Journal of Engineering and Technology Research Vol. 3(1), pp. 5-15, January 2011
Available online at http:// www.academicjournals.org/JETR
ISSN 2006-9790 ©2011 Academic Journals

Full Length Research Paper

Spiking neural network-based control chart pattern recognition

Medhat H. A. Awadalla[1]*, I. I. Ismaeil[1] and M. Abdellatif Sadek[2]

[1]Communications and Electronic Department, Faculty of Engineering, Helwan University, Egypt.
[2]Information Technology Department, High Institute of Engineering, Shorouk Academy, Egypt.

Accepted 13 December, 2010

Due to an increasing competition in products, consumers have become more critical in choosing products. The quality of products has become more important. Statistical process control (SPC) is usually used to improve the quality of products. Control charting plays the most important role in SPC. Control charts help to monitor the behavior of the process, to determine whether it is stable or not. Unnatural patterns in control charts mean that, there are some unnatural causes for variations in SPC. Spiking neural networks (SNNs) are the third generation of artificial neural networks that consider time as an important feature for information representation and processing. In this paper, spiking neural network architecture is proposed to be used for control charts pattern recognition (CCPR). Furthermore, enhancements to the SpikeProp learning algorithm are proposed. These enhancements provide additional learning rules for the synaptic delays, time constants and for the neurons thresholds. Experiments have been conducted and the achieved results show a remarkable improvement in the overall performance compared with artificial neural networks.

Key words: Spiking neural network, control chart pattern recognition, SpikeProp algorithm,

INTRODUCTION

Traditionally, statistical process control (SPC) was used only for monitoring and identifying process variation. Advances in SPC charting have moved from merely statistical and economic control to diagnosis purposes through control chart pattern identification (Ibrahim and Adnan, 2010). Control charts are useful tool in detecting out-of-control situations in process data (Hui-Ping and Chuen-Sheng, 2009). A process is considered out of control, if a point falls outside the control limits or a series of points exhibit an unnatural pattern (also known as nonrandom variation). There are seven basic CCPs, normal (NOR), systematic (SYS), cyclic (CYC), increasing trend (IT), decreasing trend (DT), upward shift (US) and downward shift (DS), as shown in Figure 1. All other patterns are either special forms of basic CCPs or mixed forms of two or more basic CCPs.

Advances in manufacturing and measurement technology have enabled real-time, rapid and integrated gauging and measurement of process and product quality (Indra et al., 2010). A typical control chart consists of a centre line (CL) corresponding to the average statistical level and two control limits, upper (UCL) and lower (LCL) normally located at ±3σ of this statistic, where σ is a measure of the spread, or standard deviation in a distribution. Selection of CCPs parameters is important for training and testing the ANN recognizers. Among the important parameters included window size, random noise, mean shift (for shift patterns), trend slope (for trend patterns), cycle amplitude and cycle period (for cyclic pattern), and systematic departure (for systematic pattern), Table 1 shows the equations that described all patterns in CCPs. The following equations were used to create the data points for the various patterns.

Control chart pattern recognition has the capability to recognize unnatural patterns (Jenn-Hwai et al., 2005). Pattern recognition is an information-reduction process which aims to classify patterns based on a priori knowledge or based on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations come from process or event (Samir, 2009).

Currently, most researchers in artificial neural networks are interested in spiking neural networks (SNNs). SNNs

*Corresponding author. E-mail: awadalla_medhat@yahoo. co.uk.

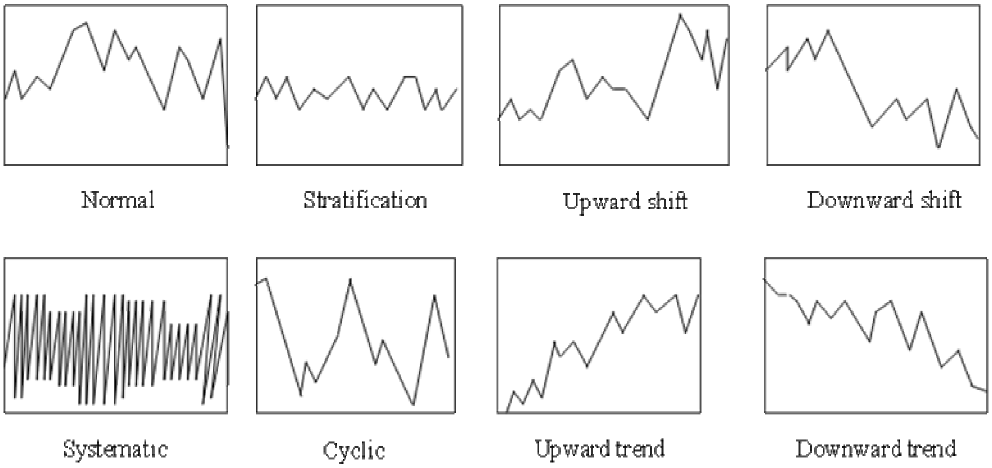

Figure 1. Common CCPs for univariate process.

Table 1. Equations for simulation control charts.

