INT. J. PROD. RES., 1 NOVEMBER 2004,
VOL. 42, NO. 21, 4581–4607

Adaptive Resonance Theory-based neural algorithms for manufacturing process quality control

M. PACELLA†*, Q. SEMERARO‡ and A. ANGLANI†

The demand for quality products in industry is continuously increasing. To produce products with consistent quality, manufacturing systems need to be closely monitored for any unnatural deviation in the state of the process. Neural networks are potential tools that can be used to improve the analysis of manufacturing processes. Indeed, neural networks have been applied successfully for detecting groups of predictable unnatural patterns in the quality measurements of manufacturing processes. The feasibility of using Adaptive Resonance Theory (ART) to implement an automatic on-line quality control method is investigated. The aim is to analyse the performance of the ART neural network as a means for recognizing any structural change in the state of the process when predictable unnatural patterns are not available for training. To reach such a goal, a simplified ART neural algorithm is discussed then studied by means of extensive Monte Carlo simulation. Comparisons between the performances of the proposed neural approach and those of well-known SPC charts are also presented. Results prove that the proposed neural network is a useful alternative to the existing control schemes.

1. Introduction

In any production process, a certain amount of variability exists in the measurements of quality parameters. Two sources of variation may affect such measurements: commonly they are referred to as unassignable and assignable causes (Montgomery 2000). The variation due to unassignable causes is the result of numerous unremarkable changes that may occur in the process. This kind of variation is to some extent inevitable without a profound revision of the production procedure. When only unassignable causes are in effect, a process is considered to be in a natural state (i.e. in control). On the other hand, the variation due to assignable causes is produced by factors that lie outside the process. New methods and different machines introduced into the system, or changes in the measurement instruments and in the turnover of labour force, are common examples of assignable causes. In such cases, the process is said to be in an unnatural state (i.e. out of control), and quality improvement is possible by detection and removal of the assignable causes.

Statistical Process Control (SPC) is a methodology based on several techniques that is aimed at monitoring variability in the measurements of quality parameters. Control charts are the most widely applied SPC tools used to reveal unnatural

Revision received April 2004.

†Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, Via per Monteroni, Lecce, I-73100 Italy.

‡Dipartimento di Meccanica, Politecnico di Milano, Via Bonardi, Milan, I-20133, Italy.

*To whom correspondence should be addressed. E-mail address: massimo.pacella@unile.it

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540410001715706

variations in the monitored measurements. They are based on the idea that if a process is in a natural state, then the series of quality measurements are predictable according to a specific statistical model.

Nowadays, with the movement towards computer-integrated manufacturing, the automation of SPC implementation is considered essential. In relatively recent years, artificial neural networks have been used for quality control applications (Zorriassantine and Tannock 1998). A neural network is a computer algorithm with the ability to learn a specific knowledge, to adapt it to new situations, and to provide reliable classifications and approximations of data (Haykin 1999). Neural networks learn a specific knowledge by iterating through a set of exemplar data. Learning can take place through internal clustering (self-organizing or competitive learning) or through paired training sets (supervised learning). A supervised approach, which requires user to prespecify the desired output, can be used for data modelling when calibrated training data are available.

Since supervised neural networks are capable of recalling learned patterns from noisy or incomplete representations, they were extensively exploited for control chart pattern recognition. Hwarng and Hubele (1993a, b) carried out extensive studies on control chart pattern recognition by training a back-propagation network (BPN) in order to detect predictable unnatural patterns. Cheng (1997) used modular neural networks to deal with higher noise-to-signal ratios in control chart pattern recognition. Guh and Tannock (1999) proposed a neural network model to recognize concurrent patterns (where more than one pattern exists together, which may be associated with different assignable causes). Guh and Hsieh (1999) presented a control system composed of several interconnected BPNs both to recognize the unnatural control chart patterns and to estimate their parameters. Perry *et al.* (2001) implemented two BPNs to detect unnatural patterns on control charts as specified by Western Electric (1956).

The results of such methodologies are promising, and the implementation of supervised neural networks for control chart pattern recognition has been shown to be successful in all these studies. However, reported approaches have assumed that the output class always corresponded with one (or more) of the predictable patterns, thus failing to identify unexpected patterns that might arise. To obtain an adequate number of training examples that mimic the series of quality measurements generated by the process in an unnatural state, an implicit assumption of the above approaches is that the group of unnatural patterns are known in advance. In actual cases, unnatural process outputs could not be manifested by the appearance of predictable patterns. Therefore, sufficient training examples of unnatural outputs may not be readily available.

The present paper proposes a different neural network approach for process monitoring. The aim is to develop a complementary neural-based approach to the existing methodologies that is capable of enhancing the effectiveness of quality control when no prior knowledge of the unnatural patterns is available for training. In particular, the proposed approach is based on the Adaptive Resonance Theory (ART) neural network as a means for recognizing any structural change in the state of a process. The monitoring procedure proposed can be useful when starting processing of new products, or with a new installed process.

The paper is structured as follows. In Section 2, the ART is briefly presented. In Section 3, the ART-based approach is illustrated. In Section 4, the operating phases of the proposed neural network for quality control are discussed. A simulation methodology is presented in Section 5, while experimental results and comparisons between neural network performances and those of well-known SPC benchmarks are provided in Section 6. Section 7 has conclusions and discusses some directions for further research. Finally, analytical and computational results on neural network performances are given in appendices A and B.

2. Adaptive Resonance Theory (ART) for quality control

ART has been introduced as a mathematical model for the description of biological brain functions such as learning, memory and pattern recognition (Hagan *et al.* 1996). This theory has led to an evolving series of neural network models, which includes ART1 (binary input), ART2 and Fuzzy ART (analogue input). An ART-based neural network can be used to cluster arbitrary data into groups with similar features. It operates by summarizing similar data into categories that are formed during training in a self-organizing manner. For further details, see Hagan *et al.* (1996).

An ART-based neural network was first exploited for quality control by Hwarng and Chong (1995). They used an ART1 network to implement a control chart pattern recognizer, which had a fast and cumulative training phase. The ART1 network was trained in a 'quasi-supervised' approach for identifying six predictable unnatural patterns (trends, stratification, cycles, systematic, mixture and shift).

Al-Ghanim (1997) exploited an ART1 network differently. The author did not develop a pattern recognizer, but a system for signalling any change in the structure of a process, i.e. a generic unnatural behaviour. A change in structure can also be manifested in the form of predictable unnatural patterns and, therefore, a detected change may result from a cycle, mixture, systematic or any other pattern. However, the detection of a specific unnatural pattern was not considered since for any input the output from Al-Ghanim's neural system indicated whether the process was in a natural or an unnatural state.

More specifically, the ART1 neural network was trained on a set of data produced by the process in a state of control (i.e. patterns of natural data). During this training phase, the network clusters natural patterns 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). After the training, the neural network can provide an indication that a structural change in process outputs has occurred when the current input does not fit to any of the learned natural clusters. The goal was to exploit an unsupervised neural system that autonomously identified the different structures of clusters in the process output, and that, through a calibration schema, provided the capability to recognize the state of the process (either natural or unnatural). The calibration of network outputs, based on knowledge of input classes, was used in order to obtain the correct classification of a few numbers of input patterns as in any supervised system.

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 supervised procedures. Similarly, our preliminary studies have demonstrated that while the ART1 network can be applied to detect high changes of the process mean, it cannot identify medium and small changes, e.g. less than two standard deviation (SD) units. This drawback can be mainly ascribed to the binary coding of the ART1 algorithm. Indeed, binary format of input data are a less flexible way of using process outputs than a method based on graded continuous number encoding.

Our research aims to extend the methodology proposed by Al-Ghanim as well as to improve the performances of an ART-based neural network for quality control. A new simplified ART algorithm (the Fuzzy ART), which does not require binary coding of input data, is investigated for quality control applications. Fuzzy ART is based on the fuzzy set theory operations, thus input values, as well as weights of the network links, can range only between zero and 1. This network is composed of two major subsystems: attentional and orienting. While in the former familiar patterns are processed, the latter resets the neural activity whenever an unfamiliar input pattern is submitted.

There are many desirable properties of learning and characteristics associated with a Fuzzy ART neural network. First, Fuzzy ART is capable of learning in both off-line (batch) and on-line (incremental) training modes. In addition, due to the nature of its neural model, responses of Fuzzy ART to input vectors can be easily explained, in contrast to other neural network models, where in general it is more difficult to explain why an input pattern produces a specific output. For properties of learning for Fuzzy ART, see Huang *et al.* (1995), Georgiopoulos *et al.* (1996, 1999) and Anagnostopoulos and Georgiopoulos (2002).

3. Outline of the proposed Fuzzy ART algorithm

By process monitoring, we mean the use of a control system that can cyclically check the desired stable state of the process. Assume that at time t the process quality characteristic is measured with reference to a constant nominal value μ. Denote by $\{Y_t\} = \{Y_1, Y_2, \ldots\}$ the time series of the quality characteristic measurements obeying the model:

$$Y_t = \mu + Z_t + S_t, \tag{1}$$

where $\{Z_t\}$ is a time series of random deviations with zero mean, which models the natural variation of process output data due to unassignable causes, and $\{S_t\}$ is an arbitrary disturbance time series that models the unnatural variation due to assignable causes.

