Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines

Wen-An Yang a,⁎, Wei Zhou a,b, Wenhe Liao a, Yu Guo a

a School of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210036, People’s Republic of China
b Nanjing Surveying and Mapping Instrument Factory, Nanjing 210003, People’s Republic of China

ARTICLE INFO

Article history:
Received 27 January 2014
Accepted 24 June 2014
Communicated by Hung-Yuan Chung

Keywords:
Statistical process control
Control chart
Pattern recognition
Extreme learning machine
Extreme-point symmetric mode decomposition

Abstract

Control chart pattern recognition (CCPR) is an important issue in statistical process control because unnatural control chart patterns (CCPs) exhibited on control charts can be associated with specific causes that adversely affect the manufacturing processes. In recent years, many machine learning techniques [e.g., artificial neural networks (ANNs) and support vector machines (SVMs)] have been successfully applied to CCPR. However, such existing research for CCPR has mostly been developed for identification of basic CCPs. Little attention has been given to the utilization of ANNs/SVMs for identification of concurrent CCPs (two or more basic CCPs occurring simultaneously) which are commonly encountered in practical manufacturing processes. In addition, existing research for CCPR cannot provide more detailed CCP parameter information, such as shift magnitude, trend slope, cycle amplitude, etc., which is very useful for quality practitioners to search the assignable causes that give rise to the out-of-control situation. This study proposes a hybrid approach that integrates extreme-point symmetric mode decomposition (ESMD) with extreme learning machine (ELM) to identify typical concurrent CCPs and in addition to accurately quantifying the major CCP parameter of the specific basic CCPs involved. The numerical results indicate that the proposed model can effectively identify not only concurrent CCPs but also basic CCPs. Meanwhile, the major CCP parameter of the identified concurrent CCP can also be accurately quantified.

© 2014 Published by Elsevier B.V.

1. Introduction

Control charts are the most widely applied statistical process control (SPC) tools used to identify assignable causes in manufacturing processes. Unnatural patterns exhibited on control charts can be associated with a specific set of assignable causes provided that appropriate process knowledge is available [1]. Since the traditional Shewhart control charts do not provide any pattern-related information, a series of supplementary rules (e.g., zone tests or run rules) has thus been developed to facilitate quality practitioners in detection of unnatural patterns [1–3]. However, utilization of all the available rules could result in excessive numbers of false alarms. Because of common-cause variation in the process, a run would still have a low probability of occurrence before a control limit breach signal the process as out-of-control [4]. As pointed out by Lucy-Bouler [5], run rules are often ineffective in recognizing control chart patterns (CCPs). In addition, expert systems (ESs) have also been tried for pattern recognition tasks [6]. Nevertheless, developing such application of ESs in CCP recognition (CCPR) still remains a very hard and time-consuming task.

Various machine learning techniques, e.g., artificial neural networks (ANNs), support vector machines (SVMs), and decision trees (DTs) have recently been utilized for detecting and recognizing typical unnatural CCPs. Wang and Chen [7] proposed a neural-fuzzy model for detecting mean shifts and classifying their magnitudes in a multivariate manufacturing process. Experimental results indicated that this model outperformed the Hotelling’s T2 control chart in terms of out-of-control average run length (ARL) (i.e., the average number of inspected samples required to signal a process shift after it has occurred) under fixed Type I error. Low et al. [8] developed a back-propagation neural network (BPN)-based model for detecting mean shifts in multivariate manufacturing processes. Experimental results indicated the superiority of the proposed neural system-based model in process control while multiple quality characteristics were simultaneously considered. Barghash and Santarisi [9] utilized ANN for pattern recognition of...
the most common CCPs. In their work, in order to identify the effect of the training parameters on the performance of the neural network, a resolution IV fractional factorial experiment was utilized to explore a portion of the range of selected parameters to obtain better performance of the neural network. The results showed that many parameters such as minimum shift, shift range, population size and shift percentage, have significant effect on the performance of the ANN, while others such as network size and window size do not have major significance on the performance of the ANN. Niaiki and Abbasi [10] developed a hybrid model, in which Hotelling’s T² control chart was used for detecting the out-of-control signals, while a BPN was used for identifying the source(s) of the out-of-control signals. Aparisi et al. [11] presented a similar research work. They evaluated the correct classification percentage, and showed that the neural network is better than traditional decomposition method. Wang and Kuo [12] proposed a hybrid framework composed of filtering module and clustering module to identify six typical types of CCPs. In particular, a multiscale wavelet filter was utilized for denoising and three fuzzy clustering algorithms were employed to compare their performance of pattern classification. Experimental results demonstrated that the proposed method performed better than the ANN-based approaches in on-line CCPR. Guh and Shiu [13] proposed a straightforward and effective model to detect the mean shifts in multivariate control charts using DT learning techniques. Experimental results showed that the proposed model could not only efficiently detect the mean shifts but also accurately identify the variables that have deviated from their original means. Jiang et al. [14] proposed a neural network–numerical fitting (NN–NF) model to recognize different control chart patterns. Specifically, a BPN was first used to recognize CCPs preliminarily, and while numerical fitting method was then adopted to estimate the parameters and specific types of the patterns. Experiment results showed that the proposed NN–NF model can not only substantially improve the recognition rate but also significantly reduce the training time. El-Midany et al. [15] proposed a framework for multivariate process CCPR. The proposed methodology used the neural networks to recognize a set of subclasses of multivariate abnormal patterns, identify the responsible variable(s) on the occurrence of abnormal pattern and classify the abnormal pattern parameters. Salehi et al. [16] proposed a hybrid learning-based model consisting of two modules. In the first module using a SVM-classifier, type of unnatural pattern can be recognized. Then by using three ANNs for shift mean, trend and cycle it can be recognized magnitude of mean shift, slope of trend and cycle amplitude for each variable simultaneously in the second module. Du et al. [17] presented a SVM ensemble approach, in which several SVMs were jointly used for classifying the source(s) of process mean shifts in multivariate control charts. Experimental results indicated that the proposed approach can perform effectively for classifying the source(s) of process mean shifts. He et al. [18] proposed a DT-based model for bivariate process mean shift monitoring and fault identification. Specifically, two DT classifiers based on the C5.0 algorithm were built, one for process monitoring and the other for fault identification. Simulation results showed that the proposed model can not only detect the mean shifts but also give information on the variable or subset of variables that cause the out-of-control signals and its/their deviate directions.

