (tuning phase). Let $\tilde{\alpha}$ be the reference Type I error rate. The rate of Type I errors is equal to $\alpha = P\{\mathrm{MAD}_t > 2l(1-\rho)|H_0\}$, and it can be considered as a function of the three configuration parameters: $\alpha(M,l,\rho)$. From Eq. (19) it results that, for each combination of the parameters M, l and ρ, the neural network rate of Type I errors is $\alpha(M,l,\rho) \leqslant \sup_{\alpha}$ where $\sup_{\alpha} = 4\sigma_0^2/9[2l(1-\rho)-\mu_0]^2$ being $\mu_0 = E[\mathrm{MAD}_t|H_0,M,l]$ and $\sigma_0^2 = \mathrm{Var}[\mathrm{MAD}_t|H_0,M,l]$ the mean and variance of MAD_t, respectively, under the hypothesis H_0 and given the values M and l.

Let (M,l,ρ) be an initial setting combination of the system parameters for which the neural network provides the expected Type I error rate: $\tilde{\alpha} = \alpha(M,l,\rho)$. By Eqs. (15) and (16), assuming M and l large enough (practically $M \geqslant 50$ and $l \geqslant 6$), it is straightforward to realise that when a small change (a perturbation) of the initial configuration (M,l) into a new setting configuration (M',l') is implemented (e.g. where $M' = M + \Delta M$ and $l' = l + \Delta l$ with $\Delta M \in \{-15,\ldots,15\}$ and $\Delta l \in [-2,2]$), the mean and the variance of MAD_t experience only irrelevant changes. In such cases, denoting with $\mu_0' = E[\mathrm{MAD}_t|H_0,M',l']$ and $\sigma_0'^2 = \mathrm{Var}[\mathrm{MAD}_t|H_0,M',l']$, respectively, the mean and variance of MAD_t for the new setting combination, we can obtain the same upper bound limit for the Type I error rate using the following vigilance parameter value:

$$\rho' = 1 - \frac{1}{2l'}\left(\mu_0' + \frac{2\sigma_0'}{3\sqrt{\sup_{\alpha}}}\right). \tag{21}$$

Since the perturbation of the initial configuration does not cause relevant changes in the probability distribution functions of MAD_t, we can expect that the Type I error rates in the two different configuration result similar

$$\alpha(M',l',\rho') \cong \alpha(M,l,\rho) \;\Rightarrow\; \alpha(M',l',\rho') \cong \tilde{\alpha}. \tag{22}$$

Therefore, given an initial solution, Eq. (21) can be used for identifying the new value of the vigilance parameter (ρ') that must be used in conjunction to a new system configuration $M' = M + \Delta M$and $l' = l + \Delta l$ in order to obtain the same Type I error rate of the initial setting point.

7.2. A numerical example

This section applies the tuning criterion of Eq. (21) in several simulation examples. Consider the initial setting configuration of the neural system with window size $M = 50$ and coding limit $l = 6$. In order to obtain a false alarm rate about equal to $\tilde{\alpha} = 0.27\%$ (i.e. the expected false alarm rate of a traditional Shewhart control chart), a trial and error approach was implemented in order to select a proper value of the vigilance parameter. From simulation results (50 batches of 2000 M-dimensional non-overlapping input vectors whose components are independently and normally distributed), it resulted that by using $\rho = 0.9128$, the neural network provides a false alarm rate equal to $\alpha(M,l,\rho) = 0.263\%$ with t-based confidence interval (coverage 95%) $[0.234\%, 0.292\%]$. For the configuration $M = 50$, $l = 6$ and $\rho = 0.9128$, the upper bound limit of the Vysochanskiĭ–Petunin inequality can be computed by Eq. (19) and it results equal to $\sup_{\alpha} = 5.25\%$.

