INT. J. PROD. RES., 1995, VOL. 33, NO. 7, 1817–1833

Detecting process non-randomness through a fast and cumulative learning ART-based pattern recognizer

H. B. HWARNG† and C. W. CHONG

An adaptive resonance theory (ART) based, general-purpose control chart pattern recognizer (CCPR) which is capable of fast and cumulative learning is presented. The implementation of this ART-based CCPR was made possible by introducing two key alternatives, that is, incorporating a synthesis layer in addition to the existing two-layer architecture and adopting a quasi-supervised training strategy. A detailed algorithm with the training and the testing modes was presented. Extensive simulations and performance evaluations were conducted and proved that this ART-based CCPR indeed possesses the capability of fast and cumulative learning. When compared with a back-propagation pattern recognizer (BPPR), the ART-based CCPR is superior on cyclic patterns, inferior on mixture patterns, and comparable on other patterns. Furthermore, an ART-based CCPR is easier to develop since it needs fewer training templates and takes less time to learn. This study not only provides a basis for understanding the capabilities of ART-based neural networks on control chart pattern recognition but re-confirms the applicability of the neural network approach.

1. Introduction

The need for identifying patterns of data on statistical quality control charts was realized early in the 1950's (Western Electric 1956). Control chart patterns can generally be classified into two categories, namely, random and non-random patterns. When a control chart exhibits a random pattern it usually indicates that the process is statistically *in-control.* On the other hand, when a control chart exhibits a non-random pattern it indicates that the process is statistically *out-of-control.* From this perspective, an out-of-control situation cannot be characterized merely by an observation falling outside the control limits. All of the observations can fall well within the control limits while still behaving non-randomly. This implies that the traditional way of using control charts, based only on the control limits, is insufficient for detecting certain types of non-randomness. Alternatively an effective and efficient system capable of detecting process non-randomness through identification of non-random patterns on control charts must be developed.

It was based on this understanding, under Western Electric's statistical quality control program, that engineers and shop floor operators were trained with additional rules to identify the existence of non-randomness in the process and then to determine its causes. The procedure is as follows:

(1) The type of pattern that a control chart represents is determined from the inspection of the control chart;
(2) This pattern is compared with a set of pre-drawn patterns;
(3) The pattern is studied and related to what is known about the process;

Revision received July 1994.
Department of Decision Sciences, National University of Singapore, Singapore 0511.
† To whom correspondence should be addressed.

0020-7543/95 $10·00 © 1995 Taylor & Francis Ltd.

(4) Based on the suggested potential causes for the pattern, an attempt is made to recall similar causes which might be operating in the process.

In Western Electric's *Statistical Quality Control Handbook* a set of run rules was devised to identify some patterns of instability. Additional verbal descriptions were also given to assist in identifying other more complex non-random patterns. These rules or variation of these rules for step 1 and 2 of the above procedure are still quite popular in today's literature and practices. It would be ideal if the above procedure could be done automatically without human intervention. Nevertheless, the implementation of the procedure is not as straightforward as we would expect. The major difficulties lie in how the pattern can be identified automatically and how the causes can be deduced from the identified pattern without the help of human interpretation. Besides, the major problems associated with these rules are their inflexibility to tolerate any deviation from the rules and their inability to explicitly identify which type of pattern.

Traditional run rules are useful in signalling if there is any structural change in the mean and/or variance of the process; the pattern recognition approach, which explicitly identifies non-random patterns, is more useful in identifying what particular type of non-randomness is occurring in the process. Additional information about non-randomness in the process then can be deduced through the identified non-random pattern. This approach is not only more effective in determining corrective actions but more flexible in a dynamic manufacturing environment. Recently the utility of neural networks in identifying process non-randomness as exhibited on statistical quality control charts has been demonstrated by a number of researchers (Lim and Ooi 1990, Lim *et al.* 1991, Guo and Dooley 1992, Hwarng and Hubele 1991, 1992, 1993a, 1993b).

In almost all of the published works, the major architecture used has been back-propagation neural networks. Although the back-propagation algorithm has been widely used and well studied, two well known problems, namely, slowness in training and inability to perform adaptive learning without re-learning, still pose some inconvenience for practical applications. Therefore, the objective of this research is to investigate and develop a general-purpose control chart pattern recognizer (CCPR) which is capable of fast and cumulative learning. It is general-purpose because it is capable of identifying multiple patterns.

In this research, we adopted adaptive resonance theory (ART) (Carpenter and Grossberg 1987, 1988) as the foundation for this CCPR. ART is adopted for two distinctive features, that is, its ability to learn fast and its ability to retain previously learned pattern classes, while adaptively learning new ones. Although there are a number of varied ART architectures, we will confine our discussions to the binary version of ART, ART1.

The paper is organized in the following manner. § 2 gives an overview of ART concerning the architecture and general operation; § 3 explains some limitations of ART when applied to control chart pattern recognition; § 4 presents and describes the proposed ART-based CCPR; § 5 discusses the performance of the ART-based CCPR. Finally, § 6 concludes the paper with a summary and some additional comments.

2. Overview of ART: architecture and operation

The design principles of adaptive resonance neural networks originated from some basic idea in Grossberg (1967). Since then a number of modifications and variations have been developed to satisfy the needs of various situations. In this paper, we will deal with control chart data which are preprocessed into a binary format. Hence, we will constrain ourselves to the simplest architecture, ART 1 of the ART family, which is designed for binary input data.

