IIE Transactions (2013) **45**, 373–391

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2012.662420

Fractal geometry statistical process control for non-linear pattern-based processes

NOA RUSCHIN-RIMINI, IRAD BEN-GAL* and ODED MAIMON

Department of Industrial Engineering, Tel-Aviv University, Tel-Aviv, Israel
E-mail: bengal@eng.tau.ac.il

Received December 2010 and accepted January 2012

This article suggests a new Statistical Process Control (SPC) approach for data-rich environments. The proposed approach is based on the theory of fractal geometry. In particular, a monitoring scheme is developed that is based on fractal representation of the monitored data at each stage to account for online changes in monitored processes. The proposed fractal-SPC enables a dynamic inspection of non-linear and state-dependent processes with a discrete and finite state space. It is aimed for use with both univariate and multivariate data. The SPC is accomplished by applying an iterated function system to represent a process as a fractal and exploiting the fractal dimension as an important monitoring attribute. It is shown that data patterns can be transformed into representing fractals in a manner that preserves their reference (in control) correlations and dependencies. The fractal statistics can then be used for anomaly detection, pattern analysis, and root cause analysis. Numerical examples and comparisons to conventional SPC methods are given.

Keywords: Anomaly detections, SPC, autocorrelated processes, control charts, analytics

1. Introduction

In recent years, developments in storage capacity, sensor usage, and information technology have created a growing need for new monitoring techniques that can cope with complex and data-rich environments. Predictive analytics, business intelligence, business activity monitoring, and complex event processing are few examples of new business methodologies that require the monitoring of a large amount of data in their implementation (Ren *et al*., 2006; Wasserkrug *et al*., 2008). In order to apply these business methodologies properly, one needs to rely on monitoring techniques that, following the taxonomy in Ben-Gal *et al.* (2003), can be classified as model-generic (nonparametric) tools for processing dependent data with minimum *a priori* assumptions. Model-generic methods, as opposed to traditional model-specific methods, do not rely on *a priori* assumptions about the monitored variable, such as an underlying analytical distribution (e.g., Shewhart charts for independent data) or a closed-form expression (e.g., ARIMA or CUSUM for more complex dependent data).

Considerable effort has been devoted to developing methods for monitoring processes with dependent (autocorrelated) data. Most of these model-specific methods for autocorrelated data were based on time series models (e.g., Box and Jenkins, 1976; Alwan and Roberts, 1988; Harris and Ross, 1991; Montgomery and Mastrangelo, 1991; Runger and Willemain, 1995; Runger *et al*., 1995; Apley and Shi, 1999; Lu and Reynolds, 1999a, 1999b). The majority of them rely on the implicit, yet generally unguaranteed, assumption that time series can closely model the monitored processes. Ben-Gal and Singer (2004) showed that conventional Statistical Process Control (SPC) methods, including those that were designed to handle autocorrelated data, usually represent a linear dependence between observations; i.e., referring to dependencies between observations that can be modeled by an ARIMA type of model. Accordingly, these conventional methods are not suited to monitoring state-dependent non-linear processes. Thus, referring to processes where the probability distribution of the next value is strongly conditional on previously observed values and can change significantly depending on the conditioning values. Markov processes are good examples of such state-dependent processes. The authors particularly refer to industrial environments, where the process parameters are adjusted by feedback control policies based on past observations. English *et al*. (2001) and Singer and Ben-Gal (2007) further emphasized that feedback policies, as well as more complicated control theory techniques, often create non-linear dynamics of the controlled observations. They provided examples such as the recipe settings of some wafers in semiconductor processes, which are adjusted by using measurements of previously produced wafers, color adjustments between

*Corresponding author

fabric batches (Shore, 1992), and management tampering with non-manufacturing environments (Broadman and Broadman, 1990).

Many of the above-mentioned state-dependent processes, certainly the Markovian processes, take values from a finite and discrete set of attributes. The need to develop control charts for discrete processes with a finite state space has been recognized in the literature, and various methods have been proposed to address it. Shore (2000) provided examples of control charts for attributes, such as the use of arcsin transformation for binomial data; the use of the Q-chart for binomial and Poisson parameters (Quesenberry, 1991a, 1991b); and the use of the g-chart and the h-chart based on the geometric distribution (Kaminski *et al*., 1992). According to Shore (1998), due to the ineffectiveness of distribution-identification procedures, these types of methods are seldom used. Moreover, these are not model-generic methods, since each distribution requires a special treatment. Shore (2000) suggested a method based on fitting a distribution that preserves the first three moments of the chosen attribute statistic. Still, his approach requires an *a priori* fitting of the process underlying distribution.

A list of differences between the characteristics of traditional SPC and the ones required in a modern data-rich environment were compiled from the presented literature survey.

1. Traditional SPC is often based on model-specific assumptions, whereas modern environments that involve many types of data sources often require a model-generic approach.
2. Traditional SPC methods are not designed to cope with state-dependent and non-linear dynamics of the observations that may result from feedback policies, as well as from complicated control implementations. Modern SPC, on the other hand, often require monitoring complex (non-linear) patterns of univariate or multivariate data.
3. Traditional SPC tools are tuned to detect anomalies/outliers in a process, yet many of them do not allow the identification of assignable causes that may impact the process in terms of detecting patterns and relationships among the system attributes.
4. Traditional control charts were designed to be executed on a paper sheet (even if nowadays it is presented on a computer screen), whereas modern monitoring requires more advanced monitoring features such as zoom-in, coloring codes, and visual inspection of complex structures (e.g., via fractals in our case). These advanced features can be used for root cause analysis tasks.

Recent research has tried to address some of these gaps. Notable contributions have been made by Alwan *et al*. (1998), Castagliola and Tsung (2005), Cheng and Thaga (2005), Perry and Pignatiello (2006), Ren *et al*. (2006), and Kim *et al*. (2007). Mason *et al*. (1995) and Runger *et al*. (1996), for example, proposed methods for assignable cause identification that can be linked with Hotelling's T^2 control chart. Ben-Gal *et al*. (2003) proposed a Context-based SPC (CSPC), as a model-generic framework that can deal with autocorrelated non-linear state-dependent processes. The CSPC implements a variable-order Markov model to represent the monitored process, without relying on *a priori* knowledge about the process parameters and without assuming a closed-form time series model.

In this article we follow that line of work and propose the fractal-SPC method. The fractal-SPC is a model-generic tool for monitoring (non-linear) dependent and independent discrete processes with a finite state space. It is aimed at monitoring both univariate and multivariate data, particularly in data-rich environments. Moreover, the method is designed to detect abnormal patterns of varying lengths. An example for such applications could be the analysis of vehicle warranty claims data history (as applied to General Motors' data provided by their research labs located in Bangalore, India) or identification of faulty operation sequences in an assemble-to-order environment (Ruschin-Rimini *et al*., 2012). Note that although the suggested method considers discrete processes, it can be applied to processes consisting of continuous numeric data that are reduced to a sufficiently discrete set of interesting ranges (Rokach *et al*., 2008), as we show by several numerical examples.

There are several intuitive reasons why we use fractals for SPC applications. Fractals are naturally tuned to represent a large number of data patterns with complex dependence structures. They are known for their ability to visually represent complex large data sets. Their construction does not require *a priori* assumptions regarding the dependencies within the patterns or the data distribution. And, as seen in the next sections, various data patterns can be mapped by the suggested iterated function system to representing fractals, regardless of their distribution or dependency structure. In order to dynamically and automatically monitor the representing fractals, we use fractal dimension statistics. We claim, and later demonstrate, that fractal dimension statistics can be successfully applied to the inspection and analysis of data-rich environments. We then compare the performance of the fractal-SPC method with that of traditional SPC methods as well as to special-purpose methods such as the CSPC method and multi-attribute control charts (Woodall, 1997; Jolayemi, 1999) and demonstrate its advantages in several examples.

The rest of the article is organized as follows: Section 2 introduces the theoretical background of Iterated Function System (IFS) and fractal dimensions. Section 3 presents the proposed algorithmic framework and illustrates it by an example. Section 4 presents an experimental study, demonstrating cases in which the suggested SPC overcomes limitations of both traditional and special-purpose SPC methods in detecting process anomalies. Section 5 gives some conclusions and suggests future research directions.

2. Theoretical background

2.1. *IFS*

The IFS concept was originally developed as a method to construct fractals, as discussed in detail in Barnsley (1988) and reviewed in Ruschin-Rimini *et al*. (2012). IFS is used as an iterative contractive mapping technique that represents either a univariate or multivariate process as vectors in $\Re^2$, as shown in the Appendix. In particular, an IFS consists of a complete metric space (X, d) and a finite set of contraction mappings $w_i : X \rightarrow X$, with respective contractivity factors s_i, with index $i = 1, 2, \ldots, m$, which is associated with each category (possible value of a random variable) in our case. A mapping $w_i(x)$ is called contractive in (X,d), if $d(w_i(\mathbf{y}), w_i(\mathbf{z})) \leq s_i \times d(\mathbf{y}, \mathbf{z}) \forall \mathbf{y}, \mathbf{z} \in X$ for some contractivity factor $0 < s_i < 1$, where $\mathbf{y} = (y_1, y_2, \ldots, y_D)$ and $\mathbf{z} = (z_1, z_2, \ldots, z_D)$ are vectors in $\Re^D$. This type of transformation of a sequence is also known as the *chaos game representation* (Barnsley, 1988). It produces a self-similar fractal-formed graph and has two main properties.

1. It provides a unique representation of a sequence and can be seen as the fingerprint of a sequence. Every point on the graph that is obtained by IFS uniquely represents all of the sequence history up to this data point; hence, an IFS representation comprises all information regarding all subsequences existing in a sequence.
2. The source of the sequence can be fully inversed from the graph; hence, there is no loss of information when needed for further analyses.

There exist several applications of IFS such as image compression (Barnsley and Hurd, 1993), texture synthesis (Chen and Chen, 2003), and genome sequence analysis.

The role of IFS in the context of SPC is to transform either a univariate or a multivariate process into a two-dimensional fractal. Since we consider processes with multiple categories, we suggest the use of an IFS of a circle transformation, based on the IFS developed by Weiss and Goeb (2008; see also Weiss (2008)). This unique IFS transformation, which is detailed in the next section, provides the flexibility of analyzing processes consisting of any number of discrete categories, while keeping representation in $\Re^2$ in order to enable visual analysis of the monitored process via a computer screen. Note that the procedure of transforming univariate or multivariate processes into a visual two-dimensional fractal via IFS is a simple iterative procedure with linear complexity. The proposed visual analysis procedure includes visual detection of process in-control and out-of-control patterns, which can lead to the identification of assignable causes.

