Computers & Industrial Engineering 66 (2013) 683–695

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines ☆

Shichang Du [a],*, Delin Huang [a], Jun Lv [b]

[a] *Department of Industrial Engineering and Logistics Management, School of Mechanical Engineering, Shanghai Jiaotong University, 800 Dongchuan Road, Shanghai 200240, PR China*
[b] *School of Business, East China Normal University, Shanghai 200241, PR China*

ARTICLE INFO

Article history:
Received 24 May 2011
Received in revised form 13 September 2013
Accepted 14 September 2013
Available online 21 September 2013

Keywords:
Concurrent control chart pattern
Support vector machine
Wavelet transform
Statistical process control

ABSTRACT

Statistical process control charts have been widely utilized for monitoring process variation in many applications. Nonrandom patterns exhibited by control charts imply certain potential assignable causes that may deteriorate the process performance. Though some effective approaches to recognition of control chart patterns (CCPs) have been developed, most of them only focus on recognition and analysis of single patterns. A hybrid approach by integrating wavelet transform and improved particle swarm optimization-based support vector machine (P-SVM) for on-line recognition of concurrent CCPs is developed in this paper. A statistical correlation coefficient is used to determine whether the input pattern is a single or concurrent CCP. Based on wavelet transform, a raw concurrent pattern signal is decomposed into two basic pattern signals, which can be recognized by multiclass SVMs. The performance of the hybrid approach is evaluated by simulation experiments, and numerical and graphical results are provided to demonstrate that the proposed approach can perform effectively and efficiently in on-line CCP recognition task.

1. Introduction

The ability to reduce process variation for quality improvement in manufacturing process plays an essential role in the success of a manufacturing enterprise in today's globally competitive marketplace (Du, Xi, Ni, Pan, & Liu, 2008; Montgomery, 2001). Statistical process control (SPC) has been widely used for monitoring process variation in many applications. Control charts are the most important tools in SPC to determine whether a process is behaving as intended or some unnatural causes of process variation exist. In the implementation of control charts, the process is considered out of control when one or more points fall outside the control limits or the control charts display nonrandom control chart patterns (CCPs). While the former condition can be easily identified, the latter is difficult to be recognized efficiently since nonrandom CCPs are normally distorted by "common cause variation" that occurs naturally in the processes. There are generally seven basic CCPs commonly exhibited by control charts (see Fig. 1), including upward shift (u.s.), downward shift (d.s.), increasing trend (i.t.), decreasing trend (d.t.), cyclic pattern (c.p.), systematic pattern (s.p.), and normal pattern (n.p.). The six unnatural patterns exhibited by control charts often contain valuable information closely relevant to process parameters and process changes and are associated with a specific set of assignable causes. Therefore, effective control chart pattern recognition (CCPR) is a critical task in SPC for determining potential assignable causes.

Traditionally, CCPs have been analyzed and interpreted manually. Many supplementary rules, like zone tests or run rules (Duncan, 1986; Grant & Leavenworth, 1996; Nelson, 1984; Western Electric Company., 1958) have been suggested to assist quality control engineers to detect unnatural CCPs. Nevertheless, the application of all these rules can result in excessive number of false alarms. Moreover, several attempts have also been made to develop expert systems (ESs) for CCPR (Cheng & Hubele, 1992; Lucy-Bouler, 1991; Swift & Mize, 1995). Although the results are promising, a common problem as reported in these studies is high rate of false recognition (Guh, 2005).

The utility of artificial neural networks (ANNs) as an effective tool for CCPR has been demonstrated by a number of researchers. Encouraging results from a comparative study between ANNs and conventional control charts reported by Pugh (1989) attracted further investigation on ANN application to CCPR. Al-Ghanim (1997), Guh (2003), Guh, Zoriassatine, Tannock, and O'Brien (1999), Guh and Hsieh (1999), Guh and Shiue (2005), Hwarng (1995), Hwarng and Chong (1995), and Pham and Oztemel (1994) addressed some important issues for implementation of ANNs in CCPR scheme. For improving the recognition performance of ANNs, feature-based and wavelet-de-noise input representation

☆ The manuscript was processed by the area editor H.-S. Jacob Tsao.
* Corresponding author. Tel.: +86 13764479720.
E-mail address: lovbin@sjtu.edu.cn (S. Du).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.09.012

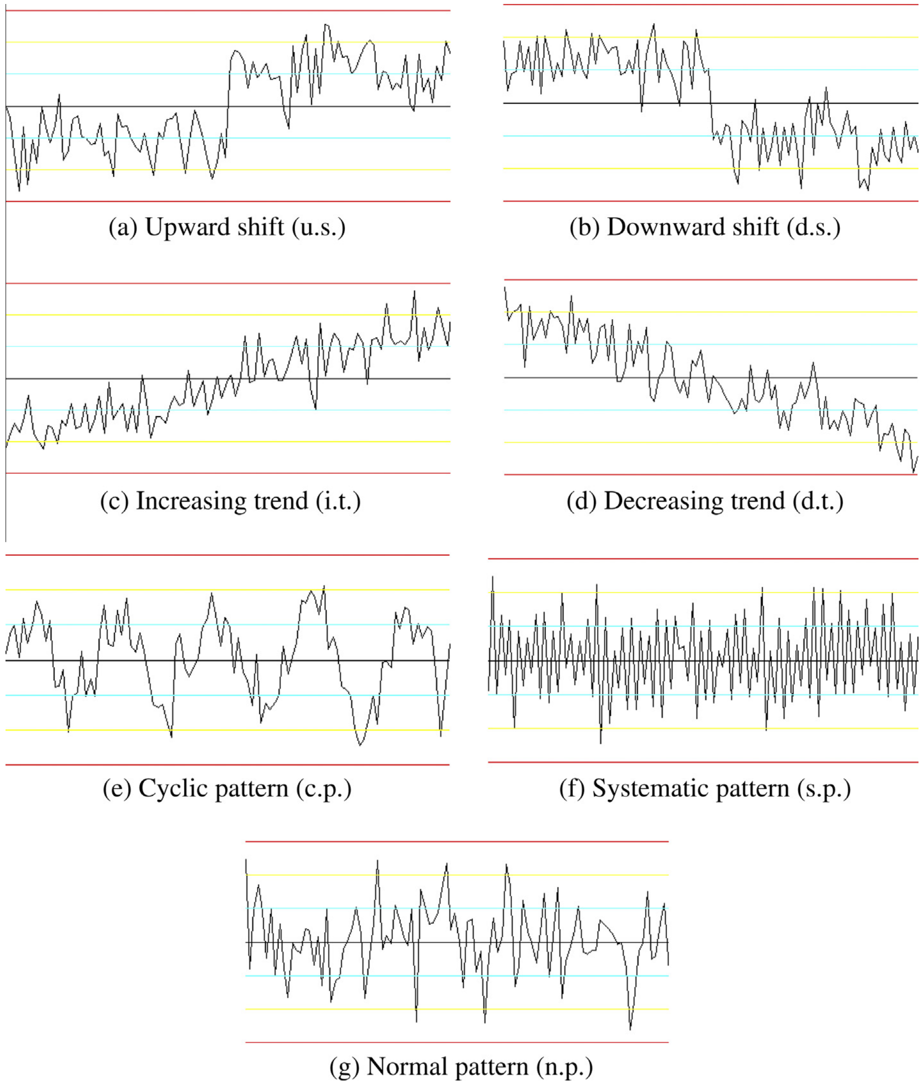

Fig. 1. Seven basic CCPs in control charts.

techniques also have been investigated. The most significant works included wavelet-ANN (Al-assaf, 2004, 2005; Assaleh & Al-Assaf, 2005; Wang, Kuo, & Qi, 2007), shape features-ANN (Gauri & Chakraborty, 2006, 2008, 2009; Pham & Wani, 1997), and statistical features-ANN (Hassan, Shariff, Shaharoun, & Jamaluddin, 2003). El-Midanya, El-Bazb, and Abd-Elwahedc (2010) proposed a framework for multivariate CCPR using ANNs. Several other ANN-based approaches have been developed and applied in intelligent monitoring and diagnosis (Chen & Wang, 2004; Cook & Chiu, 1998. Du & Xi, 2011; Niaki & Abbasi, 2005) and have been demonstrated to have higher performance in comparison to traditional SPC techniques.

However, most of the existing works are concerned with the recognition of nonrandom CCPs composed of a single abnormal variation. In real applications, concurrent patterns where two single patterns may exist together are frequently encountered; they may be associated with different assignable causes. Concurrent patterns are more difficult to recognize than single patterns due to pattern interaction and the resultant complexity. When a concurrent pattern occurs, most of existing approaches will classify the input pattern into one of predefined single CCPs. Unfortunately, any single CCP cannot represent the characteristics of a concurrent pattern, and further determination of all assignable causes becomes impossible. As a consequence, a pattern recognizer with the capability of recognizing concurrent patterns is desirable.

Limited works have been done to recognize concurrent CCPs. Yang and Yang (2005) presented one CCPR system for recognizing concurrent CCPs using a statistical correlation coefficient method, which was validated to be an effective method without a tedious training process. Wang, Dong, and Kuo (2009) developed a hybrid approach integrating independent component analysis (ICA) and decision tree (DT) to identify concurrent CCPs. Al-Assaf (2005), Chen, Lu, and Lam (2007), Guh and Tannock (1999), and Wang et al. (2007) recognized concurrent CCPs using ANNs approach. Their results indicated that ANNs approach for recognizing of concurrent CCPs can perform well. These approaches are capable of producing fast and accurate classification results and have been demonstrated to be among the most robust designs. Hachicha and Ghorbel (2012) presented an excellent survey of control-chart pattern-recognition literature (1991–2010).