It is understood that an ART network acts best as a pattern classifier. After proper learning, an ART network accepts an input pattern and classifies it according to which of the stored categories it resembles most. If the input pattern does not match any of the stored categories, the input pattern will be stored as a new category. If the mismatch between the input pattern and one of the stored categories is within a specified tolerance, the input pattern will be classified into the same category as the stored category and the critical features of the input pattern can be incorporated into the matching stored category. In other words, the previously stored category can be adaptively modified (through adaptive weights) to include the features of both the previous and the current input patterns. However, such modification of weights is not allowed if the mismatch is not within the specified tolerance.

2.1. *Architecture*

An ART network consists of two layers, namely, the comparison layer (F1) and the recognition layer (F2). Each layer comprises a number of nodes. Two layers are interconnected by two sets of connection weights, i.e., bottom-up and top-down weights. In the most simplified terms, the comparison layer acts as a feature detector which receives external input patterns and the recognition layer acts as a category classifier which receives internal inputs from the comparison layer. Gain control devices, G1 and G2, and the reset mechanism provide the control functions needed for learning and classification. Figure 1 depicts the basic components of the ART architecture.

The patterns of activity that develop over the nodes in the two layers are called shorts-term memory (STM) because they exist only in association with a single application of an input pattern. The activities are the immediate response of the system to external and internal stimuli. The bottom-up and top-down weights between F1 and F2 can be updated adaptively in response to the input patterns. This is when learning takes place. These adaptive weights are called long-term memory (LTM) because they can remain as a part of network for an extended period. For ease of reference, the ART

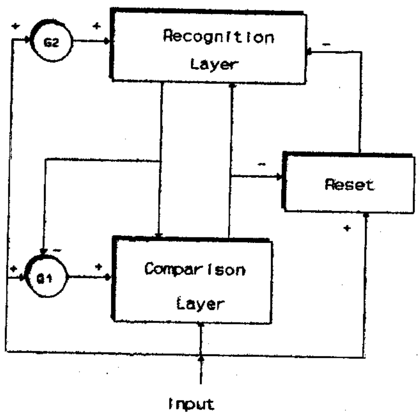

Figure 1. Basic components of the ART architecture.

architecture is usually described as two subsystems, namely, the attentional subsystem and the orienting subsystem. The attentional subsystem is composed of F1, F2, G1, G2, and the connection weights. The orienting subsystem is the reset mechanism which is controlled by a parameter known as the *vigilance parameter.*

2.2. *Operation*

When an input pattern is presented to an ART network, the data are presented to three places: F1, G1, and the reset mechanism as shown in Fig. 1. In addition to the external input, F1 receives two other sources of input, i.e., from G1 and from F2. The proper application of these three inputs at F1 is possible only if the activities at F1 follow the ⅔ rule. The ⅔ rule states that at least two out of the three sources of input must be active to supraliminally activate a node in F1.

The output of F1 is then sent to F2 and the reset mechanism. The reset mechanism not only receives inhibitory input from F1 but receives an excitatory input from the external input. The reset mechanism initiates and transmits a reset signal to F2 whenever the excitatory input overpowers the inhibitory input. This reset signal, if initiated, will stop any activities that occur in F2 and hence prepare F2 for the next incoming pattern. If the reset signal is not initiated, the previously coded pattern associated with the category node which represents the 'best match' will be transmitted back to F1 as a top-down template. An inhibitory signal will also be sent to G1 so that G1 will remain inactive. Note that two gain control devices are incorporated so that F1 can distinguish between bottom-up and top-down signals.

F1 deals only with the external input and the top-down template. Its activities are governed by the orienting subsystem. If there is not a sufficient match between the top-down template and the input pattern, the activity in F1 will decrease. A decrease in F1 activities will cause the inhibitory input to the reset mechanism to be overpowered by the excitatory input. The orienting subsystem will then generate a reset signal. On the other hand, if there is a sufficient match, a state of resonance is said to be achieved. The input pattern is then classified under the matching category. The decision concerning whether the match is sufficient or not is regulated by the vigilance parameter.

The reset signal generated by the orienting subsystem not only stops the activities in F2 but removes the inhibition on G1 such that G1 acts as a second source of stimulation to F1. This allows the input pattern to be reproduced in F1 and a re-matching process to be started. During the re-matching process, however, the category nodes in F2 which have failed to match the pattern previously are prevented from being matched again. The entire process is repeated until a matching category is found, a new category is created, or all the nodes in F2 are exhausted which means that the input pattern is not sufficiently similar to any existing categories and there is no capacity in F2 for any new category.

3. Limitations of ART in the implementation of control chart pattern recognition

While the ART architecture provides solutions for the two major concerns of back-propagation networks as will be shown later, the use of ART does not guarantee a success in control chart pattern recognition. In this section we will examine and illustrate how some of the limitations would hinder the implementation of an ART-based CCPR.

3.1. *Recoding instability*

The first problem comes from ART's recoding instability. The problem of recoding instability was not entirely solved in the original design of ART (Ryan and Winter 1987). For example, once a new category was learned in F2, the top-down weights can be gradually recoded in subsequent presentations of slightly varied new patterns. The recoding can be so substantial that eventually the network does not recognize the original pattern or patterns which are very similar to the original pattern. Figure 2 illustrates how this problem occurs in control chart pattern recognition. In the figure each control chart pattern is represented by 1's in a $c(=7)\times r(=8)$ grid from top to bottom. Data are coded into a binary format as follows. Each observation is represented by a row of c binary digits. the sequence of c binary digits corresponds to the c zones which equally divide the typical standardized range of a control chart, e.g., $[-3, +3]$. The location of the 1 within the sequence corresponds to the zone in which the standardized observation lies.