As an example, Fig. 1(a) illustrates the results of implementing an IFS of circle transformation on either a univariate process with nine possible categories 0, 1, . . . , 8 ($m = 9$) or a multivariate process of any dimension, each variable consisting of up to nine values. The interpretation of the fractal graph is based on the concept of addresses of points on a fractal (Barnsley, 1988) and is used for the purpose of analyzing the monitored process. It is based on an important attribute of fractals known as *self-similarity*. Figures 1(b) and 1(c) illustrate the self-similarity attribute

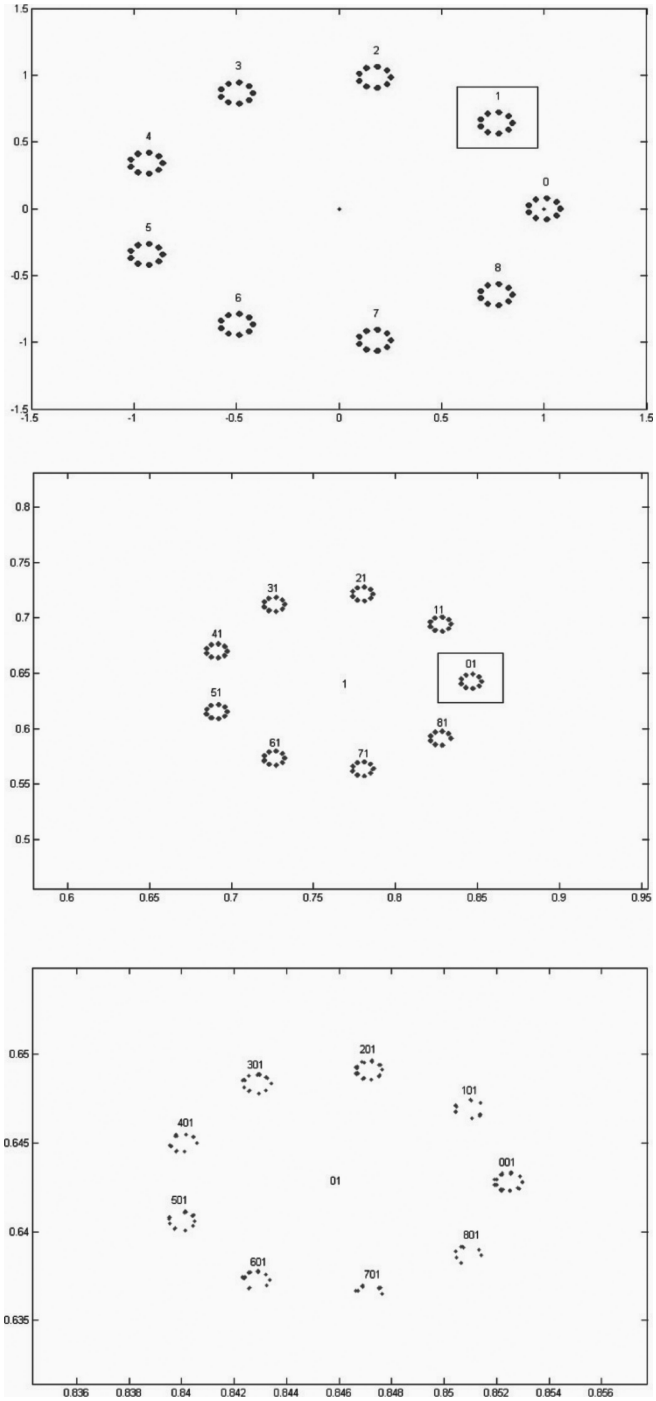


Fig. 1. The result of transforming a monitored process consisting of nine possible categories via IFS of circle transformation: (a) addresses of points for first resolution; (b) addresses of points after zooming into circle address 1; and (c) addresses of points after zooming into circle address 01 (color figure provided online).

of a fractal by zooming into the fractal in different areas that are tagged by their address.

The figures show the addresses of points on this specific circle-formed fractal, representing a certain monitored process, as previously defined. The following interpretation of the figure is based on Weiss (2008) and Weiss and Goeb (2008). Each circle in Fig. 1(a) is associated with one category (realization) out of the nine possible categories in the monitored process. Zooming into circle of address 1 (marked by a black square in Fig. 1(a)) leads to Fig. 1(b). Every circle in Fig. 1(b) represents one of the following two-length subsequences: 01, 11, 21, 31, 41, 51, 61, 71, and 81. Then, zooming into circle address 01 results in Fig. 1(c). Every circle in Fig. 1(c) represents one of the following three-length subsequences: 001, 101, 201, 301, 401, 501, 601, 701, 801, etc. The density of points in each circle indicates the frequency level of its represented sub-sequence. In order to visually identify the density level of data points in the circles, we later on provide a color code function (see Section 3.4 and Figs. 6 to 9). Such a color code enables a visual distinction between rare and frequent sub-sequences (see also Ruschin-Rimini *at al*. (2012)). Nevertheless, even without the proposed color code, one can visually identify some missing patterns. For example, one can see that zooming into circle address 701 results in empty circles, representing sub-sequences that do not exist in the monitored process such as 5701, 6701, 7701, and 8710. As illustrated, the fractal graph holds information regarding the underlying distributions and patterns of the monitored process. Hence, a change point in the monitored process would impact the fractal graph and therefore would be detected by fractal measures, as we show in the following sections.

To summarize, since the addresses of points on the fractal can represent process sub-sequences and patterns, they can enable the detection of frequent, rare, and missing patterns of various lengths in both in-control and out-of-control processes. This ability turns the root cause analysis task into a visual zoom-in and zoom-out routine; it supports the task of assignable cause identification and provides the user insights and intuition about the monitored process.

2.2. *The fractal dimension*

The fractal dimension is a statistical quantity that measures the number of dimensions "filled" by a fractal. There are several theoretical definitions of the fractal dimension. We rely on some of the most commonly used definitions, namely, the box counting dimension, the information dimension, and the correlation dimension, and use them as statistics for the proposed fractal-SPC. We demonstrate that a combination of all three fractal dimension types can provide a monitoring scheme that integrates them into a single anomaly-detection and decision-making module. It is important to note that the functionality of fractal dimension computation is offered by many standard software packages (e.g., BENOIT for MATLAB and Wolfarm Mathematica), most of them address the three types of fractal dimension chosen to monitor the statistics of the suggested fractal-SPC. Such software tools significantly increases the applicability of the suggested method.

Definitions of the fractal dimension types are now presented.

The box counting dimension: For a set of N points constructing a fractal, each of dimension D, one divides the space into grid cells of side size r (hyper-cubes of dimension D). $N(r)$ denotes the number of cells occupied by the points constructing the fractal. The box counting fractal dimension is then calculated as follows:

$$D_{\mathrm{bc}} = -\lim_{r\to 0} \frac{\log N(r)}{\log r}. \tag{1}$$

In order to relate the properties of the box counting dimension to the suggested fractal-SPC, we refer the interested reader to the studies of the Asymptotic Equipartition Property (AEP) and the properties of the typical set, as discussed in Cover and Thomas (1991). As explained in Cover and Thomas (1991), the EAP property is a direct consequence of the weak law of large numbers in information theory. The AEP states that $-1/n \log p(X_1, X_2, \ldots, X_n)$ is close to the entropy H, where $X_1, X_2, \ldots, X_n$ are independent and identically distributed random variables, and $p(X_1, X_2, \ldots, X_n)$ is the probability of the sequence $X_1, X_2, \ldots, X_n$. This observation enables the division of the set of all sequences into two sets: the typical set, where the sample entropy is close to the true entropy, and the non-typical set, which contains the other sequences.

We suggest that the box counting dimension measures the number of elements in the typical set determined by the original (in-control) process. Any change in this property is detected by the box counting dimension statistic.

Dimension of information: For a set of N points constructing a fractal, each of dimension D, one divides the space into grid cells of side size r (hyper-cubes of dimension D). $p_i(r)$ is the frequency with which points fall into the ith cell. The information dimension is obtained as follows:

$$D_{\mathrm{inf}} = \lim_{r\to 0} \frac{\sum_i p_i(r) \log p_i(r)}{\log r}. \tag{2}$$

In the context of the suggested fractal-SPC method, the dimension of information detects a change in the entropy measure for all sub-sequences within the original process, as will be detailed in Sections 3.2 and 3.4.

Dimension of correlation: For a set of N points constructing a fractal, the dimension of correlation is defined as follows (Grassberger, 1983; Grassberger and Procaccia, 1983):

$$D_{\mathrm{cor}} = -\lim_{\varepsilon\to 0} \frac{\log C(\varepsilon)}{\log \varepsilon}, \tag{3}$$

$$C(\varepsilon) = \lim_{N\to\infty} N^{-2} \times \{\text{number of pairs } (x_i, x_j), \; i \neq j = 1, \ldots, N; \text{ where } |x_i - x_j| < \varepsilon\},$$

where (x_i, x_j) denotes any existing pair of points constructing the fractal.

In the context of the suggested fractal-SPC method, the correlation dimension measures the probability for the occurrence of correlated sub-sequences in the process.

3. Suggested algorithmic framework: the fractal-SPC approach

We suggest the following framework in order to establish a fractal-based SPC. The proposed method assumes that the process measures have nominal or discrete values and consists of four stages; see Fig. 2.

1. *Fractal mapping*: An IFS scheme with circle transformation is applied to historical in-control data to obtain a fractal representation of the reference process. The fractal graph represents the reoccurring patterns in the process. Its interpretation is based on the concept of addresses of points on fractals, as discussed in Section 3.4.
2. *Selection of a fractal-based statistic*: Various types of fractal dimensions can be used as monitoring statistics of the process. We mainly focus on the information dimension out of the three described previously. Control limits are derived from fractal dimension measures either numerically or theoretically.
3. *Online process monitoring*: In the monitoring stage, each process sample is transformed into points in the fractal graph. Fractal dimension statistics are recalculated for the sampled data. Process deviation is indicated by out-of-control signals.
4. *Visual root cause analysis*: Combined analyses of the fractal graph as well as the various fractal dimension statistics are used for the purpose of identifying assignable causes that may impact the process.

The following subsections describe each of the above phases. A running example is given in Section 3.5 for illustration purposes.

3.1. *Fractal mapping*

We utilize an IFS with circle transformation, similar to the IFS proposed by Weiss and Goeb (2008; see also Weiss (2008)). This unique IFS transformation provides the flexibility to analyze processes consisting of any number of discrete categories while keeping the representation in $\Re^2$ to enable visual analysis on a computer screen. Color codes can be used to create better interpretations by the user.

As explained in Weiss (2008) and Weiss and Goeb (2008), the following is a description of the suggested IFS transformation for a process consisting of m discrete categories:

$$w_i\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix} \quad \text{for } i = 1, 2, \ldots, m,$$

where

$$\begin{aligned} \beta_i &= \cos\left(i \times \frac{2\pi}{m}\right) \quad \text{for } i = 1, 2, \ldots, m, \\ \delta_i &= \sin\left(i \times \frac{2\pi}{m}\right) \quad \text{for } i = 1, 2, \ldots, m. \end{aligned} \tag{4}$$

In order to ensure that the fractal is totally disconnected (Barnsley, 1988)—i.e., to guarantee that every point on the fractal graph has a unique address—it is required that α satisfies the following inequality:

$$\frac{\alpha}{1-\alpha} < \sin\left(\frac{\pi}{m}\right), \tag{5}$$

as proved in Ruschin Rimini *et al*. (2010).

The following steps are used to apply an IFS with circle transformation to a sequence of length N consisting of m discrete categories, as explained in Weiss (2008) and Weiss and Goeb (2008; based on the chaos algorithm of Barnsley (1988)).

1. Associate and fix each category (a possible variable realization) with one of the contractive mappings $w_i(x)$, $i \in \{1, 2, \ldots, m\}$.
2. Accordingly, represent a sequence of length N consisting of m category types by a sequence of N corresponding contractive mappings $\{w_{i(n+1)}(x_n) : i \in \{1, 2, \ldots, m\}$ and $n = 1,2, \ldots ,N\}$. $w_{i(n+1)}(x_n)$ indicates that variable x_n is mapped by the contractive mapping defined by variable x_{n+1}. For simplicity we omit one subscript and denote the mapping by $w_i(x_n)$. To start the process, the initial point $x_{(0)}$ is selected arbitrarily as a point in $\Re^2$.
3. Recursively apply each of the N contractive mappings $w_i(x_0), w_i(x_1), \ldots, w_i(x_{N-1})$ in their sequence order; i.e., apply contractive mapping $w_i(x_0)$ to obtain point $x_{(1)}$, then apply contractive mapping $w_i(x_1)$ to obtain $x_{(2)}$, etc.

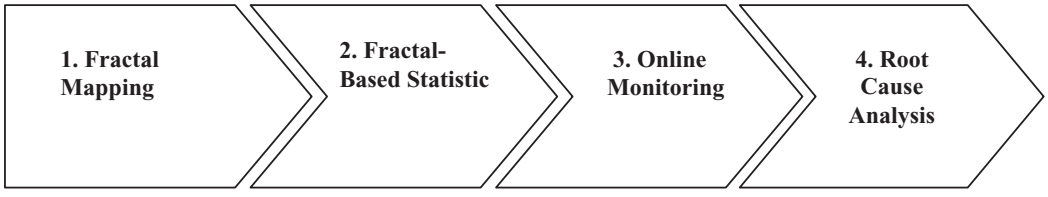


Fig. 2. The process overview.