It should be noted that the above-mentioned approaches are developed for identification of basic CCPs Therefore, they are incapable for identification of concurrent CCPs (i.e., two or more basic CCPs occurring simultaneously), which may be associated with different assignable causes. Such concurrent CCPs are commonly encountered in practical manufacturing processes. For example, in the turning process, the tool may wear (resulting in a trend pattern) and this pattern may be concurrent with a cycle pattern (resulting from other assignable causes that come and go on a regular basis, e.g., unstable material). In order to identify effectively each assignable cause that is present, it is essential to identify each “target” basic CCP separately. Due to the pattern interaction and resultant complexity, concurrent CCPs are more difficult to recognize than basic CCPs. Guh and Tannock [19] proposed a sequential neural network-based approach to detect and discriminate typical unnatural CCPs. Experimental results indicated that the proposed approach is capable of recognizing single and concurrent abnormal control chart patterns and also identifying key parameters of the specific CCP(s) involved. Chen et al. [20] proposed a hybrid approach that integrates wavelet transform and BPN to identification of concurrent CCPs. In their hybrid system, concurrent CCPs are first decomposed into different levels of basic CCPs by a wavelet transform. Then, the corresponding features are presented into BPN-based classifiers for CCPR. The same idea of using wavelet filtering for pattern decomposition and reconstruction was also investigated by Wang et al. [21] and Du et al. [22]. There are several problems with these wavelet-based methods including the determination of the proper level for decomposition, heavy computation burden involved in calculating wavelet basis, the selection of threshold for the wavelet coefficients, etc. Recently, independent component analysis (ICA) methods, which have been widely used in fields such as mobile communication [23,24], have been demonstrated as effective methods for identification of concurrent CCPs. Wang et al. [25] developed a hybrid approach based on ICA and DT to identify concurrent CCPs. Experimental results showed that the proposed approach was very successful to handle most of the concurrent CCPs. However, the developed method had two limitations in real-world applications: it needs at least two concurrent CCPs to reconstruct their constituent basic CCPs and it may be incapable to identify the concurrent CCP incurred by two correlated process (“upward shift” and “upward trend” as well as “downward shift” and “downward trend”). Lu et al. [26] proposed an ICA–SVM scheme where the FastICA algorithm [27] was first applied to decompose the concurrent CCPs. Then, a trained SVM was employed to recognize the basic CCP type for each component. However, there are several shortcomings. FastICA cannot provide a unique solution as different initial conditions will result in different solutions. Moreover, these ICA-based methods require the number of observed manufacturing process data equal to or greater than the number of independent components (ICs). Inherent permutation and scaling ambiguities are also common issues for those approaches [28], which result in the incorrectly estimated sign of recovered ICs. For the concurrent CCPs, a typical example is that the upward trend patterns will be incorrectly identified as the downward trend patterns. Therefore, these ICA-based methods may be inefficient in dealing with the concurrent CCPs. More recently, singular spectrum analysis (SSA) [29] has also been reported in the literature as another effective method for identification of concurrent CCPs. Gu et al. [30] proposed a novel approach based on SSA and learning vector quantization network (LVQ) to identify concurrent CCPs. However, the parameter selection of the SSA has not been discussed in detail. Moreover, it will be time consuming and costly to accumulate enough samples to train a LVQ in practice. Further tests indicate that the solution of LVQ is not stable in that different results will be derived even with same training parameters. These shortcomings stimulated Xie et al. [31] to investigate a new method based on SVM for this problem. Xie et al. [31] proposed a novel hybrid SSA–SVM scheme integrating SSA and SVM. The SSA allows decomposition of a mixed signal given only one observation, thus it is suitable for univariate manufacturing process. Compared with the traditional ICA-based methods, The SSA does not cause any inherent permutation and scaling ambiguity in the ICs. Therefore it has the
advantage in the separation of the concurrent CCPs. Furthermore, it is easy to reconstruct ICs by only tuning two parameters.

A review of the related literature also indicates that most of the previous work in the area of applying ANN/SVM/DT to CCPR have emphasized pattern detection and identification rather than more detailed pattern parameter information, such as shift magnitude, trend slope, cycle amplitude, etc., which is very useful for quality practitioners to search the assignable causes that give rise to the out-of-control situation. Unfortunately, there are fairly limited research of this issue in the literature [19,32–34]. According to the literature review given above, the aim of this study is to develop a hybrid approach that integrates extreme-point symmetric mode decomposition (ESMD) [35] with extreme learning machine (ELM) [36,37] to identify concurrent CCPs and in addition to accurately quantify the major CCP parameter of the specific basic CCPs involved. The experimental results indicate that the proposed model can effectively identify not only concurrent CCPs but also basic CCPs. Meanwhile, the major CCP parameter of the identified concurrent CCP can also be accurately quantified. Thus, more hidden information about the behavior of the underlying manufacturing processes can be mined. From the given results, the proposed model may be a promising tool for identification and quantification of concurrent CCPs.

2. Generation of basic and concurrent CCPs

The common types of CCP exhibited on statistical process control charts have been formalized in early literature. In most literature, the following seven typical types of basic CCP, i.e., Upward and Downward Shift Pattern (USP and DSP); Upward and Downward Trend Pattern (UTP and DTP); Cyclic Pattern (CP); Systematic Pattern (SP); Natural Pattern (NP), are usually addressed. This study uses simulations to generate the required sets of CCP examples for training, testing, and performance evaluation of the proposed model. The mathematical expressions for CCP generation are expressed in a general form that consists of process mean, common cause variation and special disturbance from assignable causes.

\[ X_t = \mu_0 + n_t + d_t \]  

where \( t \) is the time of sampling, \( X_t \) is a sample value at time \( t \), \( \mu_0 \) is the process mean when the process is in-control, \( n_t \) is a normal distribution at time \( t \) following a normal distribution with zero mean and standard deviation \( \sigma_n \), where \( \sigma_n \) is the process standard deviation when the process is in-control and is fixed at one in this study, \( d_t \) is the special disturbance at time \( t \) (zero when no unnatural pattern present).

For USPs/DSPs
\[ d_t = u s \]
where \( u \) is a parameter to determine the position of shifting (\( u = 0 \) before shifting, \( u = 1 \) after shifting) and \( s \) is the shift magnitude in terms of \( \sigma_0 \).

For UTPs/DTPs
\[ d_t = d t \]
where \( d \) is the trend slope in terms of \( \sigma_0 \).

For CPs
\[ d_t = a \sin(2\pi t / \Omega) \]
where \( a \) is the cycle amplitude in terms of \( \sigma_0 \) and \( \Omega \) is the cycle period (\( \Omega = 8 \) in this study).

For SPs
\[ d_t = g(1 - j)^i \]
where \( g \) is the magnitude of the systematic pattern in terms of \( \sigma_0 \), determining the fluctuations above or below the process mean.