The goal is to develop a neural network to signal any changes of the structure of a process. The output of the system is not intended to provide a classification of the disturbance signal $\{S_t\}$, but merely an out-of-control signal when testing the null hypothesis $H_0: Y_t = \mu + Z_t$ (the process is in control) against the alternative hypothesis $H_1: Y_t = \mu + Z_t + S_t$ (the process is out of control). As with every statistical test, errors of Type I (some action is taken although the process is in control, i.e. false alarm) and Type II (no action is taken although the process is out of control) can occur.

The proposed neural system for quality control and the reference process model are jointly shown in figure 1. At the time of index t, the control system accepts as input the quality measurement Y_t and produces the binary response b_t. The algorithm produces $b_t = 1$ if at time of index t no change in the structure of process has been detected (H_0), $b_t = 0$ otherwise (H_1).

As shown in figure 1, some pre-processing stages of input data take place before they are presented to the Fuzzy ART network. The first pre-processing stage is called *Window Forming*. It transforms the temporal series $\{Y_t\}$ of process data into

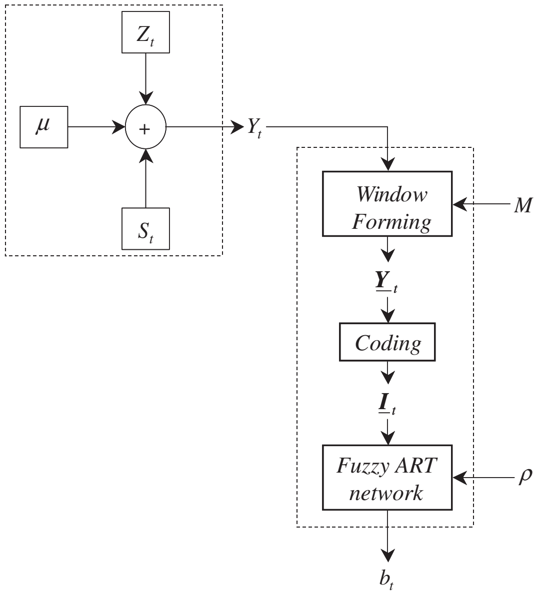

Figure 1. Proposed neural system for quality control.

M-dimensional vectors. Let $\underline{\boldsymbol{Y}}_t$ be the output of the Window Forming stage, the implemented pre-processing stage is as follows:

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

Specifically, the most recent M observations are collected to form the neural network input vector. In the most diffused literature, parameter M is referred to as the *Window Size* of the approach. 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. Nevertheless, since a single-step moving window is used, 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.

The second pre-processing stage takes as input an M-dimensional input vector $\underline{\boldsymbol{Y}}_t$ and transforms it into the corresponding M-dimensional output vector (say $\underline{\boldsymbol{I}}_t$) whose components fall into the interval $[0, 1]$. This pre-processing stage is called *Coding* and it consists of a linear rescaling of input variables into the range $[0, 1]$. *Coding* is necessary as the Fuzzy ART network can only accept input ranging between zero and 1. Let $l > 0$ be an appropriate limit for the natural absolute variation of the quality characteristic measurements $\{Y_t\}$ from the nominal value μ, the Coding stage can be summarized as follows:

$$\begin{aligned}\underline{\boldsymbol{I}}_t &= \left[I_{t-M+1}, I_{t-M+2}, \ldots, I_{t-1}, I_t\right] \\ &\times \begin{cases} I_\tau = 0 & Y_\tau \leq \mu - l; \\ I_\tau = \frac{1}{2}\left(1 + \frac{Y_\tau - \mu}{l}\right) & \mu - l < Y_\tau < \mu + l; \qquad t - M + 1 \leq \tau \leq t \\ I_\tau = 1 & \mu + l \leq Y_\tau; \end{cases}\end{aligned} \tag{3}$$

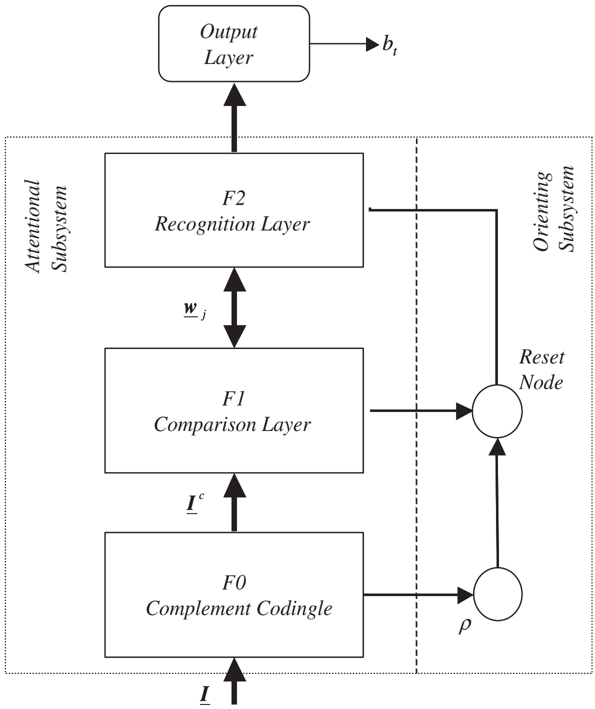

Figure 2. Proposed Fuzzy ART neural network.

The parameter $l > 0$ is a proper saturation limit for the deviations of the quality measurement from the process nominal value. Assuming that quality measurements of the process in a natural state are normally distributed, with zero mean and constant SD, σ, then $l = 3\sigma$ can be used for Coding. In this case, about 99.74% of the natural observations are expected to fall into the interval $[\mu - l, \mu + l]$.

The Fuzzy ART neural network implemented in this work is shown in figure 2. It consists of two major subsystems: attentional and orienting. Three fields of nodes denoted as $F0$, $F1$ and $F2$ compose the attentional subsystem. The $F0$ layer contains M neurones. The number of nodes in the $F1$ field is equal to $2M$, while the number of neurones in the $F2$ layer is equal to (or greater than) the number of clusters formed in the training phase. On the other hand, the orienting subsystem consists of a single node called the reset node. The output of the reset node, which depends on the vigilance parameter ρ, affects the nodes in the $F2$ layer.

In the $F0$ field, an additional pre-processing stage on the incoming input vectors $\underline{I}_t = \left[I_{t-M+1}, I_{t-M+2}, \ldots, I_{t-1}, I_t\right]$ is implemented. This pre-processing stage, called *Complement Coding*, accepts an M-dimensional vector $\underline{I}_t$ and produces the following $2M$-dimensional output vector $\underline{I}_t^c$ (where $\underline{\mathbf{1}}$ is the all-one M-dimensional vector):

$$\underline{I}_t^c = \left(\underline{I}_t, \underline{\mathbf{1}} - \underline{I}_t\right) = \left[I_{t-M+1}, I_{t-M+2}, \ldots, I_t, 1 - I_{t-M+1}, 1 - I_{t-M+2}, \ldots, 1 - I_t\right]. \quad (4)$$

Every node of index j in the $F2$ field is connected via a top-down weight with every node in the $F1$ field. The vector whose components are equal to the top-down weights emanating from node j in the $F2$ field is designated by $\underline{w}_j$. An additional layer was included into the implemented neural network: it consists of a single node that provides the output binary signal b_t of the quality control system. The proposed neural network was implemented by means of the NeuralWorks Professional II Plus software environment (NeuralWare 1997).

4. Operating phases of the proposed quality control system

4.1. *Configuration phase (choice of parameter* M)

As already mentioned, the Window Forming stage implies that the neural network cannot release a signal before the first M quality measurements have been collected. Thus, if the monitored process were in an unnatural state due to 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 window sizes can increase scrap and rework costs. On the other hand, window size can affect the recognition performance of the neural network. The larger the window size, the better the capability of the neural network in recognizing changes in the structure of the process. Since large window sizes can raise the efficiency in signalling changes of the process structure, the choice of M should be determined according to the minimum magnitude that is important to recognize, in the sense that a lower magnitude requires higher window sizes. A strategy for setting the window size is as follows.

- Identify the minimum variation of the process mean, which it is important to recognize quickly (say φ in terms of SD units).
- Choose the window size M as a function of φ. Usually, a smaller φ requires a higher window size; a rule of thumb is to choose $M \cong m/\varphi$ ($m = 75$ is the recommended value).

4.2. *Training phase*

In the proposed approach, the neural network is trained on the process nominal value μ. A single M-dimensional vector whose components are equal to the process nominal value forms the training list. Let $\underline{\boldsymbol{Y}}_\mu = [\mu, \mu, \ldots, \mu]$ be the training M-dimensional vector resulting from Window Coding. Let $\underline{\boldsymbol{I}}_\mu^c$ be the $2M$-dimensional training vector that results from the sequence of Coding and Complement Coding stages applied on $\underline{\boldsymbol{Y}}_\mu$ (each component of the vector $\underline{\boldsymbol{I}}_\mu^c$ is equal to 0.5).