In general, $x_{(n)} = w_i(x_{(n-1)})$ for $n = 1, 2, 3, \ldots, N$ and $i \in \{1, 2, 3, \ldots, m\}$. The mapping results in a sequence of N points in $\Re^2$ $\{x_{(n)} : n = 1, 2, 3, \ldots, N\}$. A running illustrative example for this procedure is given in the Appendix.

The presented procedure produces a circle-formed fractal graph that enables visual representation and analysis of any sequence with finite category (symbol) types. The number of process category types determines m. For example, if the original sequence of m category types is uniformly random and long enough, this IFS transformation results in a graph of self-similar fractals consisting of m equally filled and disconnected circles, and each circle at each resolution also consists of m circles. If the sequence is not uniformly random, the graph will reveal its underlying correlations by varying densities of points in different zones. We apply the IFS of circle transformation both to in-control process data and later to sampled data during the monitoring stage. An explanation of the minimum required initial sequence length N is given is Section 3.2.

For the purpose of illustration, a running example is introduced in Section 3.5.

3.2. *Selection of fractal-based statistic*

Fractals can be characterized by their fractal dimension statistic. As previously mentioned, there are many theoretical definitions of the fractal dimension.

Once an in-control process is mapped into a fractal, we suggest utilizing the various types of fractal dimensions as monitoring statistics. In particular, we chose to concentrate on three of the most commonly used fractal dimension types: the box counting dimension, the dimension of information, and the dimension of correlation.

The fractal dimensions of the transformed in-control process can be implemented as indicated next.

3.2.1. *Fractal dimension types implementations*

Implementation of the box counting dimension (D_{bc}): In order to implement the box counting dimension computation by the suggested method of circle transformation, we refer to Equation (1). We suggest selecting r as the radius of a circle. Moreover, one knows the exact locations and radii of the circles created by the circle transformation algorithm: the circle-formed fractal, which represents a transformed sequence of m categories, consists of m circles of radius α on the first resolution, m^2 circles of radius α^2 on the second resolution, etc. Generally, it consists of m^k circles of radius α^k on resolution k. Thus, we count the number of circles in resolution k that are occupied by one point at least, out of a total of m^k existing circles. We denote the number of occupied circles by $N(\alpha^k)$. We then calculate the box counting dimension based on Equation (1) with $r = \alpha^k$.

According to the proposed mapping method, every circle of radius α^k represents a specific k-length sub-sequence of the original process. Consequently, changes in the value of the box counting dimension take place each time that a new circle is occupied; i.e., when an unknown k-length subsequence appears in the process. It is reasonable that D_{bc} would be particularly appealing for detecting outliers such as extreme-value anomalies and minimizing their related errors (of both types). On the other hand, the D_{bc} calculation is invariant to the number of data points in each circle. Thus, it is not sensitive to changes in the distribution of data points; hence, it may suffer from Type 2 errors in such cases.

Implementation of the dimension of information (D_{inf}): Following the selection of r as the radius of a kth resolution circle, let us denote the frequency with which data points fall into the ith circle of radius α^k by $p_i(\alpha^k)$. Recall that there are a total of m^k circles of radius α^k at resolution k. Thus, $p_i(\alpha^k)$ represents the frequency of a specific sub-sequence of length k indexed by i within the original process. The information dimension is obtained as follows (derived from Equation (2)):

$$D_{\text{inf}} = \lim_{\alpha^k \to 0} \frac{\sum_{i=1}^{m^k} p_i(\alpha^k) \log p_i(\alpha^k)}{\log \alpha^k}. \tag{6}$$

Note that the numerator $\sum_{i=1}^{m^k} p_i(\alpha^k) \log p_i(\alpha^k)$ reflects Shannon's entropy measure for all k-length sub-sequences in the monitored process. Since the information dimension changes when shifts in the entropy of k-length subsequences occur in a process, it is reasonable that D_{inf} is sensitive to distribution-related errors of both types. A detailed analytical study of the box counting dimension in the context of the fractal-SPC can be found in Ruschin-Rimini *et al*. (2011a).

Implementation of the dimension of correlation (D_{cor}): Equation (3) is used to compute the correlation dimension. We select ε as the radius of a circle at resolution k, $\varepsilon = \alpha^k$. The correlation dimension D_{cor} measures a weighted frequency that two points chosen at random are located within a radius distance of each other; i.e., it increases with the correlation of k-length sub-sequences. Consequently, D_{cor} can be used to find correlation-related anomalies in the data. A detailed analytical study of the correlation dimension in the context of the fractal-SPC can be found in Ruschin-Rimini *et al*. (2011a).

In this work we use D_{inf} as our main monitoring statistic; however, we exploit both D_{bc} and D_{cor} for the purpose of root cause analysis, when an out-of-control signal is triggered (see the example in Section 3.5). Such signals rely on *a priori* control limits for the fractal dimension statistics that can be obtained numerically by using the in-control data. In the next section, we provide an analytical study of the dimension of information in the context of the suggested fractal-SPC in order to establish its analytical control limits.

3.2.2. *Distribution and control limits of the information dimension monitoring statistic: an analytical study*

In order to determine the distribution of the proposed monitoring statistic, as well as to obtain an estimation of the control limits, we refer to the studies of the natural estimator of Shannon's entropy, denoted by $\hat{H}$ (see Miller and Madow (1954) and Basarin (1959)). The statistical problem addressed is of testing and using an entropy-based model when the only data available are from comparatively small samples. Luce (1955, pp. 45–46) phrases the problem as follows:

> Let us suppose that a distribution p_i governs the selections of the n category types
> 1, 2, …, n and suppose that a sample of N independent observations of selections yields N_i cases of alternative i. The true entropy is $H = -\sum_{i=1}^{n} p_i \log p_i$ while $\hat{H} = -\sum_{i=1}^{n} (N_i/N)\log(N_i/N)$ is the estimator of the entropy obtained by replacing each p_i by its maximum likelihood estimator N_i/N.

Miller and Madow (1954) have shown that if the p_i values are not all equal, the normalized term $\sqrt{N}(H - \hat{H})$, henceforth denoted as $\tilde{H}$, has a normal limiting distribution with mean zero, $E(\tilde{H}) = 0$, and variance defined by:

$$\sigma^2(\tilde{H}) = \sum_{i=1}^{n} p_i[\log p_i + H]^2 \tag{7}$$

The authors have also shown that if $p_i = 1/n$ for every i, then $(2N/\log \ell)(H - \hat{H})$ follows a chi-square limiting distribution with $(n-1)$ degrees of freedom. For the proposed fractal-SPC, the number of degrees of freedom is given by $(m^k - 1)$, where m denotes the number of category types and k represents the resolution of the circles or, alternatively, the lengths of the selected sub-sequences. In frequent cases, $(m^k - 1)$ equals a large enough number. Consequently, the monitoring statistic can be assumed to be normally distributed even in the case of a uniform distribution when $p_i = 1/m^k$, as the chi-square distribution converges to a normal distribution when the number of degrees of freedom is large.

Miller and Madow (1954) also addressed the case of utilizing small samples. According to their studies, if small samples are used to estimate the entropy, there is a bias that can be corrected by the following equation:

$$H = E(\hat{H}) + (\log \ell)\left[\frac{n-1}{2N} - \frac{1}{12N^2} + \frac{1}{12N^2}\sum_{i=1}^{n}\frac{1}{p(i)}\right] + 0\left(\frac{1}{N^3}\right), \tag{8}$$

where $E(\hat{H})$ is the expected value of H and $0(1/N^3)$ denotes the order of the complexity.

As the dimension of the information is approximately normally and independently distributed for a long enough sequence, conventional SPC charts can be directly applied to it for a specified Type I error rate. The following procedure is used to derive the control limits.

1. Compute the information dimension of the fractal produced by mapping the in-control process data. The estimated fractal dimension is expressed by $\hat{D}_1 = \hat{H}(\alpha^k)/\log \alpha^k$.
2. For small samples, use Equation (8) to correct the bias of the estimated entropy.
3. Apply Equation (6) to find the variance of $\tilde{H}(\alpha^k)$. Multiply the result by $1/N(\log \alpha^{2k})$ so as to derive the fractal dimension statistic variance, denoted by $\sigma^2(\hat{D}_1)$ (since $\hat{D}_1 = \hat{H}(\alpha^k)/\log \alpha^k$ and $\tilde{H}(\alpha^k) = \sqrt{N}(H(\alpha^k) - \hat{H}(\alpha^k))$.
4. Specify the required Type I error rate. Since both $\sigma^2(\hat{D}_1)$ and $E(\hat{D}_1)$ were computed in previous steps, control limits can be easily determined and conventional SPC charts can be directly applied by $E(\hat{D}_1) \pm Z_{\alpha/2} \times \sigma(\hat{D}_1)$.

We suggest that the control limits estimator $\sigma^2(\hat{D}_1)$ has the following boundaries.

An upper bound: The maximum value of $\sigma^2(\tilde{H})_{\max}$ is obtained when the underlying process has one deterministic pattern; i.e., the probability of the occurrence of one type of k-length sub-sequences is one, whereas the probability of the occurrences of the rest of the k-length sub-sequences is zero. In such a case the entropy of the underlying process H equals zero and one obtains:

$$\sigma^2(\hat{D}_1) = \frac{1}{N(\log \alpha^{2k})}\sum_{i=1}^{n} p_i(\log p_i + H)^2. \tag{9}$$

The variance $\sigma^2(\hat{D}_1)$ is maximized in the case of a maximum difference between the process's true entropy and the estimated entropy. According to the maximum entropy principle, this is achieved in the case that the estimated entropy is of a uniform distribution. In this case, it follows from Equation (9) that:

$$\sigma^2(\hat{D}_1)_{\max} = \frac{\log m^{2k}}{N(\log \alpha^{2k})} \text{ (since } H = 0).$$

A lower bound: The minimal value of $\sigma^2(\tilde{H})_{\max}$ is obtained when the underlying process has a uniform distribution; i.e., $P_{(i)} = 1/n$ for all i. In this case, according to Harris (1975), $\sigma^2(\tilde{H})_{\min} = \sum_{i=1}^{n} p_i[\log p_i + H]^2 = 0$. We can derive from Equation (9) that in this case $\sigma^2(\hat{D}_1)_{\min} = 0$.

In order to determine the sequence length N—i.e., the number of data points required to construct the in-control fractal graph—we refer to the basic sampling rule suggested by Cochran (1952). This principle requires that at least 80% of the sampling bins (corresponding in this case to the occupied circles at a predetermined fractal resolution k) contain at least four data points. Experiments in which control limits are established both numerically and analytically are given in Sections 3.5 and 4.2.

3.3. *Online process monitoring*

During Phase I, the fractal graph for in-control data is generated. Moreover, the monitoring statistics, in the form of three types of fractal dimension, are computed for in-control data, as well as their analytical control limits, as explained in the previous section and in Ruschin-Rimini *et al*. (2011a). During Phase II, the monitoring stage, each monitored sample is transformed online into the fractal graph that was generated in Phase I. The fractal dimensions are recalculated whenever a new process sample is added. Process deviations are indicated by out-of-control signals, according to the fractal dimension measures and the control limits established in Phase I. The running example in Section 3.5 introduces histograms of the fractal dimension statistic, before and after a change point, as well as a fractal dimension control chart, with its control limits computed both numerically and analytically

3.4. *Visual root cause analysis*

As explained in Ruschin-Rimini *et al*. (2012) and Ruschin-Rimini and Maimon (2010), the circle-formed fractal graph can be visually interpreted by utilizing the address of points on a fractal when adding a color code function that colors data points of circles containing a relatively high density of points. A circle of high density is defined by a certain percentage threshold. The location of every point on the fractal graph remains constant during the whole sequence information stage and by that differs from traditional SPC charts. This enables us to translate areas on the fractal graph, such as empty areas, areas of relatively low density, and areas of relatively high density, into missing sub-sequences, rare sub-sequences, and frequent sub-sequences, respectively.

