Mathematical and Computational Applications, Vol. 16, No. 2, pp. 514-523, 2011.
© Association for Scientific Research

AN INTEGRATED NEURAL NETWORK STRUCTURE FOR RECOGNIZING AUTOCORRELATED AND TRENDING PROCESSES

Aslan Deniz KARAOGLAN
Department of Industrial Engineering,
Dokuz Eylul University, 35160,
Tinaztepe, Buca - Izmir, TURKEY
deniz.karaoglan@deu.edu.tr

Abstract- Data sets collected from industrial processes may have both a particular type of trend and correlation among adjacent observations (autocorrelation). In the present paper, an integrated neural network structure is used to recognize trend stationary first order autoregressive (trend AR(1)) process. The proposed integrated structure operates as follows. *(i)* First a combined neural network structure (CNN), that is composed of appropriate number of linear vector quantization (LVQ) and multi layer perceptron (MLP) neural networks, is used to recognize the trended data, *(ii)* then, the Elman's recurrent neural network (ENN) is used to diagnose the autocorrelation through the data. Correct classification rate is used as performance criteria. Results indicate that proposed structure is effective and competitive with other combined neural network structures.
Key Words- Control Chart Pattern Recognition, Neural Networks, Trend AR(1)

1. INTRODUCTION

Control charts are statistical process control (SPC) tools used to determine whether a process is in-control. The standard assumptions that are usually cited in justifying the use of control charts are that the data generated by the in-control process are normally and independently distributed (iid) by mean of μ and standard deviation of σ [1]. The most frequently reported effect on control charts of violating such assumptions is the erroneous assignment of the control limits. Alwan and Roberts (1995) showed that about 85% of a sample of 235 control chart applications displayed incorrect control limits. More than half of these displacements were due to violation of the independence assumption. Misplacement of control limits was due to serial correlation (i.e., autocorrelation) in the data. However, many processes such as those found in refinery operations, smelting operations, wood product manufacturing, waste-water processing and the operation of nuclear reactors have been shown to have autocorrelated observations [2].

In addition to autocorrelation, some types of industrial processes such as chemical processes also exhibit a particular kind of trend behavior. Trends are usually due to gradual wearing out or deterioration of a tool or some other critical process components. They can also result from human causes, such as operator fatigue or the presence of supervision. In chemical processes linear trend often occurs because of settling or separation of the components of a mixture. Such process data is usually modeled by a trend stationary first order autoregressive (trend AR(1)) model.

Analysis of unnatural patterns is an important aspect of SPC. Researchers have been investigated the use of artificial neural networks (NNs) in the application of

control chart pattern recognition (CCPR) [3]. In the present study, an integrated neural network structure is presented to recognize trend AR(1) processes effectively. The proposed structure produces desirable results under given assumptions and the results show that the proposed structure is competitive with the other approaches reviewed in the literature.

When compared to other methodologies the NN approach has certain advantages. The model development is much simpler than that for most other approaches. Instead of theoretical analysis and development for a new model the neural network tailors itself to the training data. The model can be refined at any time with the addition of new training data [4].

In recognition problems, NNs can recall patterns learned from noisy or incomplete representations, which makes them suitable for CCPR because CCPs are generally contaminated by common cause variations [3]. Pattern recognition provides a mechanism for identifying different types of predefined patterns in real time on the series of process quality measurements. The recognized patterns then serve as the primary information for identifying the causes of unnatural process behavior [2].

Various studies have demonstrated the utility of NNs in identifying CCPs. Pham and Oztemel (1992a, 1994) [5, 6] used a backpropagation network (BPN) and learning vector quantization (LVQ) network to recognize shift, trend and cycle patterns on control charts. Hwarng and Hubele (1993) [7] extensively investigated CCPR by training BPNs to detect six unnatural CCPs sudden shift, trend, cycle, stratification, systematic, and mixture. Cheng (1997) [4] developed a NN approach for the analysis of CCPs. Anagun (1998) [8] organized the training data in two different ways (direct representation and histogram representation) before introducing them to the designed NN applied to pattern recognition in SPC. Guh and Tannock (1999) [9] tried to investigate the feasibility of an NN to recognize concurrent CCPs (where more than one pattern exists together). Pham and Chan (2001) [10] described in their paper, the use of unsupervised adaptive resonance theory ART2 NNs for recognizing patterns in control charts. Al-Assaf (2004) [11] used multi-resolution wavelets analysis (MRWA) to extract distinct features for unnatural patterns by providing distinct time–frequency coefficients. Gauri and Chakraborty (2006) [12] developed two feature-based approaches using heuristics and NNs, which are capable of recognizing eight most commonly observed CCPs.