First, all the top-down weights of the neural network are initialized to 1, and all the nodes of the $F2$ layer are *uncommitted* as they are not assigned to any template. The appearance of the vector $\underline{\boldsymbol{I}}_\mu^c$ across the $F1$ field produces bottom-up inputs that affect the nodes in the $F2$ layer. One of the uncommitted nodes of the $F2$ layer is then selected to represent the $\underline{\boldsymbol{I}}_\mu^c$ training vector, and the corresponding $2M$-dimensional top-down weight vector (say $\underline{\boldsymbol{w}}_\mu^c$) is set equal to $\underline{\boldsymbol{w}}_\mu^c = \underline{\boldsymbol{I}}_\mu^c$ no matter which values are taken by the vigilance parameter. Note that $\underline{\boldsymbol{I}}_\mu^c = (\underline{\boldsymbol{I}}_\mu, \underline{\mathbf{1}} - \underline{\boldsymbol{I}}_\mu)$, where $\underline{\boldsymbol{I}}_\mu$ is an M-dimensional vector, hence $\underline{\boldsymbol{w}}_\mu^c = (\underline{\boldsymbol{w}}_\mu, \underline{\mathbf{1}} - \underline{\boldsymbol{w}}_\mu)$, where $\underline{\boldsymbol{w}}_\mu$ is an M-dimensional vector. The training stops when the $\underline{\boldsymbol{w}}_\mu^c$ template is formed, i.e. when the vector $\underline{\boldsymbol{I}}_\mu^c$ is memorized into the ideal template $\underline{\boldsymbol{w}}_\mu^c$. As a result, the neural network forms a single category (one committed node in the $F2$ layer), which matches the training vector $\underline{\boldsymbol{w}}_\mu^c = \underline{\boldsymbol{I}}_\mu^c$.

Once this category has been produced, the learning process ends because no further training vectors are used. Note too that the vigilance parameter, choice parameter and learning rate of the ART network (Hagan *et al.* 1996) do not influence training, as once the ideal cluster has been formed (first iteration) no more iterations are implemented. In other words, no further choice of $F2$ nodes (affected by the choice parameter), weights adjustment (affected by the learning rate) or reset of a committed node (affected by the vigilance parameter) are allowed.

This training represents the least supervised approach practically possible by using a single training vector. Therefore, the neural network can be used in a detection mode as follows. When the training is completed and a new vector is presented to the network, it can be either accepted as a natural input that resembles the learned natural template (no alarm is signalled by the network) or rejected otherwise (an alarm is emitted by the network). Detecting whether an input vector resembles the ideal template is the function of the *matching algorithm*. An input vector will be classified as unnatural if it does not match the ideal template.

4.3. *Matching algorithm*

To discuss the matching algorithm, some preliminary notations must be introduced. The size of a vector $\underline{\boldsymbol{x}}$ is defined as $|\underline{\boldsymbol{x}}| = \sum_i |x_i|$. The minimum between vectors $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ is defined as $\underline{\boldsymbol{x}} \wedge \underline{\boldsymbol{y}} = [\min(x_1, y_1), \ldots, \min(x_i, y_i), \ldots]$, while the maximum is defined as $\underline{\boldsymbol{x}} \vee \underline{\boldsymbol{y}} = [\max(x_1, y_1), \ldots, \max(x_i, y_i), \ldots]$. In addition, the distance between $\underline{\boldsymbol{x}}$ and $\underline{\boldsymbol{y}}$ vectors is defined as $dis(\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}) = |\underline{\boldsymbol{x}} \vee \underline{\boldsymbol{y}}| - |\underline{\boldsymbol{x}} \wedge \underline{\boldsymbol{y}}|$.

Assume that at time of index $t \geq M$, an M-dimensional input vector $\underline{\boldsymbol{I}}_t$ is presented at the $F0$ field of the ART neural network trained on the process nominal value. The appearance of the $2M$-dimensional vector $\underline{\boldsymbol{I}}_t^c$ across the $F1$ field produces the activation of the single committed node in the $F2$ layer of the network (whose top-down weights vector is denoted by $\underline{\boldsymbol{w}}_\mu^c$). The appropriateness of the natural template $\underline{\boldsymbol{w}}_\mu^c$ to represent the input vector $\underline{\boldsymbol{I}}_t^c$ is checked by comparing the ratio of equation (5) with the vigilance parameter ρ $(0 \leq \rho \leq 1)$.

$$\frac{|\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_\mu^c|}{|\underline{\boldsymbol{I}}_t^c|}. \tag{5}$$

If such ratio is not less than the vigilance parameter ρ, the current input is classified as natural (i.e. no change in the structure of the process has been recognized), and the output is set to $b_t = 1$. Otherwise, the output is $b_t = 0$, which indicates that current input is considered unnatural (i.e. a change in the process structure has been recognized).

Since $|\underline{\boldsymbol{I}}_t^c| = M$ and $|\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_\mu^c| = |(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{1}} - \underline{\boldsymbol{I}}_t) \wedge (\underline{\boldsymbol{w}}_\mu, \underline{\boldsymbol{1}} - \underline{\boldsymbol{w}}_\mu)| = |\underline{\boldsymbol{I}}_t \wedge \underline{\boldsymbol{w}}_\mu| + M - |\underline{\boldsymbol{I}}_t \vee \underline{\boldsymbol{w}}_\mu|$, i.e. $|\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_\mu^c| = M - dis(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu)$, the matching algorithm implies that an input vector is classified natural if the following condition is passed:

$$\frac{|\underline{\boldsymbol{I}}_t^c \wedge \underline{\boldsymbol{w}}_\mu^c|}{|\underline{\boldsymbol{I}}_t^c|} \geq \rho \Leftrightarrow dis(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu) \leq M(1 - \rho). \tag{6}$$

The matching algorithm can be reformulated as follows:

- Calculate the distance between the current input and the single natural template that represents the process nominal value, i.e. $dis(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu) = |\underline{\boldsymbol{I}}_t \vee \underline{\boldsymbol{w}}_\mu| - |\underline{\boldsymbol{I}}_t \wedge \underline{\boldsymbol{w}}_\mu|$.
- If such distance is greater than $M(1 - \rho)$, then the input is classified as unnatural and the output is set to $b_t = 0$.
- Otherwise, the output is set to $b_t = 1$.

Since one template has been formed during training, when the learning is disengaged, there exists no competition among alternative committed nodes. Thus, the choice parameter of the ART network has no influence on the matching algorithm (Georgiopoulos *et al.* 1996).

4.4. *Tuning phase (choice of the parameter ρ)*

The criterion of equation (6) is parametrically affected by the vigilance ρ $(0 \leq \rho \leq 1)$. In the *tuning phase*, learning is disengaged (i.e. 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. 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. In any case, only a set of natural data is needed in order to select a proper value for the vigilance parameter.

On one hand, real patterns of data can be available when the process starts in a state of natural operation and remains in this state for a given time interval. Such a hypothesis is not unusual in quality control, as it resembles the designing phase of a statistic-based control chart (i.e. the positioning of the control limits) for a new starting process.

On the other hand, a mathematical model of the in-control process state can be formulated; hence, it can be exploited to simulate vectors of natural data. The natural variation component $\{Z_t\}$ of equation (1) can be realistically modelled by a random time series whose values are normal, independent and identically distributed (*NID*) over time, with mean zero and a known constant σ. Without loss of generality, it can be assumed $\sigma = 1$, i.e. $Z_t \sim NID(0, 1)$. This model has been adopted here in order to simulate the tuning list of natural data as it gives a close approximation to many types of practical manufacturing processes. The simulation model was implemented by means of the Matlab environment (MathWorks 1991) because of the availability of an efficient pseudo-random number generator (Vattulainen *et al.* 1995).

The value of the vigilance parameter ρ is chosen to maintain the false alarm rate (Type I error) about equal to a predefined value. This serves to provide an unbiased comparison of neural network performance to any traditional charting technique when the process drifts to unnatural states (Type II error rates). Under fixed Type I error rate, the Type II error should be as small as possible in order to signal any mean changes in the process as quickly as possible.

Note from equation (6) that a higher vigilance imposes a stricter matching criterion to the natural template learned in training phase (this results in higher false alarm rates α); on the contrary, a lower vigilance tolerates greater mismatches (this results in lower false alarm rates α). In other words, there exists a monotonically increasing relation between ρ and α. Thus, when natural process data are either available or they can be simulated, the vigilance parameter can be tuned (off-line) through a straightforward 'trial-and-error' approach (higher vigilance parameters ρ causes higher Type I error rates α and vice versa lower vigilance parameters ρ causes lower Type I error rates α). The objective is to assess the Type I error for different trial values of the vigilance parameter (by using the same neural network). This approach is quite common in quality control. As an example, it resembles the designing phase of a statistic-based control chart (e.g. the choice of parameter h for a CUSUM control chart). In addition, this approach has been exploited by several neural-network-based procedures that have appeared in the recent literature for monitoring process mean shift (Cheng and Cheng 2001, Wang and Chen 2002, Hwarng 2004). In such procedures, one parameter of the neural network is empirically tuned to obtain a predefined performance with the monitoring system, which is usually the Type I error rate, i.e. the average run length for an in-control process.

In actual applications, given a set of natural data, a computer program can be used to estimate iteratively Type I error rates α produced by the ART network for several vigilance parameters ρ. For example, a *binary search method*, which iteratively divides the space of admissible values for ρ (i.e. the interval of real numbers between 0 and 1) in subintervals of length $\left(\frac{1}{2}\right)^{iteration\_number}$, can be used to find an appropriate vigilance that allows for a specific Type I error rate.