The network has been trained and encoded with two top-down templates which are upward sudden-shifts and cycles. Four inputs of noisy upward sudden-shift are presented to the network. On the fourth input, the network fails to classify it as category 1 because the top-down template has been recoded by the previous noisy upward sudden-shifts. The input pattern fails the vigilance test. As a result, it is classified into a new category.

3.2. *Inability to classify translated patterns*

The second problem arises from ART's inability to classify shifted or rotated patterns into the same category as the un-shifted or un-rotated patterns. Figure 3 shows two cyclic patterns. One of the two was shifted downward by one row. The only vigilance value which could classify both of them into the same category is zero. A possible solution to this problem is to further preprocess shifted or rotated data so as to have invariant forms. However, this will complicate data preprocessing.

3.3 *Learned categories tend to outgrow the capacity*

The third problem results from the unsupervised training environment. Under unsupervised training, when an input pattern fails to match the learned top-down templates during the vigilance test, the network always learns a new category provided sufficient capacity in F2. This might not be desirable in control chart pattern recognition, because, in the actual application, we are interested in only a number of non-random patterns. When random patterns, which are anything other than the pre-specified non-random patterns, are presented to the network, they should not be classified as new non-random patterns. Nevertheless, under unsupervised training it is very likely that many random patterns would be classified as new categories in F2, even if a low vigilance value is used. Consequently, the network might grow very large in F2. A supervised version of ART network, called ARTMAP (Carpenter *et al.* 1991), might solve this problem. However, an ARTMAP network uses a much more complicated architecture and requires more computer resource.

With the above three noted problems unsolved, the use of ART-based neural networks cannot provide satisfactory results for control chart pattern recognition. In the following, we will present a novel approach to solving these problems. The method is rather simple, yet effective.

TOP DOWN TEMPLATES vigilance=0.7

	Input pattern	1	2	3
1	0001000 0001000 0001000 0001000 1000000 1000000 1000000 1000000	0001000 0001000 0001000 0001000 1000000 1000000 1000000 1000000	0001000 0010000 0100000 0010000 0001000 0000100 0000010 0000100	
2	0001000 0001000 0000010 0001000 1000000 1000000 1000000 1000000	0001000 0001000 0000000 0001000 1000000 1000000 1000000 1000000	0001000 0010000 0100000 0010000 0001000 0000100 0000010 0000100	
3	0001000 0001000 0001000 0000100 1000000 1000000 1000000 1000000	0001000 0001000 0000000 0000000 1000000 1000000 1000000 1000000	0001000 0010000 0100000 0010000 0001000 0000100 0000010 0000100	
4	0001000 0000100 0001000 0001000 1000000 1000000 1000000 1000000	0001000 0001000 0000000 0000000 1000000 1000000 1000000 1000000	0001000 0010000 0100000 0010000 0001000 0000100 0000010 0000100	0001000 0000100 0001000 0001000 1000000 1000000 1000000 1000000

Figure 2. An illustration of recoding instability.

cycle	shifted cycle
0 0 0 1 0 0 0	0 0 0 0 1 0 0
0 0 1 0 0 0 0	0 0 0 1 0 0 0
0 1 0 0 0 0 0	0 0 1 0 0 0 0
0 0 1 0 0 0 0	0 1 0 0 0 0 0
0 0 0 1 0 0 0	0 0 1 0 0 0 0
0 0 0 0 1 0 0	0 0 0 1 0 0 0
0 0 0 0 0 1 0	0 0 0 0 1 0 0
0 0 0 0 1 0 0	0 0 0 0 0 1 0

Figure 3. Identical patterns but shifted in location.

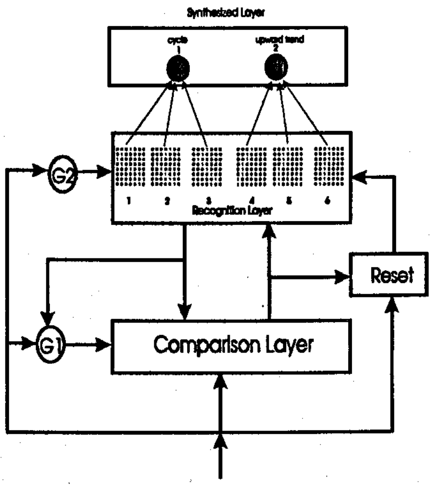

Figure 4. The ART architecture with a synthesis layer.

4. Proposed ART-based control chart pattern recognizer

4.1. *A synthesis layer*

Overcoming these problems would involve using some alternative architecture and training strategy. For the problem of recoding instability and that of inability to classify shifted or rotated patterns, adding one additional layer on top of F2 alleviates the dilemma. This layer is called the synthesis layer. Figure 4 shows the new ART architecture. The synthesis layer is configured in such a way that one node is used to connect in F2 all similar learned categories, which are either shifted or slightly varied, into one class. In other words, for any input patterns, which are shifted or slightly varied, the network will be able to classify them into a same class even though they might belong to different categories in F2. Of course, this can only be done with some a priori knowledge about the data. As an illustration in Fig. 4, the network has learned the shifted cycles and upward trends. They are represented as distinguished categories in F2. However, the shifted versions of patterns are connected to one shared node in the synthesis layer. That is, three cyclic pattern categories are connected to node 1 and three upward trend categories are connected to node 2 in the synthesis layer. The salient feature of this approach is that it simplifies input considerations and relies on synthesizing nodes to cluster shifted or rotated patterns.