Fourteen new configurations of the neural-based system are considered by setting the window size and coding limit to the new values $M' = M + \Delta M$ and $l' = l + \Delta l$, respectively, where $\Delta M = 0; \pm 5; \pm 15$ and $\Delta l = 0; \pm 2$. The tuning of such neural networks, in order to maintain the Type I error rate about equal to the

Table 4
Tuning phase. Type I error rates (simulation results 50 batches of 2000 M-dimensional vectors)

Neural network design				Simulation results	
M	l	$\sup_\alpha$	ρ	$\alpha(M,l,\rho)$	Conf. Inter. 95%
50	6	5.25%	0.9128	0.263%	[0.234%,0.292%]
M'	l'	$\rho' = 1 - \frac{1}{2l'}(\mu_0' + \frac{2\sigma_0'}{3\sqrt{\sup_\alpha}})$		$\alpha(M',l',\rho')$	Conf. Inter. 95%
35	4	0.8631		0.288%	[0.257%,0.319%]
45	4	0.8676		0.261%	[0.224%,0.298%]
50	4	0.8693		0.265%	[0.236%,0.294%]
55	4	0.8707		0.278%	[0.247%,0.309%]
65	4	0.8731		0.261%	[0.232%,0.290%]
35	6	0.9087		0.285%	[0.254%,0.316%]
45	6	0.9117		0.259%	[0.223%,0.295%]
55	6	0.9138		0.279%	[0.247%,0.311%]
65	6	0.9154		0.262%	[0.233%,0.291%]
35	8	0.9315		0.283%	[0.252%,0.314%]
45	8	0.9338		0.262%	[0.225%,0.299%]
50	8	0.9346		0.263%	[0.234%,0.292%]
55	8	0.9354		0.287%	[0.255%,0.319%]
65	8	0.9366		0.269%	[0.239%,0.299%]

reference value $\alpha = 0.27\%$, has been implemented by Eq. (21). In Table 4, for each of the new neural network configurations, the resulting values of the vigilance parameter have been reported.

In order to validate the method, the actual Type I error rates have been also estimated through simulation (50 batches of 2000 M-dimensional non-overlapping input vectors). The t-based confidence intervals (95% coverage) of the estimated Type I error rates have been reported in Table 4.

As it can be noticed, each of the new setting configurations presents an actual false alarm rate about equal to 0.27%, hence Eq. (21) allows for selecting the proper vigilance parameter value in order to maintain false alarm rate approximately unchanged.

8. Conclusions

With the widespread exploitation of automated production and inspection in several industrial applications, the tasks of SPC, traditionally performed by quality practitioners, have to be automated. As an example, applications based on continuous product manufacturing operations, including the manufacture of paper and wood products, chemicals, and cold-rolled steel products require computer-based algorithms that implement various quality control tasks automatically.

This paper is focused on an ART-based neural network for automating quality control of manufacturing processes and is mainly intended to provide a basic description of such a network to quality practitioners with a statistical background. We achieve this by deriving a statistical model of Fuzzy ART algorithm in a very specific case in order to understand the capabilities and potentials of neural networks for manufacturing quality control.

The analysed neural approach is mainly intended for identifying unnatural process behaviour by detecting changes in the state of the process. This method is quite simple to implement, and the training set can be limited to a single ideal pattern. Some important details of the proposed neural-based control schema are discussed through geometrical concepts, while the effects of three tuneable parameters on the performance are analytically examined by means of a probabilistic model. The performances of the control system reported in the paper are obtained analytically and the problem of predicting neural network performance, as upper bound limits for the errors of Types I and II, is considered.

Taking advantage of the statistical analysis described in the present work, a promising future direction of research can be related to the design of a Fuzzy ART control procedure in which the values of the tuneable parameters is achieved by optimising (i.e. minimising) a specific cost function.

Acknowledgements

This work has been partially funded by the Ministry of Education, University and Research of Italy (MIUR).

Appendix A. Proofs

In this appendix, three properties reported in Section 5 are verified.

Proof of Property 1. Consider the quality characteristic of each process output measured at time τ given by $\forall\tau$: $Y_\tau = Z_\tau + S_\tau$; $\quad Z_\tau \sim \mathrm{NID}(0,1)$.

Referring for simplicity to the case $|S_\tau| < l$ and since Y_τ is normally distributed, the expected mean value of $\min(l, |Y_\tau|)$ is given by

$$E[\min(l,|Y_\tau|)] = \int_{-l}^{+l} |x| \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x + l\Phi(-l-S_\tau) + l[1-\Phi(l-S_\tau)].$$