The concurrent CCPs were a combination of any two of the above-mentioned six basic unnatural CCPs. Thus, 13 types of concurrent CCP, namely USP mixed with UTP (USP+UTP), USP mixed with DTP (USP+DTP), USP mixed with SP (USP+SP), DSP mixed with UTP (DSP+UTP), DSP mixed with DTP (DSP+DTP), USP mixed with SP (DSP+SP), UTP mixed with CP (UTP+CP), UTP mixed with SP (UTP+SP), DTP mixed with CP (DTP+CP), DTP mixed with SP (DTP+SP), CP mixed with SP (CP+SP) are addressed in this study. A control chart with concurrent patterns may mean that the process is suffering from more than one assignable cause, although some assignable causes may produce more than one pattern. In either case it is necessary for diagnostic purposes to individually recognize each specific pattern from the concurrent pattern. Possible causes include items from different suppliers, machines, or workers. The equation for concurrent CCP generation can be derived based on the equation for basic CCP generation, i.e., Eq. (1), with small modification and is expressed as follows [19]:

\[ X(t) = \mu_0 + n_t + d_{1,t} + d_{2,t} \]  

where \( d_{1,t} \) and \( d_{2,t} \) are two special assignable causes variation that occur at the same time \( t \) Noting that here in \( d_{1,t} \) and \( d_{2,t} \) is just playing the same role in Eq. (2) as \( d_t \) did in Eq. (1), and they jointly give rise to a concurrent CCP.

This study uses simulations to generate the required sets of CCP examples for training, testing, and performance evaluation of the hybrid ESMD–ELM model. The mathematical expressions for CCP generation are detailed in Eq. (1) for basic CCPs and Eq. (2) for concurrent CCPs. Following the generation of process data, data \( X(t) \) generated using Eqs. (1) or (2) were standardized by subtracting the mean and dividing it by the standard deviation

\[ Y_t = \frac{X_t - \mu}{\sigma} \]  

where \( Y_t \) denotes the value standardized from \( X_t \), resulting in a data set with a zero mean and a standard deviation of unity. Standardization is a widespread quality control practice, through which process data are transformed to a constant range, approximately between \(-3 \) and \( 3 \), regardless of the values included in the data prior to standardization. This transformation is necessary because it can ensure the input data within a particular range.

Following standardization, an encoding scheme [19] was implemented to smooth out the effect of the random noise hidden in the original process data by dividing a selected variable range \([-7.625, +7.625]\) into 61 zones with a width of 0.25 (each returning an integer code), meanwhile retaining the main feature of the original process data. Any data falling below \(-7.625\) and above \(+7.625\) was coded as \(-31\) and \(+31\), respectively. Although the process data were set mostly within \([-3, +3]\) in this study, a relatively large range was used in the encoding procedure to allow for possible large variations due to special causes. The coding scheme is summarized as follows:

\[ Y_t \geq 7.625 \quad Z_t = +31 \]
\[ -7.625 + 0.25(i - 1) < Y_t \]
\[ -7.625 + 0.25j \quad Z_t = -31 + j \]
\[ Y_t \leq -7.625 \quad Z_t = -31 \]

Please cite this article as: W.-A. Yang et al., Identification and quantification of concurrent control chart patterns using extreme-point.... Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.068
3. Methodology

3.1. Extreme-point symmetric mode decomposition

ESMD, originally developed by Wang and Li [25], is an empirically based data-analysis method, which decomposes a signal in several independent components whose sum is the original signal. This technique is based on natural phenomenon that a high-frequency small wave ride on a low frequency big wave and the small one is almost symmetric about its crest and trough relative to the big one. In what follows the ESMD process of a signal $Y = \{y_1, y_2, ..., y_n\}$ is described briefly below.

1. Set $k$ (time of sifting) to 1.
2. Find all local extreme points of the original signal $Y = \{y_1, y_2, ..., y_n\}$ and then denote them by $E_k(1 \leq i \leq n)$, where $n$ is the number of all the local extreme points.
3. Connect all the adjacent $E_k(1 \leq i \leq n)$ with line segments, and then denote their midpoints by $F_i(1 \leq i \leq n-1)$.
4. Add a left boundary midpoint ($F_0$) and a right boundary midpoint ($F_n$) through the linear interpolation method with small modification in the interpolation styles in order to make the boundary much more stable.
5. Construct $m$ interpolating curves $L_1, L_2, ..., L_m$ ($m \geq 1$) with all these ($n+1$) midpoints, and calculate the mean value of these $m$ interpolating curves using the equation
   $$L_k = (L_1 + L_2 + ... + L_m)/m.$$  
6. Increment $k$ by 1, and repeat steps 2–5 for $|Y - L^k|$ until $|L^k| \leq \varepsilon$ or maximum number of sifting ($K$) that can be tentatively selected within the integer interval of $\{K_{min}, K_{max}\}$ has been reached, where $\varepsilon$ represents the maximum allowable error, where $\sigma_k$ is the standard deviation of $Y$. At this time, the first mode $M_1$ is thus subtracted from the original signal: $Y = Y - L^1$. The difference between $Y$ and $M_1$ is called as the first residue $r = (r_1, r_2, ..., r_k)$. It is treated as the new signal and subjected to the same sifting process. This process is repeated until the last residue $r = (r_1, r_2, ..., r_k)$ has at most one extremum (excluding the ends) or becomes constant.
7. Reset $K$ to a new candidate maximum number of sifting that can also be tentatively selected within the integer interval of $\{K_{min}, K_{max}\}$, and repeat steps 1–6. Then calculate the variance $\sigma^2$ of $(Y - r)$ and plot a figure with $v = \sigma/\sigma_0$ and $k$.
8. Find the number of $k_0$ that corresponds to the minimum $v = \sigma/\sigma_0$. Then let $k = k_0$, repeat steps 2–7, and output the whole modes. At this time, the last residue $r$ is the finally optimal adaptive global mean curve.

It should be noted that in step 5, besides the permitted error $\varepsilon$, it can also adjust the maximum sifting times $K$ to control the decomposing process. On the one hand, if $\varepsilon$, is the unique controlling parameter, it may lead the decomposition to endless loop. On the other hand, if $K$ is the unique controlling parameter, it is hard to know about the symmetric properties of each mode. Hence, Wang and Li [35] suggested using both $\varepsilon$ and $K$. To obtain a series of highly reliable modes, Wang and Li [35] also suggested setting $\varepsilon$ to be a very small value and controlling the decomposing process by selecting $K$ within a finite integer interval such that the finally last residue $r$ (adaptive global mean curve) is an optimal one. In this study, $\varepsilon$ is set as $\sigma_0/1000$, $K_{min}$ and $K_{max}$ are set as 1 and 100, respectively, the number of interpolating curves is set as 2. For additional information about the ESMD, please refer to Wang and Li [35].

3.2. Extreme learning machine

ELM is a recently emerged learning paradigm for generalized single-hidden-layer feed-forward neural networks (SLFNs) [36,37], which overcomes the problems of traditional feedforward neural network learning algorithms such as intensive human intervene, slow learning speed, poor learning scalability. The main advantage of ELM is that the hidden layer (or called feature mapping) need not be tuned. In ELM, all the hidden node parameters are independent from the target functions or the training datasets and these hidden node parameters can be randomly generated even before ELM sees the training data. Owing to its unique operating mechanism and wonderful performance, ELM has been successfully applied as a simple unified algorithm for regression, binary and multiclass classification to many real-world applications [36–43]. What follows is a brief overview of ELM for regression, binary and multiclass classification.