The IFS of circle transformation results in a graph of a self-similar fractal consisting of m disconnected circles, where each circle at each resolution is also comprised of m smaller circles. Since we associate every category with a certain contractive mapping, the address of every circle represents a category type (a process symbol). More specifically, category i, which is associated with mapping$w_i(x)$, is the address of the circle centered at

$$\begin{bmatrix} \beta_i \\ \delta_i \end{bmatrix}$$

(see Section 3.1 and Equation (4)). The address length, and thus the length of the monitored sub-sequences, is determined by the graph resolution. The addresses of the first, second, and third resolutions for the example described in the following section are displayed in Figs. 6 to 8. The process samples consist of $m = 9$ categories. Defining the addresses of points on the fractal graph enables us to suggest the following algorithm for visual detection of in-control and out-of-control process patterns.

Pattern detection algorithm

1. Set $k = 1$ as the resolution-level parameter.
2. Detect a circle of relatively high density on the kth resolution of the fractal graph; i.e., one of the m circles that contains a high percentage of points. A circle of relatively high density is defined by a certain threshold (see remark below) and can be visually detected since it consists of data points that are colored according to the predefined color code (see Figs. 6 to 9).
3. Drill into the relevant circle and set $k = k + 1$.
4. Repeat Steps 2 and 3 until the relevant circle contains points that are almost uniformly distributed between approximately m circles; i.e., with no circle of relatively high density. This is where the sequence pattern ends.
5. Compute the address of the relevant circle location in order to recover the process pattern.
6. Repeat steps 1 to 5 for a different circle of relatively high density in order to reveal another process pattern.
7. End after exploring all circles of relatively high density; i.e., after all process patterns have been revealed.

In order to improve the process of visual pattern detection, we use color codes to mark data points in circles of relatively high density. For illustration, see the running example in the following section. Nevertheless, it is important to note that as the number of categories in the process increases, interpretation of the graph becomes more challenging. Improvement of the visual representation can be of benefit in such cases and can be leveraged by new graphical techniques, such as focus and context techniques (Keim, 2002) or three-dimensional manipulations that could facilitate the users' interpretation.

Moreover, note that the threshold in Step 2 is not defined rigorously and might depend on the required sensitivity level for the triggered alerts, on the available in-control data, and on the types of possible process deviations. As a simple rule of thumb we note that if historical in-control data are available, the user can plot numerical histograms of circle densities at various k resolutions and set the corresponding thresholds by their relative (say the 90th) percentiles. In cases where there is no available information on the underlying in-control process, the user can rely on the maximum entropy rule. He (she) can assume that the probability of each category (variable realization) out of m possible categories is $1/m$; thus, the expected entropy for each type of k-length sub-sequence is $k \log m$. Accordingly, a rough starting point for a default threshold value is $(k/2) \log m$. However, since deviation types cannot be predetermined, it is proposed that this threshold value will be subject to changes and part of the visual root cause analysis procedure. A thorough study of the required threshold value, as a function of the in-control data type and the type of process deviation, should be further researched.

3.5. *Illustrative example: pattern reoccurring*

In this example we simulate a gradually deteriorating process in a data-rich environment. The process is represented by an ordered sequence of symbols that can represent observations measured in a multivariate process (each variable consisting of nine categories in this illustrative example): the monitoring of activities on a machine through a continuous stream of audit events recorded in log files (Ye *et al*., 2001; Ye *et al*., 2003; Kim *et al*., 2007), string of sensors' input, machine failures history, customers purchase history, and production processes in an assemble-to-order environment (Ruschin-Rimini *et al*., 2012).

In this illustrative case study, each sequence of $N = 15\,000$ data points was generated from a uniform distribution with an alphabet of $m = 9$ symbols. The random sequence represented an unstructured noise, with the pattern 8, 0, 4 being inserted randomly yet approximately every 15 data points. An out-of-control process was obtained by reverting the order of the reoccurring pattern to 4, 0, 8 in approximately 5% of the cases. Pattern-reoccurring processes can be used to represent machine types, part types, or production sequencing, as shown in Ruschin-Rimini *et al*. (2012). In that paper the categories represented operation types, and sequences represented optional production routes. For example, consider the production route 31452. It consists of five operations types, starting with operation 3 continuing with operation 1, and so on up to the last operation 2 in that manufacturing process. The fractal-SPC prototype was coded in MATLAB. As mentioned in Section 3.3, during the monitoring stage, each sample is mapped online into the fractal graph. The fractal dimension is recalculated whenever a new process sample is added. Process deviations are indicated by out-of-control signals. Figure 3 presents a histogram of the information dimension values, before and after the process's change point. Dimension measures for in-control samples are marked on the left-hand side and dimension measures for out-of-control samples are marked on the right-hand side. Note the significant change in the information dimension statistic, making it appealing for the monitoring of such process.

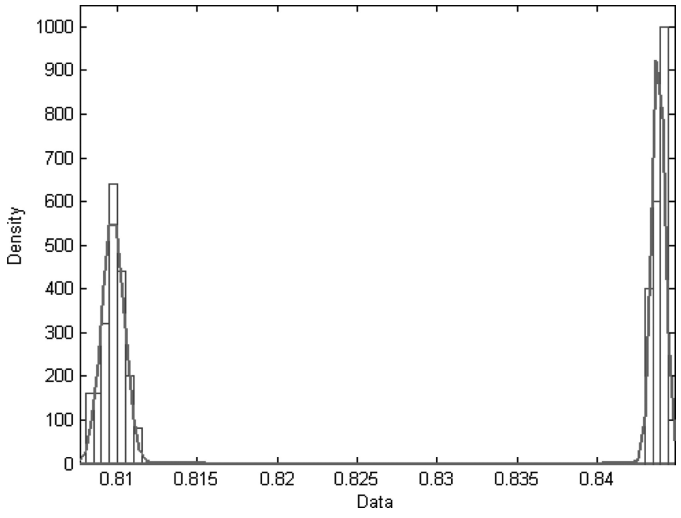


Fig. 3. Histograms of fractal dimension values for both pattern-reoccurring samples without deviation (on the left) and for pattern-reoccurring samples with ∼5% deviation (on the right) (color figure provided online).

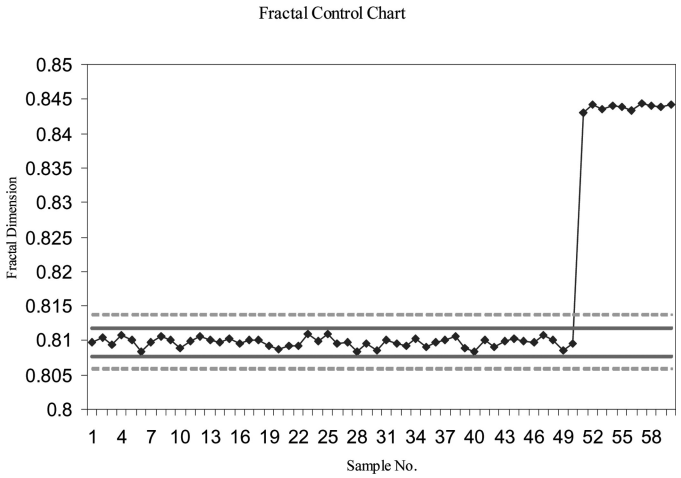
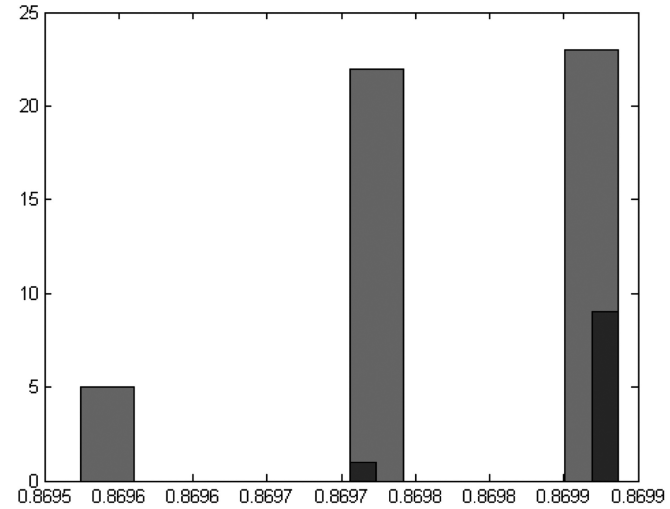
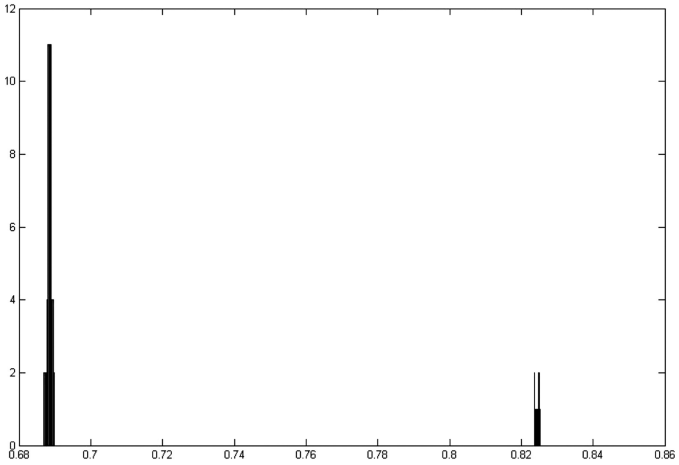


Fig. 4. A control chart of the fractal information dimension for the pattern-reoccurring process samples without deviation (samples 1 to 50) and with deviation of ∼5% (samples 51 to 60). Numerically derived control limits are marked with continuous lines. Analytically derived control limits are marked with dashed lines (color figure provided online).

Figure 4 presents the control chart for the information dimension statistic with control limits adjusted for $\alpha = 0.0027$. The information dimension values are marked blue. The limits that were obtained numerically are marked with continuous lines, whereas control limits based on the analytical computation are marked with dashed lines. The numerically drived control limits were achieved as follows: the monitoring statistic (i.e., the fractal dimension) of in-control data was computed for a large number of samples. Since we have demonstrated (see Section 3.2.2) that the monitoring statistic is approximately normally and independently distributed, we simply computed the mean and standard deviation of the fractal dimension measures for the in-control samples and applied the control limits for the specified Type I error rate. The analytically derived control limits were obtained using the procedure presented in Section 3.2.2. Both types of control limits clearly enable us to distinguish between in-control samples (samples 1 to 50) that contain the 8, 0, 4 patterns and out-of-control samples (samples 51 to 60), in which approximately 5% of the patterns are reversed to 4, 0, 8. Note that the overall number of data points in this experiment was 900 000. The out-of-control average run length (ARL_1) measure for this experiment is one. A description of the complete experiment, including ARL_1 measures of different process deviation levels, is presented in Section 4.1, under the "Pattern-reoccurring process" case.



(a)



(b)

Fig. 5. (a) A box counting dimension histogram for the pattern-reoccurring samples without deviation (samples 1 to 50, marked on left) and for those with deviation of ∼5% (samples 51 to 60, marked on right) and (b) a correlation dimension histogram for the pattern-reoccurring samples without deviation (samples 1 to 50, marked on left) and for those with deviation of ∼5% (samples 51 to 60, marked on right) (color figure provided online).

For the purpose of root cause analysis, one can analyze other fractal dimension statistics whenever a change point is detected. We continue with the above pattern-reoccurring example and start by examining histograms of both the box counting and the correlation dimension measures before and after the process change point.