The first term on the right-hand of former equation can be simply calculated as follows:

$$\begin{aligned}
&\int_{-l}^{+l} |x| \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x \\
&= \int_0^{+l} x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x - \int_{-l}^{0} x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x \\
&\overset{z=x-S_\tau}{=} \int_{-S_\tau}^{+l-S_\tau} z \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z + \int_{-S_\tau}^{+l-S_\tau} S_\tau \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z \\
&\quad - \int_{-l-S_\tau}^{-S_\tau} z \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z - \int_{-l-S_\tau}^{-S_\tau} S_\tau \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z \\
&= \left[-\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)\Big|_{-S_\tau}^{+l-S_\tau}\right] + S_\tau[\Phi(l-S_\tau) - \Phi(-S_\tau)] \\
&\quad - \left[-\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)\Big|_{-l-S_\tau}^{-S_\tau}\right] - S_\tau[\Phi(-S_\tau) - \Phi(-l-S_\tau)] \\
&= -\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(l-S_\tau)^2}{2}\right) + \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{S_\tau^2}{2}\right) + S_\tau[\Phi(l-S_\tau) - \Phi(-S_\tau)] \\
&\quad + \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{S_\tau^2}{2}\right) - \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(l+S_\tau)^2}{2}\right) - S_\tau[\Phi(-S_\tau) - \Phi(-l-S_\tau)] \\
&= \frac{1}{\sqrt{2\pi}} \left[2\exp\left(-\frac{S_\tau^2}{2}\right) - \exp\left(-\frac{(l-S_\tau)^2}{2}\right) - \exp\left(-\frac{(l+S_\tau)^2}{2}\right)\right] \\
&\quad + S_\tau[\Phi(l-S_\tau) + \Phi(-l-S_\tau) - 2\Phi(-S_\tau)],
\end{aligned}$$

where $\Phi(\cdot)$ is the cumulative standard normal distribution function. Therefore, the expected value $E[\min(l, |Y_\tau|)]$ at time of index τ is the sum of three components, which depend on the parameters l (the tuneable parameter of the *Coding* pre-processing stage) and S_τ (the unnatural offset of the process mean at time τ)

$$E[\min(l,|Y_\tau|)] = \Psi_1(l,S_\tau) + \Psi_2(l,S_\tau) + \Psi_3(l,S_\tau),$$

where the three "psi-functions" are expressed by the following equations:

$$\Psi_1(l,S_\tau) = \frac{1}{\sqrt{2\pi}} \left[2\exp\left(-\frac{S_\tau^2}{2}\right) - \exp\left(-\frac{(l-S_\tau)^2}{2}\right) - \exp\left(-\frac{(l+S_\tau)^2}{2}\right)\right],$$

$$\Psi_2(l,S_\tau) = S_\tau[\Phi(l-S_\tau) + \Phi(-l-S_\tau) - 2\Phi(-S_\tau)],$$

$$\Psi_3(l,S_\tau) = l[\Phi(-l-S_\tau) + 1 - \Phi(l-S_\tau)].$$

The second-order moment about the origin of the random variable $\min(l, |Y_\tau|)$ at time τ is given by

$$E[\min(l,|Y_\tau|)^2] = \int_{-l}^{l} \frac{1}{\sqrt{2\pi}} x^2 \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x + l^2[\Phi(-l-S_\tau) + 1 - \Phi(l-S_\tau)].$$

The first term on the right-hand of former equation can be computed by parts as follows:

$$\begin{aligned}
&\int_{-l}^{l} \frac{1}{\sqrt{2\pi}} x^2 \exp\left(-\frac{(x-S_\tau)^2}{2}\right) \mathrm{d}x \\
&\overset{z=x-S_\tau}{=} \int_{-l-S_\tau}^{l-S_\tau} \frac{1}{\sqrt{2\pi}} (z+S_\tau)^2 \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z \\
&= \int_{-l-S_\tau}^{l-S_\tau} \frac{1}{\sqrt{2\pi}} z^2 \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z + \int_{-l-S_\tau}^{l-S_\tau} \frac{1}{\sqrt{2\pi}} S_\tau^2 \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z \\
&\quad + \int_{-l-S_\tau}^{l-S_\tau} \frac{2}{\sqrt{2\pi}} S_\tau z \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z;
\end{aligned}$$