Recently, the concept of support vector machine (SVM) has been used to develop a new generation of learning systems based on recent advances on statistical learning theory for solving a variety of learning, classification and prediction problems. SVM calculates a separating hyperplane that maximizes the margin between data classes to produce good generalization abilities. The main difference between ANN and SVM is in their risk minimization (Gunn, 1998). In case of SVM, structural risk minimization principle is used to minimize an upper bound based on an expected risk, whereas in ANN, traditional empirical risk minimization is used to minimize the error in the training of data. The difference in risk minimization leads to a better generalization performance for SVMs than ANN (Gunn, 1998). Although some applications of SVM to monitor and classify process variation (Cheng & Cheng, 2008; Chinnam, 2002; Sun & Tsung 2003) or to recognize single CCPs (Ranaee, Ebrahimzadeha, & Ghaderia, 2010) have been reported, the applications of SVM in concurrent CCPR are sparse.

Different from the previous approaches, one novel hybrid approach for on-line recognition of concurrent CCPs is explored by integrating wavelet transform with improved particle swarm optimization-based support vector machines (P-SVMs). Wavelet transform is used to decompose concurrent CCPs into single nonrandom patterns. Consequently, multi-class SVMs are constructed for each binary classification to recognize the nonrandom patterns. A particle swarm optimization (PSO) algorithm is utilized for optimizing the kernel function parameters of the SVM recognizer.

The rest of this paper is organized as follows. The methodology for recognizing concurrent CCPs is presented in Section 2. More precisely, a statistical correlation coefficient approach to determining the anomaly type of the input pattern is utilized; Haar wavelet transform used to decompose an input concurrent pattern into two single patterns is presented; and the multi-class SVM recognizer for recognizing multiple binary patterns is developed. This is followed by simulation and results analysis in Section 3 and conclusions in Section 4.

2. Methodology

This section provides a practical overview with respect to online automatic CCPR system using wavelet transform and multi-class SVMs. Fig. 2 shows the schematic diagram representing the procedure of the CCPR, in which four modules are in series: module I, module II, module III, and module IV.

In module I, a similarity measure between the input pattern and a reference set, which is composed of single CCPs, is utilized to determine whether the input pattern is single or concurrent. If the input pattern is concurrent, the concurrent CCP is first decomposed into binary patterns in module II, which can be recognized by SVMs. Many methods have been developed for patterns decomposition, among which wavelet transform is one of the most widely used and well-established methods. Wavelet transform is a method that allows the signal to be viewed in multiple resolutions with each resolution representing a different frequency, and it can still keep time information. Therefore, wavelet transform allows complex information such as images and patterns to be decomposed into elementary components at different positions (time domain) and scales (frequency domain). Among many wavelet transforms, the Haar transform is a member of a class of non-sinusoidal orthogonal functions. It is a very powerful tool in signal processing and pattern decomposition and has been applied in various science and engineering fields with great success. Moreover, Haar transform is also one of the most efficient transformation algorithms in terms of computational speed and memory usage (Burrus, Gopinath, & Guo, 1998) and thus ideal for implementation as an on-line monitoring tool. Therefore, Haar transform is used in this study for decomposing concurrent patterns into single patterns.

Module III develops a multi-class SVM recognizer for recognizing the single patterns, and Module IV implements it. In order to achieve satisfactory recognition performance, the multi-class SVM recognizer has to be properly designed, trained and tested.

2.1. Similarity measure

Similarity measure needs to be developed to determine whether the input pattern is a single or concurrent pattern. Al-Ghanim and Ludeman (1997) developed a matched-filter approach using a correlation-analysis technique based on an inner product of pattern vector. Their inner-product correlation technique produced good results in recognizing trend, cycle and systematic patterns. However, this approach was poor in recognizing shift patterns. In order to improve the performance of this approach of Al-Ghanim and Ludeman (1997) and Yang and Yang (2005) presented a statistical-correlation-coefficient approach to determine the anomaly type of an input pattern based on Eq. (1).

$$corr = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} \tag{1}$$

where $x_i(y_i)$, $\bar{x}(\bar{y})$ represent the input (reference) vector and its mean.The correlation coefficient approach is a quite simple mechanism and shows good performance for determining the anomaly type of the input pattern. Therefore, the statistical correlation coefficient is also utilized as the similarity measure between the input pattern and various reference prototypes in module I. For more details, the readers are referred to Yang and Yang (2005).

2.2. Patterns decomposition using Haar wavelet transform

If the input pattern is a concurrent pattern, Haar wavelet transform method is used to decompose the input pattern into two single patterns. Haar wavelet transform is a member of a class of non-sinusoidal orthogonal functions. It consists of rectangular waves distinguished by time scaling and time shifts. The set of continuous Haar functions $\{\Phi(n,m,x)\}$is periodic, orthogonal, and complete. The sequence $\{\Phi(n,m,x)\}$is defined on the open interval $[0,1]$as:

$$\Phi(0,0,x) = 1,\ x \in [0,1] \tag{2}$$

$$\Phi(n,m,x) = \begin{cases} 2^{n/2}, & \frac{m-1}{2^n} \leqslant x < \frac{m-1/2}{2^n} \\ -2^{n/2}, & \frac{m-1/2}{2^n} \leqslant x < \frac{m}{2^n} \\ 0, & elsewhere\ \forall x \in [0,1] \end{cases} \tag{3}$$

where n is the scale of the Haar wavelet transform and $1 \leqslant m \leqslant 2^n$ for $n \geqslant 1$.

Fig. 3 shows the first eight continuous Haar functions. Points of discontinuity are defined as the average of the approached limits from both sides of the discontinuity.

The Haar functions form a complete orthonormal basis of $L^2[0,1]$. The space of functions $f(x)$ that are defined over the interval $[0,1]$ can be approximated by a partial sum $S_N(x)$ containing 2^N terms, i.e.,

$$S_N(x) = C_0^0 + \sum_{n=1}^{N}\sum_{m=1}^{2^{n-1}}\Phi(n,m,x)C_n^m \tag{4}$$

where C_n^m is called the mth Haar coefficient in scale n.

In other words, in the expansion of $f(x)$, $S_N(x)$ is a step function with 2^N steps, and the value of $S_N(x)$ at each step equals to the mean value of $f(x)$ within the step interval. It can be shown that $S_N(x)$ is the best approximation of $f(x)$ in the mean-square-error sense.

The corresponding discrete Haar functions often used in practice can be obtained by sampling continuous Haar functions in Fig. 3 at the middle of each subinterval to produce a Haar array:

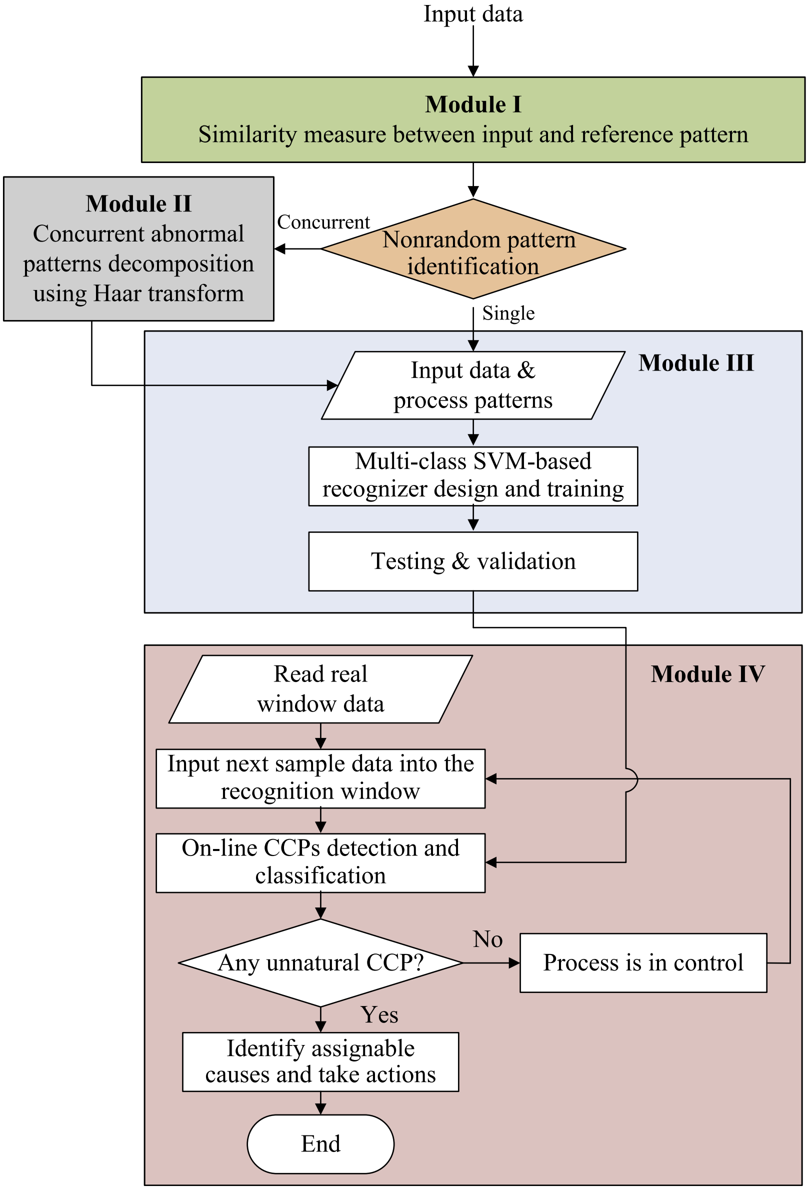
Input data
Module I
Similarity measure between input and reference pattern
Module II
Concurrent abnormal patterns decomposition using Haar transform
Concurrent
Nonrandom pattern identification
Single
Module III
Input data & process patterns
Multi-class SVM-based recognizer design and training
Testing & validation
Module IV
Read real window data
Input next sample data into the recognition window
On-line CCPs detection and classification
Any unnatural CCP?
No
Process is in control
Yes
Identify assignable causes and take actions
End