4.2. *Quasi-supervised training*

To resolve the problem of learning too many new categories at F2, some quasi-supervised training strategy based on prior knowledge about the process data should be employed. The strategy is as follows:

(1) To generate or collect a representative set of non-random patterns which covers the whole spectrum of patterns of interest and to group them according to pattern classes;
(2) To configure the size of F2 to be the number of non-random patterns in the training set;
(3) To present each pattern to the network once according to the pre-arranged sequence during training and to adjust connection weights accordingly;
(4) Upon completion of training the number of learned categories in F2 should be exactly the number of non-random patterns;
(5) In the synthesis layer one node is used to connect all similar categories in F2 into one class, called a pattern class. An additional node is created as 'other' class for all patterns that are not classified under any one of the learned categories. Thus, the size of the synthesis layer should be configured to be one plus the number of pattern classes of interest.
(6) During testing (recalling) weights are *not* modified and any patterns that are not classified under any one of the learned categories will be placed under 'other' class in the synthesis layer, i.e., random patterns.

It should be noted that although the extent of the difficulties posed by the first two problems mentioned earlier can be manipulated to a certain degree by the vigilance parameter, it is necessary to resort to alternatives such as the ones suggested here in order to eliminate the problems. With the additional synthesis layer which deals with looser boundary conditions tackling the second problem, the original recognition layer can remain at a more vigilant level, such that the first problem of recoding instability can be alleviated.

4.3. *Algorithm*

The ART neural network model is based on a set of ordinary non-linear differential equations (Carpenter and Grossberg 1987). There are two modes of operation, i.e., slow learning and fast learning. Slow learning operates according to the original result, which was derived from differential equations and is rather complicated. Fast learning assumes that the learned weights in the system will reach asymptotic stability before a new input pattern is presented and the result can be reduced into a set of simpler formulae. Therefore, it is faster. The network can be trained either with quasi-supervision or without supervision. In the following, with reference to the steps given in Lippman (1987), a quasi-supervised algorithm for the control chart pattern recognizer is presented. Note that F1 refers to the comparison layer, F2 refers to the recognition layer, and F3 refers to the synthesis layer.

INPUT: (*In training mode*)
A selective set of training data, matrix A, which is organized according to the predetermined categories, and a prespecified value for the vigilance parameter ρ. Matrix A has P rows with each representing a training pattern.
(*In testing mode*)
A window of process data $[x_1, x_2, \dots, x_N]$ and a prespecified value for the vigilance parameter ρ.

OUTPUT: (*In training mode*)
A trained ART-based CCPR which has adaptively learned all predetermined categories.
(*In testing mode*)
One of the pre-learned classes in F3 which indicates that the process exhibits a certain non-random pattern, or 'other' class which indicates that the process does not exhibit any of the pre-learned non-random patterns.

Step 1. Initialize top-down and bottom-up weights.
Top-down weights z_{ji} are initialized according to

$$z_{ji}(0) > \frac{\upsilon - 1}{\phi} \tag{1}$$

Bottom-up weights z_{ij} are initialized according to

$$0 < z_{ij}(0) < \frac{\tau}{\tau - 1 + N} \tag{2}$$

where $1 \leqslant i \leqslant N$ and N is the number of nodes in F1; $1 \leqslant j \leqslant M$ and M is the number of nodes in F2; υ, ϕ, and τ are constants satisfying the following conditions: $\phi \geqslant 0$, $\tau \geqslant 1$, and $\max\{\phi, 1\} < \upsilon < \phi + 1$.

Step 2. Apply the input pattern to F1.
A window of input pattern $[x_1, x_2, \dots, x_N]$ is applied to F1. In general, x_i is applied to the i^{th} node of F1.

Step 3. Compute the weighted output activation at F2.

$$\eta_j = \sum_{i=1}^{N} z_{ij} x_i \tag{3}$$

where η_j is the weighted output activation at node j of F2; z_{ij} is the bottom-up weight from node i of F1 to node j of F2, and $1 \leqslant j \leqslant M$.

Step 4. Select the winner at F2.
Let $\eta_{j*} = \max\{\eta_j\}, 1 \leqslant 1 \leqslant M$. Node j* with the largest value η_j in F2 is now selected as the winner.

Step 5. Evaluate the similarity (or match).
The similarity measure, s, between the top-down template of the winning node and the input pattern is computed as follows:

$$s = \frac{\sum_{i=1}^{N} z_{j*i} x_i}{\|X\|} \tag{4}$$

where z_{j*i} is the top-down weights of the winning node and $\|X\|$ is the norm of the input pattern.

5.1. If $s \geqslant \rho$, then

(*In training mode*)

go to *Step* 7.

(*In testing mode*)

the pattern has been classified under node j* of F2. It is then synthesized under class c* in F3. Go to *Step* 2 for the next new pattern.

5.2. If $s < \rho$, then

Go to *Step* 6.

Step 6. Re-matching process (when $s < \rho$).

6.1. If all the patterns stored in F2 as top-down templates have not been compared to the input pattern,

shut off the winning node and go to *Step* 3. This shut-off node will no longer take part in *Step* 4.

6.2. If all the patterns stored in F2 as top-down templates have been compared to the input pattern,
(*In training mode*)

if there are still unoccupied nodes in F2 the input pattern will be learned as a new category. Go to *Step* 7.

(*In testing mode*)

the input is unclassified in F2 and is placed under the class called 'other' in F3. Go to *Step* 2 for the next new pattern.

Step 7. Adapt new weights (*for training mode only*).
Update top-down weights z_{ji} according to

$$z_{j*i}(t+1) = z_{j*i}(t) x_i \quad i = 1 \ldots N \tag{5}$$

and bottom-up weights z_{ij} according to

$$z_{ij*}(t+1) = \frac{z_{j*i}(t) x_i}{\tau - 1 + \sum_{i=1}^{N} z_{j*i}(t) x_i} \; i = 1 \ldots N \tag{6}$$

Go to *Step* 2 for the next training pattern.