Appendix A provides some analytical results on the relationship between neural network performance and vigilance parameter. Appendix B presents computational results on the effect of Type I and II error rates when the vigilance parameter and window size are altered.

5. Performance analysis

The implementation of the proposed neural-based approach requires no knowledge of the unnatural patterns. However, to estimate the neural system performances in signalling different changes of the process structure, unnatural process data were simulated. For performance evaluation purposes, simulation with equation (1) was used to generate vectors of unnatural data. When the process starts drifting from the natural state, a special disturbance signal $\{S_t\}$ overlaps the series of process output measurements. For performance evaluation purposes, the following four disturbance models of magnitude φ were considered.

Systematic variation:

$$S_t = \varphi(-1)^t. \tag{7}$$

Cyclic: let Ω be the period, the model is as follows:

$$S_t = \varphi\cos\left(\frac{2\pi t}{\Omega}\right). \tag{8}$$

Shift: let τ be the instant of shifting. A shift is modelled by:

$$S_t = \begin{cases} 0 & t < \tau \\ \varphi & t \geq \tau \end{cases}. \tag{9}$$

Mixture:

$$S_t = \varphi m_t; \quad m_t \in \{-1, 1\}, \tag{10}$$

where m_t is the status of a binary $(-1, 1)$ Markov's chain at time of index t and $m_1 = 1$ is the initial state. Let $p_c = P\{m_t \neq m_{t-1}|m_{t-1}\}$ and $P\{m_t = m_{t-1}|m_{t-1}\} = 1 - p_c$.

The proposed neural network parallels statistical hypothesis testing and its properties can be fully described in terms of Type I and II error rates. In a simulation-based test, the rate of the Type I errors (i.e. the sample mean of the alarm signals, say $\hat{\alpha}$) occurring in process data having only common sources of variation (natural state) is a 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 control. Similarly, the rate of the Type II errors (i.e. the non-alarm signals) occurring in the process data when a special disturbance, with a specific magnitude has been introduced (say $\hat{\beta}$) is a point estimator of the parameter $\beta = P\{H_0|H_1\}$, i.e. the expected probability that the control system signals no alarms although the process is actually out of control. Obviously, $\hat{\beta}$ depends on the special disturbance signal used to simulate a specific unnatural behaviour. Generally, the objective of any quality control system is to recognize changes of the process parameters as fast as possible (few Type II errors), without too many false alarms (few Type I errors).

The method of batch means has been used in this research to estimate the Type I and II errors as well as their variability (Kleinjnen and Van Groenendaal 1992). For the simulation tests presented below, 50 batches of 2000 data were used. Such a simulation methodology has been chosen to satisfy the following two criterions: (1) independence of the batch means by passing a test for correlation at lag 1; and (2) widths of the Type I and II error interval estimators (with coverage of 95%) less than 3%.

6. Experimental results

This section is focused on the neural network performances in signalling changes of the process structure. The Type II error rates of the ART neural network were experimentally estimated in signalling simulated changes of moderate and medium magnitudes (i.e. 0.25, 0.50, 1.00, 1.50 and 2.00 SD units) and then they were compared with the performances presented by a few SPC benchmarks.

The performance evaluation procedure of the ART neural network involved the following steps:

- Configuration phase. An ART neural network of window size equal to $M = 75$ was implemented. A neural network of parameter $M = 75$ was chosen for performance evaluation purposes because the window size should be large enough to enable accurate capability in signalling alarms when changes of a moderate magnitude (e.g. ≤ 1 SD unit) occur in the process. Therefore, the neural network, implemented in NeuralWorks Professional II Plus, consisted of 75 neurones in the $F0$ layer, 150 neurones in field $F1$ and a single node in the $F2$ layer.
- Training phase. The implemented ART neural network was trained on the process nominal value $\mu = 0$, i.e. the training list consisted of the single vector $\underline{\boldsymbol{Y}}_{\mu}$ with 75 components equal to zero.
- Tuning phase. To compare the implemented neural network with 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 (Type II error rates). Therefore, the vigilance parameter ρ of the implemented ART neural network was in turn adjusted to give a comparable performance in terms of the Type I error rate ($\hat{\alpha}_{nn}$) with that of a predefined SPC benchmark ($\hat{\alpha}_{cc}$). In particular, the neural network was used to recognize 50 streams of 2000 simulated in-control data normally distributed with zero mean and $\sigma = 1$.
- Performance analysis. A sequence of simulated process data was presented to the network in a moving window of 75 data, which was incremented forward by one process measurement, representing a single sampling interval. Each window of 75 data formed an input vector to the network. Comparisons

of the neural network performances with those of a control chart benchmark were based on Type II error rates, which were experimentally estimated by introducing four controlled disturbance signals (systematic variation, cyclic $\Omega = 4$, shift $\tau = 1$ and mixture $p_c = 0.4$) at different magnitudes (0.25, 0.50, 1.00, 1.50 and 2.00). For each disturbance signal and each magnitude setting, a Type II error point and interval estimators were assessed on 50 sets of 2000 independent simulation runs.

The following SPC benchmarks were selected:

- Bilateral cumulative summation (CUSUM) control chart of parameters $k = 0.5$ and $h = 4.7749$ (Montgomery 2000). The estimated Type I error rate was $\hat{\alpha}_{cc} = 0.269\%$. The vigilance parameter of the implemented neural network that allows one to obtain a comparable Type I error rate is $\rho = 0.8475$.
- Control chart with one run rule: two of three points beyond the control limit ± 1.9307 (Klein 2000). The estimated Type I error rate was $\hat{\alpha}_{cc} = 0.281\%$. The same vigilance parameter $\rho = 0.8475$ can be chosen for this comparison.
- Shewhart control chart with Western Electric (1956) run rules. The estimated Type I error rate was $\hat{\alpha}_{cc} = 1.115\%$. The setting of the vigilance parameter that allows one to obtain a comparable Type I error rate is $\rho = 0.8575$.
- Shewhart control chart with Western Electric (1956) run rules and four additional sensitizing rules (Nelson 1984). The estimated Type I error rate was $\hat{\alpha}_{cc} = 1.617\%$. The setting of the vigilance parameter that allows one to obtain a comparable Type I error rate is $\rho = 0.8610$.

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 $t \geq M$. Numerical results and comparisons are discussed in the following sections for each SPC benchmark.

6.1. *CUSUM control chart*

Table 1 compares Type I and II errors of the CUSUM schema $k = 0.5$ and $h = 4.7749$ (Type I error $\hat{\alpha}_{cc} = 0.269\%$) with those of the ART neural network with vigilance parameter $\rho = 0.8475$ (Type I error $\hat{\alpha}_{nn} = 0.262\%$). The CUSUM parameters k and h were set optimally for signalling a shift of 1 SD in the mean with a false alarm rate about equal to that of the standard Shewhart 3-*sigma* control chart (0.27%).

To confirm the statistical significance of the difference between neural network and control chart performance, the *t*-based confidence intervals (coverage 95%) have been also presented in table 1. Specifically, the confidence interval of the difference between the neural network and control chart Type I errors ($\hat{\alpha}_{nn} - \hat{\alpha}_{cc} = -0.007\%$) is $[-0.053\%, 0.039\%]$. Such a result shows that the neural network is comparable with the CUSUM chart in terms of false alarms as the confidence interval includes the zero (i.e. there is no statistical evidence to reject the hypothesis $\alpha_{nn} = \alpha_{cc}$).

The columns marked as $\hat{\beta}_{nn} - \hat{\beta}_{cc}$ give the difference between the Type II error point estimators. The lower limit of the *t*-based confidence interval is in the column labelled $(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$, and the upper limit is in the column labelled $(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$.

	CUSUM $k=0.5$ $h=4.7749$	Fuzzy ART $M=75$ $\rho=0.8475$	Comparison neural network versus 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%	0.262%	−0.053%	−0.007%	0.039%
Systematic variation	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_{-}$	$\hat{\beta}_{nn}-\hat{\beta}_{cc}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_{+}$
0.25	99.709%	99.485%	−0.284%	−0.224%	−0.164%
0.50	99.668%	97.479%	−2.323%	−2.189%	−2.055%
1.00	99.487%	1.825%	−98.058%	−97.662%	−97.266%
1.50	99.158%	0.000%	−99.225%	−99.158%	−99.091%
2.00	98.424%	0.000%	−98.508%	−98.424%	−98.340%
Cyclic $\Omega=4$					
0.25	99.721%	99.630%	−0.145%	−0.091%	−0.037%
0.50	99.689%	99.130%	−0.630%	−0.559%	−0.488%
1.00	99.523%	84.549%	−15.667%	−14.974%	−14.281%
1.50	99.234%	0.859%	−98.607%	−98.375%	−98.143%
2.00	98.603%	0.000%	−98.688%	−98.603%	−98.518%
Shift $\tau=1$					
0.25	99.034%	99.491%	0.381%	0.457%	0.533%
0.50	92.474%	97.486%	4.639%	5.012%	5.385%
1.00	0.019%	2.092%	1.622%	2.073%	2.524%
1.50	0.000%	0.000%	0.000%	0.000%	0.000%
2.00	0.000%	0.000%	0.000%	0.000%	0.000%
Mixture $p_c=0.4$					
0.25	99.618%	99.482%	−0.196%	−0.136%	−0.076%
0.50	99.136%	97.511%	−1.799%	−1.625%	−1.451%
1.00	95.527%	2.318%	−93.699%	−93.209%	−92.719%
1.50	82.168%	0.000%	−82.548%	−82.168%	−81.788%
2.00	51.802%	0.000%	−52.413%	−51.802%	−51.191%

Table 1. Comparison between a neural network and a CUSUM control chart (simulation results, 50 sets of 2000 data).