Fig. 2. Schematic diagram of the recognition system of CCPs.

$$H=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & \sqrt{2} & \sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

The Haar array is denoted by $\mathbf{H}(N,p)$ where N is the selected scale number, and p is the smallest integer such that the number of observations does not exceed 2^p. Each row of $\mathbf{H}(N,p)$ is a discrete Haar function obtained by sampling the continuous Haar function $\{\Phi(n,m,x)\}$ shown in Eqs. (2) and (3). Let $h_{i,j}$ represent the ith row of $\mathbf{H}(N,p)$, which is generated as:

$$h_{i,j} = 2^{-p/2}\Phi(n,m,x)(j\times 2^{-p}),\quad 1\leqslant j\leqslant 2^p \tag{5}$$

where $0\leqslant n\leqslant N$, N is the selected scale number, and

$$i=\begin{cases}1 & \text{when } n=0\\ 2^{n-1}+m & \text{when } n\geqslant 1\end{cases} \tag{6}$$

The multiplier $2^{-p/2}$ is used to normalize each row of $\mathbf{H}(N,p)$. Each row of matrix $\mathbf{H}(N,p)$ is orthogonal to each other.

If $\mathbf{x}=[x_1\quad x_2\quad \ldots\quad x_{2^p}]^T$ is a sequence of 2^p discrete observed points, then the discrete Haar coefficients, $c=\left[c_0^0, c_1^1, c_2^1, c_2^2, c_3^1, \ldots, c_N^{2^N-1}\right]^T$, can be calculated as follows

$$\mathbf{c}=\mathbf{H}(N,p)\mathbf{x} \tag{7}$$

The approximation of $\mathbf{x}$ using Haar transform, $\hat{\mathbf{x}}$, is given as follows:

$$\hat{\mathbf{x}}=\mathbf{H}^T(N,p)\mathbf{c} \tag{8}$$

The Haar coefficients have a relationship with mean values of the original signal over some intervals, as shown in Eqs. (9) and (10).

For $n = 0$,

$$C_0^0 = 2^{p/2}\bar{f}(1,2^p) \tag{9}$$

For $n\geqslant 1$,

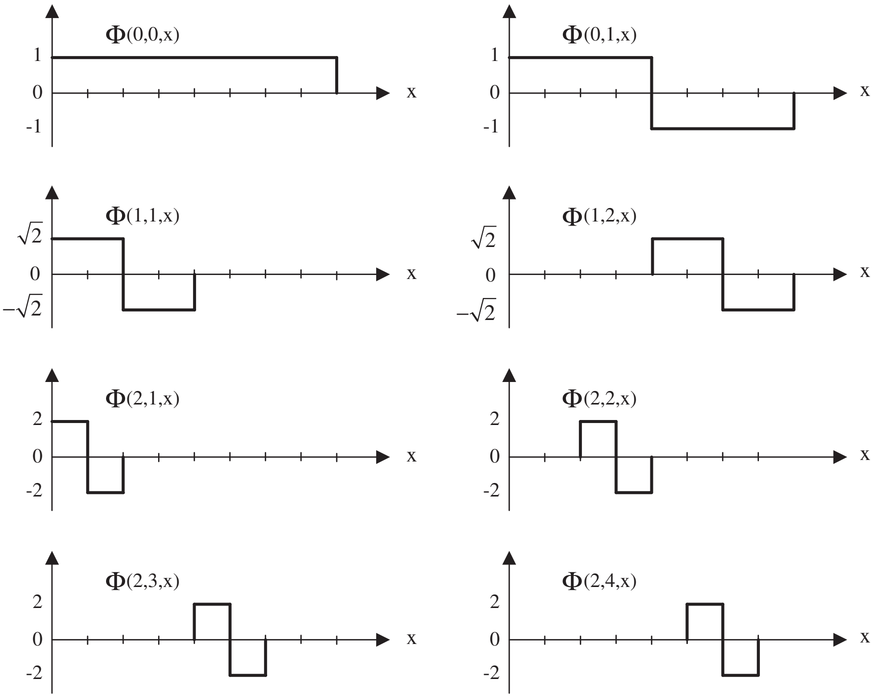

Fig. 3. Continuous Haar transform for $n = 8$ subintervals.

$$C_n^m = 2^{(p-n-1)/2}\left[\bar{f}\left\{(m-1)2^{(p-n+1)}+1,(m-1/2)2^{(p-n+1)}\right\} -\bar{f}\left\{(m-1/2)2^{(p-n+1)}+1,m2^{(p-n+1)}\right\}\right] \quad (10)$$

where $\bar{f}(i,j) = \frac{1}{j-i+1}\sum_{k=i}^{j} x_k$. Thus, C_0^0 is proportional to the mean of $f(x)$. Every C_n^m, except C_0^0, is proportional to the mean difference in two adjacent intervals.

At each level of the Haar transform, the signal data is processed through a low-pass filter and a high-pass filter. The high-pass filtered data represents the details of the signal, whereas the result of low-pass transform is called approximations, consisting of high-scale/low frequency components of the signal, which will be further used as input data to compute the next level of splitting, iteratively. The Haar wavelet transform is an octave-band representation for signals so that if a signal consists of N observations such that $N = 2^p$, then Haar wavelet results in the decomposition of the signal into p resolution levels. The approximation and detail coefficients at all resolutions are obtained by convolving the signal with a pair of quadrature mirror filters (QMF) at different scales (Burrus et al., 1998). The filters are composed of a high pass filter and a low filter that are used to obtain, respectively, the detail and approximation components of the function. Fig. 4 shows an example of a five-level wavelet decomposition of a discrete-time signal. The first level of decomposition separates the signal into a high-pass and a low-pass component according to the QMF filters. For the subsequent decomposition levels, the responses of the QMF filter pair are related to those of the previous level by replicating every sample of the response. By using Haar transform, the concurrent CCPs can be decomposed into the single CCPs, and then P-SVMs recognizer is developed for pattern recognition.

2.3. P-SVMs recognizer

2.3.1. Binary SVM

The basic idea of SVM is to transform the data to a higher dimensional feature space and find the optimal hyperplane in the space that maximizes the margin between the two classes. Consider a training data set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_M, y_M)\}$, $i = 1, 2, \ldots, M$, where M is the total number of training vectors, $\mathbf{x}_i \in R^d$ is the ith d-dimensional input vector, and $y_i \in \{1, -1\}$ is known target. The training of SVM involves the solution of the following quadratic optimization problem:

$$\text{Minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{M}\xi_i \quad (11)$$

$$\text{Subject to } y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0 \quad (12)$$

where ξ_i are slack variables, measuring the degree of misclassification of the sample $\mathbf{x}_i$, C is the error penalty factor, penalizing the non-zero ξ_i, the bias b is a scalar, representing the bias of the hyperplane, $\mathbf{w}$ is the vector of hyperplane coefficients, defining a direction perpendicular to the hyperplane, the index i labels the M training cases, and the map function ϕ is a transformation to map the input vectors into a high-dimensional feature space (see Fig. 5). The optimization problem becomes a trade-off between the margin maximization and training errors minimization.