Figures 5(a) and 5(b) present histograms of the box counting dimension and the correlation dimension measures, respectively. Both histograms show the dimension values before and after the change point in the process. Dimension measures for in-control samples are marked in red, while dimension measures for out-of-control samples are marked blue. If the figures are in grayscale, please refer

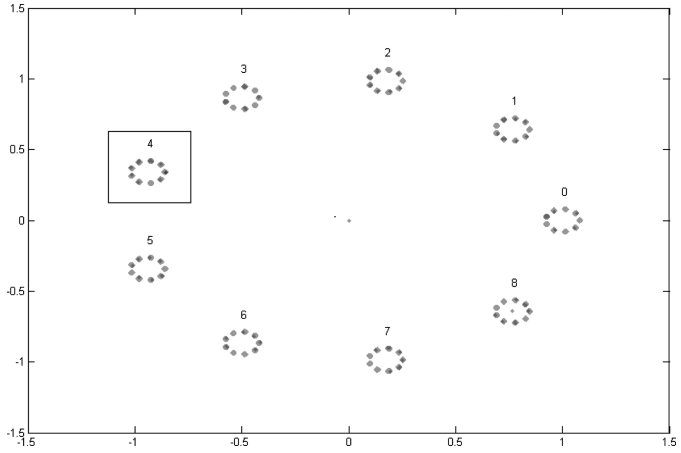


Fig. 6. Fractal-SPC monitoring chart. Green points indicate in-control process samples; blue points indicate patterns of in-control process samples; red point indicates out-of-control samples; pink points indicate patterns of out-of-control samples. For demonstration purposes, we explore the blue marked area detected in circle address 4 (color figure provided online).

to the online version of the article, where the figures appear in color.

Looking at the histograms, one can draw the following conclusions regarding the process deviation: it is caused by significant correlation changes within the process subsequences (in Fig. 5(b), D_{cor} shows an excellent separation); however, it does not involve the appearance of new subsequences (since D_{BC} shows a weak separation, as presented in Fig. 5(a)).

The fractal-SPC chart is presented in Figs. 6 to 8. As can be seen, it has the advantage of capturing both in-control and out-of-control patterns that may affect the process. Thus, the fractal-SPC can be considered both as a special

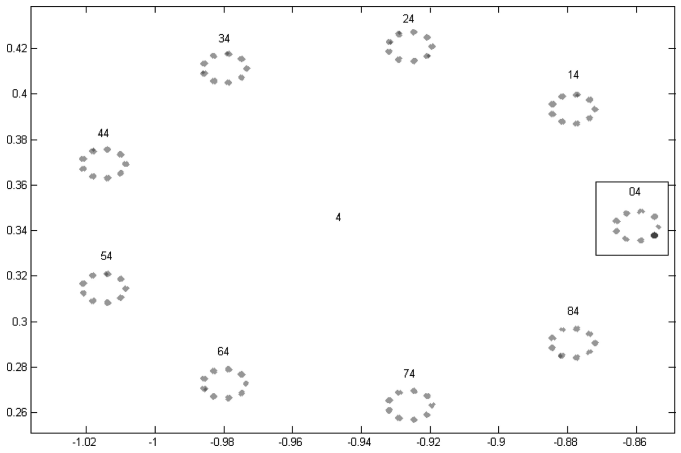


Fig. 7. Zooming into circle address 4. A blue marked area revealing the in-control pattern is detected in circle address 04 (color figure provided online).

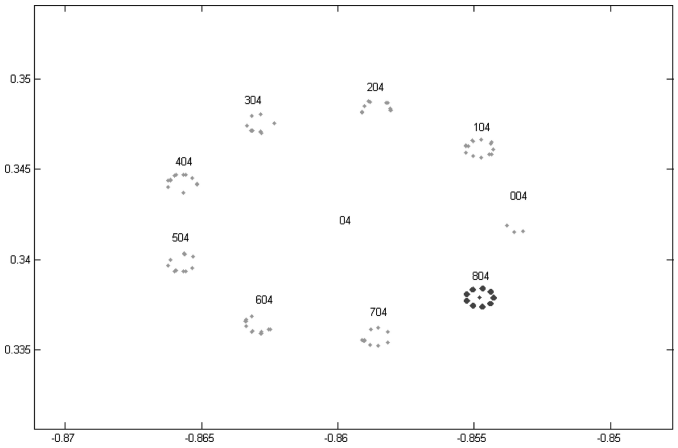


Fig. 8. Zooming in to circle address 04. A blue marked area revealing the rest of the in-control pattern is detected in circle address 804 (color figure provided online).

cause chart and as a casual cause chart, as defined by Alwan and Roberts (1988).

For illustration, we suggest the following color scheme to ease the process of visual process monitoring and root cause analysis: points of in-control process samples in green; patterns detected for in-control process samples in blue; points of out-of-control samples in red; patterns detected for out-of-control samples in pink. If figures are in grayscale, please refer to the online version of the article, where the figures appear in color.

Figures 6 to 8 demonstrate how an in-control pattern can be revealed.

Similarly, out-of-control patterns are revealed by zooming in to the pink marked circles. Figure 9 shows the third resolution graph revealing the out-of-control pattern. Thus, the presented fractal-SPC control chart enables us to visually learn and analyze the process. One can reveal underlying patterns and correlations in both in-control and

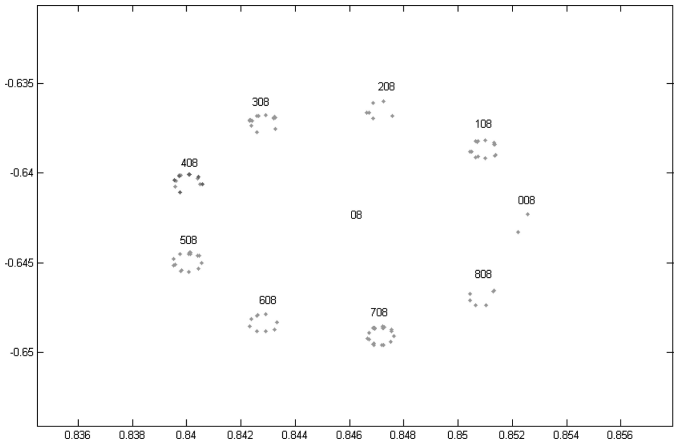


Fig. 9. A third resolution fractal graph. The pink marked area reveals the out-of-control pattern in circle address 408 (color figure provided online).

out-of-control stages and use it for root cause analysis and assignable causes identification.

4. Comparative study with known methods

4.1. *Fractal-SPC versus conventional robust SPC approaches*

In this section we study the performance of the proposed fractal-SPC by evaluating the out-of-control average run length (ARL) in various process-deviation scenarios. We compare the fractal-SPC with several conventional SPC procedures: in particular, the Shewhart, Exponentially Weighted Moving Average (EWMA), and the CUSUM SPC that are considered to be relatively robust to underlying independence assumptions. The control limits were calculated by using a distribution fit of the observed statistics and were set such that the in-control ARL would satisfy a required level of 370.4. Shore (2000) demonstrated that for non-normal data, the standard normal percentiles can be used to compute control limits in cases where the skewness of the process is low. Nonetheless, we found that despite the fact that the generated processes had a low skewness, the conventional robust SPC procedures failed to control the processes, as a result of the violation of the independence assumption. These methods generated control limits that were widely spread and resulted in wrong (much higher) in-control (as well as out-of-control) ARL values. We did not compare the fractal-SPC to the ARIMA SPC family, since Ben-Gal and Singer (2004) have already demonstrated that ARIMA models are inadequate for the monitoring of state-dependant Markovian processes with discrete measures, as is the case for most of the processes considered here.

Three types of processes were generated in the study: Markov, pattern reoccurring, and a distribution-based process. Each process was represented by an ordered sequence of symbols that can be seen also as a subset of observations measures from a multivariate process. Note that the conventional inverse transform method was used to generate a discrete data set from a given (continuous) distribution. In all three cases, each point in the control chart was derived from a sample that contained 15 000 data points representing a data-rich environment. Five levels of process deviation were defined, relying on the percentage of points that deviate from the underlying process in the sample. These levels were used to simulate a gradually deteriorating process. These deviation levels, ranging between 0.66% and 13%, are indicated in the first columns of Table 1, which summarizes the experiment results. For example, a deviation level of 0.66% represents a sample in which only 100 points are generated by a deviated out-of-control process, with the remaining 14 900 points being generated by the underlying in-control process. For each level of process deviation, the first 50 samples were in control and the last 10 samples consisted of the defined percentage of deviated

Table 1. Out-of-control ARL for three types of processes and five deviation levels

Percentage of deviated points (%)	*ARL_1 of Shewhart, EWMA, CUSUM, for all three process types*	*ARL_1 of the fractal-SPC*		
		Markovian process	*Pattern-based process*	*Distribution-based process*
0.66	d.n.i.	3.6	1	96.15
1.66	d.n.i.	1.05	1	28.5
3.33	d.n.i.	1	1	11.9
6.66	d.n.i.	1	1	1.5
13.33	d.n.i.	1	1	1

data points. Note that for each experiment, and thus each level of process deviation, 900 000 data points (60 samples of 15 000 data points) were generated (i.e., for each process scenario, $5 \times 900\,000 = 4500\,000$ data points were generated), to create a data-rich environment. Each simulation run was replicated 10 times and returned similar results in each run. Note that the suggested fractal-SPC is also relevant to smaller samples, as demonstrated in Section 4.2.

Table 1 summarizes the experimental results for the three types of processes as follows. Note that the various types of process deviation were not identified by the Shewhart, EWMA, or CUSUM SPC methods (abbreviated to d.n.i.; i.e., deviation not identified). This could be explained by the dependencies in the generated processes that resulted in spread control limits that contained both in-control and out-of-control samples.

Markov process: A Markov process is generated from a given transition matrix. Deviation from the generating process is obtained by changing the transition matrix in a manner that relatively preserves the sample average. Note that Markov processes have been used to represent many real-life settings, including queuing systems, buffer monitoring in manufacturing lines with known production probabilities, and feedback-controlled processes (Singer and Ben-Gal, 2007).

Pattern-reoccurring process: Each sample consists of unstructured noise, generated from a uniform distribution, with the pattern 8, 0, 4 being randomly inserted, approximately every 15 data points. Deviation from the generating process is obtained by changing the order of the reoccurring pattern to 4, 0, 8. Pattern-reoccurring processes can be used to represent machine types, part types, or production sequencing, as shown in Ruschin-Rimini *et al*. (2012).

Distribution-based process: Each sample is generated randomly from an underlying normal distribution. Deviation from the generated process is obtained by changing the underlying distribution to a uniform distribution and discretizing its values. Such a distribution change can appear, for example, when monitoring an independent measure of the process or when monitoring residuals in residual-based control charts for non-normal situations (see Castagliola and Tsung (2005)).

We refer the interested reader who would like to reproduce the experiments to a technical reference (Ruschin-Rimini *et al*., 2011b), that contains the generated data and the relevant transition matrix utilized for data generation of the Markov process scenario.

4.1.1. *The fractal information dimension computation results*

The dimension of information measure was computed for each of the 60 samples and per each experiment (i.e., for each deviation level and per each process type). For illustration purposes, Fig. 10 demonstrates measures of the fractal information dimensions in the case of a deviation level of 3.33% in the Markov process for (i) the first 50 in-control samples (marked on the left-hand side); and (ii) the last 10 samples consisting of the defined percentage of deviated data points (marked on the right-hand side). Figure 11 presents the fractal dimension chart with its control limits for this case. Control limits achieved numerically are marked with continuous lines, whereas control limits based on analytically derived computation according to the procedure presented in Section 3.2 are marked with dashed lines. As demonstrated, both methods for control limit computations clearly distinguish between in-control

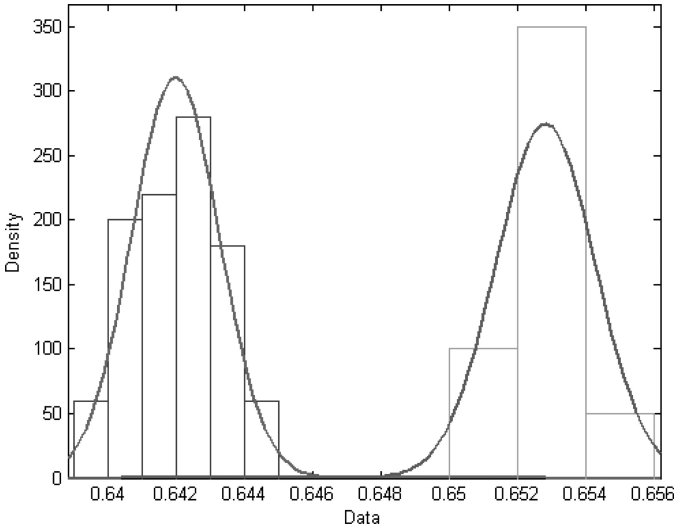


Fig. 10. A fractal dimension histogram for the Markov process without deviation (on the left) and with deviation of 3.33% of the data (on the right) (color figure provided online).