It is well known that the combined NNs that consist of three or more NNs give more successful solutions when compared with single NN structure. Each NN are trained independently; and depending on unanimity, the outputs of these NNs are combined by the help of a collective decision making module (CDMM) and a collective decision is performed [13]. In the present work, it is aimed to recognize trend AR(1) process by means of NNs. The proposed network structure operates as follows; *(i)* first, a combined neural network structure (CNN), that is composed of two linear vector quantization NNs (LVQ NNs) and a multi layer perceptron NN (MLP NN) is used to recognize the trend in process data, *(ii)* Second, the Elman's recurrent NN (ENN) is used to diagnose the autocorrelation through the trended data.

Rest of the paper is organized as follows. Next section describes the NN aided pattern recognition for trend AR(1) process. In Section 3, operation of proposed NN structure is described. Conclusions are pointed out in Section 4.

2. THE NN AIDED PATTERN RECOGNITION FOR TREND AR(1) PROCESS

A stationary autoregressive process of lag 1 (AR(1)) time series model is the representative model for the autocorrelated process. In an AR(1) process, the current observation is correlated with its previous observation. Past studies emphasize the role of AR(1) processes in process control [3]. An AR(1) time series can be utilized to model many data sets collected from manufacturing operations, such as the papermaking and the viscosity data. An AR(1) model can be expressed as follows:

$$x_t = \xi + \phi x_{t-1} + \varepsilon_t \quad (1)$$

where t is the time of sampling, x_t is the sample value at time t, ξ is the constant, ϕ is the autoregressive coefficient ($-1<\phi<1$), and ε_t is the independent random error term (common cause variation) at time t following $N(0, \sigma_\varepsilon^2)$. Let X_t with an increasing linear trend (trend for short) is represented by:

$$X_t = x_t + dt \quad (2)$$

where d is the trend slope in terms of t. Let x_t with a shift or jump is given by:

$$Z_t = X_t + \delta_\mu \quad (3)$$

where δ_μ is the magnitude of upward mean shift. The aim of this study is to diagnose $\{Z_t\}$ by using NNs.

Figure 1 displays the NN aided pattern recognition and process monitoring procedure. As can be seen, for recognizing the autocorrelated and trending patterns, a combined NN architecture is used to provide a collective authority in decision for trending data, and called it combined neural network recognizer (CNNR), then ENN is employed to recognize the autocorrelation that is filtered by CNNR. CNNR is composed of appropriate number of LVQ and MLP networks. In the relevant literature, the MLP and LVQ have been shown to be useful for statistical pattern recognition. Each member of CNNR produces its decision independently and the unanimously elected decision is accepted as the decision of CNNR. A recurrent ENN where the recurrency allows the network to remember cues from the recent past is suitable for recognizing time series data and monitoring process shifts in the presence of autocorrelation [2].

Topologies of feedforward MLP and LVQ are given in Figure 2 and Figure 3, respectively. Each neuron represented by truncated cylinders in a layer is connected with all neurons of the next layer by arcs. Each arc has a weight. Threshold value prevents the neurons to produce zero value. In Figure 3, the weights of arcs between Kohonen and output layers are equal to one (see [12] and [13] for details). The output of MLP and LVQ is one of the control chart pattern (CCP). The desired CCP for LVQ and MLP for the input data under consideration is increasing linear trend that is one of the six CCP types (natural pattern: NP; upward shift: US; downward shift: DS; increasing trend: IT; decreasing trend: DT; periodic shift: PS).