Patterns class	Description	Equations
1	Systematic	$P(t) = \eta + r_i(t)\sigma + d \times (-1)^i$
2	Cyclic	$P(t) = \eta + r_i(t)\sigma + a\ sin\ (2\pi t/T)$
3	Increasing Trend	$P(t) = \eta + r_i(t)\sigma + gt$
4	Decreasing Trend	$P(t) = \eta + r_i(t)\sigma - gt$
5	Upward Shift	$P(t) = \eta + r_i(t)\sigma + bs$
6	Downward Shift	$P(t) = \eta + r_i(t)\sigma - bs$

where η is the nominal mean value of the process variable under observation (set to 80), σ is the standard deviation of the process variable (set to 5), a is the amplitude of cyclic variations in a cyclic pattern (set to 15 or less), g is the gradient of an increasing trend pattern or a decreasing trend pattern (set in the range 0.2 to 0.5), b indicates the shift position in an upward shift pattern and a downward shift pattern ($b = 0$ before the shift and $b = 1$ at the shift and thereafter), s is the magnitude of the shift (set between 7.5 and 20), $r_i(t)$ is a function that generates random numbers normally distributed between -3 and 3, t is the discrete time at which the monitored process variable is sampled (set within the range 0 to 20), T is the period of a cycle in a cyclic pattern (set between 4 and 12 sampling intervals) and $P(t)$ is the value of the sampled data point at time t .

are often referred to as the third generation of neural networks which have potential to solve problems related to biological stimuli. They derive their strength and interest from an accurate modeling of synaptic interactions between neurons, taking into account the time of spike emission. SNNs overcome the computational power of neural networks made of threshold or sigmoid units. Based on dynamic event-driven processing, they open up new horizons for developing models with an exponential capacity of memorizing and a strong ability to fast adaptation. Moreover, SNNs add a new dimension, the temporal axis, to the representation capacity and the processing abilities of neural networks (Meftah et al., 2008).

One of the key problems with spiking neural networks is the training algorithm. Much research relied on biologically inspired local learning rules, but these rules can only be implemented using unsupervised learning. However, in supervised learning SpikeProp algorithm operates on networks of spiking neurons, that use exact spike time temporal coding. This means that the exact spike time of input and output spikes encode the input and output values (Bohte et al., 2000). SpikeProp is an error-back propagation learning rule suited for supervised learning of spiking neurons that use exact spike time coding. SpikeProp assumes a special network topology. Globally the network looks like a classical feedforward network, but every connection consists of a fixed number of delayed synaptic terminals, different weights and different delays. However, the delays are fixed, and only the weights can be trained. Because the delayed synaptic terminals are fixed, this network topology has to be largely over-specified to make all possible weight/delay combinations possible (Natschlager and Ruf, 1998).

Enhancements to the SpikeProp algorithm such that, the delay and the time constant of every connection and the threshold of the neurons can be trained; because the delays can be trained, fewer synaptic terminals are necessary, effectively will reduce the number of weights and thus, the simulation time of the network.

SPIKING NEURAL NETWORKS ARCHITECTURE

Spiking neural networks (SNNs) have a similar architecture to traditional neural networks. Elements that differentiate in the architecture are the numbers of synaptic terminals between each layer of neurons and also there are synaptic delays. Several mathematical models have been proposed to describe the behavior of spiking neurons such as Leakey Integrate-and-Fire model (LIFN) (Mass, 1997) and Spike Response model (SRM) (Bialek et al., 1991). Figures 2(a) and (b) show the network structure proposed by Natschlager and Ruf (1998).

This structure consists of a feedforward fully connected spiking neural network with multiple delayed synaptic terminals. The different layers are labeled H,I and J for the input, hidden, and output layer respectively as shown in Figure 3. The adopted spiking neurons are based on the Spike Response Model (SRM) to describe the relationship between input spikes and the internal state variable. Consider a neuron J, having a set Dj of immediate pre-synaptic neurons, receives a set of spikes with firing times firing times $t_i, i \in D_j$. It is assumed that any neuron can generate at most one spike during the simulation interval and discharges when the internal state variable reaches a threshold. The dynamics of the internal state variable xj (t) are described by the following function:

$$x_j(t) = \sum_{i \in D_j} w_{ij} y_j(t) \qquad (1)$$

yj (t) is the un-weighted contribution of a single synaptic terminal to the state variable which described as a pre-synaptic spike at a synaptic terminal k as a PSP of standard height with delay d^k.

$$y_i^k = \varepsilon(t - t_i - d^k) \qquad (2)$$

The time t_i is the firing time of pre-synaptic neuron i, and d^k the delay associated with the synaptic terminal k. by considering the multiple synapses per a connection, the state variable $x_j(t)$ of neuron j is receiving inputs from all preceding neurons and then described as the weighted sum of the pre-synaptic contributions as follow:

$$x_j(t) = \sum_{i \in D_j} \sum_{k=1}^{m} w_{ij}^k y_i^k(t) \qquad (3)$$

The effect of the input spikes is described by the function ε(t) and so is called the spike response function, and $w_{ij}$is the weight describing the synaptic strengths. The spike response function ε(t) is modeled with the α- function, thus implementing a leaky-integrate-and-fire spiking neuron, is given by:

$$\varepsilon(t) = \frac{t}{\tau} e^{1 - \frac{t}{\tau}} \text{, for } t > 0, \text{ else } \varepsilon(t) = 0 \qquad (4)$$

Where τ is the time constant, which defines the rise and the decay time of the postsynaptic potential (PSP). The individual connection described in Jenn-Hwai and Miin-Shen (2005), consists of a fixed number of m synaptic terminals.