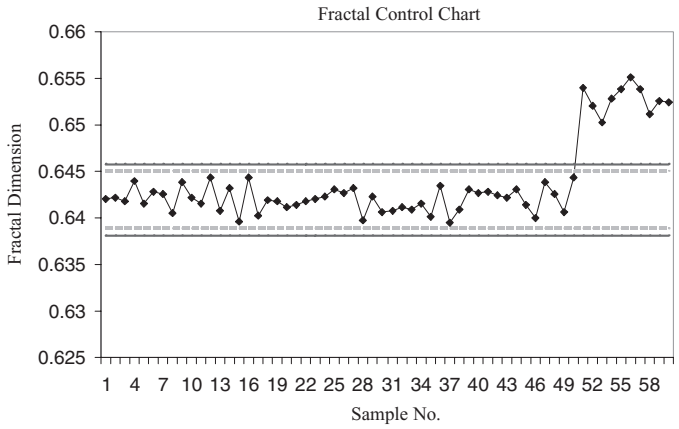


Fig. 11. Information-dimension chart for the Markov process samples without deviation (samples 1 to 50) and for samples with a deviation level of 3.33% (samples 51 to 60). Numerically derived control limits (∼0.638–0.646) are marked with continuous lines. Analytically derived control limits (∼0.6439–0.646) are marked with dashed lines (color figure provided online).

samples (samples 1 to 50) and out-of-control samples (samples 51 to 60).

As Fig. 10 shows, the obtained distributions reflect a good separation between the two populations by the fractal dimension statistic. In Fig. 12 we present both the Shewhart and the EWMA charts with the same deviation level of 3.33% of the Markov process for both (i) the in-control process (samples 1 to 50) and (ii) the out-of-control process (samples 51 to 60). It can be clearly seen that these conventional SPC approaches cannot reveal the change point in the process. The inherent deviation of the process is significantly large such that the process appears within the control limits both before and after the change point.

Our experimental results include other cases in which the dimension of information succeeds in identifying process deviations, whereas traditional control charts fail to do so. However, certain types of process deviations are not well indicated by the fractal-SPC, such as the case of small deviations in certain distribution-based processes. In these cases, further work is needed to fine-tune the fractal-SPC.

4.2. *Fractal-SPC versus CSPC multi-attribute control chart*

As previously mentioned, Ben-Gal *et al*. (2003) proposed the CSPC as a model-generic framework that can deal with autocorrelated non-linear state-dependent processes. Since the proposed fractal-SPC method addresses the same requirements of a model-generic framework, we compare the performance of fractal-SPC with the CSPC method by evaluating the out-of-control ARL in various process scenarios.

Jolayemi (1999) proposed the Multi-Attribute Control Chart (MACC) as a multivariate control chart technique aimed at monitoring multi-attribute data by a single chart. The MACC model is based on an approximation for the convolution of independent binomial variables (Jolayemi, 1992) and on an extension of *np*-control charts. Since the proposed fractal-SPC method also addresses multivariate processes with a finite state space, we compare the performance of fractal-SPC with that for the MACC model by evaluating the out-of-control ARL in various process scenarios.

We also compare all three SPC methods with the same conventional SPC procedures mentioned in Section 4.1. Four types of processes were generated in this study: (i) Markovian; (ii) pattern reoccurring; (iii) pattern interrupted by unstructured noise; and (iv) distribution based. In all four cases, the learning phase (Phase 1 of SPC processes) included 15 000 data points of a historical in-control process, representing a data-rich environment. The monitoring phase (Phase 2 of SPC processes) included samples of 100 data points each to reflect an online mode of sampling with less data. For each process scenario, the first 50 samples were in the in-control state and the last 10 samples consisted of out-of-control data points that simulate a step change in the process.

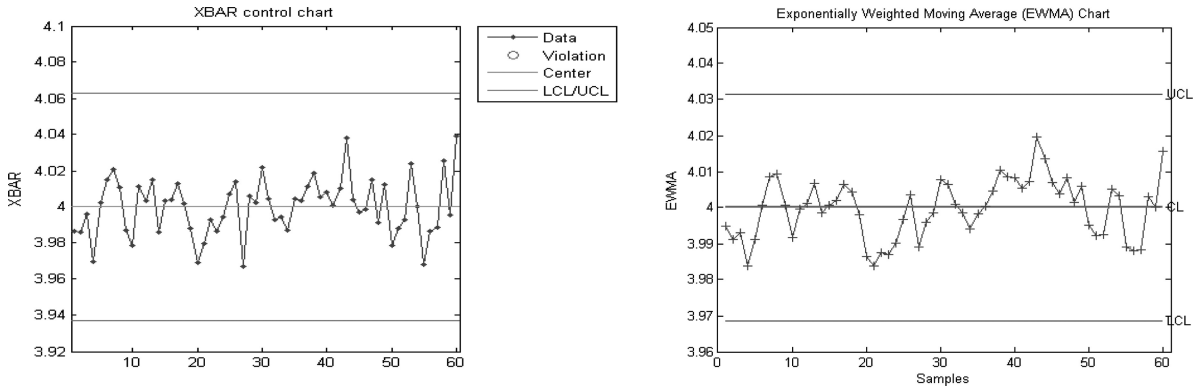


Fig. 12. The Shewhart and EWMA control charts for the Markov process without deviation (samples 1 to 50) and with deviation of 3.33% (samples 51 to 60) (color figure provided online).

Table 2. Out-of-control ARL for four types of processes

	ARL_1					
Process type	*Shewhart*	*EWMA*	*CUSUM*	*Fractal SPC*	*CSPC*	*MACC*
Fixed-order Markov	d.n.i.	d.n.i.	d.n.i.	1.02	1	d.n.i.
Pattern reoccurring	d.n.i.	d.n.i.	d.n.i.	1.04	3.15	d.n.i.
Pattern interrupted by unstructured noise	d.n.i.	d.n.i.	d.n.i.	1.07	6.92	d.n.i.
Distribution based	d.n.i.	d.n.i.	d.n.i.	1.08	1	d.n.i.

Table 2 summarizes the experimental results for the four types of processes. Note that the various types of process deviation were not identified by the Shewhart, EWMA, CUSUM, or MACC SPC methods (abbreviated d.n.i., i.e., deviation not identified). The Markov process, pattern-reoccurring process, and distribution-based process were generated as explained in the previous section. The pattern interrupted by unstructured noise process was generated as follows: each sample was generated from a uniform distribution to simulate unstructured noise. The pattern 8-X-0-X-4 (where X represents any symbol generated from the underlying distribution) was inserted approximately every 15 data points. Deviation from the generating process was obtained by changing the order of the reoccurring pattern to 4-X-0-X-8. Such an interrupted pattern was chosen in order to simulate realistic conditions regarding reordering or shuffles of real processes (see the illustrative example in Section 3.5), as well as to examine the robustness of both SPC schemes to varying order models.

4.2.1. *Computation results*

Fractal SPC versus MACC. Our experimental results demonstrate cases in which the various types of process deviation were not identified by the MACC model. This result is due to the fact that in order to monitor multiple attributes in a single chart, the MACC model utilizes a single monitoring statistic that averages the *np* values of each attribute; hence, roughly speaking it "loses information" regarding each attribute independently. Moreover, even if each attribute is monitored separately, the detection of deviations such as change in patterns and changes in correlations between attributes is not guaranteed, since attributes are assumed to be independent.

In order to emphasize the difference between the MACC model and the fractal-SPC model, one can consider three levels of monitoring multivariate data, as follows.

1. Methods that utilize a monitoring statistic that averages the attributes' data.
2. Methods that monitor each attribute independently.
3. Methods that monitor each attribute yet can identify autocorrelations and patterns between them, such as the fractal-SPC, which monitors all data in one single chart.

The first level comprises the least information regarding the attributes. Averaging attributes' data results in loss of

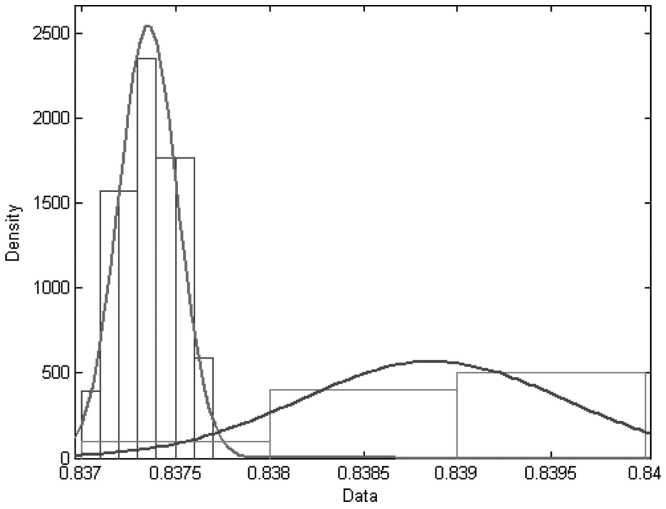


Fig. 13. A histogram of the fractal-SPC monitoring statistic for the case of patterns interrupted by unstructured noise: without deviation (on the left) and with deviation (on the right) (color figure provided online).

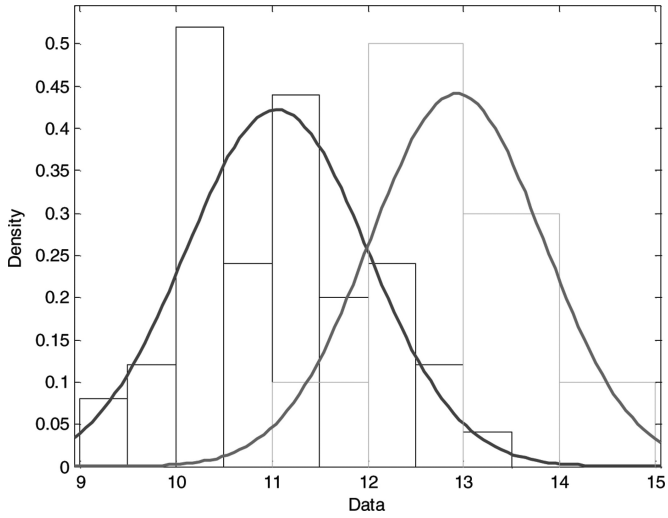


Fig. 14. The CSPC monitoring statistic histogram for the case of patterns interrupted by unstructured noise: without deviation (on the left) and with deviation (on the right) (color figure provided online).

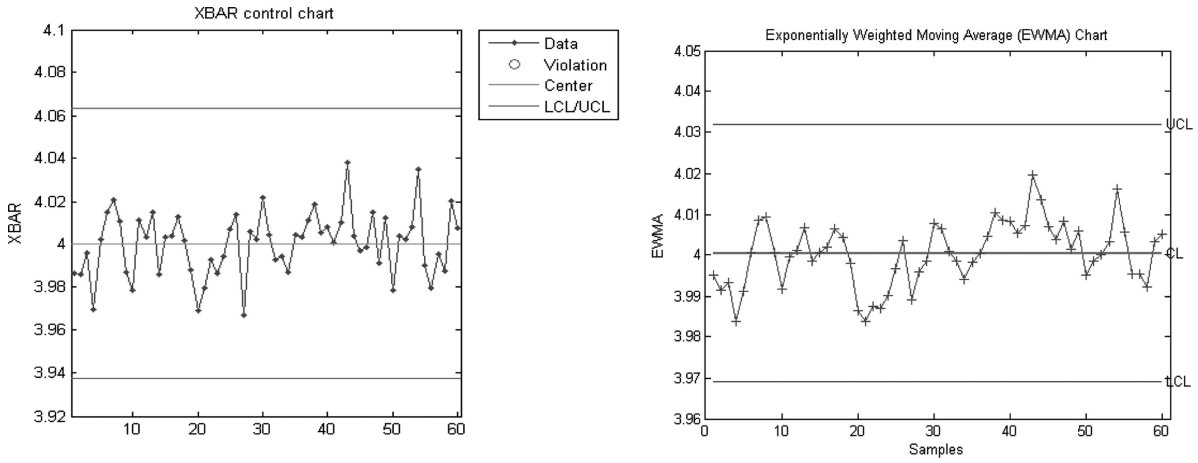


Fig. 15. The Shewhart and EWMA control charts for the pattern-reoccurring process without deviation (samples 1 to 50) and with deviation (samples 51 to 60) (color figure provided online).

information regarding each attribute independently. The MACC model matches this level of monitoring. The second monitoring level matches traditional control charts for attributes such as the p, np, c, and u, which are aimed at univariate data. When using this level of monitoring in cases of multivariate processes, it is suggested to monitor each attribute independently. Such monitoring does not suit cases in which data are autocorrelated, as some of the processes demonstrated in the presented comparative study. Moreover, monitoring of each attribute separately increases the statistical Type I error and is more complex to manage and maintain than a single chart scheme. Consequently, the fractal-SPC, which matches the third of the defined monitoring levels, outperforms traditional approaches for monitoring multi-attribute data in cases in which autocorrelation exists.