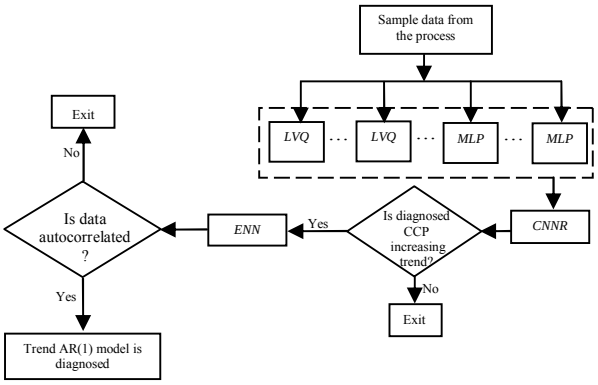


Figure 1. Proposed NN aided pattern recognition and process monitoring procedure

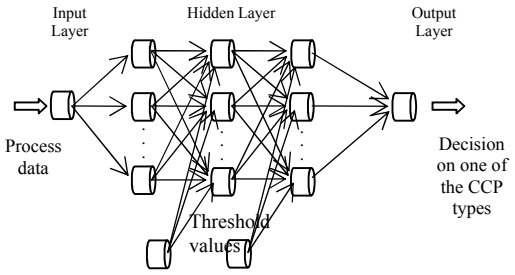


Figure 2. Feedforward MLP NN [13, 14]

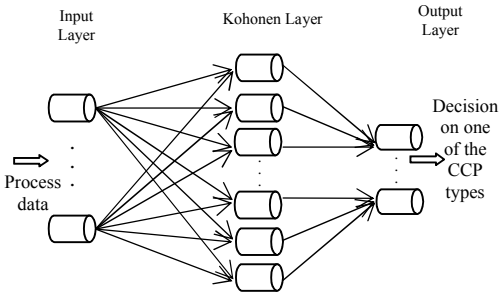


Figure 3. LVQ NN [3, 13, 15]

Topology of ENN is given in Figure 4. It is a recurrent supervised neural network like MLP. But, unlike MLP, it has the dynamic memory property and its input layer does not include transfer function. The ENN employs feedback connections and addresses the temporal relationship of its inputs by maintaining an internal state. The output of ENN is that if the trended process data filtered by CNNR is autocorrelated or not.

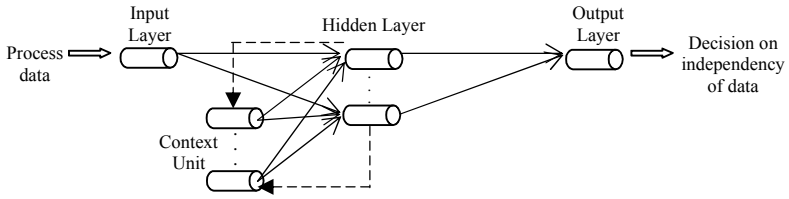


Figure 4 Elman NN [13, 15]

If trend and autocorrelation are diagnosed in data, then model parameters of trend AR(1) can be identified. In the relevant literature, for the autocorrelated processes, NNs are used to recognize if the input pattern is one of the CCP types (first task) and is it autocorrelated (second task), simultaneously. Because of the complexity of autocorrelated processes with one of CCP types, the training process of networks can be hard, while the correct classification rate decreases. By the proposed integrated structure, the mentioned two tasks are distributed to different networks. The first task is performed by CNNR that is composed of appropriate number of LVQ and MLP networks, while the second task is performed by the ENN that is advisable to recognize autocorrelation. Executing only one of the given tasks, correct classification rate of each network increases and training these networks are simplified; and then by combining

the results of each network, the performance of proposed network structure is increased when it is compared with the results represented in the literature.

3. OPERATION OF PROPOSED NN STRUCTURE

In this section, we illustrate operation of the integrated NN structure step by step beginning from training neural-based structure to the identification of trend and autocorrelation in data and its performance.

3.1 Generating sample data

In this section we explain how training data sets were generated to show the pattern recognition by the MLP, LVQ and detecting the serial correlation by the ENN. The process simulator coded in MATLAB 7.4.0 was used for generating the training data sets. The details of process simulator are given in appendix.

3.1.1 Training data set for CNNR

For each type of the six CCPs 400 sample data sets were generated. Each data set is composed of 500 observations. While one half of the sets are uncorrelated, the other half contains both correlated and uncorrelated data sets. The former sets are collected in *set1*, and the latter in *set2*. For each CCP type in *set2*, 40 data sets were generated using each of five ϕ values such as 0.95, 0.475, 0.0, -0.475, and -0.95. That is, the first 40 sets were simulated using ϕ=0.95, the second 40 sets for ϕ=0.475, and so on. *Set1* and *set2* were generated by different process simulators that are given in appendix. *Set1* and *set2* were used together for training of MLP and LVQ NNs. Parameter values used for process simulators are displayed in Table 1.