the first term in the previous equation is equal to

$$\begin{aligned}
&\int_{-l-S_\tau}^{l-S_\tau} \frac{1}{\sqrt{2\pi}} z^2 \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z \\
&= \int_{-l-S_\tau}^{l-S_\tau} -\frac{1}{\sqrt{2\pi}} \mathrm{d}z \left[\exp\left(-\frac{z^2}{2}\right)\right] \\
&= -\frac{1}{\sqrt{2\pi}} \left[z\exp\left(-\frac{z^2}{2}\right)\Big|_{-l-S_\tau}^{l-S_\tau} - \int_{-l-S_\tau}^{l-S_\tau} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z\right] \\
&= -\frac{1}{\sqrt{2\pi}} \begin{Bmatrix} (l-S_\tau)\exp\left(-\dfrac{(l-S_\tau)^2}{2}\right) \\ -(-l-S_\tau)\exp\left(-\dfrac{(-l-S_\tau)^2}{2}\right) \\ -\sqrt{2\pi}[\Phi(l-S_\tau) - \Phi(-l-S_\tau)] \end{Bmatrix};
\end{aligned}$$

the second term is equal to

$$\int_{-l-S_\tau}^{l-S_\tau} \frac{1}{\sqrt{2\pi}} S_\tau^2 \exp\left(-\frac{z^2}{2}\right) dz = S_\tau^2[\Phi(l-S_\tau) - \Phi(-l-S_\tau)];$$

the last term is

$$\int_{-l-S_\tau}^{l-S_\tau} \frac{2}{\sqrt{2\pi}} S_\tau z \exp\left(-\frac{z^2}{2}\right) dz = 2S_\tau\left[-\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2}{2}\right)\Big|_{-l-S_\tau}^{l-S_\tau}\right] = S_\tau\sqrt{\frac{2}{\pi}}\left[-\exp\left(-\frac{(l-S_\tau)^2}{2}\right) + \exp\left(-\frac{(-l-S_\tau)^2}{2}\right)\right].$$

Therefore, the second-order moment about the origin of the random variable $\min(l, |Y_\tau|)$ at time of index τ is the sum of four components, which depend on the parameters l (the tuneable parameter of the *Coding* pre-processing stage) and S_τ (the unnatural offset of the process mean at time τ)

$$E[\min(l, |Y_\tau|)^2] = \Psi_4(l, S_\tau) + \Psi_5(l, S_\tau) + \Psi_6(l, S_\tau) + \Psi_7(l, S_\tau),$$

where the additional four psi-functions are expressed by following formulae:

$$\Psi_4(l, S_\tau) = -\frac{1}{\sqrt{2\pi}}\left[(l-S_\tau)\exp\left(-\frac{(l-S_\tau)^2}{2}\right) - (-l-S_\tau)\exp\left(-\frac{(-l-S_\tau)^2}{2}\right)\right],$$

$$\Psi_5(l, S_\tau) = (1+S_\tau^2)[\Phi(l-S_\tau) - \Phi(-l-S_\tau)],$$

$$\Psi_6(l, S_\tau) = S_\tau\sqrt{\frac{2}{\pi}}\left[-\exp\left(-\frac{(l-S_\tau)^2}{2}\right) + \exp\left(-\frac{(-l-S_\tau)^2}{2}\right)\right],$$

$$\Psi_7(l, S_\tau) = l^2[\Phi(-l-S_\tau) + 1 - \Phi(l-S_\tau)].$$

Finally, the variance of the random variable $\min(l, |Y_\tau|)$ at time τ can be simply computed as follows:

$$\mathrm{Var}[\min(l, |Y_\tau|)] = -E^2[\min(l, |Y_\tau|)] + E\lfloor \min(l, |Y_\tau|)^2 \rfloor.$$

It should be pointed out that results of Property 1 hold for any value of S_τ. □

Proof of Property 2. Property 2 is a corollary of Property 1. Indeed, Property 2 is immediately proved by Property 1 by assuming $S_\tau = 0$. □

Proof of Property 3. If the process is in an unnatural state (i.e. $\forall\tau: S_\tau = \pm\varphi$ where φ is a constant value greater than zero and less than l), and considering a fixed value of limit l, then the psi-functions are symmetric functions of the variable φ. In particular it holds that

$$\Psi_1(l, -\varphi) = \frac{1}{\sqrt{2\pi}}\left[2\exp\left(-\frac{\varphi^2}{2}\right) - \exp\left(-\frac{(l+\varphi)^2}{2}\right) - \exp\left(-\frac{(l-\varphi)^2}{2}\right)\right] = \Psi_1(l, \varphi),$$

$$\begin{aligned}\Psi_2(l, -\varphi) &= -\varphi[\Phi(l+\varphi) + \Phi(-l+\varphi) - 2\Phi(\varphi)] \\ &= -\varphi[1 - \Phi(-l-\varphi) + 1 - \Phi(l-\varphi) - 2 + 2\Phi(-\varphi)] \\ &= \varphi[\Phi(l-\varphi) + \Phi(-l-\varphi) - 2\Phi(-\varphi)] \\ &= \Psi_2(l, \varphi),\end{aligned}$$