Fractal-SPC versus CSPC. For illustration purposes, Fig. 13 demonstrates measures of the fractal-SPC monitoring statistic in the case of the patterns interrupted by unstructured noise process: (i) the first 50 in-control samples (marked on the left-hand side) and (ii) the last 10 samples consisting of deviated data points (marked on the right-hand side). For illustration purposes, Fig. 14 demonstrates the measures of the CSPC monitoring statistic in the same case: (i) the first 50 in-control samples (on the left-hand side) and (ii) the last 10 samples consisting of deviated data points (on the right-hand side).

Figure 15 presents both Shewhart and EWMA charts for the case of patterns interrupted by unstructured noise in both (i) the in-control process (samples 1 to 50) and (ii) the out-of-control process (deviated samples 51 to 60). As seen, these conventional SPC approaches do not reveal the change point in the process.

Our experimental results demonstrate cases in which the fractal-SPC and CSPC methods succeed in identifying deviations in pattern-based processes, whereas traditional control charts fail to do so. As indicated in Table 2, the main advantage of the fractal-SPC over the CSPC is when monitoring processes that are populated by deterministic patterns with or without unstructured noise. The CSPC method is advantageous when the correlation structure is more complex and unknown; i.e., when the order of the reoccurring pattern varies and the dependence order is inhomogeneous. The CSPC does not require knowledge of the monitoring order (resolution) in advance and thus can represent processes that are generated by variable-order Markov processes or by context-specific Bayesian networks in the case of inhomogeneous models.

Additional preliminary results, presented in Ruschin-Rimini *et al.* (2012), mainly focus on monitoring the effects of various operational settings on the quality of production. The method was implemented in a world-leading automotive manufacturer. It provides a realistic example where the proposed method can benefit a service or an industrial organization and supports root cause analysis applications.

5. Conclusions and further research

In this article we proposed a fractal-SPC method that has several attractive features. It can learn the process data dependence and its underlying distribution without assuming *a priori* information. Thus, it is a model-generic (nonparametric) approach and thus extends the current scope of control charts to non-linear state-dependent processes (Ben-Gal *et al*., 2003). This advantage over traditional control charts is particularly appealing when monitoring data-rich processes with an unknown underlying model.

The obtained fractals can be used to visually track anomalies in data-rich patterns whose order is of several magnitudes larger than the one used in traditional SPC tools. The fractal representation copes well with modern monitoring schemes that are executed on PC screens rather than on paper sheets: the proposed IFS transformation projects the multidimensional patterns into a

two-dimensional space. Moreover, it applies the ability of zooming in to areas of interest to better analyze patterns and support root cause analysis tasks.

In order to automate the fractal monitoring process, we compute the fractal dimension as a representing statistic that is used to dynamically monitor the process behavior. Our selection of the fractal dimension is encouraged by the theoretical relations that are established with information theory and data compression techniques, which are known as viable for data-rich applications. We demonstrate that various definitions of the fractal dimension can support a multi-level inspection for simultaneously representing both common and rare patterns in the inspected process.

Despite these advantages, the suggested fractal-SPC is limited in its current form to discrete processes with a finite state space (alphabet). The proposed control chart requires a relatively large amount of data to construct the initial in-control fractal model; hence, it is best suited for data-rich environments. Future research could extend the suggested method to handle continuous problems. It could improve the fractal visualization chart to exploit a greater portion of the screen's available pixels and use modern graphical techniques. Finally, it could define a better default threshold value for the root cause analysis phase, as a function of the in-control data type and the type of process deviation.

Another research direction could be to focus on studying the various forms of the fractal dimension in terms of computation tractability versus its sensitivity and specificity performance. Such research should analyze the relations among these statistics to provide a unified monitoring scheme that integrates all of them into a single anomaly-detection and decision-making tool. We anticipate that such integration can provide excellent inputs for root cause analyses, especially if it is accompanied by machine-learning procedures.

Acknowledgements

This research was supported by The Israel Science Foundation (grant 1362/10).

References

Alwan, L.C., Ebrahimi, N., and Soofi, E.S. (1998) Information theoretic framework for process control. *European Journal of Operations Research*, **111**, 526–542.

Alwan, L.C. and Roberts, H.V. (1988) Time-series modeling for statistical process control. *Journal of Business & Economics Statistics*, **6**(1) 87–95.

Apley, D.W. and Shi, J. (1999) The GLRT for statistical process control of autocorrelated processes. *IIE Transactions*, **31**, 1123–1134.

Barnsley, M. (1988) *Fractals Everywhere*, Academic Press, Boston, MA.

Barnsley, M. and Hurd, L.P. (1993) *Fractal Image Compression*, A.K. Peters, Boston, MA.

Basarin, G.R. (1959) On a statistical estimate for the entropy of a sequence of independent random variables. *Teor Verojanost. i Primenen.*, **4**, 361–364.

Ben-Gal, I., Morag, G. and Shmilovici, A. (2003) CSPC: A monitoring procedure for state dependent processes. *Technometrics*, **45**(4), 293–311.

Ben-Gal, I. and Singer, G. (2004) Statistical process control via context modeling of finite state processes: an application to production monitoring. *IIE Transactions on Quality and Reliability*, **36**(5), 401–415.

Boardman, T.J. and Boardman, E.C. (1990) Don't touch that funnel. *Quality Progress*, **23**, 56–69.

Box, G.E.P. and Jenkins, G.M. (1976) *Times Series Analysis, Forecasting and Control*, Holden-Day, Oakland, CA.

Castagliola, P. and Tsung, F. (2005) Autocorrelated SPC for non-normal situations. *Quality and Reliability Engineering International*, **21**, 131–161.

Chen, C.C. and Chen, C.C. (2003) Texture synthesis, a review and experiments. *Journal of Information Science and Engineering*, **19**, 371–380.

Cheng, S. and Thaga, K. (2005) Max-CUSUM chart for autocorrelated processes. *Statistical Sinica*, **15**(2), 527–546.

Cochran, W.G. (1952) The chi-square test of goodness of fit. *The Annals of Mathematical Statistics*, **23**, 315–345.

Cover, T.M. and Thomas, J.A. (1995) *Element of Information Theory*, 2nd ed., John Wiley & Sons Inc., Hoboken, NJ.

English, J.R., Martin, T., Yaz, E., and Elsayed, E. (2001) Change point detection and control using statistical process control and automatic process control. Presented at the *IIE Annual Conference*, Dallas, TX.

Grassberger, P. (1983) Characterization of strange attractors. *Physical Review Letters*, **50**, 346–349.

Grassberger, P. and Procacciaa, I. (1983) Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, **9**(1-2), 189–208.

Harris, B. (1975) The statistical estimation of entropy in the non-parametric case. Technical Summary Report no. 1605, Mathematics Research Center, University of Wisconsin–Madison, Madison.

Harris, T.J. and Ross, W.H. (1991) Statistical process control procedures for correlated observations. *Canadian Journal of Chemical Engineering*, **69**(1), 48–57.

Jolayemi, J.K. (1992) Convolution of independent binomial variables: an approximation method and a comparative study. *Computation Statistics and Data Analysis*, **18**, 403–417.

Jolayemi, J.K. (1999) A statistical model for the design of multiattribute control chart. *The Indian Journal of Statistics*, **61**(2), 351–365.

Kaminski, F.C., Benneyan, J.C., Davis, R.D. and Burke, R.J. (1992) Statistical control charts based on a geometric distribution. *Journal of Quality Technology*, **24**, 63–69.

Keim, D.A. (2002) Information visualization and visual data mining. *IEEE Transactions of Visualization and Computer Graphics*, **7**(1), 100–107.

Kim, S.H., Alexopoulos, C., Tsui, K.L. and Wilson, J.R. (2007) A distribution free tabular CUSUM chart for autocorrelated data. *IIE Transactions*, **39**, 317–330.

Lu, C.W. and Reynolds, M.R. (1999a) Control charts for monitoring the mean and variance of autocorrelated processes. *Journal of Quality Technology*, **31**(3), 259–274.

Lu, C.W. and Reynolds, M.R. (1999b) EWMA control charts to monitor the mean of autocorrelated processes. *Journal of Quality Technology*, **31**(2), 166–188.

Luce, R.D. (1955) The theory of selective information and some of its behavioral applications, in *Developments in Mathematical Psychology*, Luce, R.D. (ed), The Free Press, Glencoe, IL, pp. 45–46.

Mason, R.L., Tracy, N.D. and Young, J.C. (1995) Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, **27**, 99–108.

Miller, G.A. and Madow, W.G. (1954), On the maximum likelihood estimate of the Shannon-Weaver measure of information. Technical Report, Air Force Cambridge Research Center.

Montgomery, D.C. and Mastrangelo, C.M. (1991) Some statistical process control methods for autocorrelated data. *Journal of Quality Technology*, **23**(3), 179–204.

Perry, M.B. and Pignatiello, J.J. (2006) Estimation of the change point of a normal process mean with a linear step disturbance in SPC. *Quality Technology and Quantitive Management*, **3**(3), 325–334.

Quesenberry, C.P. (1991a) SPC Q charts for a binomial parameter: short or long runs. *Journal of Quality Technology*, **23**, 239–246.

Quesenberry, C.P. (1991b) SPC Q charts for a Poisson parameter *k*: short or long runs. *Journal of Quality Technology*, **23**, 296–303.

Ren, Y., Ding, Y. and Zhou, S. (2006) A data-mining approach to study the significance of nonlinearity in multi-station assembly processes. *IIE Transactions*, **38**(12), 10691083.

Rokach, L., Romano, R. and Maimon, O. (2008) Mining manufacturing databases to discover the effect of operation sequence on the product quality. *Journal of Intelligent Manufacturing*, **19**(3), 313–325.

Runger, G.C., Alt, F.B., and Montgomery, D.C. (1996) Contributors to a multivariate SPC chart signal. *Communications in Statistics, Theory and Methods*, **25**, 2203–2213.

Runger, G.C. and Willemain, T. (1995) Model-based and model-free control of autocorrelated processes. *Journal of Quality Technology*, **27**, 283–292.

Runger, G.C., Willemain, T. and Prabhu, S. (1995), Average-run-lengths for CUSUM control charts applied to residuals. *Communications in Statistics, Part A—Theory and Methods*, **24**, 273–282.

Ruschin-Rimini, N., Ben-Gal, I. and Maimon, O. (2011a) Technical paper: fractal dimension types in the context of SPC: analytical study, available at http://www.eng.tau.ac.il/∼bengal/fractal2.pdf

Ruschin-Rimini, N., Ben-Gal, I., and Maimon, O. (2011b) Web appendix: data generated in experiments, available at http://www.eng.tau.ac.il/∼bengal/fractalData.xlsx

Ruschin-Rimini, N. and Maimon, O. (2010) Visual analysis of sequences using fractal geometry, in *Data Mining and Knowledge Discovery Handbook*, second edition, Springer, New York, pp. 591–601.

Ruschin-Rimini, N., Maimon, O. and Romano, R. (2012) Visual analysis of quality-related manufacturing data using fractal geometry. *Journal of Intelligent Manufacturing*, **23**(3), 481–495.