5. Simulation and evaluation

In order to test the ART-based CCPR's ability to identify multiple patterns, eight pattern classes, namely, upward trends, downward trends, cycles, systematic variables, stratification, mixtures, upward sudden-shifts, and downward sudden-shifts were incorporated in the training. The definitions of these patterns and their associated potential assignable causes can be found in the *Statistical Quality Control Handbook* (Western Electric 1956). All the data used in the following simulation and discussion were generated by the mathematical formulae defined in Hwarng and Hubele (1993a). To test the ART-based CCPR's ability to recognize a legitimate pattern with the least number of observations, eight observations were considered as a classifying window. Every classifying attempt is based on a window of eight observations. Using fewer observations as a classifying window would not provide sufficient data to identify a pattern with confidence or would simply result in frequent false alarms.

5.1. *One-pass training*

According to the definitions of the eight non-random patterns, a training data set, as shown in Fig. 5, consisting of 28 templates, was produced. Templates 1 to 3 belong to upward trends; templates 4 to 6 belong to downward trends; templates 7 to 9 belong to cycles; templates 10 to 12 belong to systematic variables; template 13 belongs to stratifications; templates 14 to 16 belong to mixtures; templates 17 to 25 belong

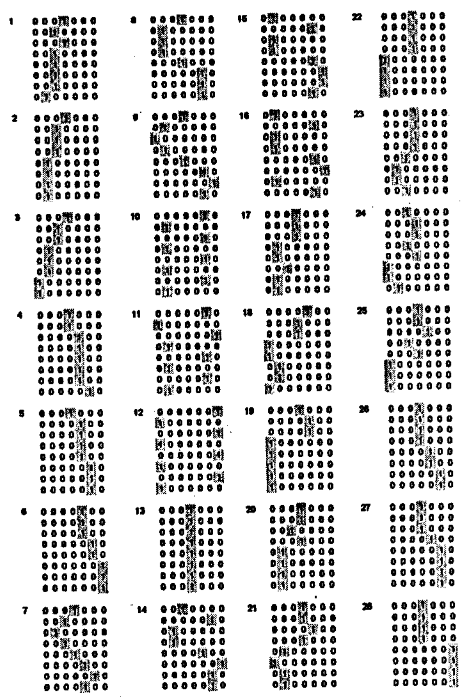

Figure 5. A training data set consists of 28 templates from eight pattern classes.

to upward sudden-shifts; and templates 26 to 28 belong to downward sudden-shifts. Two major principles of choosing the templates are coverage-broad and conflict-free. Coverage-broad means that sufficient templates should be included to cover the whole spectrum of patterns *within* each pattern class. Conflict-free requires that templates with conflicting features *between* pattern classes should be avoided. With this training set, an ART-based CCPR was configured to have 56 nodes in F1, 28 nodes in F2, and 9 nodes in F3. Nodes in F3 represent eight non-random pattern classes and one class termed 'other'. The training was carried out by using the training mode of the quasi-supervised algorithm presented in §4.3. The training time for this ART-based CCPR is rather brief because the training simply goes through a single pass of the training data set. Each training template eventually occupied one node in F2 ($\rho = 0{\cdot}7$). However, each class of templates were synthesized into one single class in F3.

5.2. *Performance measures and results*

In order to compare the performance of the ART-based CCPR with that of a previously developed back-propagation pattern recognizer, three performance measures used here are *rate of target* (R_t), *average target pattern run length* (ARL_t), and *average target pattern run length index* (ARL_{tx}) (Hwarng and Hubele 1993a). R_t is the percentage of sequences of data in which the target pattern was first detected within a sequence of data of a limited length, i.e., measuring how frequently the CCPR detects the target pattern. ARL_t is the average run length of detecting the target pattern within a sequence of data, i.e., measuring how quickly the target pattern can be detected in a sequence of data. ARL_{tx} is defined as ARL_t/R_t which measures the average run length while considering the fact that some of the patterns cannot be detected within the limited length of the data sequence.

In the following, performance evaluation is based on separately, independently generated new testing data. R_t is calculated based on 100 independent sequences of data. Each sequence consists of 30 observations. ARL_t is calculated in terms of the number of classifying attempts. Also shown in the tables are the results produced by back-propagation pattern recognizer (BPPR) (Hwarng and Hubele 1993a). The two vigilance values for the ART-based CCPR are 0·6 and 0·7, while the two activation cutoff values for the BPPR are 0·85 and 0·90. These parameters were chosen because they produced approximately the same level of Type I errors.

The performance of the ART-based CCPR on trends is summarized in Table 1. As shown, R_t is sensitive to the level of random noise, i.e., the higher the noise, the lower the R_t. This is because high-noise trends tend to resemble sudden shifts. The effect of vigilance parameter ρ is less significant when the slope is smaller. When a trend is detected first, the trend can usually be detected within two attempts, i.e., $ARL_t < 2$, except one case. In general, the ART-based CCPR is comparable to the BPPR.

The performance on cycles is summarized in Table 2. Similar to trends, R_t is also sensitive to the effect of random noise, i.e., the higher the noise, the lower the R_t. However, it is possible to improve the performance on noisier cycles by learning additional templates. R_t tends to be slightly better as the amplitude increases. The effect of vigilance parameter ρ is more significant when the noise level is higher. Most cycles can be detected with an ARL_t of two at low noise level. Even at high noise level, cycles can be detected within an ARL_t of 8. The ART-based CCPR slightly outperforms BPPR on cycles with a higher amplitude and/or a higher level of random noise.