It appears that the neural network performance is better (i.e. smaller Type II errors) than that of the CUSUM chart for systematic, cyclic and mixture, no matter which magnitude is considered (0.25, 0.50, 1.00, 1.50 and 2.00). Indeed, the confidence intervals on the difference between β_{nn} and β_{cc} include only negative values and thus we can statistically conclude that $\beta_{nn} < \beta_{cc}$ for each of the above tests. For example, to signal an unnatural change in the process, which is simulated by a systematic variation of magnitude equal to $\varphi = 1.00$, the ART neural network presents a Type II error rate of about $\hat{\beta}_{nn} = 1.825\%$ versus $\hat{\beta}_{cc} = 99.487\%$ of the CUSUM chart. In the case of a cyclic variation of the same magnitude, the neural network has a worse performance ($\hat{\beta}_{nn} = 84.549\%$); nevertheless, it still outperforms the CUSUM chart ($\hat{\beta}_{cc} = 99.523\%$).

On the other hand, the neural network has a slightly worse performance if compared with the CUSUM chart (i.e. higher Type II errors) for shifts of 0.25, 0.50 and 1.00 SD units. For higher shifts (1.50 and 2.00 SD units), the neural network performance and that of the CUSUM chart are similar.

Table 1 shows that the CUSUM test performs very poorly in signalling alarms when systematic variations, cycles or mixtures overlap the series of quality measurements, while it performs better for mean shifts. This means that the CUSUM schema cannot be adopted as the sole tool for signalling a generic modification in the state of the process. On the other hand, the ART neural network appears able to recognize different kinds of change with the same capability. More specifically, note that the neural network performances in tackling systematic variations, shifts and mixtures are approximately the same for each level of magnitude, while for cycles the performances are slightly worse.

The comparison results can be practically generalized as follows. While the CUSUM chart can signal only a particular unnatural kind of variation (i.e. a constant shift), the ART neural network can extend the recognition ability to a wide set of potential unnatural changes of the process structure. Therefore, from the above comparison, it is fair to conclude that the proposed neural algorithm has better discriminatory capability to recognize unnatural changes of the process state that CUSUM charts cannot address.

6.2. *Control chart with one run rule*

Table 2 compares the performances of a control chart with one run rule (Klein 2000) with those of the neural network previously analysed ($\rho = 0.8475$). The run rule proposed by Klein is as follows: an alarm signal is released when at least two of three consecutive points exceed the ± 1.9307-*sigma* limits from the nominal value. The advantage of using one rule is that appropriate upper and lower control limits can be easily found so that the Type I error rate is equal to about 0.27%, the same value as that of the standard Shewhart control chart.

The results in table 2 show that the neural network is comparable with the SPC chart in terms of Type I error rates. Moreover, it appears that the neural network performances are better (i.e. smaller Type II errors) than those of the SPC chart for systematic, cyclic, shift and mixture disturbances, no matter which magnitude level has been considered (0.25, 0.50, 1.00, 1.50 and 2.00).

The results prove that the neural-based procedure achieves slight improvements over Klein's control schema when tackling changes of the process structure of small magnitude (0.25 and 0.50). On the other hand, the proposed neural system achieves

	Control Chart 2/3 ± 1.9307	Fuzzy ART $M = 75$ $\rho = 0.8475$	Comparison neural network versus control chart		
Natural	$\hat{\alpha}_{cc}$	$\hat{\alpha}_{nn}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_-$	$\hat{\alpha}_{nn} - \hat{\alpha}_{cc}$	$(\hat{\alpha}_{nn} - \hat{\alpha}_{cc})_+$
	0.281%	0.262%	−0.063%	−0.019%	0.025%
Systematic variation	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
0.25	99.609%	99.485%	−0.182%	−0.124%	−0.066%
0.50	99.349%	97.479%	−2.009%	−1.870%	−1.731%
1.00	97.173%	1.825%	−95.754%	−95.348%	−94.942%
1.50	91.161%	0.000%	−91.333%	−91.161%	−90.989%
2.00	78.476%	0.000%	−78.741%	−78.476%	−78.211%
Cyclic $\Omega = 4$					
0.25	99.699%	99.630%	−0.121%	−0.069%	−0.017%
0.50	99.659%	99.130%	−0.604%	−0.529%	−0.454%
1.00	99.448%	84.549%	−15.592%	−14.899%	−14.206%
1.50	99.029%	0.859%	−98.404%	−98.170%	−97.936%
2.00	98.438%	0.000%	−98.508%	−98.438%	−98.368%
Shift $\tau = 1$					
0.25	99.574%	99.491%	−0.146%	−0.083%	−0.020%
0.50	98.964%	97.486%	−1.631%	−1.478%	−1.325%
1.00	95.210%	2.092%	−93.583%	−93.118%	−92.653%
1.50	84.668%	0.000%	−84.945%	−84.668%	−84.391%
2.00	61.975%	0.000%	−62.308%	−61.975%	−61.642%
Mixture $p_c = 0.4$					
0.25	99.635%	99.482%	−0.213%	−0.153%	−0.093%
0.50	99.308%	97.511%	−1.968%	−1.797%	−1.626%
1.00	97.104%	2.318%	−95.259%	−94.786%	−94.313%
1.50	90.103%	0.000%	−90.275%	−90.103%	−89.931%
2.00	74.827%	0.000%	−75.189%	−74.827%	−74.465%

Table 2. Comparison between a neural network and a control chart with Klein (2000) run rule (simulation results, 50 sets of 2000 data).

	Shewhart WE RRs	Fuzzy ART $M = 75$ $\rho = 0.8575$	Comparison neural network versus 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%	1.058%	−0.154%	−0.057%	0.040%
Systematic variation	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_-$	$\hat{\beta}_{nn} - \hat{\beta}_{cc}$	$(\hat{\beta}_{nn} - \hat{\beta}_{cc})_+$
0.25	98.723%	98.157%	−0.698%	−0.566%	−0.434%
0.50	98.407%	90.994%	−7.904%	−7.413%	−6.922%
1.00	95.635%	0.141%	−95.644%	−95.494%	−95.344%
1.50	87.731%	0.000%	−87.901%	−87.731%	−87.561%
2.00	72.220%	0.000%	−72.536%	−72.220%	−71.904%
Cyclic $\Omega = 4$					
0.25	98.867%	98.627%	−0.349%	−0.240%	−0.131%
0.50	98.693%	96.873%	−2.017%	−1.820%	−1.623%
1.00	97.785%	49.005%	−50.089%	−48.780%	−47.471%
1.50	95.198%	0.037%	−95.283%	−95.161%	−95.039%
2.00	89.905%	0.000%	−90.094%	−89.905%	−89.716%
Shift $\tau = 1$					
0.25	98.143%	98.160%	−0.125%	0.017%	0.159%
0.50	95.704%	90.686%	−5.462%	−5.018%	−4.574%
1.00	80.820%	0.190%	−80.965%	−80.630%	−80.295%
1.50	42.902%	0.000%	−43.427%	−42.902%	−42.377%
2.00	8.335%	0.000%	−8.609%	−8.335%	−8.061%
Mixture $p_c = 0.4$					
0.25	98.651%	98.108%	−0.674%	−0.543%	−0.412%
0.50	97.790%	90.963%	−7.310%	−6.827%	−6.344%
1.00	93.260%	0.111%	−93.318%	−93.149%	−92.980%
1.50	80.829%	0.000%	−81.090%	−80.829%	−80.568%
2.00	58.286%	0.000%	−58.657%	−58.286%	−57.915%

Table 3. Comparison between a neural network and a Shewhart control chart with standard Western Electric (1956) run rules (simulation results, 50 sets of 2000 data).

significant improvements over Klein's control performances for the other change magnitudes (1.00, 1.50 and 2.00).

6.3. *Shewhart control chart with Western Electric run rules*

Table 3 compares Type I and II errors of the Shewhart chart with the three standard 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 centre line), with those of the neural network.

While the simultaneous tests proposed in Western Electric (1956) improve the performance of the Shewhart control chart in recognizing moderate changes of the process mean, they do so at the cost of increases in false out-of-control signals, as shown by Champ and Woodall (1987). Therefore, in this case, a higher vigilance parameter ($\rho = 0.8575$) was adopted to obtain a neural network false alarm rate that is comparable with the increased Type I error of the benchmark.

The performances of the neural network and those of the Shewhart control chart with Western Electric run rules are similar in recognizing small shifts of process mean: $\hat{\beta}_{nn} = 98.160\%$ and $\hat{\beta}_{cc} = 98.143\%$, respectively, with a confidence interval on the difference equal to $[-0.125\%, 0.159\%]$. On the other hand, the proposed neural network achieves better performances (lower Type II error rates) than those of the SPC benchmark in recognizing higher shifts of the mean (0.50, 1.00, 1.50 and 2.00 SD units) as well as of any other disturbance signals.