Shore, H. (1992) *Total Quality, Quality Control and Quality by Design* (in Hebrew), 2nd ed., 1995. Self-published.

Shore, H. (1998) A new approach to analyzing non-normal data with application to process capability analysis. *International Journal of Production Research*, **36**(7), 1917–1933.

Shore, H. (2000) General control charts for attributes. *IIE Transactions*, **32**, 1149–1160.

Singer, G. and Ben-Gal, I. (2007) The funnel experiment: the Markov-based SPC approach. *Quality and Reliability Engineering International*, **23**, 899–913.

Wasserkrug, S., Gal, A., Etzion, O. and Turchin, Y. (2008) Complex event processing over uncertain data. Presented at the *IEEE Second International Conference on Distributed Event-Based Systems*, Rome, Italy.

Weiss, C.H. (2008) Visual analysis of categorical time series. *Statistical Methodology*, **5**, 56–71.

Weiss, C.H. and Goeb, B. (2008), Discover patterns in categorical time series. *Computational Statistics & Data Analysis*, **52**(9), 4369–4379.

Woodall, W.H. (1997) Control charts based on attribute data: bibliography and review. *Journal of Quality Technology*, **29**(2), 172–183.

Ye, N., Li, X., Chen, Q., Emran, S.M., and Xu, M. (2001) Probabilistic techniques for intrusion detection based on computer audit data. *IEEE Transactions on System, Man, and Cybernetics-Part A: Systems and Humans*, **3**1, 266–274.

Ye, N., Vilbert, S., and Chen, Q. (2003) Computer intrusion detection through EWMA for autocorrelated and uncorrelated data. *IEEE Transactions on Reliability*, **52**, 75–82.

Appendix

Fractal mapping procedure—a running example

In this example we demonstrate the mapping procedure of a sampled sequence consisting of the following three ordered observations: 0, 3, 6. These observations can be taken either from a sliding window of a univariate process with nine possible categories (realizations) 0, 1, . . . , 8 (i.e., $m = 9$) or from a sliding window of a multivariate vector (of any dimension larger than three), where the range of each variable contains up to nine values (categories).

Step 1: Each of the process' categories is associated with one contractive mapping. In this example we associate variable 1 with contractive mapping w_1; variable 2 with contractive mapping $w_2, \ldots$, variable 8 with contractive mapping w_8; and variable 0 with contractive mapping w_9. Following are the appropriate contractive mappings according to Equation (4), where $\alpha = 0.08$:

$$w_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \cos\left(\frac{2\pi}{9}\right) \\ \sin\left(\frac{2\pi}{9}\right) \end{bmatrix},$$

$$w_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \cos\left(2 \times \frac{2\pi}{9}\right) \\ \sin\left(2 \times \frac{2\pi}{9}\right) \end{bmatrix},$$

.
.

$$w_9\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \cos\left(9 \times \frac{2\pi}{9}\right) \\ \sin\left(9 \times \frac{2\pi}{9}\right) \end{bmatrix}.$$

Step 2: Accordingly, the sequence 0, 3, 6 is represented by a sequence of three corresponding contractive mappings: $\{w_9, w_3, w_6\}$; $x_{(0)}$ is arbitrarily selected to be plotted in

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Step 3: We recursively apply each of the three contractive mappings $w_9(x_0)$, $w_3(x_1)$, $w_6(x_2)$ by their order in the sequence. We start by applying contractive mapping $w_9(x_0)$ to obtain point $x_{(1)}$ as follows:

$$x_{(1)} = w_9\left(\left[x_{(0)}\right]\right) = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\left(9 \times \frac{2\pi}{9}\right) \\ \sin\left(9 \times \frac{2\pi}{9}\right) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

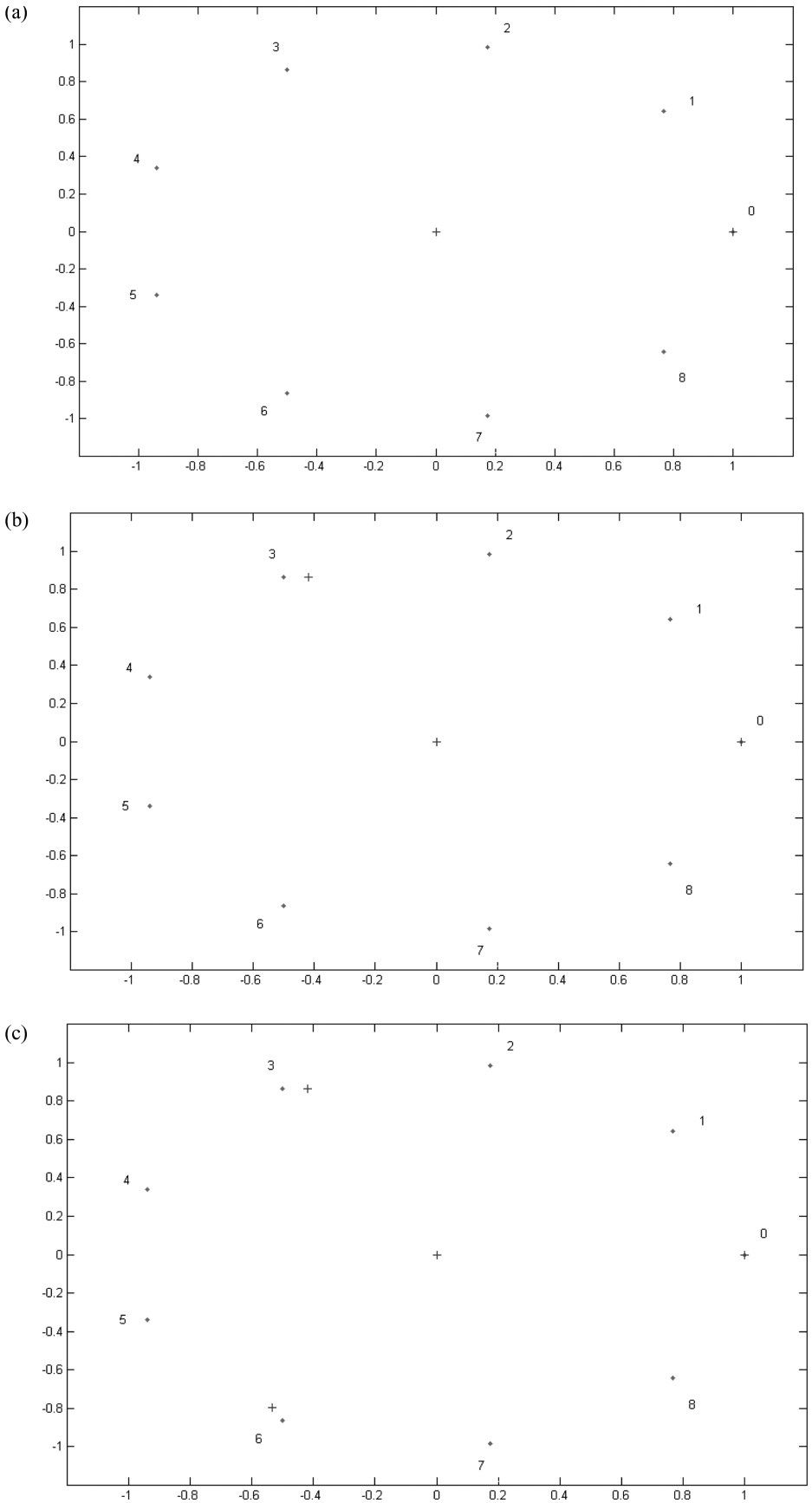


Fig. A1. The plotting of three fractal mapping iterations for the ordered sample 0 3 6: (a) points $x_{(0)}$ and $x_{(1)}$; (b) points $x_{(0)}$, $x_{(1)}$, and $x_{(2)}$; and (c) points $x_{(0)}$, $x_{(1)}$, $x_{(2)}$, and $x_{(3)}$ (color figure provided online).

We then apply contractive mapping $w_3(x_1)$ to obtain $x_{(2)}$, as follows:

$$x_{(2)} = w_3\left(\left[x_{(1)}\right]\right) = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\left(3 \times \frac{2\pi}{9}\right) \\ \sin\left(3 \times \frac{2\pi}{9}\right) \end{bmatrix} = \begin{bmatrix} -0.42 \\ 0.866 \end{bmatrix}.$$

Similarly, we obtain $x_{(3)}$. Figures A1(a) to A1(c) demonstrate the plotting outcome of all three iterations. respectively: Fig. A1(a) presents points $x_{(0)}$ and $x_{(1)}$, Fig. A1(b) presents points $x_{(0)}$, $x_{(1)}$, and $x_{(2)}$, etc. Points $x_{(0)}$ to $x_{(3)}$ are marked by blue cross signs. For illustration purposes, all nine circle centers of the fractal graph are marked by red points. In order to illustrate the fractal interpretation procedure, one can focus on Fig. A1(b). Note that point $x_{(2)}$ is positioned close to circle address 3, signifying variable 3. Moreover, note that by zooming into circle address 3, point $x_{(2)}$ is positioned in circle address 03, indicative of its preceding variable 0.

Biographies

Noa Ruschin-Rimini holds a Ph.D. degree from the Engineering Faculty at Tel Aviv University. Her research interests include anomaly and pattern detection methods for non-linear autocorrelated processes, Big Data (analysis, learning, tools and applications), and predictive analytics. She holds both B.Sc. (1998) and M.Sc. (2007) degrees from the Industrial Engineering Department, Tel-Aviv University. Her research received grants from the Israeli Science Foundation (ISF) and from General Motors. She has received several best papers awards. Her papers have been published in IIE Transactions, Journal of Intelligent Manufacturing, Data Mining and Knowledge Discovery Handbook (2nd Edition) and presented in conferences such as INFORMS 2011 Annual Meeting, ENBIS 2011 Annual Meeting and more. Furthermore, she held a lecturer position in the Industrial Engineering Department in Tel-Aviv University. Before joining Tel-Aviv University, she held several positions in Oracle Israel Ltd. including a Product Manager and the Supervisor of the Presales Department. She was also a Solutions Sales Manager at IBM Israel Ltd., and a Business Development Manager in an Israeli Start-Up company.

Irad Ben-Gal is an Associate Professor at Tel Aviv University. His research interests include statistical methods for control and analysis of stochastic processes and applications of information theory and machine learning to industrial and service systems. He holds a B.Sc. (1992) degree from Tel-Aviv University and M.Sc. (1996) and Ph.D. (1998) degrees from Boston University. He has written and edited five books; published more than 80 scientific papers, patents, and book chapters; and received several best papers awards. His papers have been published in *IIE Transactions*, *Technometrics*, *IEEE Transactions*, *Quality and Reliability Engineering International*, *Journal of Statistical Planning and Inference*, *IJPR*, *Bioinformatics*, and *BMC Bioinformatics*. He is a member of the Institute for Operations Research and Management Sciences, the Institute of Industrial Engineers, The European Network for Business and Industrial Statistics, and an elected member in the International Statistical Institute. He is a Department Editor for *IIE Transactions on Quality and Reliability* and serves on the editorial boards of several other professional journals. He has supervised dozens of graduate students and received several research grants, among them from General Motors, IEEE, the Israeli Ministry of Science, and the European Community. He worked with such companies as Pratt & Whitney, Siemens, Proctor and Gamble, Kimberly-Clark, Applied Materials, Erikson, IBM, and others.

Oded Maimon received B.Sc. degrees in Industrial Engineering and Mechanical Engineering and an M.Sc. degree in Operations Research from The Technion, Haifa, Israel, and a Ph.D. degree from Purdue University, West Lafayette, Indian. He is a Professor and former Chair of the Industrial Engineering Department, Tel-Aviv University. Before joining Tel-Aviv University, he was a Research Scientist at the Massachusetts Institute of Technology, Cambridge, Massachusetts, and a Project Leader at Digital Equipment Corporation. He co-authored the book *Decomposition Methodology for Knowledge Discovery and Data Mining* (World Scientific, Singapore, 2005) and co-edited the handbook *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Research Scientists and Practitioners* (Springer, New York, 2005).