In particular, if the data are perfectly linearly separable, then $\xi_i = 0$, and the separating hyperplane that creates the maximum distance between the plane and the nearest data (i.e., the maximum-margin equals $\|\mathbf{w}\|^{-2}$) is the optimal separating hyperplane. To solve non-linear classification tasks, a mapping function is usually employed to map the training samples from the input space into a higher-dimensional feature space. This allows the SVM to fit the maximum-margin hyperplane in the transformed feature space. Any function that satisfies Mercer's theorem (Cristianini & Shawe-Taylor, 2000) can be used as a kernel function. Some popular SVM kernel functions include:

$$\text{linear function}: \; K(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^T \times \mathbf{y}_j \quad (13)$$

$$\text{Gaussian radial basis function(GRBF)}: K(\mathbf{x}_i, \mathbf{y}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{y}_j\|^2) \quad (14)$$

$$\text{polynomial function with degree } d: K(\mathbf{x}_i, \mathbf{y}_j) = ((\mathbf{x}_i \times \mathbf{y}_j) + 1)^d \quad (15)$$

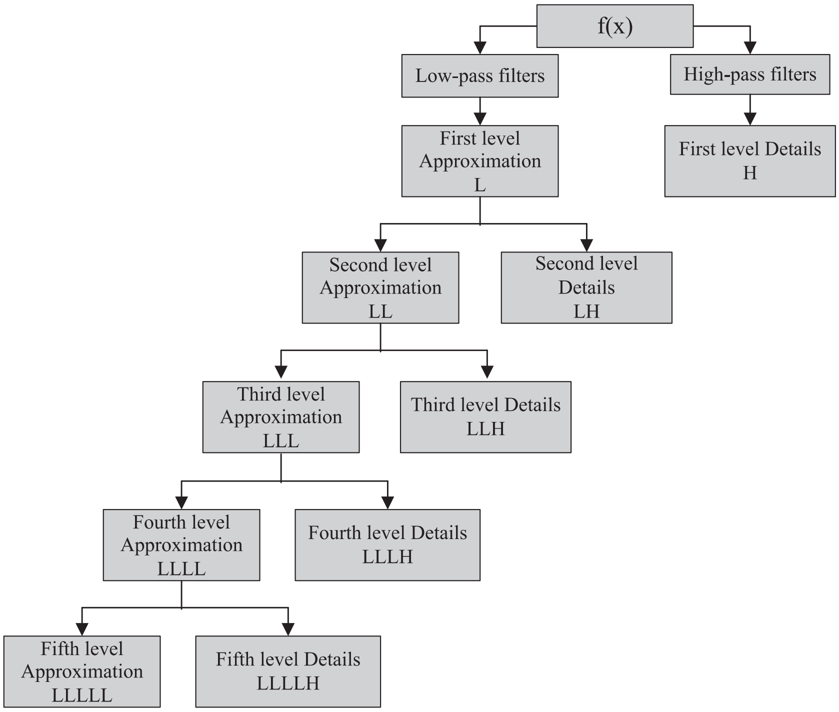
f(x)
Low-pass filters
High-pass filters
First level Approximation L
First level Details H
Second level Approximation LL
Second level Details LH
Third level Approximation LLL
Third level Details LLH
Fourth level Approximation LLLL
Fourth level Details LLLH
Fifth level Approximation LLLLL
Fifth level Details LLLLH

Fig. 4. A five-level wavelet decomposition.

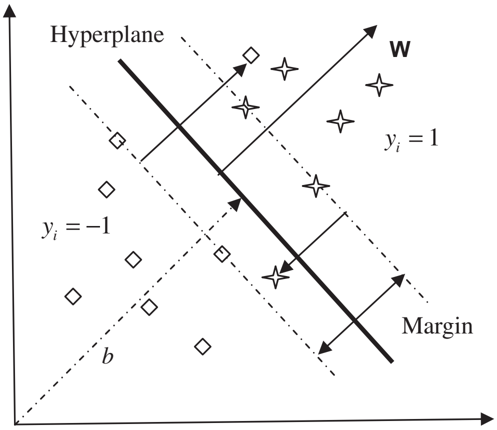

Fig. 5. A geometric interpretation of the classification of SVM for data set with two classes.

2.3.2. Multi-class SVMs recognizer

SVMs are originally designed for binary classification, which are not suitable for recognizing multiple CCPs. Through Haar wavelet transform, the concurrent CCPs are decomposed into several single patterns, which is a typical multi-classification problem. Multi-classification can be obtained by the combination of binary classification. In order to extend a single SVM to solve multi-classification problem, "one-against-one", "one-against-all" and direct acyclic graph (DAG) are three popular methods. Hsu and Lin (2002) conducted a comprehensive comparison of these three multi-class SVM classification methods, and they suggested that the "one-against-one" method is most suited for practical use than other methods. Therefore, "one-against-one" method is used to recognizing CCPs in this study.

For k-class single unnatural patterns, the "one-against-one" method constructs $M = C_k^2 = k(k-1)/2$ binary classifiers, and each of them is trained by binary-class data. For example, there are generally six unnatural basic CCPs exhibited by a control chart (see Fig. 1), and thus $M = C_6^2 = 6(6-1)/2 = 15$ binary classifiers need to be constructed. For training data from the ith and the jth class, the following binary classification problem needs to be solved:

$$\min \frac{1}{2}(\mathbf{w}^{ij})^T \mathbf{w}^{ij} + C\sum_t \xi_n^{ij}(\mathbf{w}^{ij})^T$$

$$\text{s.t.}\begin{cases} (\mathbf{w}^{ij})^T K(x_n) + b^{ij} > 1 - \xi_n^{ij} & \mathit{if} \quad y_n = i \\ (\mathbf{w}^{ij})^T K(x_n) + b^{ij} \leqslant \xi_n^{ij} - 1 & \mathit{if} \quad y_n = j \\ \xi_n^{ij} \geqslant 0 \end{cases} \tag{16}$$

where similarly to binary classification SVMs, $K(x_n)$ is the kernel function, (x_n, y_n) is the ith or jth training sample, $\mathbf{w} \in R^N$ and $b \in R$ are the weighting factors, ξ_n^{ij} is the slack variable and C is the penalty parameter.

There are some different methods for doing the future testing after all SVM recognizers are constructed. After some tests, the decision is made using the following strategy: if $(\mathbf{w}^{ij})^T K(x_n) + b^{ij} > 1 - \xi_n^{ij}$ is true, $\mathbf{x}$ belongs to the ith class, then the vote for the i^{th} class is added by one. Otherwise, the jth is increased by one. Then, $\mathbf{x}$ is predicted to be in the class using the largest vote. The voting approach described above is also called as Max Win strategy. For more details, readers are referred to the literature (Vapnik, 1995).

2.3.3. Selection of kernel function

The performance of SVMs varies depending on the choice of the kernel function and its parameters. Therefore, one suitable kernel function should be first selected for implementing SVM. How to choose a good kernel function is highly problem-dependent, and it is an important factor in SVM applications. There are usually two kernel functions for nonlinear SVM including the GRBF and the polynomial function (see Eq. (14) and (15)). The GRBF kernel nonlinearly maps the samples into a higher dimensional space unlike the linear kernel, so it can handle the case where the relation between class labels and attributes is nonlinear. The polynomial function takes a longer time in the training stage of SVM and has more hyper-parameters than the GRBF kernel has.

Hybrid kernel is a way to combine the heterogeneous complementary characteristics of different kernels, and it was originally proposed by Tan and Wang (2004) and Vert et al.,(2004). Recently, some successful applications of hybrid kernels were reported (Conforti & Guido, 2010; Liu, Lu, Jin, Li, & Chen, 2006; Zhao & Sun, 2009; Zheng, Liu, & Tian, 2005). The results demonstrate that the use of the hybrid kernel results in a better performance than those with a single common kernel. Therefore, one hybrid kernel function combining GRBF and polynomial kernel in a generalized kernel form is presented for CCPR in this study.

$$\textit{hybrid kernel} = \beta \times \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{y}_j\|^2\right) + (1-\beta)((\mathbf{x}_i \times \mathbf{y}_j) + 1)^d \tag{17}$$

The parameter γ and d represent kernel width and degree, and β is coefficient. Since the hybrid kernel still satisfies Mercer's condition, it can be used as a kernel function for SVM. For instance, one case of the kernel of degree 2, width 2, input vector $\mathbf{x}_{\mathrm{i}} = [x_1, x_2]$, $\mathbf{x}_{\mathrm{j}} = [x'_1, x'_2]$ of only two dimension, and β 0.5 can be represented: $\textit{hybrid kernel} = 0.5 \times \exp(-2\|x_1 - x'_1 + x_2 - x'_2\|^2) + 0.5(1 + 2x_1x'_1 + 2x_2x'_2 + 2x_1x_2x'_1x'_2 + (x_1x'_1)^2 + (x_2x'_2)^2)$.

The recognition performance of GRBF, the polynomial function and hybrid kernel function are compared and analyzed in Section 3.

2.3.4. Optimization of kernel function parameters

For the hybrid kernel, three parameters, including penalty parameter C, kernel width σ and degree d, need to be optimized in order to obtain the best classification result. The exhaustive search method for discrete parameters and the grid search method for continuous parameters are two most popular methods for selection of model parameters.

Recently, some intelligent evolution algorithms show distinguished performance for parameters optimization (Du & Xi, 2010; Du, Xi, Yu, & Sun, 2010; Eberhart & Kennedy, 1995). Particle swarm optimization (PSO) is a population-based search algorithm, which is inspired by the social behavior of bird flocks and was originally developed by Eberhart and Kennedy (1995). It is widely reported that the PSO algorithm is very easy to implement and has fewer parameters to adjust when compared to other evolutionary algorithms. In this study, PSO technique is used to obtain an optimal subset of parameters. Each particle is "flown" through the multidimensional search space, adjusting its position in search space according to its own experience and that of neighboring particles. The particle therefore makes use of both the best position encountered by itself and the best position encountered by its neighbors to position itself toward an optimal solution. The performance of each particle is evaluated using a predefined fitness function, which encapsulates the characteristics of the optimization problem.

It is an iterative process in which the change in weights of a particle at the beginning of an iteration are calculated using Eq. (18) and the new position of every particle is found using Eq. (19).

$$v_i(t+1) = w_p v_i(t) + c_1 r_1 (p_{id} - x_i(t)) + c_2 r_2 (p_{gd} - x_i(t)) \tag{18}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{19}$$

where t is the current step number, w_p is the inertia weight, c_1 and c_2 are the acceleration constants, r_1 and r_2 are two random numbers in the range $[0, 1]$, $x_i(t)$ is the position of the particle at time step t, $v_i(t)$ is the rate of movement (velocity) for particle i at time step t, p_{id} is the best one of the solutions this particle has reached, p_{gd} is the best one of the solutions all the particles have reached.