Slope (σ)	Noise (σ)	ART Vig. (ρ)	ART R_t (%)	ART ARL_t	ART ARL_{tx}	Back-Propagation Act. Cut.	Back-Propagation R_t (%)	Back-Propagation ARL_t	Back-Propagation ARL_{tx}
0·15	0·1	0·6	99	1·00	1·01	0·85	100	1·50	1·50
		0·7	99	1·42	1·42	0·90	100	1·50	1·50
0·25	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·02	1·02	0·90	100	1·01	1·01
0·35	0·1	0·6	100	1·00	1·00	0·85	91	1·00	1·09
		0·7	100	1·00	1·00	0·90	91	1·00	1·07
0·15	0·3	0·6	78	1·15	1·48	0·85	93	2·23	2·40
		0·7	82	1·69	2·07	0·90	88	2·27	2·58
0·25	0·3	0·6	94	1·05	1·12	0·85	98	1·04	1·06
		0·7	90	1·43	1·59	0·90	96	1·07	1·11
0·35	0·3	0·6	94	1·02	1·09	0·85	62	1·02	1·65
		0·7	78	1·13	1·45	0·90	57	1·02	1·79
0·15	0·5	0·6	55	1·44	2·61	0·85	77	2·89	3·75
		0·7	56	2·68	4·78	0·90	71	3·17	4·46
0·25	0·5	0·6	74	1·34	1·81	0·85	76	1·50	1·97
		0·7	56	1·80	3·22	0·90	71	1·62	2·28
0·35	0·5	0·6	76	1·18	1·56	0·85	44	1·09	2·48
		0·7	47	1·15	2·44	0·90	37	1·05	2·84

Table 1. Performance evaluation of ART-based CCPR: upward trend patterns with various slope and noise values.

Amp. (σ)	Noise (σ)	ART Vig. (ρ)	ART R_t (%)	ART ARL_t	ART ARL_{tx}	Back-Propagation Act. Cut.	Back-Propagation R_t (%)	Back-Propagation ARL_t	Back-Propagation ARL_{tx}
1·50	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
2·00	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
2·50	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
1·50	0·3	0·6	98	1·57	1·60	0·85	100	1·56	1·56
		0·7	99	3·02	3·05	0·90	100	1·72	1·72
2·00	0·3	0·6	96	1·67	1·74	0·85	94	1·21	1·29
		0·7	97	3·64	3·75	0·90	91	1·17	1·29
2·50	0·3	0·6	100	1·24	1·24	0·85	86	1·00	1·16
		0·7	100	1·56	1·56	0·90	82	1·05	1·28
1·50	0·5	0·6	92	3·85	4·18	0·85	95	3·58	3·77
		0·7	72	5·78	8·02	0·90	96	5·35	5·57
2·00	0·5	0·6	93	3·26	3·50	0·85	71	1·73	2·44
		0·7	78	7·56	9·69	0·90	73	2·15	2·95
2·50	0·5	0·6	93	3·51	3·77	0·85	53	1·46	2·75
		0·7	81	5·84	7·21	0·90	49	1·98	4·04

Table 2. Performance evaluation of ART-based CCPR: cyclic patterns (period = 8) with various amplitude and noise values.

The performance on systematic variable and on stratification is summarized in Tables 3 and 4 respectively. Though there is a very minor indication of lower R_t for higher noise, the performance is quite outstanding and consistent over the range of the parameters tested. It produces very few false alarms. This might be due to the fact that there are fewer variations within these two pattern classes compared to other unnatural patterns. Most systematic variables can be detected within an ARL_t of 4, while most stratification patterns can be detected within two attempts. Based on the given training data, the BPPR is more consistent on these two patterns than the ART-based CCPR.

With a very limited number of templates, the performance on mixtures is quite poor due to the random nature of the mixture patterns. This result is consistent with the observation and the guideline from our previous experience (Hwarng and Hubele 1993b). That is, the amount of training data should be approximately proportional to the complexity and irregularity of the pattern. To further verify it, we augmented the training data set by incorporating 21 additional templates for mixture patterns. The network was subsequently trained with these additional templates. Table 5 summarizes the performance based on the original and the augmented data sets. As evidenced, the performance is quite consistent and improved substantially. R_t seems to be better as the magnitude increases. In general, when a mixture pattern is detected, the mixture pattern can usually be detected within an ARL_t of 7, except in one case.

5.3. *Cumulative learning with an augmented training set*

One of the advantages for using ART in this application is its ability to preserve previously learned categories while learning new categories. This characteristic, cumulative learning, was verified by the above training with the augmented data set. After the second round of training, the class for mixtures was enforced as just discussed. The concern now is whether the network's ability to detect other pattern classes has degenerated. Further testing proved that the capability of recognizing previously learned categories was preserved. Table 6 lists the results of the verification test on some of the previous testing data for systematic variables and cycles. As evidenced, the effect of these augmented mixture templates on these two previously learned categories is negligible. The seeming drop of R_t for systematic ($1{\cdot}5\sigma$, $0{\cdot}5\sigma$) is because some of the new mixture templates are similar to systematic variables. This phenomenon once again affirms the principle of conflict-free. That is, the recognition capability can be degraded or never be achieved when conflicting natures between pattern classes are included in the training data.