$$\begin{aligned}\Psi_3(l, -\varphi) &= l[\Phi(-l+\varphi) + 1 - \Phi(l+\varphi)] \\ &= l[1 - \Phi(l-\varphi) + 1 - 1 + \Phi(-l-\varphi)] \\ &= l[\Phi(-l-\varphi) + 1 - \Phi(l-\varphi)] \\ &= \Psi_3(l, \varphi),\end{aligned}$$

$$\begin{aligned}\Psi_4(l, -\varphi) &= -\frac{1}{\sqrt{2\pi}}\left[(l+\varphi)\exp\left(-\frac{(l+\varphi)^2}{2}\right) - (-l+\varphi)\exp\left(-\frac{(-l+\varphi)^2}{2}\right)\right] \\ &= -\frac{1}{\sqrt{2\pi}}\left[(l-\varphi)\exp\left(-\frac{(l-\varphi)^2}{2}\right) - (-l-\varphi)\exp\left(-\frac{(-l-\varphi)^2}{2}\right)\right] \\ &= \Psi_4(l, \varphi),\end{aligned}$$

$$\begin{aligned}\Psi_5(l, -\varphi) &= (1+\varphi^2)[\Phi(l+\varphi) - \Phi(-l+\varphi)] \\ &= (1+\varphi^2)[1 - \Phi(-l-\varphi) - 1 + \Phi(l-\varphi)] \\ &= (1+\varphi^2)[\Phi(l-\varphi) - \Phi(-l-\varphi)] \\ &= \Psi_5(l, \varphi),\end{aligned}$$

$$\begin{aligned}\Psi_6(l, -\varphi) &= -\varphi\sqrt{\frac{2}{\pi}}\left[-\exp\left(-\frac{(l+\varphi)^2}{2}\right) + \exp\left(-\frac{(-l+\varphi)^2}{2}\right)\right] \\ &= \varphi\sqrt{\frac{2}{\pi}}\left[-\exp\left(-\frac{(l-\varphi)^2}{2}\right) + \exp\left(-\frac{(-l-\varphi)^2}{2}\right)\right] \\ &= \Psi_6(l, \varphi),\end{aligned}$$

$$\begin{aligned}\Psi_7(l, -\varphi) &= l^2[\Phi(-l+\varphi) + 1 - \Phi(l+\varphi)] \\ &= l^2[1 - \Phi(l-\varphi) + 1 - 1 + \Phi(-l-\varphi)] \\ &= l^2[\Phi(-l-\varphi) + 1 - \Phi(l-\varphi)] \\ &= \Psi_7(l, \varphi).\end{aligned}$$

Hence, Property 3 is a corollary of Property 1 when $S_\tau = \pm\varphi$. □

Appendix B. Simulation validation

This appendix provides the reader with a short comparison between the neural network performances and those of traditional SPC tools. The Type II error rates presented by the ART network are estimated for process mean changes of 1.0, 1.5 and 2.0 unit of standard deviations and then, they are compared to those of three SPC benchmarks. The evaluation procedure of the neural network involves the following steps:

- *Configuration phase*: The ART neural network of Section 7.2 has been used for performance evaluation ($M = 50$, $l = 6$). It consists of 50 neurones in the $F0$ layer, 100 neurones in the field $F1$, and a single node in the $F2$ layer.
- *Training phase*: The implemented ART neural network has been trained on the process nominal value $\mu = 0$.
- *Tuning phase*: In order to compare the neural network to any traditional charting technique it is required that performances must be identical when the process is in a natural state (Type I error rates). This serves to provide an unbiased comparison when the process drifts to unnatural states. Hence, the vigilance parameter ρ of the ART neural network was in turn adjusted in order to give a comparable performance in terms of the Type I error rate ($\hat{\alpha}_{nn}$) to that of a predefined SPC benchmark ($\hat{\alpha}_{cc}$). In particular, the neural network has been tuned on 50 streams of 2000 simulated in-control data normally distributed with zero mean and standard deviation one.
- *Performance analysis*: Comparisons of the neural network performances to those of a control chart benchmark are based on Type II error rates, which have been experimentally estimated by introducing two controlled disturbance signals: systematic variation and shift (Section 3). For each disturbance signal and each magnitude setting, Type II error point and interval estimators were assessed on 50 batches of 2000 independent simulation runs. This simulation methodology has been chosen in order to obtain independence of the batch means by passing a test for correlation at lag 1.