Let $(x_i, y_i), x_i \in \mathbb{R}^p, y_i \in \mathbb{R}, i = 1, 2, ..., N$ be a set of training samples, which contains $N$ distinct examples, and each example has $n$ inputs and $m$ outputs. Then, for the randomly assigned values for the weight vector as $a_i = (a_{i1}, a_{i2}, ..., a_{in})^T \in \mathbb{R}^p$ and the bias $b_i \in \mathbb{R}$ connecting the input layer to the $i$th hidden node, the SLFNs with $n$ number of hidden nodes approximate the input examples with zero error if and only if there exists an output weight vector $w = (w_1, w_2, ..., w_n)^T \in \mathbb{R}^m$ connecting the hidden nodes to the output node such that the following condition holds:

$$y_i = \sum_{i=1}^{c} w_i G(a_i, b_i, x_i) \quad (i = 1, 2, ..., m)$$  

where $G(a_i, b_i, x_i)$ is the output of the $i$th hidden node for the input example $x_i$. The above system of linear equations can be, equivalently, written in matrix form as

$$HW = y$$  

where is the hidden layer output matrix of the network that can be expressed as

$$H = \begin{bmatrix} G(a_1, b_1, x_1) & G(a_2, b_2, x_1) & \cdots & G(a_m, b_1, x_1) \\ G(a_1, b_1, x_2) & G(a_2, b_2, x_2) & \cdots & G(a_m, b_2, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ G(a_1, b_1, x_m) & G(a_2, b_2, x_m) & \cdots & G(a_m, b_m, x_m) \end{bmatrix}$$

and $y = (y_1, y_2, ..., y_m)^T \in \mathbb{R}^m$ is the vector of desired outputs.

For the randomly assigned values of the parameters $a_i = (a_{i1}, a_{i2}, ..., a_{in})^T \in \mathbb{R}^p$ and $b_i \in \mathbb{R}$, training the SLFN is equivalent to obtaining a least squares solution $w$ of the linear system, namely Eq. (5). In fact, $w$ is determined to be the minimum norm least squares solution of Eq. (5) which can be explicitly obtained to be [36]

$$w = H^+ y$$

where $H^+$ is the Moore–Penrose generalized inverse [44] of the matrix $H$. Finally, by obtaining the solution $w = (w_1, w_2, ..., w_n)^T \in \mathbb{R}$, the regression estimation function $f(\cdot)$ is determined to be for any input example $x \in \mathbb{R}^p$

$$f(x) = (G(a_1, b_1, x), G(a_1, b_1, x), \ldots, G(a_1, b_1, x)) w$$  

However, for binary classification problem, the decision function $f(\cdot)$ will become

$$f(x) = \text{sign}(G(a_1, b_1, x), G(a_1, b_1, x), \ldots, G(a_1, b_1, x)) w$$

For multiclass classification with $k$ number of classes, let ELM have $k$ number of nodes in the output layer. Then, for every input example $x_i \in \mathbb{R}^p$, the network output will be equal to the target output $(y_{i1}, y_{i2}, ..., y_{ik})^T \in \mathbb{R}^k$ if

$$HW = Y$$  

Please cite this article as: W.-A. Yang et al., Identification and quantification of concurrent control chart patterns using extreme-point,..., Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.068
where

$$W = [w_1, w_2, \ldots, w_k] = \begin{bmatrix}
w_{11} & w_{12} & \cdots & w_{1k} \\
w_{21} & w_{22} & \cdots & w_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
w_{n1} & w_{n2} & \cdots & w_{nk}
\end{bmatrix}$$ (12)

$$Y = [y_1, y_2, \ldots, y_k] = \begin{bmatrix}
y_{11} & y_{12} & \cdots & y_{1k} \\
y_{21} & y_{22} & \cdots & y_{2k} \\
\vdots & \vdots & \ddots & \vdots \\
y_{n1} & y_{n2} & \cdots & y_{nk}
\end{bmatrix}$$ (13)

where \(w = (w_{1j}, w_{2j}, \ldots, w_{nj})^T \in \mathbb{R}^n\) is the weight vector connecting the hidden nodes with the jth output node and \(y_j = (w_{1j}, w_{2j}, \ldots, w_{mj})^T \in \mathbb{R}^m\) is the output vector corresponding to the jth output node. For any test example \(x \in \mathbb{R}^n\), its predicted class label will be determined by

$$\arg \max_{f_j(x)}$$

where

$$f_j(x) = (G(a_1, b_1, x), G(a_2, b_2, x), \ldots, G(a_m, b_m, x))$$

Note that, once the values of the weight vector \(a_i = (a_{i1}, a_{i2}, \ldots, a_{in})^T \in \mathbb{R}^n\) and \(b_i \in \mathbb{R}\) are randomly assigned at the beginning of the learning algorithm they remain fixed and therefore the matrix \(H\) remains unchanged. Further, since the sigmoidal, radial basis, sine, cosine and exponential functions are infinitely differentiable in any interval of definition they can be chosen as activation functions [36].

4. The proposed hybrid ESMD–ELM model

In this section, a hybrid ESMD–ELM model that integrates ESMD with ELM is designed to be applied to an automated identification and quantification of concurrent CCPs without the need for human intervention. In today’s manufacturing environment, with rapid development in data acquisition and inspection technologies, a high-speed automatic data acquisition and inspection system can obtain dozens or even hundreds of measurements of every unit. Therefore, this study addresses the individual data. Although the utilization of individual data usually yield a short in-control ARL (equivalent to a high Type I error), it can generally detect unnatural CCPs quickly (i.e. low Type II error). In a high-speed automatic production scenario, detecting and correcting the incipient problem as early as possible is important to avoid the creation of additional defective products. Moreover, individual data are often collected for process monitoring in computer integrated manufacturing environments. Consequently, the X chart is the major chart considered in the present study. In addition, three assumptions are made here. First, only one quality characteristic variable is considered. Second, the process standard variation (\(\sigma\)) does not change. Third, two- and only two-basic CCPs began at the same time.

4.1. Model architecture

The proposed hybrid ESMD–ELM model consists of two sequential modules – Module I and Module II (Fig. 1). Module I consists of a ESMD and an ELM for multiclass classification (ELM #1) that was designed and trained to identify various types of concurrent CCPs through two major procedures, namely decomposing and recognizing. In the first procedure, each concurrent CCPs sample is decomposed by the ESMD into several sources of basic CCPs. Those separated components are presented to the trained ELM (ELM #1) and the corresponding basic CCPs are then identified. If the identified results contain the target basic unnatural CCPs, and the others are identified as NP, the concurrent CCP sample is correctly classified. Otherwise, it is misclassified.