5.4. *Other issues*

In interpreting the results presented here, some explanation on Type I errors should be in place as well, otherwise the utility of this approach would be questionable. Type I errors were measured by testing the CCPR on random patterns and calculating the average run length of signalling a non-random pattern. When vigilance parameter $\rho = 0{\cdot}6$, the ARL_{tx} is 7·2—an apparent indication of unacceptable Type I errors. When vigilance parameter $\rho = 0{\cdot}7$, the ARL_{tx} was improved to 25—a somewhat acceptable value as most of the ARL_{tx}'s for non-random patterns are under 3 or 4. It is natural to argue that this level of Type I errors is still too high. Upon close examination of the false signals generated, it was found that a majority of these falsely identified patterns were classified under stratification. Noting that stratification is characterized by observations hugging the process mean with a small deviation from

		ART				Back-Propagation			
Mag. (σ)	Noise (σ)	Vig. (ρ)	R_t (%)	ARL_t	ARL_{tx}	Act. Cut.	R_t (%)	ARL_t	ARL_{tx}
1·50	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
2·00	0·1	0·6	100	1·02	1·02	0·85	100	1·00	1·00
		0·7	100	1·08	1·08	0·90	100	1·00	1·00
2·50	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
1·50	0·3	0·6	99	1·48	1·50	0·85	100	1·52	1·52
		0·7	99	3·40	3·44	0·90	100	1·90	1·90
2·00	0·3	0·6	99	1·24	1·26	0·85	100	1·00	1·00
		0·7	98	2·90	2·96	0·90	100	1·01	1·01
2·50	0·3	0·6	100	1·02	1·02	0·85	100	1·00	1·00
		0·7	99	1·22	1·24	0·90	100	1·00	1·00
1·50	0·5	0·6	98	4·39	4·48	0·85	100	1·85	1·85
		0·7	87	6·03	6·94	0·90	100	2·59	2·59
2·00	0·5	0·6	96	2·02	2·11	0·85	100	1·01	1·01
		0·7	96	2·02	2·11	0·90	100	1·05	1·05
2·50	0·5	0·6	100	1·10	1·10	0·85	100	1·01	1·01
		0·7	100	2·56	2·56	0·90	100	1·01	1·01

Table 3. Performance evaluation of ART-based CCPR: systematic patterns with various magnitude and noise values.

		ART				Back-Propagation			
Offset from mean (σ)	Noise (σ)	Vig. (ρ)	R_t (%)	ARL_t	ARL_{tx}	Act. Cut.	R_t (%)	ARL_t	ARL_{tx}
− 0·20	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
0·00	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
0·20	0·1	0·6	100	1·00	1·00	0·85	100	1·00	1·00
		0·7	100	1·00	1·00	0·90	100	1·00	1·00
− 0·20	0·2	0·6	97	1·00	1·03	0·85	100	1·24	1·24
		0·7	100	1·14	1·14	0·90	100	1·34	1·34
0·00	0·2	0·6	100	1·00	1·00	0·85	100	1·03	1·03
		0·7	100	1·00	1·00	0·90	100	1·04	1·04
0·20	0·2	0·6	99	1·02	1·03	0·85	100	1·14	1·14
		0·7	99	1·30	1·31	0·90	100	1·21	1·21
− 0·20	0·3	0·6	90	1·28	1·42	0·85	98	2·11	2·15
		0·7	94	2·21	2·35	0·90	100	2·76	2·76
0·00	0·3	0·6	99	1·02	1·03	0·85	100	1·26	1·26
		0·7	100	1·53	1·53	0·90	100	1·45	1·45
0·20	0·3	0·6	95	1·08	1·14	0·85	100	1·76	1·76
		0·7	96	2·08	2·17	0·90	100	2·08	2·08

Table 4. Performance evaluation of ART-based CCPR: stratification patterns with various magnitude and noise values.

Mag. (σ)	Noise (σ)	Vig. (ρ)	Original training set R$_t$ (%)	Original training set ARL$_t$	Original training set ARL$_{tx}$	Augmented training set R$_t$ (%)	Augmented training set ARL$_t$	Augmented training set ARL$_{tx}$
1·75	0·1	0·6	32	1·25	3·91	89	1·18	1·32
		0·7	27	4·26	15·80	85	2·56	3·02
2·00	0·1	0·6	47	1·83	3·90	97	1·01	1·04
		0·7	32	6·13	19·10	92	1·64	1·78
2·25	0·1	0·6	1	8·00	8·00	93	1·00	1·07
		0·7	0	—	—	93	1·13	1·21
1·75	0·2	0·6	33	1·30	3·95	92	2·17	2·36
		0·7	44	1·70	3·87	85	6·53	7·68
2·00	0·2	0·6	54	3·98	7·37	91	1·69	1·86
		0·7	36	8·89	24·70	87	4·43	5·09
2·25	0·2	0·6	8	11·50	143·80	96	1·03	1·07
		0·7	5	12·00	240·00	98	1·65	1·69
1·75	0·3	0·6	43	2·44	5·68	87	2·24	2·58
		0·7	39	9·23	23·70	83	8·54	10·30
2·00	0·3	0·6	47	4·15	8·82	93	1·80	1·93
		0·7	26	9·92	38·20	95	6·98	7·35
2·25	0·3	0·6	14	11·60	82·70	93	1·13	1·21
		0·7	5	14·40	288·00	95	3·16	3·32

Table 5. Performance evaluation of ART-based CCPR: mixture patterns with various magnitude and noise values.

Mag. (σ)	Noise (σ)	Training data set	R$_t$ (%)	ARL$_t$	ARL$_{tx}$
1·5	0·3	Original	99	1·48	1·50
		Augmented	96	1·35	1·41
1·5	0·5	Original	98	4·39	4·48
		Augmented	82	3·61	4·40

(*a*) systematic patterns

Amp. (σ)	Noise (σ)	Training data set	R$_t$ (%)	ARL$_t$	ARL$_{tx}$
1·5	0·1	Original	100	1·00	1·00
		Augmented	100	1·00	1·00
1·5	0·3	Original	98	1·57	1·60
		Augmented	96	1·42	1·48

(*b*) cyclic patterns

Table 6. Effects of augmented training data set on performance ($\rho = 0{\cdot}6$).

the mean, it is reasonable that stratification with noise could be easily produced even under a random process. However, the chance of producing a long sequence of stratification or other patterns is small under a random process. Therefore, a solution to improving Type I errors under the current setting is to increase the number of observations in a classifying window. Recall that a small window size was used in order to test the ART-based CCPR's ability to recognize a legitimate pattern with the *least* number of observations. In practice, this window size should be adjusted (most likely increased) according to the nature of the process. In our simulation, the ARL_{tx} was improved to 116 when the CCPR signalled only if two patterns in a row were detected. With more observations in a classifying window, the classified results are expected to be more reliable as well.