The following SPC benchmarks have been selected.

1. Bilateral cumulative summation (CUSUM) control chart with parameters $k = 0.5$ and $h = 4.7749$ (Montgomery, 2000). Estimated Type I error rate $\hat{\alpha}_{cc} = 0.269\%$.
2. Shewhart control chart with Western Electric run rules (Western Electric, 1956). Estimated Type I error rate $\hat{\alpha}_{cc} = 1.115\%$.
3. Shewhart control chart with Western Electric run rules and four additional sensitising rules (Nelson, 1984). Estimated Type I error rate $\hat{\alpha}_{cc} = 1.617\%$.

A one-step moving window of size M has been exploited for Type II error estimation. For the comparison to be unbiased, the alarms of a control chart occurred during the first $M - 1$ observations were neglected, and the performances were estimated for time indexes $t \geqslant M$. Numerical results and comparisons are discussed in the following sections for each of the SPC benchmarks.

B.1. CUSUM control chart

Table 5 compares Types I and II errors of the CUSUM schema $k = 0.5$ and $h = 4.7749$ (Type I error $\hat{\alpha}_{cc} = 0.269\%$) to those of the ART neural network with vigilance parameter $\rho = 0.9128$ (Type I error $\hat{\alpha}_{nn} = 0.263\%$). The values of the CUSUM parameters k and h have been set for signalling a shift of one standard deviation in the mean with a false alarm rate about equal to that of the standard Shewhart 3-sigma control chart (0.27%).

In order to confirm the statistical significance of the difference between neural network and control chart performances, the t-based confidence intervals (coverage 95%) have been also presented in the same table. The columns marked as $\hat{\beta}_{nn} - \hat{\beta}_{cc}$ gives the difference between the Type II error point estimators. The lower limit of the t-based confidence intervals is reported in the column labelled as $(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$, while the upper limit in the column labelled as $(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$.

The neural network performance is better (i.e. smaller Type II errors) than that of the CUSUM chart for signalling systematic variations of the process mean. On the other hand, the neural network has a worse performance if compared to the CUSUM chart (i.e. higher Type II errors) for shifts of 1.0 unit of standard deviation. For higher shifts, the performances are similar.

The results of Table 5 prove that the CUSUM schema cannot be adopted as the sole tool for signalling a generic modification in the state of the process (e.g. it performs poorly in signalling alarms for a systematic variation of the mean, while it performs better for a constant shift of the mean). On the other hand, the ART neural network appears able to recognise different kinds of change with the same capability. Indeed, the neural network performances in tackling systematic variations and shifts of the mean are similar for each level of magnitude.

Table 5
Comparison between neural network and CUSUM control chart (simulation results, 50 sets of 2000 data)

	CUSUM $k = 0.5$ $h = 4.7749$	Fuzzy ART $M = 50$ $\rho = 0.9128$	Comparison Neural network vs. control chart		
Natural	$\hat{\alpha}_{cc}$	$\hat{\alpha}_{nn}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_-$	$\hat{\alpha}_{nn} - \hat{\alpha}_{cc}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_+$
	0.269%	0.263%	−0.050%	−0.006%	0.038%
Sys. Var.	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
1.0	99.487%	67.645%	−32.741%	−31.842%	−30.943%
1.5	99.158%	0.000%	−99.225%	−99.158%	−99.091%
2.0	98.424%	0.000%	−98.508%	−98.424%	−98.340%
Shift	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
1.0	0.019%	67.989%	67.045%	67.970%	68.895%
1.5	0.000%	0.012%	0.000%	0.012%	0.030%
2.0	0.000%	0.000%	0.000%	0.000%	0.000%

Table 6
Comparison between neural network and Shewhart control chart with standard Western Electric (1956) run rules (simulation results, 50 sets of 2000 data)