Each terminal serves as a sub-connection that is associated with a different delay and weight (Figure 3b). The delay d^k of a synaptic terminal k is defined as the difference between the firing time of the pre synaptic neuron and the time when the postsynaptic potential starts rising. The threshold θ is constant and equal for all neurons in the network.

An overview of neural coding schemes

In real biological systems, signals are encoded by information using specific coding methods. Basically, there are three different coding methods: rate coding, temporal coding, and population coding. Rate coding is the earliest neural coding method. The essential information is encoded in the firing rates and the rate is counted as a spike in an interval T divided by T (averaged over time). More recently, there has been growing recognition that the traditional view of mean firing encoding is often inadequate. Experiments on the visual system of a fly and studies of the middle temporal (MT) area of the monkey have indicated that the precise timing of spikes can be used to encode information. Such a scheme is called temporal coding (Bohte, 2002; Bohte et al., 2000). In temporal coding, the timing of single spikes is used to encode information. It is considered that, the timing of the first spike contains most of the relevant information needed for processing. Population coding is another coding scheme in which information is encoded in the activity of a given population of neurons firing within a small temporal window. This work adopts temporal coding as the code used by neurons to transmit information.

Spiking neural network for supervised learning procedure

Authors worked on supervised learning (Rumelhart et al., 1986) proposed a network of spiking neurons, that encodes information in the timing of individual spike times. They derived a supervised learning rule, SpikeProp, akin to traditional error back propagation. They utilized a fully connected feed-forward spiking neural network with layers labeled H(input), I (hidden) and J (output). Each connection between two neurons corresponds to multi sub-connections. Each sub-connection is characterized with a different delay and weight. They demonstrated how networks of spiking neurons with biologically reasonable action potentials can perform complex non-linear classification in fast temporal coding just as well as rate-coded networks and the resulting algorithm can be applied equally well to networks with many hidden layers. The target of the algorithm is to learn a set of target firing times, denoted at the output neurons j∈J for a given set of input patterns

$$\{P[t_1, t_2, \dots, t_h]\}$$

where defines a single input pattern described by single spike-times for each neuron h∈H . The least mean squares error is choosing as the error-function. Given the desired spike times $\{t_j^d\}$ and actual firing times $\{t_j^a\}$, this error-function is defined by:

$$E_v = \frac{1}{2} \sum_j \left(t_j^a - t_j^d\right)^2 \qquad (5)$$

The general form of the error derived by a connection's weight is:

$$\frac{\frac{\partial E}{\partial w_{ij}^k} = \frac{\frac{\partial E}{(\partial t_j)(t_j^a)(\partial t_j)}}{(\partial a_j(t))(t_j^a)(\partial a_j(t))}}{(\partial w_{ij}^k)(t_j^a)}$$

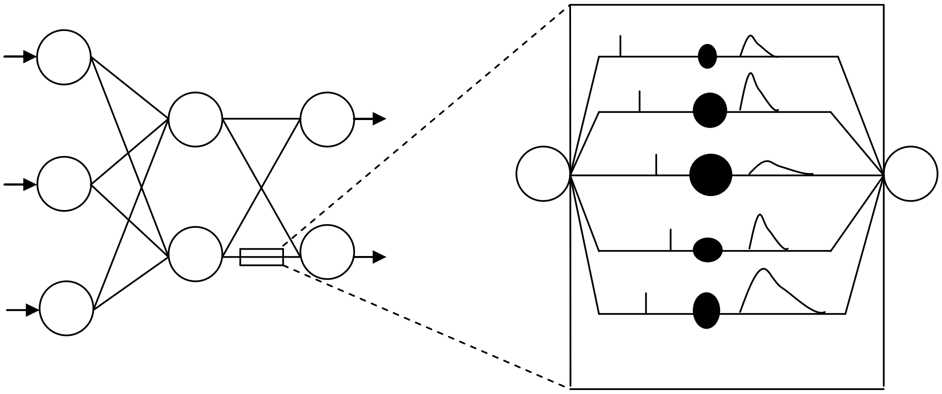

Figure 2. a) Feed forward spiking neural network b) Connection consisting of multiple synaptic terminals.