Table 1. CCP training set for the LVQ and MLP networks

Pattern type	Parameter range for *set1*	Parameter range for *set2*	Set name and number of observations
Natural Pattern	$t=[1,500]$, $\mu=10$, $\sigma=2$, $u=$ Uniform$[0,1]$	$t=[1,500]$, $\xi=0$, $x_1=10$, $\varepsilon\sim N(0,4)$, ϕ : 0.95, 0.475, 0.0, -0.475, -0.95	*Set1:*200 *Set2:*200
Increasing trend	$g=$ Uniform$[0.1,0.125]$ (for the first 100 data set) $g=0.2$ (otherwise)	$d=0.2$	*Set1:*200 *Set2:*200
Decreasing trend	$(-g)$	$(-d)$	*Set1:*200 *Set2:*200
Sliding up	$k_{sliding}=0$ (if $t<50$) $k_{sliding}=1$ (if $t\geq 50$) $s_{displacement}=$Uniform [2,4]	$k_{sliding}=0$ (if $t<150$) $k_{sliding}=1$ (if $t\geq 150$) $s_{displacement}=(4.5)$	*Set1:*200 *Set2:*200
Sliding down	$(-k_{sliding})$	$(-k_{sliding})$	*Set1:*200 *Set2:*200
Periodical shifting	$N=500$ $\pi=180$ $\alpha_{periodic}=$randint [0,30]	$N=500$ $\pi=180$ $\alpha_{periodic}=(0.25,1.0,3.0)$	*Set1:*200 *Set2:*200

3.1.2 Training data set for ENN

To train ENN, both autocorrelated and uncorrelated data sets, with increasing and decreasing linear trend, were generated. Each set is composed of 500 data points. Underlying model for autocorrelated process is given in Equation (1). Totally 800 sets

were generated for autocorrelated process; one half for increasing trend and other half for decreasing trend. On the other hand, each 100 of autocorrelated sets have a specific ϕ value listed in the previous section. Underlying model for uncorrelated process with increasing trend is given below.

$$X_t = \mu_t + \varepsilon_t + dt \tag{4}$$

For the process with decreasing trend sign of the dt in Equation (4) will be negative. 400 sets were generated for each of uncorrelated process with increasing trend and decreasing trend, respectively. Parameter values used for CCP training data sets are depicted in Table 2.

Table 2. Training set for the ENN

Parameter range for *AR*(1) process data	Parameter range for uncorrelated process data
$\xi = 0,\ x_1 = 10$, $\varepsilon \sim N(0,4)$, $t = [1,500]$, $d = 0.2$,	$\varepsilon \sim N(0,4)$, $t = [1,500]$
$(-d)$ (for decreasing trend), ϕ : 0.95, 0.475, -0.475, -0.95	$g = 0.2$, $(-g)$ (for decreasing trend)

3.2 Recognition of trend and autocorrelation in data

3.2.1 CNNR to detect six unnatural CCPs

Three different ANN based recognizers were developed. First two of these recognizers (ANN1 and ANN2) use LVQ algorithm. Architectures of ANN1 and ANN2 differ in number of neurons at Kohonen layer. The third member of CNNR uses MLP structure. Configurations of the MLP and the LVQ networks which are the members of the CNNR are depicted in Table 3 and Table 4, respectively. Preliminary investigations are conducted to choose a suitable network topology and training algorithm for each member of CNNR. According to the experimental design performed for investigating the appropriate network topology of MLP network, the design parameters for number of hidden layers (from 1 to 4 increasing one by one), number of neurons at each layer (from 8 to 72 increasing eight by eight), learning rates (0.01, 0.02) and momentum constants (from 0.02 to 0.05 increasing with step size 0.01) are used and minimum mse is reached for the design parameters that are given in Table 3. Similarly, for LVQ networks, the design parameters for learning rate (0.01, 0.03, 0.06, 0.09) respect to number of neurons at Kohonen layer (from 4 to 40 neurons increasing four by four) are used and minimum mse is reached with the design parameters that are given in Table 4.