	Shewhart WE RRs	Fuzzy ART $M = 50$ $\rho = 0.9168$	Comparison Neural network vs. control chart		
Natural	$\hat{\alpha}_{cc}$	$\hat{\alpha}_{nn}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_-$	$\hat{\alpha}_{nn} - \hat{\alpha}_{cc}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_+$
	1.115%	1.133%	−0.075%	0.018%	0.111%
Sys. Var.	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
1.0	95.635%	42.181%	−54.654%	−53.454%	−52.254%
1.5	87.731%	0.000%	−87.901%	−87.731%	−87.561%
2.0	72.220%	0.000%	−72.536%	−72.220%	−71.904%
Shift	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
1.0	80.820%	43.423%	−38.506%	−37.397%	−36.288%
1.5	42.902%	0.008%	−43.419%	−42.894%	−42.369%
2.0	8.335%	0.000%	−8.609%	−8.335%	−8.061%

B.2. Shewhart control chart with Western Electric run rules

Table 6 compares Type II errors of the Shewhart chart with the three Western Electric run rules (two of three consecutive points outside the ±2-sigma limits; four of five consecutive points beyond the ±1-sigma limits; a run of eight consecutive points on one side of the centreline), to those of the neural network.

While the simultaneous tests proposed in the Western Electric Statistical Quality Control Handbook improve the performance of the Shewhart control chart in recognising changes of the process mean, they do so at the cost of increases in false out-of-control signals. Therefore, in this case a higher value of the vigilance parameter ($\rho = 0.9168$) has been adopted in order to obtain a neural network false alarm rate that is comparable to the increased Type I error of the benchmark.

The results of Table 6 prove that the proposed neural network achieves better performances (lower Type II error rates) than those of the SPC benchmark in recognising any disturbance signals.

B.3. Shewhart control chart with Western Electric run rules and sensitising rules

Table 7 compares Type II errors of the Shewhart control chart with seven run rules, to those given by the ART neural network with vigilance parameter $\rho = 0.9179$. The run rules implemented in the SPC benchmark are the standard three tests described in Western Electric (1956) and four additional sensitising rules proposed by Nelson (1984): six points in a row steadily

Table 7
Comparison between neural network and Shewhart control chart with standard Western Electric (1956) and Nelson (1984) sensitising run rules (simulation results, 50 sets of 2000 data)

	Shewhart WE+SR RRs	Fuzzy ART $M=50$ $\rho=0.9179$	Comparison Neural network vs. control chart		
Natural	$\hat{\alpha}_{cc}$	$\hat{\alpha}_{nn}$	$(\hat{\alpha}_{nn}-\hat{\alpha}_{cc})_-$	$\hat{\alpha}_{nn}-\hat{\alpha}_{cc}$	$(\hat{\alpha}_{nn}-\hat{\alpha}_{cc})_+$
	1.671%	1.653%	−0.135%	−0.018%	0.099%
Sys. Var.	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_-$	$\hat{\beta}_{nn}-\hat{\beta}_{cc}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_+$
1.0	87.888%	34.829%	−54.326%	−53.059%	−51.792%
1.5	65.761%	0.000%	−66.193%	−65.761%	−65.329%
2.0	32.188%	0.000%	−32.652%	−32.188%	−31.724%
Shift	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_-$	$\hat{\beta}_{nn}-\hat{\beta}_{cc}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_+$
1.0	80.255%	35.669%	−45.727%	−44.586%	−43.445%
1.5	42.546%	0.005%	−43.017%	−42.541%	−42.065%
2.0	8.265%	0.000%	−8.521%	−8.265%	−8.009%

increasing or decreasing; fifteen points in a row within the ±1-sigma limits; fourteen points in a row alternating up and down; eight points in a row on both sides beyond the ±1-sigma limits. The use of four additional run rules increases the false alarm rate of the SPC benchmark, thus, a higher value of the vigilance parameter ($\rho = 0.9179$) has been adopted in this case.

The proposed neural network has better performances (lower Type II error rates) than those of the SPC chart when recognising the disturbance signals, for each magnitude level considered in the test.

B.4. Discussion

From the experimental results and comparisons, it is fair to conclude that the proposed ART-based control system is superior to (or in par with) several SPC charts in terms of Type II error rates. In particular, test comparisons show that the proposed method is a good control procedure for tackling different kinds of alteration in the process mean. For example, the neural network possesses superior detection capability against fluctuations of the process mean (systematic variations) than the CUSUM test, while it presents a comparable (or slightly worse) ability in signalling constant shifts. At the same time, the neural network outperforms Shewhart control charts with a set of run rules and sensitising rules.