6.4. *Shewhart control chart with Western Electric run rules and sensitizing rules*

Table 4 compares Type I and II errors of the Shewhart control chart with seven run rules with those given by the ART neural network with vigilance parameter $\rho = 0.8610$. The run rules implemented in the SPC benchmark are the standard three tests described in Western Electric (1956) and four additional sensitizing rules proposed by Nelson (1984). Specifically, six points in a row steadily increasing or decreasing; 15 points in a row within the ± 1-*sigma* limits; 14 points in a row alternating up and down; and eight points in a row on both sides beyond the ± 1-*sigma* limits.

Since the use of four additional run rules increases the false alarm rate of the SPC chart, a higher vigilance parameter ($\rho = 0.8610$) has been adopted in this case.

From the results of table 4, it is still possible to draw similar conclusions to those presented for the comparison of the neural network with a control chart with a single run rule. More specifically, the proposed neural network has better performances (lower Type II error rates) than those of the SPC chart when recognizing each of the disturbance signals, for each magnitude level considered in the test (0.25, 0.50, 1.00, 1.50 and 2.00).

6.5. *Discussion*

Simulation results indicate that the proposed neural network achieves comparable performances in signalling a constant shift of the process mean with those of a CUSUM control chart. At the same time, the neural network appears able to extend the recognition ability to a wide set of potential unnatural changes of the process structure that cannot be addressed by a CUSUM chart. Moreover, for other types of change such as systematic, cycle or mixtures, the neural network outperforms

	Shewhart WE + SR RRs	Fuzzy ART $M=75$ $\rho=0.8610$	Comparison neural network versus 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%	1.706%	−0.095%	0.035%	0.165%
Systematic variation	$\hat{\beta}_{cc}$	$\hat{\beta}_{nn}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_-$	$\hat{\beta}_{nn}-\hat{\beta}_{cc}$	$(\hat{\beta}_{nn}-\hat{\beta}_{cc})_+$
0.25	97.985%	97.029%	−1.163%	−0.956%	−0.749%
0.50	96.658%	84.911%	−12.467%	−11.747%	−11.027%
1.00	87.888%	0.016%	−88.102%	−87.872%	−87.642%
1.50	65.761%	0.000%	−66.193%	−65.761%	−65.329%
2.00	32.188%	0.000%	−32.652%	−32.188%	−31.724%
Cyclic $\Omega=4$					
0.25	98.361%	97.724%	−0.796%	−0.637%	−0.478%
0.50	98.394%	95.082%	−3.613%	−3.312%	−3.011%
1.00	97.624%	29.719%	−69.334%	−67.905%	−66.476%
1.50	95.079%	0.026%	−95.168%	−95.053%	−94.938%
2.00	89.784%	0.000%	−89.970%	−89.784%	−89.598%
Shift $\tau=1$					
0.25	97.660%	96.984%	−0.884%	−0.676%	−0.468%
0.50	95.239%	84.795%	−11.158%	−10.444%	−9.730%
1.00	80.255%	0.050%	−80.541%	−80.205%	−79.869%
1.50	42.546%	0.000%	−43.022%	−42.546%	−42.070%
2.00	8.265%	0.000%	−8.521%	−8.265%	−8.009%
Mixture $p_c=0.4$					
0.25	98.121%	97.036%	−1.260%	−1.085%	−0.910%
0.50	97.300%	85.197%	−12.715%	−12.103%	−11.491%
1.00	92.843%	0.022%	−92.984%	−92.821%	−92.658%
1.50	79.948%	0.000%	−80.247%	−79.948%	−79.649%
2.00	53.682%	0.000%	−54.136%	−53.682%	−53.228%

Table 4. Comparison between a neural network and a Shewhart control chart with standard Western Electric (1956) and Nelson (1984) sensitising run rules (simulation results, 50 sets of 2000 data).

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.

Therefore, 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, designed to recognize different kinds of change in the process structure. This means that the proposed approach can be exploited as the sole tool for signalling a generic modification in the state of the process. 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.

7. Conclusions

This research is concerned with developing a new neural-based approach for quality control. The application of the ART for quality control has been discussed and analysed by means of Monte Carlo simulation.

The ART neural approach is mainly intended for signalling unnatural process behaviour by merely recognizing changes in the state of the process rather than by detecting specific unnatural patterns. The neural algorithm is quite simple to implement and the training set can be restricted to a few data vectors. It has been demonstrated that the training set can even be limited to a single vector whose components are equal to the process nominal value. Therefore, the first advantage of the ART-based network that makes it a practical tool for quality control is the reduced training time. One more advantage of this approach is that it requires no previous information about unnatural pattern appearances and related mathematical models. Moreover, a significant benefit of the neural approach is that it can model multiple control strategies simultaneously. 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.

Simulation has been used for performance measure. 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, cycles or mixtures) than the CUSUM test, while it presents a comparable ability in signalling constant shifts. At the same time, the neural network outperforms Shewhart control charts with a set of run rules and sensitizing rules in signalling changes such as systematic, cycle or mixtures.

The proposed method can improve the efficacy of quality control. However, since the ART-based approach can only signal generic unnatural process behaviours, it should be stressed that the proposed system cannot substitute existing 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. As an example, the proposed system can be realistically used in industrial applications when starting processing of new products, or with a new installed process.

Finally, there are two main possible directions for future research. First, the effect of departures from normality and independence for the reference manufacturing process model can be investigated. Second, the neural network system can be improved in order to recognize not just a generic pattern of unnatural data, but also one or more particular unnatural patterns.

Appendix A

An analytical method is provided for deciding on the vigilance parameter value. It is assumed that the quality measurements being monitored are normally distributed with a common variance and additive error structure: $Y_t = Z_t \pm \varphi$, where $Z_t \sim NID(0, 1)$ and φ is an arbitrary constant change of the process mean (e.g. a shift, systematic variation or a mixture pattern). As observed in Section 3, this model gives a close approximation to many types of practical manufacturing processes. In situations where these assumptions are violated, a power transformation technique (Sakia 1992) can be applied to reduce anomalies such as non-normality and the heteroscedasticity of the monitored quality measurements.

With these assumptions, it can be demonstrated (Pacella and Semeraro 2003) that given the proposed Fuzzy ART network of window size M, with coding limit l large enough (e.g. $l \geq 3$) and values of the vigilance parameter in the range:

$$1 - \frac{1}{2l}\left(\mu_1 - \sqrt{\frac{8}{3} \cdot \frac{1 + \varphi^2 - \mu_1^2}{M}}\right) < \rho < 1 - \frac{1}{2l}\left(\mu_0 + \sqrt{\frac{8}{3} \cdot \frac{1 - \mu_0^2}{M}}\right), \quad (11)$$

where the constants μ_0 and μ_1 are as in equation (12) (and $\Phi(\cdot)$ is the cumulative standard normal distribution function)

$$\mu_0 = \sqrt{\frac{2}{\pi}} \quad \mu_1 = \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\varphi^2}{2}\right) + \varphi[1 - 2\Phi(-\varphi)] \quad (12)$$

the expected Type I and II error rates of the proposed Fuzzy ART algorithm are not greater than the limits of equations (13) and (14) respectively:

$$\alpha \leq \frac{4}{9} \cdot \frac{1 - \mu_0^2}{M[2l(1 - \rho) - \mu_0]^2} \quad (13)$$

$$\beta \leq \frac{4}{9} \cdot \frac{1 + \varphi^2 - \mu_1^2}{M[\mu_1 - 2l(1 - \rho)]^2}. \quad (14)$$

Given a predefined upper limit for Type I error (say $\sup_\alpha$), equation (13) can be exploited to find an appropriate vigilance. Equation (15) shows the relationship between ρ and $\sup_\alpha$: given a specific neural network of window size M and coding limit l, the vigilance parameter of equation (15) can be used to obtain Type I error rates not greater than the predefined upper limit $\sup_\alpha$.

$$\rho = 1 - \frac{1}{2l}\left(\sqrt{\frac{2}{\pi}} + \frac{2}{3}\cdot\frac{1-2/\pi}{M\sqrt{\sup_\alpha}}\right). \tag{15}$$

To obtain a given α (typically $\alpha < 1.25\%$), a practical configuration approach is to use equation (15) by replacing $\sup_\alpha$ with $(20/3 \cdot \alpha \cdot \sqrt{M})$. As an example, referring to the ART neural network implemented in Section 5 ($M = 75$ and $l = 3$), to obtain a Type I error rate about equal to $\alpha \cong 0.27\%$ and $\alpha \cong 1.15\%$, the criterion of equation (15) provides the following vigilance parameters:

$$\left.\begin{array}{l} M = 75 \\ l = 3 \\ \alpha \cong 0.27\% \\ \sup_\alpha = \left(\frac{20}{3}\cdot\alpha\cdot\sqrt{M}\right) \end{array}\right\} \Rightarrow \rho = 0.8475 \qquad \left.\begin{array}{l} M = 75 \\ l = 3 \\ \alpha \cong 1.15\% \\ \sup_\alpha = \left(\frac{20}{3}\cdot\alpha\cdot\sqrt{M}\right) \end{array}\right\} \Rightarrow \rho = 0.8575$$

In the cases of unnatural behaviour (e.g. for changes in the process mean of magnitude $|\varphi| = 1, 1.5, 2$), an upper limit with the Type II error given by the neural network can be obtained by equation (14):

$$\left.\begin{array}{l} M = 75 \\ l = 3 \\ \rho = 0.8475 \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \beta(\varphi = 1.0) \le 5.980\% \\ \beta(\varphi = 1.5) \le 1.174\% \\ \beta(\varphi = 2.0) \le 0.455\% \end{array}\right. \qquad \left.\begin{array}{l} M = 75 \\ l = 3 \\ \rho = 0.8575 \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \beta(\varphi = 1.0) \le 3.899\% \\ \beta(\varphi = 1.5) \le 0.982\% \\ \beta(\varphi = 2.0 \le 0.409\% \end{array}\right.$$

These results can be compared with those obtained from simulation (tables 1–3). In addition, table 5 reports the results of the designing criterion for several neural network configurations with $M = 25, 40, 50, 75, 85$ and $l = 3, 3.5$. To validate the theoretical results, simulation results for the neural network are also included in table 5.