Table 3. Network configuration for MLP NN

Member of CNNR :	ANN3
Type of ANN	MLP
Number of hidden layers	3
Number of neurons at each hidden layer respectively	12, 64, 64
Number of neurons at output layer	1
Transfer functions of hidden layers respectively	Purelin, Tangent sigmoid, Tangent sigmoid
Transfer function of output layer	Purelin
Performance function / Training set	mse (mean square error) / *Set1+ Set2*
Error goal / Maximum number of epochs	$1x10^{-2}$ / 30000
Momentum constant / Learning rate	0.4 / 0.01

Table 4. Network configurations for LVQ NNs

Members of CNNR :	ANN1	ANN2
Type of ANN	LVQ	LVQ
Number of neurons at Kohonen layer	24	36
Learning rate / Max number of epochs / Error goal	0.01 / 200 / $1x10^{-17}$ (1e-17)	0.01 / 200 / $1x10^{-17}$ (1e-17)
Training example set	*Set1+ Set2*	*Set1+ Set2*
Training example set class percentages	are same and equal to (1/6) for each of six CCP type	

The generated data were presented to each NN independently and also each NN was trained independently. The coding is performed in MATLAB 7.4.0. Then outputs of these NNs were combined and by the help of the CDMM and a collective decision is performed (see [13] for detailed information). The training was stopped whenever either the error goal has been achieved or the maximum allowable number of training epochs has been met. Now verification sets are needed for testing the performance of NNs which are the members of CNNR. New sets of *verification samples* of size 2400 each were generated by using the parameters given in Table 1 again. The generated samples for verification were used to test the performance of CNNR members. The recognition performance of all these ANN-based recognizers was tested using different sets of test samples. The verification results of the ANN-based recognizers are displayed through Table 5 - Table 7, and the performance of CNNR is given in Table 8. The elements in these tables are the classification rates (CR) of networks by percentages. The columns represent the expected classification for the input pattern, while rows represent the actual classification rate of network for the given test set. For example in Table 5, for the first column the expected classification is NP, but as it can be seen from the last row of Table 5, the correct classification rate of ANN1 is 92.5%.

Table 5. Testing results of the ANN1 for autocorrelated data

CR (%)		Required classification					
		NP	US	DS	IT	DT	PS
Network classification	PS	0.0000	0.0000	0.0000	0.0000	0.0000	0.7565
	DT	0.0000	0.0000	0.0052	0.0000	0.9982	0.0000
	IT	0.0000	0.0105	0.0000	0.9884	0.0000	0.0000
	DS	0.0750	0.0020	0.9848	0.0000	0.0018	0.0000
	US	0.0000	0.9875	0.0000	0.0116	0.0000	0.0000
	NP	0.9250	0.0000	0.0100	0.0000	0.0000	0.2435

Table 6. Testing results of the ANN2 for autocorrelated data

CR (%)		Required classification					
		NP	US	DS	IT	DT	PS
Network classification	PS	0.0000	0.0000	0.0000	0.0000	0.0000	0.7550
	DT	0.0000	0.0000	0.0010	0.0000	0.9975	0.0000
	IT	0.0000	0.0000	0.0000	0.9876	0.0000	0.0000
	DS	0.0600	0.0150	0.9990	0.0000	0.0025	0.0000
	US	0.0000	0.9850	0.0000	0.0124	0.0000	0.0000
	NP	0.9400	0.0000	0.0000	0.0000	0.0000	0.2450

Table 7. Testing results of the ANN3 for autocorrelated data

CR (%)		Required classification					
		NP	US	DS	IT	DT	PS
Network classification	PS	0.0000	0.0100	0.0000	0.0000	0.0000	0.9990
	DT	0.0000	0.0050	0.0100	0.0350	0.9950	0.0000
	IT	0.0050	0.0000	0.0000	0.9500	0.0000	0.0000
	DS	0.0000	0.0350	0.9900	0.0150	0.0050	0.0000
	US	0.0000	0.9500	0.0000	0.0000	0.0000	0.0000
	NP	0.9950	0.0000	0.0000	0.0000	0.0000	0.0010

Table 8. Testing results of the CNNR for autocorrelated data and comparisons with the results presented by Guh (2008)