Module II consists of six specialist ELMs (ELMs #2, #3, #4, #5, #6 and #7) for regression that are designed and trained to quantify the major parameter of the unnatural CCPs. ELMs #2 and #3 were designed to quantify the upward and downward shift magnitude, respectively. ELMs #4 and #5 were designed to quantify the upward and downward trend slope, respectively. ELMs #6 was designed to quantify the cycle amplitude. ELMs #7 was designed to quantify the systematic magnitude. For example, if an out-of-control situation identified by Module I is “USP-DTP”, the input data vector will be further analyzed by ELMs #2 and #5, to quantify the upward shift magnitude and the downward trend, respectively. The main reason for adopting this modular design strategy was to split the original problem of quantifying the major parameter of the unnatural CCPs into more manageable sub-problems, and thus improve the performance of Module II. If only one ELM for regression was used to perform all of the quantification functions required, the ELM for regression would have to be large and complex and therefore making the ELM training very difficult due to interference among different CCP categories. The training and testing results of Module II demonstrated that this design strategy was successful.

The hybrid ESMD–ELM model monitors the manufacturing process on-line by using a moving window analysis approach [45]. Data vectors transformed from a sequence of process data are presented to the hybrid ESMD–ELM model in the moving observation window, which is moved forward by one process observation at a time, representing a single sampling interval. In other words, in the monitoring procedure, the input to the hybrid ESMD–ELM model is a sequence of moving windows of process data. When concurrent CCPs begin to appear in the discrete data series, the pattern features strengthen gradually as the observation window moves forward through the process data stream.

4.2. Selection of observation window size

As aforementioned, the manufacturing process is monitored by the hybrid ESMD–ELM model using the moving window analysis method. The number of data numbers contained in a specific concurrent CCP example, referred to as the observation window size herein, can significantly influence the identification and quantification performance of the proposed hybrid ESMD–ELM model. A small observation window size will typically detect unnatural CCPs quickly, and may also cause a short in-control ARL (equivalent to a high Type I error), because the amount of data numbers available is insufficient (1) to enable the ESM of Module I to accurately extract several separated empirical modes from the original input data vector; (2) to present all recognition features of the unnatural CCP(s) to the ELM #1 of Module I, and (3) to expose all major parameter features of the unnatural CCP(s) to ELMs #2, #3, #4, #5, #6 and #7 of Module II. However, a large observation window size usually prolongs the response time required to detect CCPs (high Type II error, or long out-of-control ARL). Moreover, a large observation window size can also enlarge the size of the ELMs quickly. The appropriate observation window size here should balance Type I and Type II errors. In the literature, several observation window sizes (ranging from 5 to 60) have been suggested. In this study, a relatively large observation window size of 56 got selected after considerable preliminary experimentation, reflecting the special concern that concurrent CCPs need more data points to be included in the observation window in order to fully expose their complicated features. A similar observation window size was successfully used by Guh and Tannock [19], Guh and Hsieh [32] and Pham and Oztanzeil [46,47].
4.3. ELM training and testing

4.3.1. Training and testing results of Module I

The training example set for Module I (ELM #1) consists of two portions, namely concurrent CCP examples and basic CCP examples. There were 9750 concurrent CCP examples, comprising 13 groups of mixed cases (see Table 1), in the training set. According to the preliminary research, even if the training set does contain basic CCP examples only, without any concurrent CCP examples, the trained ELM #1 will still identify the concurrent CCPs. This finding may be partially explained by the fact that the trained ELM #1 can to some extent generalize the capability of identifying basic CCPs to identifying concurrent CCPs. Therefore, in order to enhance the identification performance of the ELM #1 here, basic CCP examples were also included in the training set. There were 4860 basic CCP examples in the training set (see Table 2). The parameter ranges of basic CCP examples were similar to those of concurrent CCP examples, with smaller parameter increments. Together with 2500 natural patterns, the training set of the ELM #1 contained a total of 17,110 example CCPs. The parameter settings in Tables 1 and 2 were reasonable ranges of magnitudes encountered in practice. On the one hand, CCPs with parameters smaller than these ranges are very difficult to identify. On the other hand, CCPs with larger parameters can be identified easily by traditional methods (e.g., control limits). Notably, the CCP parameters are measured in terms of the $\sigma_0$ of the observations. It is worth noting that each concurrent CCP example stream herein contains 56 data numbers, representing 56 consecutive data points in a process.

To allow large shifts to be detected quickly, the shift position was set at point 44 from the left edge of the observation window ($s \geq 2.25\sigma_0$ or $s \leq -2.25\sigma_0$). For moderate and small shifts ($s \leq 2.00\sigma_0$ or $s \geq -2.00\sigma_0$), the shift position was set in the middle of the observation window (point 30 from the left edge of the window). For the same reason, in order to ensure the large trends ($d \geq 0.18\sigma_0$ or $d \leq -0.18\sigma_0$) to be detected quickly, the starting points for large trends were set at point 16. With respect to trends with small slopes ($d \leq 0.16\sigma_0$ or $d \geq -0.16\sigma_0$), the starting points were set at the beginning of the observation window (point 1). As for the cycle patterns, the starting point can start at any point within the period. Note that although the pattern starting points of shift and trend patterns were fixed during ELM training, the trained ELM #1 can identify the shift and trend patterns with other starting points in the observation window through its generalization characteristic. This generalization capability is critically important because the starting point of process abnormality is generally unknown in a practical process-monitoring scenario.

The popular Gaussian kernel function $K(x_i, x_j) = \exp(-\gamma|x_i-x_j|^2)$ is applied as kernel function for the ELM #1 of Module I. There are two parameters associated with Gaussian kernel function: the penalty parameter $C$ and the kernel parameter $\gamma$. These parameters control the error-margin trade-off and play a crucial role in ELM performance [37]. In order to achieve good generalization performance, the penalty parameter $C$ and the kernel parameter $\gamma$ of ELM need to be chosen appropriately. In this study, the grid-search [48] is used for determining the best combination of $C$ and $\gamma$ so that the ELM #1 can accurately recognize unseen data. All possible combinations of $(C, \gamma)$ with exponentially growing sequences, $(2^{-24}, 2^{-23}, \ldots, 2^{-4}, 2^4)$, were tried, and the one with the best accuracy was chosen as the optimal parameter. There are two advantages for using the grid-search. First, the grid-search is an exhaustive parameter search. Second, the computational time to find the optimal parameter values by the grid-search is not much more than those of advanced methods (e.g., genetic algorithm and particle swarm optimization) since there are only two parameters.

Using the simulated training and testing examples, the optimal $(C, \gamma)$ was determined as $(2^{13}, 2^{55})$ through the grid-search procedure. The classification rate (i.e., the number of correctly identified training examples/total number of training examples)
The three values in parentheses correspond to initial value, and increment, respectively.

Table 2 Basic CCP training examples for the ELM#1.