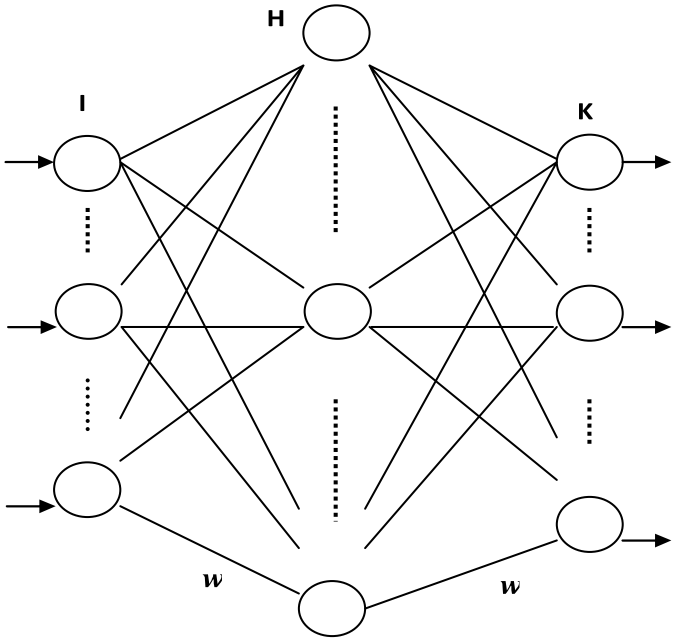

Figure 3. The MLP neural network structure for CCPR.

$$= \varepsilon_{ij}^{k}\left(t_j^a - t_i^a - d_{ij}^{k}\right)\delta_j \qquad (6)$$

Here δ_j is not the same for neurons in the output layer and neurons in the hidden layers. The set is used to represent all the direct pre-synaptic neurons of neuron j, while the set represents all the direct successors of neuron j. For a neuron in the output layer, j∈o , δ_j is equal to:

$$\delta_j = \frac{-\left(t_j^a - t_j^d\right)}{\sum_{i\in\Gamma_j}\sum_k w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_j^a - t_i^a - d_{ij}^k\right)}} \qquad (7)$$

For hidden neurons j∈H, is equal to:

$$\delta_j = \frac{\sum_{i\in\Gamma^j}\delta_i \sum_k w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_i^a - t_j^a - d_{ij}^k\right)}}{\sum_{i\in\Gamma_j}\sum_k w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_j^a - t_i^a - d_{ij}^k\right)}} \qquad (8)$$

Here, the error-backpropagation is clearly visible because is dependent on all ’s of the successors neuron j.

The adaptation rule for weights is

$$w_{ij}(T+1) = w_{ij}(T) + \Delta w_{ij}(T). \qquad (9)$$

Where T is the T^{th} step of algorithm (T^{th} is simulation interval), and

$$w_{new} = w_{old} + \Delta w_{ij} \qquad (10)$$

Table 2. Control chart patterns and neural network outputs.

		ANNs outputs					
		Node					
Pattern class	**Description**	**1**	**2**	**3**	**4**	**5**	**6**
1	Systematic	1	0	0	0	0	0
2	Cyclic	0	1	0	0	0	0
3	Increasing trend	0	0	1	0	0	0
4	Decreasing trend	0	0	0	1	0	0
5	Upward shift	0	0	0	0	1	0
6	Downward shift	0	0	0	0	0	1

$$\Delta w_{ij}(T) = -\eta \frac{\partial E}{(\partial w_{ij})(T)} \qquad (11)$$

The symbol η is the learning rate. The error is minimized by changing the weight according to the negative local gradient.

Enhancements to SpikeProp algorithm

The following enhancements have been proposed to provide additional learning rules for the synaptic delays, time constants and the neurons' thresholds.

Learning synaptic delays

The partial derivative of the error function to the synaptic delay d_{ij}^k is determined:

$$\frac{\partial E}{\partial d_{ij}^k} = \frac{\frac{\partial E}{(\partial t_j)(t_j^a)(\partial t_j)}}{\frac{(\partial a_j(t))(t_j^a)(\partial a_j(t))}{(\partial d_{ij}^k)(t_j^a)}} \qquad (12)$$

The first two terms are the same as for the weight update rule, only the last term is different:

$$\frac{\partial a_j(t)}{\partial d_{ij}^k}(t_j^a) = -w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_j^a - t_i^a - d_{ij}^k\right)} \qquad (13)$$

$$= -w_{ij}^k \varepsilon_{ij}^k \left(t_j^a - t_i^a - d_{ij}^k\right)\left[\frac{1}{t_j^a - t_i^a - d_{ij}^k} - \frac{1}{\tau_{ij}^k}\right]$$

By substitution and using the definition of in Equation 8 (which is different for output neurons and hidden neurons) we can get:

$$\frac{\partial E}{\partial d_{ij}^k} = -w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_j^a - t_i^a - d_{ij}^k\right)\delta_j} \qquad (14)$$

The final update rule for the delays is:

$$\Delta d_{ij}^k = -\eta_d \frac{\partial E}{\partial d_{ij}^k} \qquad (15)$$

Where η_d is the learning rate for the delays.