Concerning ease-of-use between ART-based and BP-based CCPRs, it can be discussed from the following aspects. First, ART is more selective in the selection of training patterns, however, it needs much fewer templates than BP. Second, it is rather easy to configure an ART-based CCPR since it does not need to configure the hidden layer(s) as in BP. Third, it is much faster to train an ART and the learning behaviour is more predictable as well. Fourth, testing times of various patterns for ART are varied while they are constant for BP. Fifth, ART is easier to adapt to new applications since it can learn fast and cumulatively. Finally, ART has fewer critical parameters to set. The settings of vigilance parameter ρ during training and testing are similar to the settings of training-terminating errors and output activation-cutoff values in BP.

6. Conclusions

Among various approaches to detecting process non-randomness, the pattern recognition approach applied to control charts is a promising one. With their learning capability and computing power, neural networks seem to have produced a pragmatic platform for implementing control chart pattern recognition in real-world situations. However, as often seen in some network architectures, slowness in learning and inability to perform adaptive learning without re-learning still pose some inconvenience for practical applications. Hence, the objective of current research is to investigate and develop a control chart pattern recognizer which is capable of fast and cumulative learning.

Taking advantage of its design principles and architecture, ART was adopted as a foundation. However, inherent limitations of ART cause some intolerable difficulties in the implementation of control chart pattern recognition. In this paper, we proposed some alternatives and presented an ART-based, general-purpose control chart pattern recognizer which is capable of fast and cumulative learning. Unlike supervised ARTMAP, this quasi-supervised approach adopts a rather simple architecture yet achieves flexible and effective clustering. The implementation of this ART-based CCPR was made possible by introducing two key alternatives, that is, incorporating a synthesis layer in addition to the existing two-layer architecture and adopting a quasi-supervised training strategy. A detailed algorithm with the training and the testing modes was presented. Extensive simulation and performance evaluation were conducted and proved that this ART-based CCPR indeed possesses the capability of fast and cumulative learning. When compared with a back-propagation pattern recognizer (BPPR), the ART-based CCPR is superior on cyclic patterns, inferior on mixture patterns, and comparable on other patterns. However, it should be noted that the comparison made here is based on two particular training data sets and can only serve as a basis for understanding the capabilities of these two neural network architectures on control chart pattern recognition.

Although this paper demonstrated the feasibility and the utility of a general-purpose CCPR, an ART-based CCPR can be designed to accommodate the specific needs of a particular process and may produce even more satisfactory results. It is believed that an ART-based CCPR such as the one proposed here can be readily integrated into a dynamic manufacturing environment as a key component of an automated SPC system.

Acknowledgment

The authors would like to express their gratitude to the two anonymous referees for their valuable comments. This research was supported in part by National University of Singapore research grant No. 3920070.

References

CARPENTER, G. A., and GROSSBERG, S., 1987, A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, **37**, 54–115.

CARPENTER, G. A., and GROSSBERG, S., 1988, Self-organizing neural network architectures for real-time adaptive pattern recognition. *IEEE Computer*, **21** (3(, 77–88.

CARPENTER, G. A., GROSSBERG, S., and REYNOLDS, J. H., 1991, ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, **4**, 565–588.

GROSSBERG, S., 1976, Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.

GUO, Y., and DOOLEY, K. J., 1992, Identification of change structure in statistical process control. *International Journal of Production Research*, **30** (7), 1655–1669.

HWARNG, H. B., and HUBELE, N. F., 1991, X-bar chart pattern recognition using neural nets. *ASQC Quality Congress Transactions*, Milwaukee, Wisconsin, pp. 884–889.

HWARNG, H. B., and HUBELE, N. F., 1992, Boltzmann machines that learn to recognize patterns on control charts. *Statistics and Computing*, **2** (4), 191–202.

HWARNG, H. B., and HUBELE, N. F., 1993a, Back-propagation pattern recognizers for $\bar{X}$ control charts: methodology and performance. *Computers and Industrial Engineering*, **24** (2), 219–235.

HWARNG, H. B., and HUBELE, N. F., 1993b, $\bar{X}$ control chart pattern identification through efficient off-line neural network training. *IIE Transactions*, **25** (3), 27–40.

LIM, M. H., GWEE, B. H., and GOH, T. H., 1991, Cause associator network for fuzzily deduced conclusion in process control. *International Joint Conference on Neural Networks*, Singapore, **II**, 1248–1253.

LIM, M. H., and OOI, T. H., 1990, Statistical process control using fuzzy logic. *Condition Monitoring and Diagnostic Technology*, **1** (2), 64–70.

LIPPMAN, R., 1987, An introduction to computing with neural network. *IEEE ASSP Magazine*, **4** (2), 4–22.

RYAN, T. W., and WINTER, C. L., 1987, Variations on adaptive resonance theory. *International Joint Conference on Neural Networks*, **II**, 767–775.

WESTERN ELECTRIC CO., INC., 1956, *Statistical Quality Control Handbook* (New York: Western Electric).