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Parameter description</th>
<th>Example quantity of each parameter setting</th>
<th>Example quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>USP</td>
<td>s=(1.0, 3.0, 0.25)</td>
<td>90</td>
<td>810</td>
</tr>
<tr>
<td>DSP</td>
<td>s=(−3.0, −1.0, 0.25)</td>
<td>90</td>
<td>810</td>
</tr>
<tr>
<td>UTP</td>
<td>d=(0.10, 0.26, 0.02)</td>
<td>90</td>
<td>810</td>
</tr>
<tr>
<td>DTP</td>
<td>d=(−0.26, −0.30, 0.02)</td>
<td>90</td>
<td>810</td>
</tr>
<tr>
<td>CP</td>
<td>a=(1.0, 3.0, 0.25)</td>
<td>90</td>
<td>810</td>
</tr>
<tr>
<td>SP</td>
<td>g=(1.0, 3.0, 0.25)</td>
<td>90</td>
<td>810</td>
</tr>
</tbody>
</table>

The three values in parentheses correspond to initial value, final value, and increment, respectively.

for the training examples was 98.1%, indicating that the training was successful. The trained ELM #1 was then tested using the testing example set, generated by Eqs. (1) and (2), with different random number seeds. The classification rate for the testing example set was 96.6%, indicating that ELM #1 exhibited strong capability for identifying concurrent CCP types. Note that these testing results represent the performance of the proposed model only in an off-line (static) recognition scenario in which Module I identifies each testing example CCP in its entirety. In a practical process-monitoring scenario, however, the off-line recognition scheme is not realistic because the starting point of a CCP is usually unknown, making the analyzed CCPs generally incomplete. Hence, the testing results cannot fully reflect the performance of the proposed model in an on-line (dynamic) process-monitoring scenario. Section 5 evaluates the performance of the proposed model in an on-line (dynamic) process-monitoring scenario using the moving window analysis.

As mentioned in Section 4.1, each concurrent CCPs sample is decomposed by the ESMD into several sources of basic CCPs, and then those decomposed components are classified by the trained ELM #1 respectively. Hence, the final classification rate for concurrent CCPs identification depends on performance of the trained ELM #1 for the typical type of basic CCPs.

4.3.2 Training and testing results of Module II

The function of ELMs #2, #3, #4, #5, #6 and #7 in Module II is to further quantify the major parameter of these unnatural basic CCPs identified by the ELM #1 of Module I. In training the ELMs #2, #3, #4, #5, #6 and #7 in Module II, only basic CCP examples were employed. For each CCP parameter setting, 400 CCP examples were generated using Eq. (1). Thus, 4400 upward shift (s = 0.75 − 3.25), 4400 downward shift (s = −3.25 to −0.75), 4400 upward trend (d = 0.08 − 0.28), 4400 downward trend (d = −0.28 to −0.08), 4400 cycle (a = 0.75 − 3.25) and 4400 systematic (g = 0.75 − 3.25) CCP examples were employed as the training set to train the ELMs #2, #3, #4, #5, #6 and #7, respectively. Details of the training examples are given in Table 3. Also it should be worth noting that each concurrent CCP example stream herein contains 56 data numbers, representing 56 consecutive data points in a process.

The popular Gaussian kernel function \( K(x_i, x_j) = \exp(-\gamma|x_i - x_j|^2) \) is also applied as kernel function for ELMs #1 in Module II. In the training process, the penalty parameter \( C \) and the kernel parameter \( \gamma \) for ELMs #2, #3, #4, #5, #6 and #7 were also chosen by applying the grid-search method. The optimal \((C, \gamma)\) for ELMs #2, #3, #4, #5, #6 and #7 are \( (2^2, 2^2), (2^2, 2^2), (2^2, 2^2), (2^2, 2^2) \) and \( (2^4, 2^4) \), respectively. Each trained ELM was then tested using the testing example sets, also generated by Eq. (1), but with different random number seeds. The experimental results indicated that the most significant factor affecting the training result in the situation addressed herein is the output range setting of the ELMs #2, #3, #4, #5, #6 and #7. With the penalty parameter \( C \) and the kernel parameter \( \gamma \) kept unchanged, the average quantification errors were minimized by scaling the data of the training examples into a ELM input range between \([-1, +1]\) and an ELM output range between \([-0.5, +0.5]\) and \([-0.3, +0.3]\) [−0.3, +0.3], \([-0.3, +0.3]\) [−0.04, +0.04], \([-0.2, +0.2]\] and \([-0.3, +0.3]\) before presentation to ELMs #2, #3, #4, #5, #6 and #7 respectively. After ELMs #2, #3, #4, #5, #6 and #7 had generated a scaled output result, this result was de-scaled into the original unit. The test results showed that the overall errors of the major CCP parameter quantification obtained by ELMs #2, #3, #4, #5, #6 and #7 were 0.0841, 0.0819, 0.0053, 0.0068, 0.1356 and 0.1297, respectively. The overall error of the major CCP parameter quantification was 0.0739. Therefore, Module II can effectively quantify the major parameters of CCPs.

The ELM program executed in a MATLAB 8.0 development environment was applied to develop the ELMs #1, #2, #3, #4, #5, #6 and #7. ELM source codes are freely available from the website (http://www.ntu.edu.sg/home/egbhuang/).

5. Performance evaluation

After the training and testing of individual ELM was completed, the performance of the proposed hybrid ESMD–ELM model was evaluated in a dynamic process-monitoring scenario. A sequence of simulated process data sets \( X_t (i = t − 55, t − 54, …, t − 1, t) \) was presented to the model in the moving window, which was incremented forward by one process observation representing a single sampling interval. The process began under an in-control condition. The initial observation window contains no unnatural CCP data. Unnatural CCPs start to appear as the observation window moves through the discrete time series (i.e. unnatural CCPs begin at the 57th point in the discrete time series). This approach is considered to be practical because, with on-line...
process monitoring applications, an out-of-control condition often occurs after a period in which the process is in-control, and the actual starting point of the out-of-control condition is generally unknown. The performance evaluation involves the following steps:

1. Set \( t \) (time of sampling) to 56.
2. Collect the most recent 56 coded sample data sets, \( X_t \) (\( t = 55, 54, \ldots, 1 \)), from the process.
3. Standardize the input data \( X_t \) with Eq. (3). Denote the standardized data by \( Z_t \).
4. Encode the process data \( Z_t \) with the coding scheme in Eq. (4).
5. Generate an input vector \( V_t \), which contains the coded process data \( Z_t - Z_{55}, Z_{54}, \ldots, Z_1, Z_t \).
6. Analyze the \( V_t \) using the ESMD of Module I, and thus several separated empirical modes and a residue \( r \) are generated.
7. Those separated modes and the residue \( r \) are classified by the trained ELM #1.
8. If the identified results contain the target basic unnatural CCPs, and the others are identified as NP, conclude that an unnatural concurrent CCP has been detected in the current observation window and go to step 9. Otherwise, increment \( t \) by 1, and return to step 2. Repeat this procedure until an unnatural concurrent CCP is detected.
9. Determine the ARL of the shift. ARL = alarm point – shift starting point (=57 in this study) + 1.
10. Present those separated modes in the current observation window to the corresponding ELMs of Module II to quantify the major parameters of the concurrent CCP identified by Module I.