Learning synaptic time constants

The partial derivative of the error function to the time constant of the α-function

$$\frac{\partial E}{\partial \tau_{ij}^k} = \frac{\partial E}{\partial t_j}(t_j^a)\frac{\partial t_j}{\partial a_j(t)}(t_j^a)\frac{\partial a_j(t)}{(\partial \tau_{ij}^k)(t_j^a)} \qquad (16)$$

The third term can be written as:

$$\frac{\partial a_j(t)}{\partial \tau_{ij}^k}(t_j^a) = w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{(\partial \tau_{ij}^k)\left(t_j^a - t_i^a - d_{ij}^k\right)} \qquad (17)$$

$$= w_{ij}^k \varepsilon_{ij}^k \left(t_j^a - t_i^a - d_{ij}^k\right)\left[\frac{\left(t_j^a - t_i^a - d_{ij}^k\right)}{\left(\tau_{ij}^k\right)^2} - \frac{1}{\tau_{ij}^k}\right]$$

By substitution and using the definition of δ_j in Equation 8 we can get:

$$\frac{\partial E}{\partial \tau_{ij}^k} = w_{ij}^k \varepsilon_{ij}^k \left(t_j^a - t_i^a - d_{ij}^k\right)\left[\frac{\left(t_j^a - t_i^a - d_{ij}^k\right)}{\left(\tau_{ij}^k\right)^2} - \frac{1}{\tau_{ij}^k}\right]\delta_j \qquad (18)$$

The final update rule for the synaptic time constants becomes:

$$\Delta \tau_{ij}^k = -\eta_\tau \frac{\partial E}{\partial \tau_{ij}^k} \qquad (19)$$

Where η_τ is the learning rate for the synaptic time constants.

Learning neuron's threshold

Last we derive the error function for the neuron's threshold:

$$\frac{\partial E}{\partial \theta_j} = \frac{\partial E}{\partial t_j}(t_j^a)\frac{\partial t_j}{(\partial \theta_j)(t_j^a)} \qquad (20)$$

$$\frac{\partial t_j}{\partial \theta_j}(t_j^a) = \frac{1}{\frac{\partial a_j(t)}{\partial t\,(t_j^a)}} \qquad (21)$$

$$= \frac{1}{\sum_{i\epsilon\Gamma_j}\sum_k w_{ij}^k \frac{\partial \varepsilon_{ij}^k}{\partial t\left(t_j^a - t_i^a - d_{ij}^k\right)}}$$

Table 3. The network parameters for CCPR.

Parameters	Values
Number of input neurons	20
Number of hidden neurons	6
Number of output neurons	6
Threshold θ	0.3
Goal error	0.001
Coding interval ΔT	0-20 ms
Synaptic time constant τ	170 ms
Number of synaptic terminals k	3
Learning rate for weight updating η_w	0.0075
Learning rate for delay updating η_d	0.0065
Learning rate for synaptic time constant updating η_τ	0.0055
Learning rate for neuron threshold updating η_θ	0.0035

Table 5. Targeted recognizer based on weight update with ANNs.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	Description	1	2	3	4	5	6
1	Systematic	86	2	0	0	0	12
2	Cyclic	1	84	0	3	12	0
3	Increasing trend	0	0	87	0	13	0
4	Decreasing trend	0	0	10	88.1	0	1.9
5	Upward shift	12	0	0.8	0	87.2	0
6	Downward shift	0	0	10	1.7	0	88.3

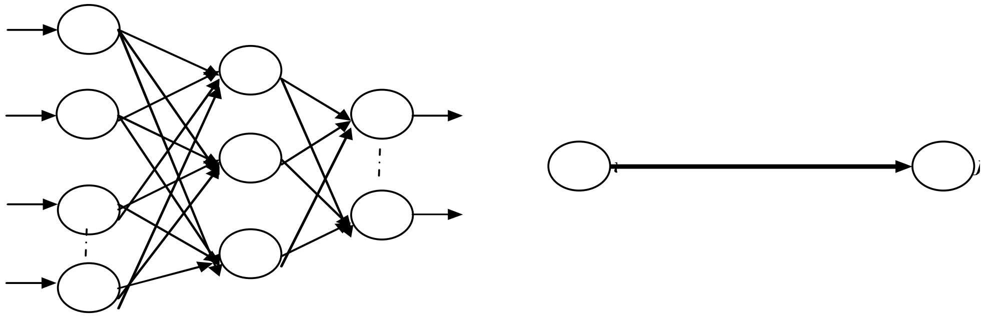

Figure 4. The proposed SNNs for CCPR with single synaptic termanial.

By adding the first term, we can get the negative of δ_j as learning rule:

$$\frac{\partial E}{\partial \theta_j} = -\delta_j \quad (22)$$

The final update rule for the neuron threshold is:

$$\Delta\theta_j = -\eta_\theta \frac{\partial E}{\partial \theta_j} \quad (23)$$

Where η_θ is the learning rate for the neuron's threshold.

ANN-Based CCPR schemes

First, an artificial neural network has been developed for control chart pattern recognition for comparison with the spiking neural network. A multilayer perceptions (MLPs) architecture comprises an input layer with 20 neurons, one hidden layer with 6 neurons and an output layer with 6 neurons, one for each patterns of CCPs is used, as shown in Figure 3. Table 2 depicts the control chart patterns and representation of the desired neural network outputs.