CR (%)		Required Classification											
		NP		US		DS		IT		DT		PS	
		Proposed CNNR	Guh (2008)	Proposed CNNR	Guh (2008)	Proposed CNNR	Guh (2008)	Proposed CNNR	Guh (2008)	Proposed CNNR	Guh (2008)	Proposed CNNR	Guh (2008)
Network Classification	PS	0.0000	0.0092	0.0000	0.0075	0.0000	0.0084	0.0000	0.0000	0.0000	0.0033	0.7650	**0.9606**
	DT	0.0000	0.0000	0.0000	0.0000	0.0024	0.0496	0.0000	0.0000	**0.9968**	0.9619	0.0000	0.0000
	IT	0.0000	0.0000	0.0090	0.0395	0.0000	0.0000	**0.9947**	0.9705	0.0000	0.0000	0.0000	0.0000
	DS	0.0600	0.0080	0.0000	0.0000	**0.9976**	0.9161	0.0000	0.0000	0.0032	0.0349	0.0000	0.0054
	US	0.0000	0.0127	**0.9910**	0.9495	0.0000	0.0000	0.0053	0.0295	0.0000	0.0000	0.0000	0.0032
	NP	0.9400	**0.9521**	0.0000	0.0035	0.0000	0.0172	0.0000	0.0000	0.0000	0.0000	0.2350	0.0110

As can be seen from these tables, at the training and verification phases, recognition performances of three members of CNNR do not differ significantly for trended patterns and are highly correct for all NN based recognizers. As can be seen from Table 8, the correct classification rate of CNNR is higher then its members and 99.47% for increasing trend. These correctly recognized trended patterns are used as input for ENN. Also the results that are displayed in Table 8 for CNNR are compared with the related results presented by Guh (2008) [3]. According to Table 8 it is clearly observed that proposed system performs well for US, DS, IT, and DT when it is compared with the results of Guh (2008) [3].

3.2.2 Network configuration of ENN

Initially, all weight values were chosen randomly and then they were optimized during the training stage. Configuration for ENN is given in Table 9.

Table 9. Network configuration for ENN

Number of neurons at each layer respectively	6 (for hidden layer), 1 (for output layer)
Transfer functions of hidden layer and output layer	Tangent Sigmoid, Purelin
Performance function / Training example set	mse (mean square error) / Presented in Table 2
Error goal / Maximum number of epochs	$1x10^{-2}$ / 500

Desired output of ENN is either 1 if the autocorrelation has been detected or 0 otherwise. Due to the random noise and to different values of actual inputs, the output of ENN is a number ranging approximately between 0 and 1. Therefore, an activation cut off value must be defined to release an alarm if the network output is greater than the cutoff. Similar to the approach used by Pacella and Semeraro [2], cut off values (C) 0.60 for increasing trend and 0.5290 for decreasing trend are defined. The trained NN signals greater than or equal to cutoff value point out that the tested data set has serial correlation, otherwise has no serial correlation. Table 10 summarizes the performance of ENN for autocorrelated data with increasing linear trend that is filtered by CNNR.

Table 10. Testing results of the ENN

CR (%)		Required classification	
		Autocorrelated	Uncorrelated
Network Classification	Autocorrelated	0.9846	0.0003
	Uncorrelated	0.0154	0.9997

The results displayed in Table 10 indicate that trend AR(1) is recognized with 98.46 % accuracy rate for the cut off value 0.60. This means that autocorrelation has

been detected by ENN with 98.46% accuracy through the set that is correctly recognized by CNNR (see Table 8) with 99.47% accuracy from trend AR(1) test set (this means 97.94% accuracy). Based on preliminary investigation, no evident improvement in performance was attained by extending the training set beyond 400 examples for each type. According to results presented by Guh (2008) that are listed in Table 8, autocorrelated and increasingly trending patterns are recognized with 97.05% and this result shows that proposed CNNR with 97.94% accuracy outperforms results presented by Guh (2008) [3].

4. CONCLUSION

Recently researchers' interest focused on using neural-based approaches to control chart pattern recognition in the case of autocorrelated data. In the present work it is aimed to recognize trend AR(1) model through autocorrelated process observations. The numerical results from an extensive simulation study indicate that the proposed integrated neural network structure can work reasonably well even in cases where the process data are highly positively or negatively correlated. Empirical comparisons reveal that the proposed NN structure performs more efficiently than presented combined NN structures in the literature in recognizing autocorrelated and trending process observations (for details see the literature given in references to compare the performance of proposed integrated network structure with previous combined network studies carried out for the same type of problems). Future research can be extended to other CCP types.

Acknowledgment- The author would like to thank Prof. Dr. G. Mirac BAYHAN for her helpful comments. Also the author wishes to thank the editor and the referees for their thoughtful and detailed suggestions that improved the final version of this paper.