The procedure of PSO used in this study is summarized as follows:

- Step 1: Initialize a population of particles with random positions and velocities, where each particle contains q variables. In this study, $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_{2^p}]^T$ is a sequence of 2^p discrete observed points, and there are three variables that need to be optimized, namely regularization parameter C, kernel width γ and degree d. Therefore the dimension of a particle is $2^p + 3$.
- Step 2: Evaluate the objective values of all particles. Let own best experience (p_{best}) of each particle and its objective value be equal to its current position and objective value, and let the group best experience (g_{best}) and its objective value be equal to the position and objective value of the best initial particle.
- Step 3: Update the velocity and position of each particle according to equations (18) and (19).
- Step 4: Evaluate the objective values of all particles.
- Step 5: For each particle, compare its current objective value with the objective value of its p_{best}. If the current value is better, then update p_{best} and its objective value with the current position and objective value.
- Step 6: Determine the best particle of the current swarm with the best objective value of g_{best}, update g_{best} and its objective value with the position and objective value of the current best particle.
- Step 7: If a stopping criterion is met, give output g_{best} and its objective value; otherwise, go to Step 3.

3. Simulation and results analysis

3.1. Data generation and processing

Without of loss generality, eight types of concurrent CCPs and normal pattern were simulated in this study, including upward shift and increasing trend (u.s. plus i.t.), downward shift and decreasing trend (d.s. plus d.t.), increasing trend and cyclic pattern (i.t. plus c.p.), decreasing trend and cyclic pattern (d.t. plus c.p.), upward shift and cyclic pattern (u.s. plus c.p.), and downward shift and cyclic pattern (d.s. plus c.p.), upward shift and systematic pattern (u.s. plus s.p.), and downward shift and systematic pattern (d.s. plus s.p.).

Due to the demand for large numbers of training examples, the Monte-Carlo simulation approach was used to generate training data sets for CCPR. The parameters are summarized in Table 1, and the data was generated according to the reference (Guh & Tannock, 1999):

$$X(t) = N(t) + u_i(t)d_i(t) + u_j(t)d_j(t) \tag{20}$$

where $X(t)$ is the sample at time t, and $N(t)$ is a common cause variation at time t following a normal distribution with zero mean and standard deviation σ. σ follows a uniform distribution in the range $0 \leqslant \sigma \leqslant 5$. $d_i(t)$ and $d_j(t)$ are special assignable cause variation that start at time i or j. For normal patterns: $d_i(t) = d_j(t) = 0$

$$u_i(t) = \begin{cases} 0, & t < i \\ 1, & t \geqslant i \end{cases} \tag{21}$$

$$u_j(t) = \begin{cases} 0, & t < j \\ 1, & t \geqslant j \end{cases} \tag{22}$$

where i and j are selected randomly due to the data-window size.

Table 1
The parameters used for eight typical concurrent CCPs.

Pattern	Parameter	Sample No.	Total No.
u.s. plus i.t.	$\lambda = (-2.5, -1.75, 0.25), (1.75, 2.5, 0.25)$ $k = (0.1, 0.2, 0.02), (-0.2, -0.1, 0.02)$	30	2880
d.s. plus d.t.	$\lambda = (-1.75, -2.5, -0.25), (2.5, 1.75, -0.25)$ $k = (0.2, 0.1, -0.02), (-0.1, -0.2, -0.02)$	30	2880
i.t. plus c.p.	$k = (0.1, 0.2, 0.02), (-0.2, -0.1, 0.02)$ $A = (2, 2.5, 0.25)$	60	2160
d.t. plus c.p.	$k = (0.2, 0.1, -0.02), (-0.1, -0.2, -0.02)$ $A = (2, 2.5, 0.25)$	60	2160
u.s. plus c.p.	$\lambda = (-2.5, -1.75, 0.25), (1.75, 2.5, 0.25)$ $A = (2, 2.5, 0.25)$	80	1920
d.s. plus c.p.	$\lambda = (-1.75, -2.5, -0.25), (2.5, 1.75, -0.25)$ $A = (2, 2.5, 0.25)$	80	1920
u.s. plus s.p.	$\lambda = (-2.5, -1.75, 0.25), (1.75, 2.5, 0.25)$ $d = (1.5, 3, 0.3)$	64	2560
d.s. plus s.p.	$\lambda = (-1.75, -2.5, -0.25), (2.5, 1.75, -0.25)$ $d = (1.5, 3, 0.3)$	64	2560

For single shift patterns: $d_i(t) = \pm\lambda$, where λ is the magnitude of the shift which is randomly selected and uniformly distributed, such that $-2.5\sigma \leqslant \lambda \leqslant 2.5\sigma$.

For single trend patterns: $d_i(t) = \pm k(t - i)$, where k is the trend slope and follows a normal distribution, such that $-0.22\sigma \leqslant k \leqslant 0.22\sigma$.

For single cyclic patterns: $d_i(t) = A \times \sin(2\pi t/\Omega)$, where A is the cycle amplitude which is randomly selected, such that $1.0\sigma \leqslant A \leqslant 2.5\sigma$. The cycle frequency, Ω, is fixed to the value of 0.8 in this study.

For single systematic pattern: $x(t) = N(t) + (-1)^t d(t)$, where $1.5 \leqslant d(t) \leqslant 3$ denotes the magnitude of shift.

For shift and cyclic concurrent patterns: $d_i(t) = \pm\lambda$ and $d_j(t) = A \times \sin(2\pi t/\Omega)$.

For shift and trend concurrent patterns: $d_i(t) = \pm\lambda$ and $d_j(t) = \pm k(t - j)$.

For trend and cyclic concurrent patterns: $d_i(t) = \pm k(t - i)$ and $d_j(t) = A \times \sin(2\pi t/\Omega)$.

For shift and systematic concurrent pattern: $d_i(t) = \pm\lambda$ and $x(t) = N(t) + (-1)^t d_i(t)$.

In Table 1, the notation (x, y, z) represents (initial value, final value, increment). The Sample No. and Total No. represent the sample number for each concurrent pattern and the total sample number.

The pattern parameters are chosen so that the simulated patterns are mostly within the range of $\pm 3\sigma$, since an important reason for the use of SVM for CCPR is to identify patterns that do not cause data points to exceed control chart limits. Furthermore, only univariate quality characteristic is used and only two concurrent patterns occur at any one time.

3.2. Parameters of P-SVMs

The performance of learning in P-SVMs is influenced by its parameters. There are no absolute rules for tuning these parameters, which are dependent on the characteristics of the real-world problems and are experimentally specified through try and error. The parameters of P-SVM having better performances during training in this study are summarized as follows.

(1) Hybrid kernel function in Eq. (17) is used in this study, and $\beta = 0.5$.
(2) The inertia weights w_p are generated at random in the range [0.2, 0.8].
(3) The speed of particles are generated at random in the range [−4, 4].
(4) The initial velocities of the initial particles are generated at random in the range [−4, 4].

Table 2
Classification performance for the proposed approach using four-level decomposition (%).

Input	Recognized pattern														
	u.s.	d.s.	i.t.	d.t.	s.p.	c.p.	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.
u.s. + i.t.	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
u.s. + c.p.	1.75	0	0	0	0	1.20	0	97.05	0	0	0	0	0	0	0
d.s. + d.t.	0	1.42	0	2.09	0	0	0	0	96.49	0	0	0	0	0	0
d.s. + c.p.	0	3.33	0	0	0	1.38	0	0	0	95.29	0	0	0	0	0
i.t. + c.p.	0	0	1.75	0	0	2.91	0	0	0	0	95.34	0	0	0	0
d.t. + c.p.	0	0	0	4.17	0	1.21	0	0	0	0	0	94.62	0	0	0
u.s. + s.p.	1.93	0	0	0	1.83	0	0	0	0	0	0	0	96.24	0	0
d.s. + s.p.	0	2.58	0	0	2.04	0	0	0	0	0	0	0	0	95.38	0
n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table 3
Classification performance for the proposed approach using five-level decomposition (%).

Input	Recognized pattern														
	u.s.	d.s.	i.t.	d.t.	s.p.	c.p.	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.
u.s. + i.t.	0.5	0	0.75	0	0	0	98.75	0	0	0	0	0	0	0	0
u.s. + c.p.	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
d.s. + d.t.	0	1.11	0	0.65	0	0	0	0	98.24	0	0	0	0	0	0
d.s. + c.p.	0	1.07	0	0	0	1.48	0	0	0	97.45	0	0	0	0	0
i.t. + c.p.	0	0	1.33	0	0	1.36	0	0	0	0	97.31	0	0	0	0
d.t. + c.p.	0	0	0	0.47	0	1.38	0	0	0	0	0	98.15	0	0	0
u.s. + s.p.	1.56	0	0	0	0.96	0	0	0	0	0	0	0	97.48	0	0
d.s. + s.p.	0	0.84	0	0	1.03	0	0	0	0	0	0	0	0	98.13	0
n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Table 4
Classification performance for the proposed approach using six-level decomposition (%).