Table 6. Targeted recognizer based on weight update with SNNs single synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	Description	1	2	3	4	5	6
1	Systematic	90	2	0	0	0	8
2	Cyclic	1	89	0	0	10	0
3	Increasing trend	0	0	92.5	0.5	7	0
4	Decreasing trend	0	0	2.9	91.1	0	6
5	Upward shift	9	0	0.8	0	90.2	0
6	Downward shift	0	0	7	0	0	93

Table 7. Targeted recognizer based on synaptic delay update with SNNs single synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	Description	1	2	3	4	5	6
1	Systematic	92	0.8	0	0	0	7
2	Cyclic	3	90	0	0	7	0
3	Increasing trend	0	0	94.5	0.5	5	0
4	Decreasing trend	0	0	0.8	93.2	0	6
5	Upward shift	9	0	0	0	91	0
6	Downward shift	0	0	6	0	0	94

Sample patterns

Sample patterns should be collected from a real manufacturing process. Since, a large number of patterns are required for developing and validating a CCP recognizer, and as those are not economically available, simulated data are often used. Since a large window size can decrease the recognition efficiency by increasing the time required to detect the patterns, an observation window with 20 data points is considered here. A set of 720 (120 x 6) sample patterns are generated from 120 series of standard normal variants. It may be noted that each set contains equal number of samples for each pattern class. The equations used for simulating the six CCPs. Table 3 shows the network parameters that were used in our simulations for two proposed networks.

SNN-Based CCPR schemes

SNN with single connection and multi-synaptic terminals have been developed, as shown in Figures 4 and 5, respectively. Table 4 depicts the control chart patterns and representation of the desired spiking neural network outputs.

The proposed SNN network architecture

Here, we proposed a new architecture for spiking neural network in application to control charts. The proposed architecture consists of a fully connected feed-forward network of spiking neurons. An individual connection consists of a fixed number of synaptic terminals, where each terminal serves as a sub-connection that is associated with a different delay and weight between the input, hidden and output layers. The weights of the synaptic terminals are set randomly between -1 to +1. The network adopted 20 input neurons in the input layer which mean that, the input patterns consisted of the 20 most recent mean values of the process variable to be controlled, therefore, one input neuron was dedicated for each mean value, six output neurons with one for each pattern category, and six hidden neurons where the numbers of the hidden neurons here adopted on the number of classes.

RESULTS AND DISCUSSION

CCPR using artificial neural networks

Table 5 shows the obtained results of control chart pattern recognition based on artificial neural network. It was noted during training that ANN-based recognizers were more easily trained. This table shows that 86.76% of the patterns were correctly recognized.

CCPR using single synaptic terminals SNN

Synaptic weights update

First, the synaptic weights only are updated while the other parameters are fixed. The achieved results are presented in Table 6. It is obviously that there is an improvement in the performance accuracy for all recognized patterns compared with ANN, the performance accuracy is increased to 90.9%.

Updating of the synaptic delay

Again, synaptic delays are updated while the other parameters are fixed and the values of the synaptic weights are optimum weights w_opt obtained from the

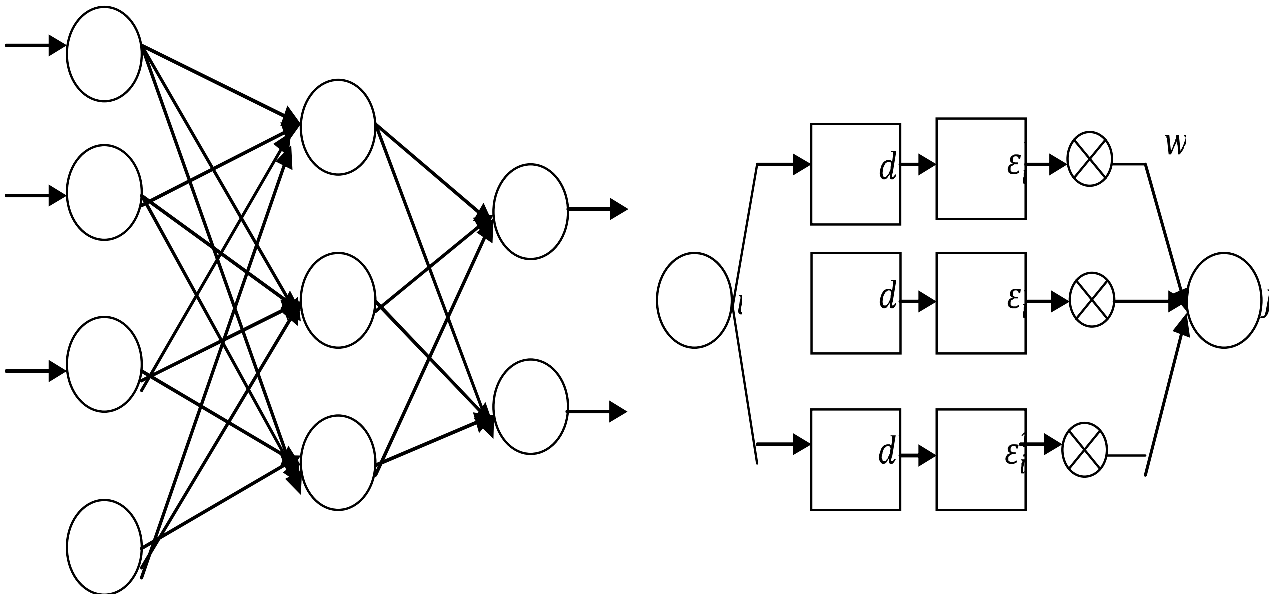

Figure 5. The proposed SNNs for CCPR with multi synaptic termanial.