Early detection of unnatural CCPs can help minimize troubleshooting, which can be achieved by the accurate identification and quantification of unnatural CCPs provided that appropriate process knowledge is available. This study employs the percentage of correctly identified concurrent CCPs (CIC), ARL, the standard deviation of run length (SRL) and overall quantification error (OQE) as performance indexes. The ARL can be regarded as the identification speed or sensitivity of the proposed model. A small ARL (i.e., early detection of unnatural CCPs) makes the proposed model easier and faster to identify the assignable cause that adversely affects the behavior of the manufacturing process, leading to a rapid rate of continuous quality improvement. The CIC can be regarded as the identification accuracy of the proposed model. A high CIC makes the proposed model more accurate to identify the types of unnatural CCP. The SRL indicates ARL dispersion and can be regarded as the stability of the identification speed. A small SRL implies that the proposed model detects the unnatural CCPs with risk factors in a more precise and predictable manner. The OQE can be regarded as the quantification accuracy of the proposed model. A low OQE makes the proposed model more accurate to quantify the major parameter of the CCP identified by Module I.

### 5.1. Performance evaluation for concurrent CCPs

The performance of the proposed hybrid ESMD–ELM model for concurrent CCPs was evaluated with 1000 independent simulation runs for each parameter combination setting of each mixed case (i.e., two- and only two-basis CCPs occurring simultaneously). As in the training example set of the ELMs, 13 groups of mixed cases were employed (see Table 1). To evaluate the performance of the proposed hybrid ESMD–ELM model in a dynamic process-monitoring scenario concisely, the performance evaluation for concurrent CCPs in this study focused only on the mixed cases in which one basic CCP and another basic CCP begin at the same time. If the two basic CCPs do not begin at the same time, the first basic CCP could have been detected and removed practically as a single CCP before the second basic CCP occurs. Therefore, it was quite reasonable to assume herein that the two basic CCPs begin at the same time. Table 4 summarizes the CIC (%), ARL and SRL performances of the proposed model for concurrent CCPs in the column “ELM with the sub-column Gaussian kernel”. Only average performance values are listed here because of the very large number of parameter combinations of concurrent CCPs. Noting that the ARL, SRL and CIC were calculated only for the correctly identified concurrent CCP (i.e., two basic CCPs were respectively classified correctly by the ELM #1 of Module I). As seen in Table 4, the proposed hybrid ESMD–ELM approach possesses good identification capability for concurrent CCPs, but not without weak points for some specific pattern combinations. It was relatively difficult for ELM #1 to identify a USP mixed with a UTP, a DTP mixed with a DTP, a UTP mixed with a CP, and a CP mixed with a CP. The average CIC(%) values for concurrent CCPs with these four pattern combinations were less than 90%. This low recognition accuracy can be explained by the fact that there exists to some extent feature similarity between different basic CCPs in a real-time process-monitoring scenario.

Table 5 summarizes the average errors of the CCP parameter quantification obtained by ELMs #2, #3, #4, #5, #6 and #7 of Module II in the column “ELM with the sub-column Gaussian kernel”. It can be seen in Table 5 that the overall performance of the CCP parameter quantification is fairly good. Noting that the OQE results herein were calculated only for the concurrent CCPs where two constituent basic CCPs were individually identified correctly by Module I. Also noting that only basic CCP examples were employed to train ELMs #2, #3, #4, #5, #6 and #7 of Module II, and thus the quantification performance herein might be improved by means of additional training. Concurrent CCP examples could be attached to the previous basic CCP training set to improve the quantification capability of the ELMs #2, #3, #4, #5, #6 and #7, respectively, with regard to concurrent CCPs.

### 5.2. Performance evaluation for basic CCPs

The input vector containing only one basic CCP is also frequently encountered in practical manufacturing processes, it is thus necessary to investigate the robustness of the proposed hybrid ESMD–ELM model for basic CCPs. In this section, after performance evaluation for concurrent CCPs was complete, the proposed hybrid ESMD–ELM model was then evaluated using basic CCPs. Similar to 4860 basic CCP examples generated in the

**Table 3**

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>Parameter description</th>
<th>Example quantity</th>
<th>Example quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>USP</td>
<td>( s = (0.75, 3.25, 0.25) )</td>
<td>400</td>
<td>4400</td>
</tr>
<tr>
<td>DSP</td>
<td>( s = (-3.25, -0.75, 0.25) )</td>
<td>400</td>
<td>4400</td>
</tr>
<tr>
<td>UTP</td>
<td>( d = (0.08, 0.28, 0.02) )</td>
<td>400</td>
<td>4400</td>
</tr>
<tr>
<td>DTP</td>
<td>( d = (-0.28, -0.28, 0.02) )</td>
<td>400</td>
<td>4400</td>
</tr>
<tr>
<td>CP</td>
<td>( a = (0.75, 3.25, 0.25) )</td>
<td>400</td>
<td>4400</td>
</tr>
<tr>
<td>SP</td>
<td>( g = (0.75, 3.25, 0.25) )</td>
<td>400</td>
<td>4400</td>
</tr>
</tbody>
</table>

The three values in parentheses correspond to initial value, final value, and increment, respectively.
testing and testing procedure, another 4860 basic CCP examples (see Table 2) also generated using Eq. (1) but with different random seeds were presented into the proposed hybrid ESMD–ELM model. Table 6 presents evaluation results for basic CCPs. The ARL and SRL values were calculated only for the correctly identified CCPs. As seen in Table 6, the proposed hybrid ESMD

5.3. Effects of activation function on model performance

With other parameter settings (such as the observation window size, the number of interpolating curves) held unchanged, a comparison of the ARLs, SRLs, CICs and OQEs of the ELM with Gaussian kernel and the ELM with random Sigmoid hidden nodes and multiquadric radial basis function (RBF) hidden nodes with regard to the identification and quantification of concurrent CCPs is made in this section. For ELM with Sigmoid additive hidden node and multiquadric RBF hidden node, a feature mapping (hidden-layer output vector) \( h(x) = \{G(a_1, b_1, x), G(a_2, b_2, x), \ldots, G(a_n, b_n, x)\} \), where \( G(a, b, x) = 1/(1 + \exp(-ax + b)) \) for Sigmoid additive hidden node or \( G(a, b, x) = (|x - a|^2 + b^2)^{1/2} \) for multiquadric RBF hidden node. All the hidden node parameters \((a_i, b_i)\) are two user-specified parameters. For achieving a fair comparison, the grid-search method is also used for determining the optimal value of \( C \) so that the ELM with random Sigmoid hidden nodes and multiquadric RBF hidden nodes can perform the task as well as possible. All possible values of \( C \) with exponentially growing

Please cite this article as: W.-A. Yang, et al., Identification and quantification of concurrent control chart patterns using extreme-point... Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.068
sequences, \(2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\), was tried, and the one with the best accuracy was chosen as the optimal parameter for ELM with Sigmoid additive hidden node and multiquadric RBF hidden node. The performance of ELM with Sigmoid additive hidden node and multiquadric RBF hidden node is not sensitive to \(C\) as long as \(L\) is set large enough. In this comparison, therefore, \(L\) is always set to 1000 for ELM with Sigmoid additive hidden node and multiquadric RBF hidden node.