Input	Recognized pattern														
	u.s.	d.s.	i.t.	d.t.	s.p.	c.p.	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.
u.s. + i.t.	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
u.s. + c.p.	0.53	0	0	0	0	0.63	0	98.84	0	0	0	0	0	0	0
d.s. + d.t.	0	0.32	0	1.12	0	0	0	0	98.56	0	0	0	0	0	0
d.s. + c.p.	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
i.t. + c.p.	0	0	0.45	0	0	0.84	0	0	0	0	98.71	0	0	0	0
d.t. + c.p.	0	0	0	1.76	0	0.91	0	0	0	0	0	97.33	0	0	0
u.s. + s.p.	0.71	0	0	0	0.55	0	0	0	0	0	0	0	98.74	0	0
d.s. + s.p.	0	0.46	0	0	0.02	0	0	0	0	0	0	0	0	98.52	0
n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

(5) The study factors c_1 and c_2 equal 2.

3.3. Performance analysis

Several experiments have been done to verify the effectiveness of the proposed hybrid approach. The influences of the key factors of wavelet transform and P-SVM recognizer on recognition performance are analyzed. The analysis can help us to obtain the suitable parameter setting to improve the general performance of the hybrid approach. In each of the following tests, only one factor is varied while the remaining ones are kept constant. The aggregate correct classification rate (CCR: the number of correctly recognized tested patterns/total number of tested pattern) is used as the evaluation index.

3.3.1. Performance for recognizing concurrent patterns using different decomposition level

The objective of this experiment is to compare the recognition performance of different decomposition level of Haar wavelet. The hybrid kernel function is used for P-SVM recognizer. In Haar wavelet analysis, a raw signal is decomposed into different levels with each level having two components: details and approximations. Each details component separated from the signal of previous level contain information at relatively high frequencies. For a concurrent pattern, such as trend plus cyclic pattern, the part of cyclic signal is embedded somewhere in those details and the trend part is included in the approximations.

In practice, one needs to select a suitable number of levels based on the nature of the signal. The number of levels is closely related to the data-window size. In this study, the data-window size is chosen to be 16, 32, and 64 samples, which are considered to be reasonable for on-line monitoring applications (Cheng, 1997; Pham and Oztemel, 1994). The corresponding wavelet coefficients up to four-level, five-level and six-level wavelet decompositions are calculated, and the pattern recognition results are compared (see Tables 2–4). The rows in the tables represent the input patterns (i.e. target patterns), while columns indicate the output patterns (i.e. recognized patterns) as identified by the proposed approach. Moreover, it is assumed that the process starts in control and then two assignable causes would start randomly at any sample between the 6th and 14th sample for the 16-sample window size, between the 10th and 26th sample for the 32-sample window size, and between the 20th and 58th sample for the 64-sample window size, respectively. The unnatural pattern then continues until the end of the observed data window. This is considered to be more practical since with on-line monitoring applications, the unnatural pattern often occurs after a period in which the process is in-control, and the starting point of the unnatural pattern is usually unknown (Al-Assaf, 2004; Cheng, 1997).

From Tables 2–4, the average correct classification rates for P-SVM recognizer using three types of decomposition levels are 96.71%, 98.39%, and 98.97%, respectively. The simulation results show that the proposed hybrid approach integrating Haar wavelet and SVMs works very well for recognizing concurrent control chart patterns for the cases tested.

A small window size generally detects unnatural patterns more quickly, but might result in a higher Type I error. A larger window size can accommodate more shifted points with shift magnitudes so that more unnatural pattern features can be exposed to the recognizer. However, a larger window size can reduce the recognition efficiency by increasing the time required to detect patterns. Therefore, in order to find a compromise size to balance errors and recognition efficiency, one experiment has been conducted to compare different sizes of moving window and choose a suitable one. Different window sizes with 16, 32, 64, and 128 samples are considered. Furthermore, the computation time (CT) is also taken as another evaluation index to evaluate the performance of these approaches. The unit of CT is second. All of the classification tasks have been done on a microcomputer with Inter (R) core (TM) 2.33 GHz and 996 MB internal memory.

The comparison results are presented in Table 5. According to the results, the window size with 64 samples and 128 samples produced almost the same best CCR (98.97% and 98.99%), and there is no significant improvement on CCR from the window size with 64 samples to the window size with 128 samples. However, the CT for the window size with 128 samples (almost 1 s) is much longer than the CT for the window size with 64 samples (0.027 s). Therefore, the window size 64 is suggested in this study.

Before P-SVM recognizes the CCPs, the concurrent CCPs are decomposed into single CCP. The results of using Haar transform to decompose the concurrent CCPs are presented in Table 6. The decomposition results of the concurrent CCP (i.t. + c.p.), one concurrent CCP as example, using different level wavelet transform are presented in Table 7.

Table 6 suggests that the Haar wavelet transform has different decomposing capability for three types of decomposition levels. The six-level decomposition tends to produce better classification accuracy at 99.85% on average, compared with 99.31% for four-level and 99.51% for five-level decomposition. Tables 2–4, 6 and 7 demonstrate that though the proposed approach has better performance when the number of decomposition level is bigger, a large decomposition level results in high time cost of training and

Table 5
Performance comparison for different window sizes.

Window size	CCR (%)	CT (s)
16	96.71	0.004
32	98.39	0.011
64	98.97	0.027
128	98.99	0.903

Table 6
Decomposition performance for Haar transform with different level decomposition (%).

	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.	Average CCR (%)
Four-level decomposition	100	100	99.32	99.01	98.98	98.96	98.84	98.70	100	99.31
Five-level decomposition	100	100	99.93	98.97	98.96	99.42	99.01	99.32	100	99.51
Six-level decomposition	100	100	99.97	100	99.98	99.94	99.24	99.51	100	99.85

Table 7
Decomposition results of the concurrent CCP (i.t. + c.p.) using different level wavelet transform (%).

	Decomposed pattern														
	u.s.	d.s.	i.t.	d.t.	s.p.	c.p.	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.
Four-level	0	0	0.18	0	0	0.84	0	0	0	0	98.98	0	0	0	0
Five-level	0	0	0.14	0	0	0.90	0	0	0	0	98.96	0	0	0	0
Six-level	0	0	0.0	0	0	0.02	0	0	0	0	99.98	0	0	0	0

Table 8
Shows the aggregate correct classifications rate for different parameters of hybrid kernel function.

Run	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	Average
C	296.7	428.3	338.4	305.7	452.9	412.1	289.4	324.7	401.5	348.2	421.3	397.1	292.8	387.4	412.9	367.29
γ	0.0109	0.0093	0.0078	0.0097	0.0042	0.0097	0.0108	0.0103	0.0099	0.0089	0.0085	0.0108	0.0084	0.0075	0.0049	0.00877
d	2	4	4	3	5	5	2	2	5	4	4	5	2	3	3	
CR (%)	98.91	98.92	98.97	98.94	98.97	98.89	98.94	98.97	98.87	98.92	98.97	98.84	98.91	98.97	98.96	98.93

testing, and it also decreases the effectiveness of timely detecting out-of-control. Therefore, the number of decomposition level of the proposed approach must be varied based on the balance between time cost and CCRs. Moreover, the concurrent pattern may be incorrectly recognized as single patterns. For example, the CCR for the concurrent d.s. + d.t. is 98.56% for six-level decomposition where 0.32% patterns are incorrectly recognized to be d.s. and 1.12% patterns are incorrectly recognized to be d.t. as shown in Table 4. All of these incorrectly recognized patterns for d.s. + d.t. are either d.s. and d.t. This result is reasonable.

3.3.2. Performance for recognizing concurrent patterns using different kernel function parameters

In this experiment, fifteen runs have been performed to find optimal parameters for hybrid kernel function, which was demonstrated in the previous experiments to be an effective and appropriate kernel for concurrent CCPR. Six-level decomposition of Haar wavelet was applied since it exhibited better performance in previous experiments, and PSO is applied to find the best combination of the parameters of the P-SVM recognizer and to gain the CCR maximum. The related parameters C and $\gamma$are varied in the arbitrarily chosen ranges [1, 1000], and [0, 1] so as to cover high and small regularization of the classification model, and fat as well as thin kernels, respectively. The degree d is varied in the range [2, 5] with integer value in order to span polynomials with low and high flexibility. The detailed results about the optimal values of the P-SVM recognizer parameters (i.e., the penalty parameter C and kernel width γ and degree d) obtained by the proposed P-SVM are shown in Table 8. The proposed approach successfully finds the global optimum just with 150 iterations. This result is repeated in multiple runs of the program. Values of the parameters C and γ of the P-SVM recognizer in five different runs of the program with 150 iterations are presented in Fig. 6. In each different run of program, PSO first generates random values for C, γ and d, and then it searches for better values of them that produce better CCR. Usually, after 50 iterations, the approach produces be best parameters values of the P-SVM recognizer.