Appendix B

By combining equations (3) and (6), the neural network does not emit an alarm if the relationship $dis(\underline{\boldsymbol{I}}_t, \underline{\boldsymbol{w}}_\mu) \le M(1-\rho) \Leftrightarrow \sum_{r=1}^{M} \min\left(l, \left|Y_{t-M+r} - \mu\right|\right) \le 2lM(1-\rho)$ is verified. Therefore, if $\rho = 0$, the network does not release any alarm (Type I error rate $\alpha = 0\%$, and Type II error rate $\beta = 100\%$). On the other hand, when $\rho = 1$, the neural network constantly does release alarms (Type I error rate $\alpha = 100\%$, Type II error rate $\beta = 0\%$).

The effects of Type I and II errors when ρ is altered have been assessed by means of simulation. The analysis is accomplished at four levels of window size $M = 10, 25, 50, 75$. Table 6 shows Type I error estimator points ($\hat{\alpha}$), and intervals (with 95% coverage and indicated by the notation $[\hat{\alpha}_-, \hat{\alpha}_+]$), are both presented.

From the results in table 6, it appears that as the vigilance parameter increases, so does the Type I error. Furthermore, as the vigilance parameter approaches the upper limit $\rho = 1$, then the Type I error approaches the upper limit $\hat{\alpha} = 100\%$ at any window size (M). On the other hand, as the vigilance

Neural network design $\alpha = 0.27\%,\ \sup_\alpha = \left(\frac{20}{3}\cdot\alpha\cdot\sqrt{M}\right)$			Simulation results	
M	l	$\rho = 1 - \frac{1}{2l}\left(\sqrt{\frac{2}{\pi}} + \frac{2}{3}\cdot\frac{1-2/\pi}{M\sqrt{\sup_\alpha}}\right)$	α (%)	Confidence interval 95%
25	3.0	0.8218	0.271	[0.240%, 0.302%]
40	3.0	0.8352	0.252	[0.224%, 0.280%]
50	3.0	0.8401	0.275	[0.241%, 0.309%]
75	3.0	0.8475	0.262	[0.229%, 0.295%]
85	3.0	0.8490	0.242	[0.205%, 0.279%]
25	3.5	0.8473	0.281	[0.250%, 0.312%]
40	3.5	0.8588	0.263	[0.232%, 0.294%]
50	3.5	0.8630	0.288	[0.252%, 0.324%]
75	3.5	0.8690	0.260	[0.227%, 0.293%]
85	3.5	0.8705	0.257	[0.219%, 0.295%]

Table 5. Neural network configuration to obtain a theoretical Type I error of about 0.27%. Simulation results validate the approach.

	$M = 10$			$M = 25$		
ρ	$\hat{\alpha}_-$	$\hat{\alpha}$	$\hat{\alpha}_+$	$\hat{\alpha}_-$	$\hat{\alpha}$	$\hat{\alpha}_+$
0.800	0.72%	**0.77%**	0.82%	0.02%	**0.02%**	0.03%
0.825	2.59%	**2.70%**	2.81%	0.32%	**0.36%**	0.39%
0.850	8.16%	**8.38%**	8.59%	2.48%	**2.61%**	2.73%
0.875	23.97%	**24.39%**	24.81%	16.67%	**17.15%**	17.64%
0.900	61.01%	**61.62%**	62.23%	79.05%	**79.88%**	80.70%
0.925	93.75%	**94.11%**	94.47%	99.64%	**99.74%**	99.83%
0.950	99.83%	**99.86%**	99.89%	100.00%	**100.00%**	100.00%
0.975	100.00%	**100.00%**	100.00%	100.00%	**100.00%**	100.00%
	$M = 50$			$M = 75$		
0.800	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
0.825	0.02%	**0.03%**	0.03%	0.00%	**0.00%**	0.01%
0.850	0.78%	**0.84%**	0.89%	0.34%	**0.38%**	0.43%
0.875	12.91%	**13.48%**	14.05%	11.75%	**12.31%**	12.88%
0.900	94.22%	**94.78%**	95.33%	98.83%	**99.05%**	99.26%
0.925	99.99%	**100.00%**	100.00%	100.00%	**100.00%**	100.00%
0.950	100.00%	**100.00%**	100.00%	100.00%	**100.00%**	100.00%
0.975	100.00%	**100.00%**	100.00%	100.00%	**100.00%**	100.00%

Table 6. Neural network Type I error point and interval estimators (simulation results, 50 sets of 2000 data).

parameter decreases, as does the Type I error, and it approaches the lower limit $\hat{\alpha} = 0\%$ at any window size (M). However, the growing rate presented by the Type I error as a function of the vigilance parameter increases as the window size increases. Figure 3(a) shows a chart of point estimator $\hat{\alpha}$ as a function of vigilance parameter ρ. The chart area enclosed in the rectangle (i.e. for ρ between 0.825 and 0.875, $\hat{\alpha}$ between 0 and 20%) is shown in detail in figure 3(b), in which the interval estimators of Type I error rate have been also included.

The point and interval estimators of Type II errors are presented in table 7. The Type II errors are estimated on data produced by a process with a shifted mean. The magnitude of the shift in equation (9) has been fixed to 1 unit SD ($\varphi = 1$), and the starting point has been fixed to the first observation ($\tau = 1$). The notation $[\hat{\beta}_-, \hat{\beta}_+]$ has been exploited to indicate the 95% confidence interval of the Type II error point estimator $\hat{\beta}$.

From the results presented in table 7, it can be deduced that as the vigilance parameter increases, the Type II error decreases. In particular, the Type II error approaches the lower limit $\hat{\beta} = 0\%$ as the vigilance parameter approaches the upper limit $\rho = 1$. However, the decreasing trend is smaller for low window size and it is higher for a high window size.

Figure 4(a) shows a chart of point estimators $\hat{\beta}$ as a function of vigilance. The chart area enclosed in the rectangle (ρ ranging between 0.825 and 0.875, $\hat{\beta}$ ranging between 0 and 20%) is shown in detail in figure 4(b), in which the interval estimators of Type II error rate have been included.

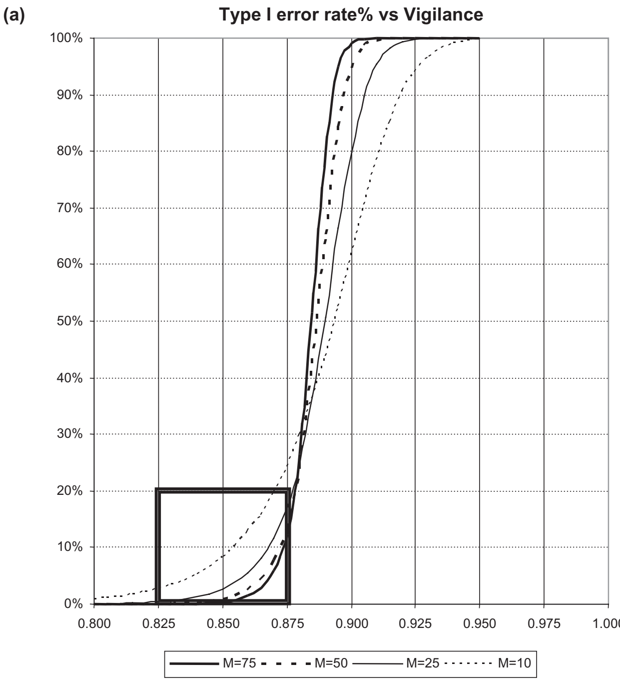

Figure 3. Neural network Type I point estimators (ordinate) versus vigilance parameter (abscissa) at four window sizes (simulation results, 50 sets of 2000 data): (a) abscissa range [0.800, 1.000], ordinate range [0%, 100%]; (b) abscissa range [0.825, 0.875], ordinate range [0%, 20%] and interval estimators (coverage 95%).