Table 8. Targeted recognizer based on synaptic time constant update with SNNs single synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	**Description**	**1**	**2**	**3**	**4**	**5**	**6**
1	Systematic	94	0.8	0	0	0	5.2
2	Cyclic	2	91	0	0	7	0
3	Increasing trend	0	0	94.8	0.2	5	0
4	Decreasing trend	0	0	0.7	94.3	0	5
5	Upward shift	7	0	0.7	0	92.3	0
6	Downward shift	0	0	5.4	0	0	94.6

Table 9. Targeted recognizer based on neurons threshold update with SNNs single synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	**Description**	**1**	**2**	**3**	**4**	**5**	**6**
1	Systematic	95	0.8	0	0	0	4.2
2	Cyclic	3	91.4	0	0	5.6	0
3	Increasing trend	0	0	9	0.2	4.8	0
4	Decreasing trend	0	0	0.8	95.2	0	4
5	Upward shift	7	0	0.2	0	92.8	0
6	Downward shift	0	0	5	0	0	95

previous learning part. Table 7 shows the obtained results based on the adaptation of the synaptic delay. The achieved results show that 92.4% of the patterns were correctly recognized. It is obviously that there is an increment in the performance accuracy for all recognized patterns; the performance accuracy is getting better,

Table 10. Targeted recognizer based on weight update with SNNs multi synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	**Description**	**1**	**2**	**3**	**4**	**5**	**6**
1	Systematic	97	0.8	0	0	0	2.2
2	Cyclic	3	93	0	0	4	0
3	Increasing trend	0	0	96	4	0	0
4	Decreasing trend	0	0	0.8	96.2	0	3
5	Upward shift	4	0	0.6	0	95.4	0
6	Downward shift	3	0	0.5	0	0	96.5

Table 11. Targeted recognizer based on synaptic delay update with SNNs multi synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	**Description**	**1**	**2**	**3**	**4**	**5**	**6**
1	Systematic	98	0.6	0	0	0	1.4
2	Cyclic	3	95	0	0	5	0
3	Increasing trend	0	0	96.4	3.6	0	0
4	Decreasing trend	0	0	0.2	96.8	0	3
5	Upward shift	3	0	0.6	0	96	0.4
6	Downward shift	3	0	0.1	0	0	96.9

increased to 92.4%.

Synaptic time constant update

The synaptic time constant is updated only while other parameters are fixed, synaptic weights are the optimum weights w_opt and the synaptic delay is the optimum synaptic delay d_opt obtained in previous steps. Table 8 shows the obtained results based on the adaptation of the synaptic time constant.

Again, the performance accuracy for all recognized patterns is improved; the performance accuracy is increased to 93.5%.

Neuron threshold update

The neuron's threshold is updated while the other parameters are fixed. Synaptic weights, synaptic delay, and synaptic time constant are the optimum values previously obtained; the obtained results are shown in Table 9. Also there is a remarkably improvement especially for downward shift pattern. From Table 9, we can observe that the recognition rate is still increasing with neuron threshold updating about 94.06%.

SNNs for CCPR with multi synaptic terminals

The existence of multiple synapses is biologically plausible, since in brain areas like the neocortex a single pre-synaptic axon makes several independent contacts with the pot-synaptic neuron. Instead of a single synapse, with its specific delay and weight, this synapse model consists of many sub-synapses, each one of them has its own weight and delay as shown in Figure 2b. The use of multiple synapses enables an adequate delay selection using the learning rule. In this proposed SNN, multiple synapses per a connection are used. The network topology is feed-forward SNN with multiple synaptic (seven sub-connections per synaptic terminal, k=3) terminals per connection with different weights and delays.

Again the same procedure conducted in the SSN with single synaptic connection is repeated with the same chosen parameters.

Synaptic weights update

The obtained results with the proposed architecture and the SpikeProp learning procedure for control chart pattern recognition are presented in Table 10. It is obviously that there better performance accuracy for all recognized patterns. The results from Table 10 shows that, the accuracy of the network is still increasing from 94.06 to 95%.

Updating of the synaptic delay

Again Table 11 shows the obtained results based on the

Table 12. Targeted recognizer based on synaptic time constant update with SNNs multi synaptic.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	Description	1	2	3	4	5	6
1	Systematic	98	0.6	0	0	0	1.4
2	Cyclic	3	96	0	0	4	0
3	Increasing trend	0	0	97	3	0	0
4	Decreasing trend	0	0	0	98	0	2
5	Upward shift	3	0	0.5	0	96.5	0
6	Downward shift	3	0	0	0	0	97

Table 13. Targeted recognizer outputs based on neuron's threshold update.

		Targeted recognizers outputs (%)					
		Node					
Pattern class	Description	1	2	3	4	5	6
1	Systematic	99	0.6	0	0	0	1
2	Cyclic	2	98.2	0	0	0.8	0
3	Increasing trend	0	0	98	0	0	2
4	Decreasing trend	0	0	0	98.5	0	1.5
5	Upward shift	0	0	1	0	99	0
6	Downward shift	0.4	0	0	0.6	0	99

adaptation of the synaptic delay. From the results shown in Table 11, there is an increment in the performance accuracy for all recognized patterns. The network performance still increases.