The combination of grid-search method is also used for determining the optimal parameters. Table 4 summarizes the ARLs, SRLs and CICs of the ELM and the SVM with random Sigmoid hidden nodes and multiquadric RBF hidden nodes in the column “ELM” and the SVM with Gaussian kernel and the ELM with random Sigmoid hidden nodes and multiquadric RBF hidden nodes in the column “ELM”. Table 5 summarizes the OQEs of the ELM with Gaussian kernel and the ELM with random Sigmoid hidden nodes and multiquadric RBF hidden nodes in the column “ELM”. Tables 4 and 5 indicate that the ELM with Gaussian kernel has comparable or better identification capability than the ELM with random Sigmoid hidden nodes and multiquadric RBF nodes. Hence, it is appropriate to equip the ELMs with Gaussian kernel rather than other activation functions in order to have efficient implementation of the proposed hybrid ESMD–ELM model, although any different activation functions of ELM can be used in ELM #1 of Module I or ELMs #2, #3, #4, #5, #6 and #7 of Module II.

### 5.4. Comparison of the model’s performances with ELM and SVM

With other parameter settings (such as the observation window size, the number of interpolating curves) held unchanged, a comparison of the ARLs, SRLs, CICs and OQEs of the ELM and the SVM with regard to the identification and quantification of concurrent CCPs is made in this section. It is well known that the generalization performance of SVM with Gaussian kernel is highly sensitive to the combination of \((C, \gamma)\). Thus, for achieving a fair comparison, the best combination of \((C, \gamma)\) of SVM with Gaussian kernel needs to be chosen. In this comparison, the grid-search method is also used for determining the optimal combination of \((C, \gamma)\) so that the ELMs can perform the task as well as possible. All possible combinations of \((C, \gamma)\) with exponentially growing sequences, \(2^{-24}, 2^{-23}, \ldots, 2^{24}, 2^{25}\), were tried, and the one with the best accuracy was chosen as the optimal parameter. Table 4 summarizes the ARLs, SRLs and CICs of the ELM and the SVM for multiclass classification in the column “ELM” with the sub-column “Gaussian kernel” and in the column “SVM”, respectively. Table 5 summarizes the OQEs of the ELM and the SVM for regression in the column “ELM” with the sub-column “Gaussian kernel” and in the column “SVM”, respectively. As seen in Tables 4 and 5, compared to SVM, ELM achieves similar or better generalization capability of detecting and classifying the typical concurrent CCPs and much better generalization capability of quantifying the major parameter of the concurrent CCPs. SVM source codes can be freely obtained from the website (http://www.csie.ntu.edu.tw/cjlin/libsvm/).

### 6. Conclusions

Correct and timely identification of CCPs is important for manufacturing practitioners since the unnatural CCPs are normally associated with specific assignable causes for the out-of-control processes. Most of existing research focus on identification of basic CCPs. However, it is very common in practices that concurrent CCPs may affect the manufacturing processes simultaneously. In addition, these existing research for CCPR cannot provide more detailed CCP parameter information, such as shift magnitude, trend slope, cycle amplitude, etc., which is very useful for quality practitioners to search the assignable causes that give rise to the out-of-control situation. A hybrid ESMD–ELM model that integrates ESMD with ELM has been proposed in this study to identify typical concurrent CCPs and also to accurately quantify the major CCP parameter of the specific basic CCPs involved. The experimental results indicate that the proposed model can effectively identify not only concurrent CCPs but also basic CCPs. Meanwhile, the major CCP parameter of the identified concurrent CCP can also be accurately quantified.

In this study, only six unnatural basic CCP types were adopted, thus the proposed hybrid ESMD–ELM model cannot identify other basic types of basic CCPs (e.g., mixture). Therefore, for applying the proposed hybrid ESMD–ELM model in a practical process-monitoring scenario, other common types of basic CCP should also be included in the training examples of the ELMs to overcome the usage limitation of the proposed model.

Three future research directions may be worth pursuing. First, with the rapid development of the automatic data acquisition and inspection technologies, interest in the simultaneous scrutiny of several interrelated quality characteristics has increased substantially. The hybrid ESMD–ELM model proposed for univariate manufacturing processes in this study can be modified and extended to deal with CCPR problems for multivariate manufacturing processes. Second, as individual measurements are used, data are likely to be autocorrelated [49]. This proposed hybrid ESMD–ELM model can also be extended to deal with observations from an autocorrelated manufacturing process. Third, in this study, the normalized direct (raw) data of the time series was used as the input vector of ELMs. However, the direct data-based approach is very sensitive to noise (i.e., the natural random fluctuations of the manufacturing process), especially when the major CCP parameters are small [19,45,50]. Using hybrid data (i.e., direct data combined statistical features (e.g., mean, standard deviation, skewness and kurtosis) extracted from the manufacturing process) as inputs of ELMs could reduce the impact of noise on the proposed model.

### Acknowledgments

This work was supported by grants from the Defense Industrial Technology Development Program under Grant A2520110003 and the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT0968. The authors would also like to thank Professor Jin-Liang Wang from Qingdao Technological University, People’s Republic of China, for his constructive and inspiring comments on our research on ESMD.

### References


Table 6

<table>
<thead>
<tr>
<th>Pattern type</th>
<th>CIC</th>
<th>ARL</th>
<th>SRL</th>
<th>Pattern type</th>
<th>CIC</th>
<th>ARL</th>
<th>SRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>USP</td>
<td>98.3</td>
<td>4.7</td>
<td>1.35</td>
<td>DTP</td>
<td>98.1</td>
<td>7.4</td>
<td>1.65</td>
</tr>
<tr>
<td>DSP</td>
<td>97.5</td>
<td>5.3</td>
<td>1.10</td>
<td>CP</td>
<td>93.8</td>
<td>9.8</td>
<td>1.81</td>
</tr>
<tr>
<td>UTP</td>
<td>98.7</td>
<td>7.1</td>
<td>1.47</td>
<td>SP</td>
<td>95.9</td>
<td>9.5</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Please cite this article as: W.-A. Yang et al., Identification and quantification of concurrent control chart patterns using extreme-point... Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.06.068
W.-A. Yang et al. / Neurocomputing 11