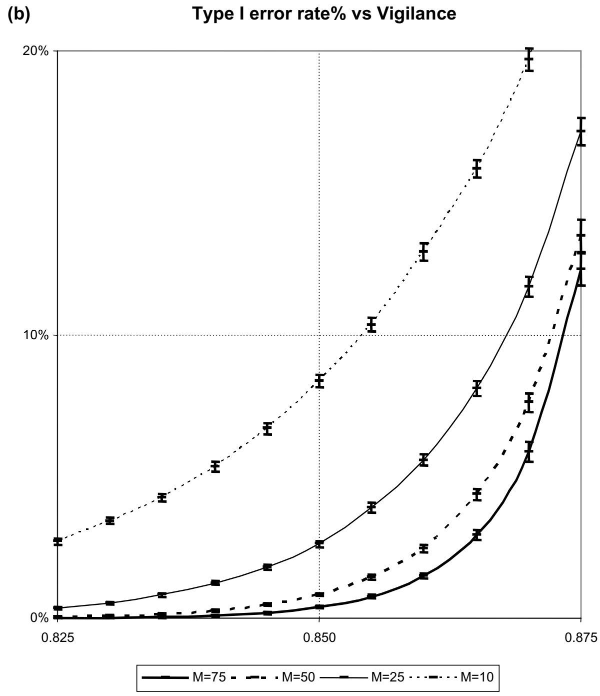

Figure 3. Continued.

ρ	$M=10$ $\hat{\beta}_-$	$\hat{\beta}$	$\hat{\beta}_+$	$M=25$ $\hat{\beta}_-$	$\hat{\beta}$	$\hat{\beta}_+$
0.800	85.18%	**85.49%**	85.80%	93.28%	**93.53%**	93.77%
0.825	67.35%	**67.78%**	68.21%	71.26%	**71.94%**	72.61%
0.850	37.16%	**37.82%**	38.48%	20.68%	**21.36%**	22.04%
0.875	10.84%	**11.21%**	11.58%	0.99%	**1.18%**	1.36%
0.900	1.37%	**1.53%**	1.68%	0.00%	**0.02%**	0.04%
0.925	0.06%	**0.09%**	0.12%	0.00%	**0.00%**	0.00%
0.950	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
0.975	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
	$M=50$			$M=75$		
0.800	96.70%	**96.86%**	97.02%	97.92%	**98.06%**	98.19%
0.825	70.86%	**71.68%**	72.49%	67.97%	**69.04%**	70.11%
0.850	4.96%	**5.51%**	6.05%	0.83%	**1.10%**	1.37%
0.875	0.00%	**0.03%**	0.06%	0.00%	**0.00%**	0.00%
0.900	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
0.925	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
0.950	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%
0.975	0.00%	**0.00%**	0.00%	0.00%	**0.00%**	0.00%

Table 7. Neural network Type II error point and interval estimators (shift of 1 SD unit simulation results, 50 sets of 2000 data).

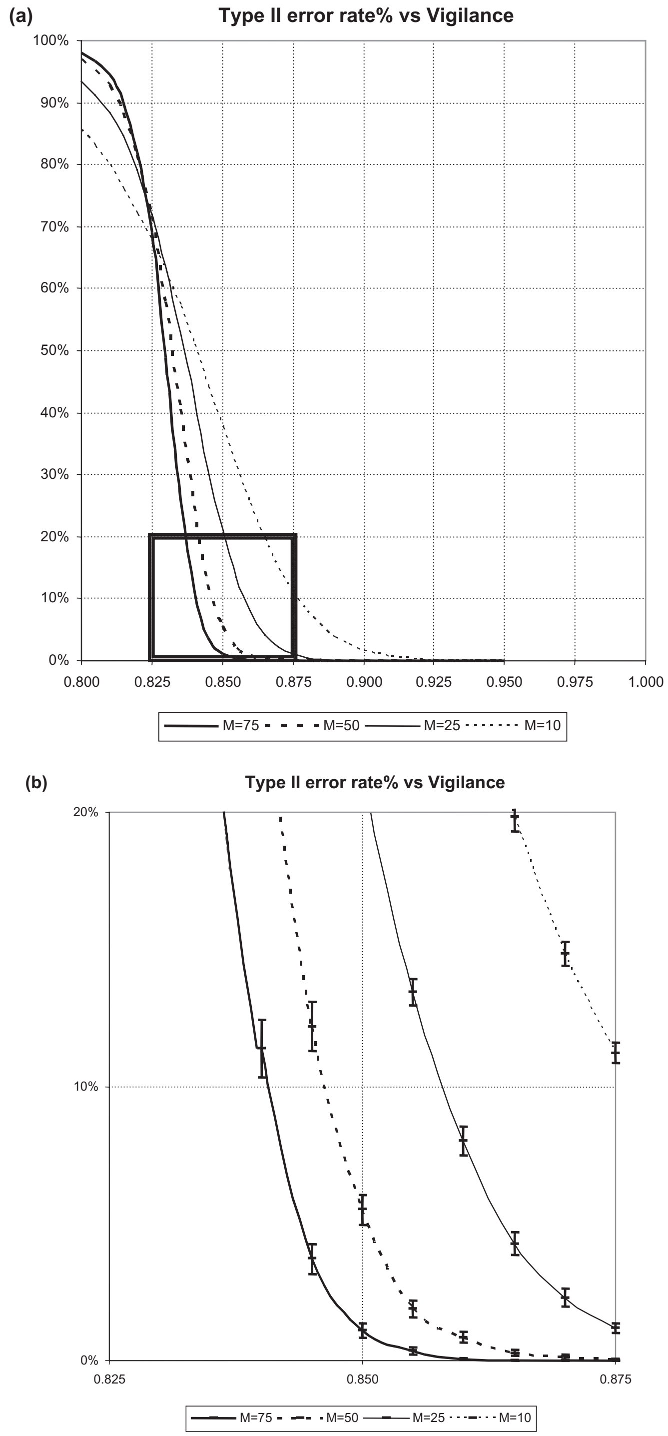

Figure 4. Neural network Type II point estimators (ordinate) for a shift of 1 unit standard deviation versus vigilance parameter (abscissa) at four window sizes (simulation results, 50 sets of 2000 data): (a) abscissa range [0.800, 1.000], ordinate range [0%, 100%]; (b) abscissa range [0.825, 0.875], ordinate range [0%,20%] and interval estimators (coverage 95%).

Acknowledgement

This work was partially funded by the Ministry of Education, University and Research of Italy (MIUR). The authors thank the reviewers who contributed to the quality of the paper.

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. and Georgiopoulos, M., 2002, Category regions as new geometrical concepts in Fuzzy-ART and Fuzzy-ARTMAP. *Neural Networks*, **15**, 1205–1221.

Champ, C. W. and Woodall, W. H., 1987, Exact results for Shewhart control charts with supplementary run rules. *Technometrics*, **29**, 393–399.

Cheng, C.-S., 1997, A neural network approach for the analysis of control chart patterns. *International Journal of Production Research*, **35**, 667–697.

Cheng, C.-S. and Cheng, S.-S., 2001, A neural network based procedure for the monitoring of exponential mean. *Computers and Industrial Engineering*, **28**, 51–61.

Georgiopoulos, M., Dagher, I., Heileman, G. L. and Bebis, G., 1999, Properties of learning of a Fuzzy ART variant. *Neural Networks*, **12**, 837–850.

Georgiopoulos, M., Fernlund, H., Bebis, G. and Heileman, G. L., 1996, Order of search in Fuzzy ART and Fuzzy ARTMAP: effect of the choice parameter. *Neural Networks*, **9**, 1541–1559.

Guh, R. S. and 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. and Tannock, J. D. T., 1999, Recognition of control chart concurrent patterns using a neural network approach. *International Journal of Production Research*, **37**, 1743–1765.

Guh, R. S., Zorriassatine, F., Tannock, J. D. T. and O'Brien C., 1999, On-line control chart pattern detection and discrimination—a neural network approach. *Artificial Intelligence in Engineering*, **13**, 413–425.

Hagan, M. T., Demuth, H. B. and Beale, M., 1996, *Neural Network Design* (Boston, MA, USA: PWS Publishing Co.).

Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, 2nd edn (Englewood Cliffs: Prentice-Hall).

Huang, J., Georgiopoulos, M. and Heileman, G. L., 1995, Fuzzy ART proprieties. *Neural Networks*, **8**, 203–213.

Hwarng, H. B., 2004, Detecting process mean shift in the presence of autocorrelation: a neural-network based monitoring scheme. *International Journal of Production Research*, **42**, 573–595.

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

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

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

Klein, M., 2000, Two alternatives to the Shewhart X control chart. *Journal of Quality Technology*, **32**, 427–431.

Kleinjnen, J. P. C. and Van Groenendaal, W., 1992, *Simulation. A Statistical Perspective* (Chichester: Wiley).

MathWorks, 1991, *MATLAB User's Guide* (South Natick: MathWorks, Inc.).

Montgomery, D. C., 2000, *Introduction to Statistical Quality Control*, 4th edn (Chichester: Wiley).

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

NeuralWare, 1997, *NeuralWorks Professional II Plus Reference Guide* (Pittsburgh: NeuralWare, Inc.).

Pacella, M. and Semeraro, Q., 2003, A statistical survey of a neural network based procedure for manufacturing process quality monitoring (submitted and under review).

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

Sakia, R. M., 1992, The Box–Cox transformation technique: a review. *Statistician*, **41**, 168–179.

Vattulainen, I., Kankaala, K., Saarinen, J. and Ala-Nissila, T., 1995, A comparative study of some pseudorandom number generators. *Computer Physics Communications*, **86**, 209–226.

Wang, T.-Y. and Chen, L.-H., 2002, Mean shifts detections and classifications in multivariate processes: a neural-fuzzy approach. *Journal of Intelligent Manufacturing*, **13**, 211–221.

Western Electric, 1956, *Statistical Quality Control Handbook* (Indianapolis: Western Electric).

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