Simulation results prove that the proposed approach can model different control strategies simultaneously: e.g., those of a CUSUM and of a Shewhart control chart with run rules, which were designed to recognise different kinds of change in the process structure (steady shifts of moderate magnitude and sudden fluctuation of high magnitude in the process mean, respectively). Indeed, the neural network can be potentially adopted to signal any types of unnatural pattern, so it provides a powerful diagnostic tool for detecting assignable causes in real processes.

Thus, the main advantage of the proposed approach over traditional ones is that it can be exploited as the sole tool for signalling a generic modification in the state of the process. Indeed, the proposed neural network can be useful when starting processing of new products, or with a new installed process, for which no prior knowledge of the unnatural changes are available in advance in order to design a proper control strategy. However, since the ART-based approach can only signal generic unnatural process behaviours, the proposed system cannot substitute existing neural-based methodologies for detecting and classifying predictable unnatural patterns on control charts. It is a complementary promising tool capable of enhancing the effectiveness of quality control using neural network when no prior knowledge of the unnatural patterns is available for training.

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, 627–639.

Anagnostopoulos, G.C., Georgiopouolos, M., 2002. Category regions as a new geometrical concepts in Fuzzy-ART and Fuzzy-ARTMAP. Neural Networks 15, 1205–1221.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Chang, S.I., Aw, C.A., 1996. A neural fuzzy control chart for detecting and classifying process mean shifts. International Journal of Production Research 34 (8), 2265–2278.

Chang, S.I., Ho, E.S., 1999. A two-stage neural network approach for process variance change detection and classification. International Journal of Production Research 37 (7), 1581–1599.

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, B., Titterington, D.M., 1994. Neural networks: a review from a statistical perspective. Statistical Science 9 (1), 2–30.

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.

De Veaux, R.D., Schumi, J., Schweinsberg, J., Ungar, L.H., 1998. Prediction intervals for neural networks via nonlinear regression. Technometrics 40 (4), 273–282.

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.

Guh, R.S., Hsieh, Y.C., 1999. A neural network based model for abnormal pattern recognition of control charts. Computers and Industrial Engineering 36, 97–108.

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.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation, second ed. Prentice-Hall, Englewood Cliffs, NJ.

Huang, J., Georgiopoulos, M., Heileman, G.L., 1995. Fuzzy ART proprieties. Neural Networks 8 (2), 203–213.

Hwang, J.T.G., Ding, A.A., 1997. Prediction intervals for artificial neural networks. Journal of the American Statistical Association 92 (438), 748–757.

Hwarng, H.B., Chong, C.W., 1995. Detecting process non-randomness through a fast and cumulative learning ART-based pattern recognizer. International Journal of Production Research 33 (7), 1817–1833.

Hwarng, H.B., Hubele, N.F., 1993a. Back-propagation pattern recognizers for X-bar control charts: methodology and performance. Computers and Industrial Engineering 24 (2), 219–235.

Hwarng, H.B., Hubele, N.F., 1993b. X-bar control chart pattern identification through efficient off-line neural network training. IIE Transactions 25 (3), 27–40.

Montgomery, D.C., 2000. Introduction to Statistical Quality Control, fourth ed. Wiley, New York.

Nelson, L.S., 1984. The Shewhart control chart-tests for special causes. Journal of Quality Technology 16 (4), 237–239.

Perry, M.B., Spoerre, J.K., Velasco, T., 2001. Control chart pattern recognition using back propagation artificial neural networks. International Journal of Production Research 39 (15), 3399–3418.

Pacella, M., Semeraro, Q., Anglani, A., 2004a. Manufacturing quality control by means of a Fuzzy ART network trained on natural process data. Engineering Applications of Artificial Intelligence 17 (1), 83–96.

Pacella, M., Semeraro, Q., Anglani, A., 2004b. Adaptive resonance theory-based neural algorithms for manufacturing process quality control. International Journal of Production Research 40 (21), 4581–4607.

Pukelsheim, F., 1994. The three sigma rule. The American Statistician 48 (2), 88–91.

Ripley, B.D., 1994. Neural networks and related methods for classification. Journal of the Royal Statistical Society. Series B (Methodological) 56 (3), 409–456.

Stern, H.L., 1996. Neural networks in applied statistics. Technometrics 38 (3), 205–214.

Western Electric, 1956. Statistical Quality Control Handbook. Western Electric Corporation, Indianapolis, IN.

Zorriassantine, F., Tannock, J.D.T., 1998. A review of neural networks for statistical process control. Journal of Intelligent Manufacturing 9, 209–224.