Synaptic time constant update

Table 12 shows the obtained results based on the adaptation of the synaptic time constant. The obtained results show that, there is an improvement in the performance accuracy.

Neuron threshold update

The obtained results are shown in Table 13. Also, there is an improvement especially for downward shift pattern. Generally, the results of the overall percentages of correct recognition of random patterns in Tables 10 (95.68%), 11 (96.51%), 12 (97.08%) and Table 13 (98.61%) suggest that, there are better performance accuracy for all recognized patterns with the proposed architecture of SNNs with multi-connection compared with SNN-based recognizers with single connection. Table 14, shows the comparison between three networks topologies for performance accuracy of six unnatural patterns.

Furthermore, Table 15 shows the comparison between spiking neural networks based on LVQ algorithm and the work presented in this paper for control chart pattern recognition. From Table 15, it can be summarized that the SNNs with three synaptic terminals achieve better performance in the control chart pattern recognition.

Conclusion

In this paper, a spiking neural network architecture is developed and used for control charts pattern recognition (CCPR). It has a good capability in data smoothing and generalization. The overall mean percentages of correct recognition of SNN-based recognizers were 98.61%. This shows clearly that, the superior

Table 14. Recognition performance comparison between ANN and SNN recognizers.

Patterns	Percentage correction recognition								
	ANNs (%)	SNNs single connection				SNNs Multi Connections			
		Weight (%)	Delay (%)	Time constant (%)	Neuron threshold (%)	Weight (%)	Delay (%)	Time constant (%)	Neuron threshold (%)
Systematic	86	90	92	94	95	97	98	98	99
Cyclic	84	89	90	91	91.4	93	95	96	98.2
Inc. Trend	87	92.5	94.5	94.8	95	96	96.4	97	98
De. Trend	88.1	91.1	93.2	94.3	95.2	96.2	96.8	98	98.5
Up. Shift	87.2	90.2	91	92.3	92.8	95.4	96	96.5	99
Dow. Shift	88.3	93	94	94.6	95	96.5	96.9	97	99

Table 15. Results of four different pattern recognizers applied to control chart data set.

Pattern recognizers	No. of training epochs	Learning performance (%)	Test performance (%)
SNN-based LVQ	20	100	97.70
ANN-based CCPR	100	98	86.76
SNN-based CCPR Single Connection	150	99	94.06
SNN-based CCPR Multi Connection	200	100	98.61

performance of the spiking neural networks technique in an application to control chart data over the other procedures using traditional neural network. Furthermore, enhancements to the SpikeProp learning algorithm are presented. These enhancements provide additional learning rules for the synaptic delays, time constants and for the neurons thresholds. Experiments have been conducted and the achieved results show a remarkable improvement in the overall performance. This work can also be extended to investigate online learning and address the effect of costs on the decisions in terms of computational time and complexity.

REFERENCES

Bialek W, Rieke F, Steveninck R, Warland D (1991). Reading a neural code. Science, 252(5014): 1854-1857.

Bohte M (2002). Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks. IEEE Transaction on Neural Networks, 13(2): 426-435.

Bohte M, la Poutre H, Kok JN (2002). SpikeProp: Error-Backpropagation for networks of spiking neurons. ESANN'2000, Bruges (Belgium), pp. 419-425.

Bohte SM, Poutre HL, Kok JN (2002). Error-backpropagation in temporally encoded networks of spiking neurons, NeuroComputting, 48: 17-37.

Hui-Ping C, Chuen-Sheng C (2009). Control Chart Pattern Recognition Using Wavelet Analysis and Neural Networks, J. Qlty., 16(5): 311-319.

Ibrahim M, Adnan H (2010). Issues in Development of Artificial Neural Network- Based Control Chart Pattern Recognition Schemes, Eur. J. Sci. Res., 39(3): 336-355.

Indra K, Pramila DM, Lakshmi G (2010). Effective Control Chart Pattern Recognition Using Artificial Neural Networks, IJCSNS Int. J. Compt. Sci. Network Security, 10(3): 194-198.

Jenn-Hwai Y, Miin-Shen Y (2005). A Control Chart Pattern Mass W (1997). Networks of spiking neurons: the third generation of neural network models, Neural Networks, 10(9): 1659-1671.

Meftah B, Benyettou A, Lezoray O, Debakla M (2008). Image

Natschlager T, Ruf B (1998). Spatial and Temporal Pattern analysis Via Spiking Neurons, Network: Computation in Neural Systems. 9(3): 319-332.

Recognition System Using a Statistical Correlation Coefficient method, Comput. Ind. Eng., 48: 205-221.

Rumelhart DE, Hinton GE, Williams RJ (1986). Learning representations by back-propagation errors, Nature, 323: 533-536.

Samir BB (2009). Fast and Accuracy Control Chart Pattern Recognition Using a New Cluster-K-Nearest Neighbor, World Academy of Science, Engineering and Technology, 49: 1022-1025.

Segmentation with Spiking Neuron Network, 1st Mediterranean Conference on Intelligent Systems and Automation, pp. 15-18.