3.3.3. Performance for recognizing concurrent patterns using different kernel functions

The objective of this experiment is to compare the classification performance of hybrid kernel function with the commonly used

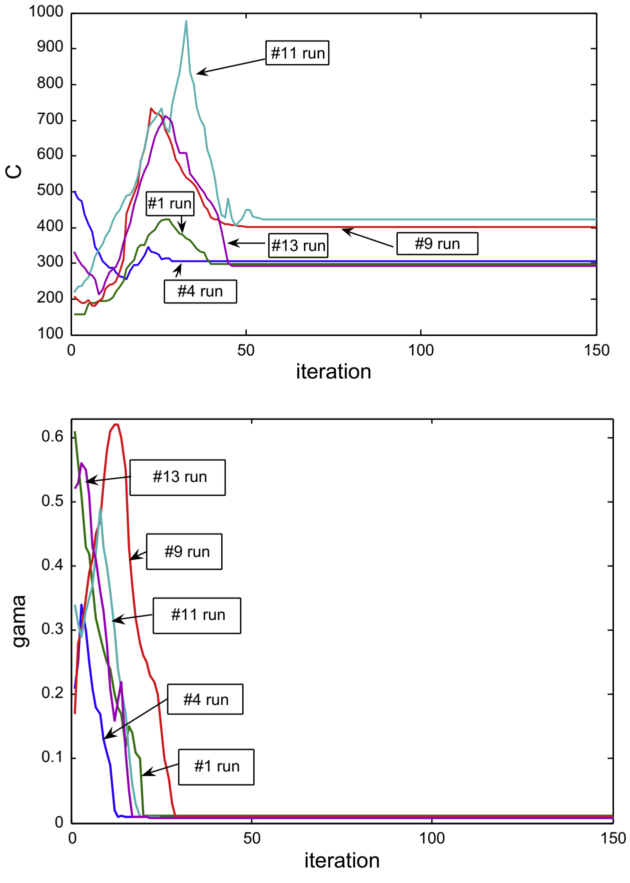

Fig. 6. The parameters C and γ of hybrid kernel function for five different runs.

Table 9
The aggregate correct classification rate for different kernel function.

Kernel	Input	Recognized pattern															CCR (%)
		u.s.	d.s.	i.t.	d.t.	s.p.	c.p.	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.	
Hybrid kernel ($C = 338.4$, $\beta = 0.5$, $\gamma = 0.0078$, $d = 4$)	u.s. + i.t.	0.19	0	0.36	0	0	0	99.45	0	0	0	0	0	0	0	0	98.97
	u.s. + c.p.	0	0	0	0	0	0	0	100	0	0		0	0	0	0	
	d.s. + d.t.	0	0.32	0	0.58	0	0	0	0	99.1	0	0	0	0	0	0	
	d.s. + c.p.	0	1.14	0		0	0.38	0	0		98.48	0	0	0	0	0	
	i.t. + c.p.	0	0	0.7	0	0	0.91	0	0	0	0	98.39	0	0	0	0	
	d.t. + c.p.	0	0	0	0.73	0	1.25	0	0	0	0	0	98.02	0	0	0	
	u.s. + s.p.	0.71	0	0	0	0.55	0	0	0	0	0	0	0	98.74	0	0	
	d.s. + s.p.	0	0.46	0	0	0.02	0	0	0	0	0	0	0	0	98.52	0	
	n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	
GRBF kernel ($\gamma = 0.0078$)	u.s. + i.t.	0.72	0	1.27	0	0	0	98.01	0	0	0	0	0	0	0	0	97.93
	u.s. + c.p.	1.31	0	0	0	0	2.35	0	96.34	0	0		0	0	0	0	
	d.s. + d.t.	0	0.8	0	1.29	0	0	0	0	97.91	0	0	0	0	0	0	
	d.s. + c.p.	0	1.07	0		0	0.81	0	0	0	98.12	0	0	0	0	0	
	i.t. + c.p.	0	0	2.08	0	0	1.05	0	0	0	0	96.87	0	0	0	0	
	d.t. + c.p.	0	0	0	0.25	0	1.12	0	0	0	0	0	98.63	0	0	0	
	u.s. + s.p.	1.23	0	0	0	1.32	0	0	0	0	0	0	0	97.35	0	0	
	d.s. + s.p.	0	1.01	0	0	0.85	0	0	0	0	0	0	0	0	98.14	0	
	n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	
Polynomial kernel ($d = 4$)	u.s. + i.t.	0.97	0	1.65	0	0	0	97.38	0	0	0	0	0	0	0	0	96.07
	u.s. + c.p.	3.49	0	0	0	0	2.24	0	94.27	0	0	0	0	0	0	0	
	d.s. + d.t.	0	1.78	0	1.68	0	0	0	0	96.54	0	0	0	0	0	0	
	d.s. + c.p.	0	0.78	0		0	1.56	0	0		97.66	0	0	0	0	0	
	i.t. + c.p.	0	0	3.88	0	0	1.83	0	0	0	0	94.29	0	0	0	0	
	d.t. + c.p.	0	0	0	1.27	0	2.88	0	0	0	0	0	95.85	0	0	0	
	u.s. + s.p.	3.26	0	0	0	2.62	0	0	0	0	0	0	0	94.12	0	0	
	d.s. + s.p.	0	1.32	0	0	4.17	0	0	0	0	0	0	0	0	94.51	0	
	n.p.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	

Table 10
Recognition performance with and without Haar wavelet decomposition.

	u.s. + i.t.	u.s. + c.p.	d.s. + d.t.	d.s. + c.p.	i.t. + c.p.	d.t. + c.p.	u.s. + s.p.	d.s. + s.p.	n.p.	Average CCR (%)
With wavelet decomposition	100	98.84	98.56	100	98.71	97.33	98.74	98.52	100	98.97
Without wavelet decomposition	93.18	88.47	90.43	79.42	85.48	87.09	84.15	86.32	96.72	87.92

Table 11
Recognition performance comparison of P-SVM with other approaches.

Approach	Parameters	CCR (%)	CT (s)
PNN	Spread = 50	93.42	0.005
RBFNN	Spread = 64, goal = 0.01	93.78	0.004
MLP(BP)	Hidden neurons = 32	94.66	0.001
MLP(RP)	Hidden neurons = 24	94.15	0.004
P-SVM	Hybrid kernel function $C = 338.4$, $\beta = 0.5$, $\gamma = 0.0078$, $d = 4$	98.97	0.027

GRBF (see Eq. (14)) and polynomial function (see Eq. (15)). Table 9 shows the aggregate percentage of correct classifications. Based on the simulations results, it is found that the P-SVM recognizer with hybrid kernel function has better recognizing capability than those with the other two kernels since hybrid kernel function has the merits of both GRBF and polynomial function.

3.3.4. Comparison for recognizing concurrent patterns with wavelet decomposition and without wavelet decomposition

The objective of this experiment is to compare the classification performance of the approach with Haar wavelet decomposition and its counterpart without Haar wavelet decomposition. If the input pattern is a concurrent pattern, Haar wavelet transform method is used to decompose the input pattern into two single patterns. Therefore, the inputs of the P-SVM model with wavelet transform are the sample data points with single patterns. The P-SVM recognizer also can be applied solely to classify the concurrent patterns, which are not decomposed into single patterns. For the P-SVM model without wavelet transform, the input vectors consist of 64 consecutive raw sample data points over time series with concurrent pattern. The detailed results are presented in Table 10. It can be seen that the correct classification rate of P-SVM with six-level wavelet decomposition (98.97%) is much higher than that of P-SVM without wavelet decomposition (87.92%).

3.4. Comparison with other approaches

In order to further evaluate the performance of the proposed approach, it is compared with other recognizers including probabilistic neural networks (PNN) (Specht, 1990), radial basis function neural networks (RBFNNs) (Chen, Cowan, & Grant, 1991) and multilayered perceptron (MLP) neural networks (Pal & Mitra, 1992) with different training algorithms such as back propagation (BP) learning algorithm and with the resilient propagation (RP) learning algorithm, as indicated in Table 11. They involve parameters that should be readjusted in any new classification. Those parameters regulate the classifiers to be best fitted in the classification task. In most cases, there is no classical method for obtaining their values, and therefore they are experimentally specified through try and error.

It can be seen from Table 10 that the proposed hybrid approach outperforms the PNN recognizer by 5.61%, RBFNN by 5.24%, MLP (BP) by 4.35%, and MLP (RP) by 4.87%. These results suggest that the proposed hybrid approach has better performance compared with those of the commonly used ANNs approach. With regard to CT index, MLP (BP) has best performance, and it takes approximately 0.001 s for the MLP (BP) approach to do a classification operation. P-SVM has worst performance and it takes approximately 0.027 s to do a same task, however, the computation time (approximately 0.027 s) is not too long compared to the processing time (usually approximately several minutes) for a machining operation of a workpiece. This feature enables the use of the proposed model in an on-line real time mode. Moreover, with rapid development of electronic and computer techniques, the time required for computation of P-SVM can be further met in on-line real applications.

4. Conclusions

Recognizing concurrent control chart patterns has been a challenging task for traditional pattern recognition techniques. New approaches need to be developed to accommodate high correct classification rate, and recognize the concurrent patterns as soon as possible. This paper addresses a novel hybrid approach to tackle the concurrent pattern recognition problem by incorporating SVM and Haar wavelet transform. The main merit of the proposed hybrid approach is that the approach demonstrates improved performance for recognizing concurrent CCPs and outperforms the commonly used approaches. One hybrid kernel function is presented for SVM recognizer, and the simulation result shows that the use of the hybrid kernel results in a better performance than those with a single common kernel. This result could help to better design and select kernel functions when using SVM in other applications. Furthermore, the analysis from this study provides guidelines in developing multiclass SVMs-based control chart pattern recognition systems in manufacturing processes. From Table 10, it can see that the main limitation is that the proposed hybrid approach has worse performance with regard to CT index. Though the computation time is not too long compared to the processing time, the faster and more efficient algorithms are desired to be developed for reducing the computation time.