REFERENCES

1. D. C. Montgomery, *Introduction to Statistical Quality Control*, John Wiley & Sons Inc., New York, 1997.
2. M. Pacella and Q. Semeraro, Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring, *Computers & Industrial Engineering* **52**, 502–520, 2007.
3. R. S. Guh, Real-time recognition of control chart patterns in autocorrelated processes using a learning vector quantization network-based approach, *International Journal of Production Research* **46**, 3959-3991, 2008.
4. C. S. Cheng, A neural network approach for the analysis of control chart patterns, *International Journal of Production Research* **35**, 667-697, 1997.
5. D. T. Pham and E. Oztemel, Control chart pattern recognition using neural Networks, *Journal of Systems Engineering* **2**, 256-262, 1992a.
6. D. T. Pham and E. Oztemel, Control chart pattern recognition using learning vector quantization networks, *International Journal of Production Research* **32**,721-729, 1994.
7. H. B. Hwarng and N. F. Hubele, Backpropagation pattern recognizers for (x) over-bar control charts - methodology and performance, *Computers & Industrial Engineering* **24**, 219-235, 1993.

8. A. S. Anagun, A neural network applied to pattern recognition in statistical process control, *Computers & Industrial Engineering* **35**, 185-188, 1998.
9. R. S. Guh and J. D. T. Tannock, Recognition of control chart concurrent patterns using a neural network approach, *International Journal of Production Research* **37**, 1743-1765, 1999.
10. D. T. Pham and A. B. Chan, Unsupervised adaptive resonance theory neural networks for control chart pattern recognition, *Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture* **215**, 59-67, 2001.
11. Y. Al-Assaf, Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks, *Computers & Industrial Engineering* **47**, 17-29, 2004.
12. S. K. Gauri and S. Chakraborty, Feature-based recognition of control chart patterns, *Computers & Industrial Engineering* **51**, 726-742, 2006.
13. E. Oztemel, *Artificial neural networks*, Papatya press, Istanbul, 2003.
14. S. K. Gauri and S. Chakraborty, Improved recognition of control chart patterns using artificial neural networks, *International Journal of Advanced Manufacturing Technology* **36**, 1191-1201, 2008.
15. F. M. Ham and I. Kostanic, *Principles of neurocomputing for science and engineering*, McGraw-Hill, New York, 2001.

APPENDIX - THE PROCESS SIMULATOR

The process simulator for generating the uncorrelated CCP examples [13]:

$$y_t = \mu + r_t\sigma \text{ and } r_t = \sqrt{-2\ln(u)}\cos 2\pi u \text{ (normal distribution)} \quad \text{(A1)}$$

where u is the Uniform[0,1] random variable and π is the *Pi* constant.

$$y_t = \mu + r_t\sigma \mp gt \text{ (increasing or decreasing trend)} \quad \text{(A2)}$$

where g is the trend slope of uncorrelated process

$$y_t = \mu + r_t\sigma \mp k_{sliding}s_{displacement} \text{ (sliding up or sliding down)} \quad \text{(A3)}$$

where $k_{sliding}$ is sliding coefficient at the sliding moment and $s_{displacement}$ is the displacement of process mean in terms of standard deviation

$$y_t = \mu + r_t\sigma + \alpha_{periodic}\sin(2\pi t/N) \text{ (periodic shift)} \quad \text{(A4)}$$

where $\alpha_{periodic}$ is the dimension of the periodic shifting (random integer variable).

The process simulator for generating the CCP examples in the AR(1) process [3]:

$$x_t = \xi + \phi x_{t-1} + \varepsilon_t \text{ and } X_t = x_t + d_t \quad \text{(A5)}$$

where d_t is the special disturbance at time t (zero when no unnatural pattern present).

$$d_t = 0 \text{ (natural pattern)} \quad \text{(A6)}$$

$$d_t = \mp dt \text{ (increasing or decreasing trend)} \quad \text{(A7)}$$

where d is the trend slope of trend AR(1) model,

$$d_t = \mp k_{sliding}s_{displacement} \text{ (sliding up or sliding down)} \quad \text{(A8)}$$

$$d_t = \alpha_{periodic}\sin(2\pi t/N) \text{ (periodic shift)} \quad \text{(A9)}$$