Acknowledgements

The authors greatly acknowledge the editor and the referees for their valuable comments and suggestions that have led to a substantial improvement of the paper. This work was supported by the National Natural Science Foundation of China (Grant No. 51275558 and 50905114), the research fund of Aerospace Advanced Technology Research Institute and Shanghai Jiao Tong University (Grant No. 12GFZ-JJ08-011), and Shanghai Rising-Star Program (Grant No. 13QA1402100).

References

Al-Assaf, Y. (2004). Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks. *Computers and Industrial Engineering, 47*, 17–29.

Al-Assaf, Y. (2005). Multi-Resolution Wavelets analysis approach for the recognition of concurrent control chart patterns. *Quality Engineering, 17*, 11–21.

Al-Ghanim, A. M. (1997). An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. *Computers and Industrial Engineering, 32*, 627–639.

Al-Ghanim, A. M., & Ludeman, L. C. (1997). Automated unnatural pattern recognition on control charts using correlation analysis techniques. *Computers and Industrial Engineering, 32*(3), 679–690.

Assaleh, K., & Al-Assaf, Y. (2005). Features extraction and analysis for classifying causable patterns in control charts. *Computers and Industrial Engineering, 49*, 168–181.

Burrus, C. S., Gopinath, R. A., & Guo, H. (1998). *Introduction to wavelets and wavelet transforms: A primer*. Upper Saddle River: Prentice Hall.

Chen, L. H., & Wang, T. Y. (2004). Artificial neural network to classify mean shifts form Multivariate χ^2 chart signals. *Computers and Industrial Engineering, 47*, 195–205.

Chen, S., Cowan, C. F. N., & Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transaction on Neural Networks, 2*(2), 302–309.

Chen, Z., Lu, S., & Lam, S. (2007). A hybrid system for SPC concurrent pattern recognition. *Advanced Engineering Informatics, 21*, 303–310.

Cheng, C. (1997). A neural network approach for the analysis of control chart patterns. *International Journal of Production Research, 35*(3), 667–697.

Cheng, C. S., & Cheng, H. P. (2008). Identifying the source of variance shifts in the multivariate process using neural networks and support vector machines. *Expert Systems with Applications, 35*(1–3), 198–206.

Cheng, C. S., & Hubele, N. F. (1992). Design of a knowledge-based expert system for statistical process control. *Computers and Industrial Engineering, 22*(4), 501–517.

Chinnam, R. B. (2002). Support vector machines for recognizing shifts in correlated and other manufacturing processes. *International Journal of Production Research, 40*, 4449–4466.

Cook, D. F., & Chiu, C. C. (1998). Using radial basis function neural networks to recognize shifts in correlated manufacturing process parameters. *IIE Transactions, 20*, 227–234.

Cristianini, N., & Shawe-Taylor, J. (2000). *A introduction to support vector machines and other kernel-based learning methods*. Cambridge, UK: Cambridge University Press.

Conforti, Domenico, & Guido, Rosita (2010). Kernel based support vector machine via semidefinite programming: Application to medical diagnosis. *Computers & Operations Research, 37*(8), 1389–1394.

Du, S., & Xi, L. (2010). An integrated system for on-line intelligent monitoring and identifying process variability and its application. *International Journal of Computer Integrated Manufacturing, 23*(6), 529–542.

Du, S., & Xi, L. (2011). Fault diagnosis in assembly processes based on engineering-driven rules and PSOSAEN algorithm. *Computers and Industrial Engineering, 60*, 77–88.

Du, S., Xi, L., Ni, J., Pan, E., & Liu, C. R. (2008). Product lifecycle-oriented quality and productivity improvement based on stream of variation methodology. *Computers in Industry, 59*(2–3), 180–192.

Du, S., Xi, L., Yu, J., & Sun, J. (2010). On-line intelligent monitoring and diagnosis of aircraft horizontal stabilizer assemble processes. *International Journal of Advanced Manufacturing Technology, 50*(1–4), 377–389.

Duncan, A. J. (1986). *Quality control and industrial statistics* (5th ed.). Irwin.

Eberhart, C. R., & Kennedy, J. (1995). Particle swarm optimization. In *Proc. IEEE International Conference on Neural Networks* (pp. 1942–1948), Piscataway, NJ.

El-Midanya, T. T., El-Bazb, M. A., & Abd-Elwahedc, M. S. (2010). A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks. *Expert Systems with Applications, 37*(2), 1035–1042.

Gauri, S. K., & Chakraborty, S. (2006). Feature-based recognition of control chart patterns. *Computers and Industrial Engineering, 51*, 726–742.

Gauri, S. K., & Chakraborty, S. (2008). Improved recognition of control chart patterns using artificial neural networks. *International Journal of Advanced Manufacturing Technology, 36*, 1191–1201.

Gauri, S. K., & Chakraborty, S. (2009). Recognition of control chart patterns using improved selection of features. *Computer and Industrial Engineering, 56*, 1577–1588.

Grant, E. L., & Leavenworth, R. S. (1996). *Statistical quality control* (7th ed.). New York: McGraw-Hill Book Company.

Guh, R. S. (2003). Integrating artificial intelligence into on-line statistical process control. *Quality and Reliability Engineering International, 19*, 1–20.

Guh, R. S. (2005). A hybrid learning-based model for on-line detection and analysis of control chart patterns. *Computers and Industrial Engineering, 49*(1), 35–62.

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

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

Guh, R. S., & Shiue, Y. R. (2005). On-line identification of control chart patterns using self-organizing approaches. *International Journal of Production Research, 43*, 1225–1254.

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

Gunn, S. R. (1998). *Support vector machines for classification and regression*, Technical Report, University of Southampton.

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. *Computers and Industrial Engineering, 63*, 204–222.

Hassan, A., Shariff, N. B. M., Shaharoun, A. M., & Jamaluddin, H. (2003). Improved SPC chart pattern recognition using statistical features. *International Journal of Production Research, 41*(7). 1687-1603.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 415–425.

Hwarng, H. B. (1995). Multilayer perceptron for detecting cyclic data on control charts. *International Journal of Production Research, 33*, 3101–3117.

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

Liu, X., Lu, W. C., Jin, S. L., Li, Y. W., & Chen, N. Y. (2006). Supportvector regression applied to materials optimization of sialon ceramics. *Chemometrics and Intelligent Laboratory Systems, 82*, 8–14.

Lucy-Bouler, T. L. (1991). *Using autocorrelations, CUSUMs and runs rules for control chart pattern recognition: An expert system approach*. PhD dissertation. Tuscaloosa, Alabama: University of Alabama.

Montgomery, D. C. (2001). *Statistical quality control* (4th ed.). New York: Wiley.

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

Niaki, S. T. A., & Abbasi, B. (2005). Fault diagnosis in multivariate control charts using artificial neural networks. *Quality and Reliability Engineering International, 21*, 825–840.

Pal, S. K., & Mitra, S. (1992). Multilayer perceptron, fuzzy sets, and classification. *IEEE Transaction on Neural Networks, 3*(5), 683–697.

Pham, D. T., & Oztemel, E. (1994). Control chart pattern recognition using learning vector quantization networks. *International Journal of Production Research, 32*, 721–729.

Pham, D. T., & Wani, M. A. (1997). Feature-based control chart pattern recognition. *International Journal of Production Research, 35*, 1875–1890.

Pugh, G. A. (1989). Synthetic neural networks for process control. *Computers and Industrial Engineering, 17*(1–4), 24–26.

Ranaee, V., Ebrahimzadeha, A., & Ghaderia, R. (2010). Application of the PSO–SVM model for recognition of control chart patterns. *ISA Transaction, 49*, 577–586.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks, 3*(1), 109–118.

Sun, R., & Tsung, F. (2003). A Kernel-distance-based multivariate control Chart using Support Vector Methods. *International Journal of Production Research, 41*, 2975–2989.

Swift, J. A., & Mize, J. H. (1995). Out-of-control pattern recognition and analysis for quality control charts using Lisp-based system. *Computers and Industrial Engineering, 28*(1), 81–91.

Tan, Y., & Wang, J. (2004). A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension. *IEEE Transactions on Knowledge and Data Engineering, 16*(4), 285–295.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer.

Vert, J. P., Tsuda, K., & Schölkopf, B. (Eds.). (2004). *Kernel methods in computational biology*. Cambridge, MA: MIT Press. 35–70.

Wang, C. H., Kuo, W., & Qi, H. (2007). An integrated approach for process monitoring using wavelet analysis and competitive neural network. *International Journal of Production Research, 45*, 227–244.

Wang, C. H., Dong, T. P., & Kuo, W. (2009). A hybrid approach for identification of concurrent control chart patterns. *Journal of Intelligent Manufacturing, 20*, 409–419.

Western Electric Company (1958). *Statistical quality control handbook*. Western Electric.

Yang, J. H., & Yang, M. S. (2005). A control chart pattern recognition system using a statistical correlation coefficient method. *Computers and Industrial Engineering, 48*(2), 205–221.

Zhao, Lu., & Sun, Jing. (2009). Non-Mercer hybridkernel for linear programming supportvector regression in nonlinear systems identification. *Applied Soft Computing, 9*(1), 94–99.

Zheng, S., Liu, J., & Tian, J. W. (2005). An efficient star acquisition method based on SVM with mixtures. *Pattern Recognition Letters, 26*, 147–165.