ANALYSIS OF PROCESS CONTROL BASELINE DATA USING DATA MINING

by

HANG ZHANG

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under the direction of

Susan L. Albin

And approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October 2007

ABSTRACT OF THE DISSERTATION

Analysis of process control baseline data using data mining

By HANG ZHANG

Dissertation Director:

Susan L. Albin

There are two phases in multivariate statistical process control (MSPC). In phase I, we model baseline data off-line to characterize the process. Baseline data is a collection of observations describing successful manufacturing. In phase II, we compare on-line observations to these models to determine whether the process is in control. This dissertation addresses four questions to improve phase I analysis: (1) How many operational modes are in baseline data? (2) In a large historical dataset collected over a long time period, which periods are the baseline? (3) In profile baseline data, are there outlier profiles? (4) When should the phase I model be updated?

Each operational mode appears as a cluster in baseline data. To address the first question, we propose a new method to determine the number of clusters with all of the following critical features: it determines if there is only one cluster, the most common case; it identifies convex or non-convex clusters; and it is insensitive to user-specified parameters. No existing method has them all. Simulations show that the proposed method works well.

We propose a new method to address the second question, where historical data may be collected during both baseline and unsuccessful periods. The identified baseline periods are reasonably long, and have the best product quality with a stable distribution. Through simulated and real datasets, the proposed method shows its robustness to various distributions, in contrast to the existing change point identification method that is very sensitive to the distribution.

We address the third question in the context of complex profiles. We treat complex profiles as high-dimension vectors. We apply the χ^2 control chart to identify outliers. Applied to simulated and real datasets, it demonstrates better performance on complex profiles than the existing nonlinear regression method.

We address the fourth question by testing whether the correlation matrix changes from the baseline. The correlation matrix describes relationships among variables. We propose a new method to diagnose the responsible variables when the change is indicated.

We also discuss the future work of applying MSPC and data mining technologies on data from a brain neural system.

Key words: Statistical Process Control, Data Mining, Phase I, Operational Mode, Profile, Outlier Detection, Number of Clusters, Correlation Matrix.

Acknowledgements

I would like to thank my family. My parents and brother were a constant source of love and support. I am grateful to them for their encouragement and belief in me.

I would especially like to thank my advisor, Dr. Susan L. Albin, for her generous commitment and constant encouragement. This dissertation cannot be completed without her insightful advice. It is such an honor for me to have an advisor and friend like her. She helped me to develop independent thinking and research skills. She continually stimulated my analytical thinking and greatly assisted me with scientific writing. Whenever I got lost in research, her profound insights led me out of the puzzle and opened my horizon. Whenever I felt frustrated, her optimism inspired me and kept me moving forward. What I have learned from her is not only what is presented in this dissertation, but also the positive and optimistic attitude to work and life, especially in face of difficulties. The spirit is now embedded in me. It is the thing that makes me different from five years ago when I first stepped on the soil of America. It is the thing that will make me achieve more in the future than what I have imagined before.

I am also very grateful for having an exceptional dissertation committee and wish to thank Dr. Elsayed A. Elsayed, Dr. Wanpracha (Art) Chaovalitwongse, and Dr. Di Xu for giving valuable advices and thoughtful critiques to my dissertation.

I would also like to extend many thanks to my colleagues and friends. They accompanied me along in this journey. I am happy to have them with me.

Table of contents

List of tables

List of illustration

1 Introduction

Statistical process control (SPC) is a method of monitoring the performance of a manufacturing process by comparing the current state of the process against "successful working conditions". We call working conditions successful if the process is consistently producing products with good quality. Originally, SPC was applied on each critical variable separately, called univariate SPC. With the advent of more and more complex manufacturing processes, and the progress in measurement technology, more critical variables (maybe hundreds) can be monitored on-line. Univariate SPC is not effective in monitoring these processes because there are too many critical variables to be monitored, and correlations among these variables are ignored. Multivariate SPC (MSPC), by building a few statistics to monitor all these critical variables simultaneously, is a much more effective methodology.

There are two phases in the implementation of SPC. In phase I, conducted off-line, statistical models are built to characterize baseline data, which is collected when the process is manufacturing under successful working conditions. Statistics are chosen to measure the dissimilarity between observations and baseline models. Control limits for these statistics are calculated such that the statistics of observations from successful working conditions stay within their corresponding control limits with high probability (0.9 or 0.95, for example). Phase II occurs on-line and compares new observations to baseline models. If any of the statistics of new observations exceeds its corresponding control limits, we know that the new observations do not match closely enough with baseline models. We conclude that the process is out of

statistical control and is in need of adjustments or corrections, and that the output may be unacceptable.

Two types of variables are concerned in MSPC: process and product variables. Process variables characterize the condition of the manufacturing process. Product variables, which are measured much less frequently than process variables, describe the quality of products.

In some processes, the working condition is characterized by observations of multivariate variables. For example, in an industrial oven process, there are 14 thermocouples measuring temperatures at different locations inside the oven. The working condition of this oven at one time is described by the readings of these 14 thermocouples at that time. In some other processes, profiles are used to describe the working condition of process or the quality of product. Successful working conditions or good quality of product requires profiles to have desired patterns. A profile is usually defined as a set of responses as a function of one or more explanatory variables. Examples of profiles include the percent of a drug dissolved as a function of time, and the density of a wood product as a function of the depth into the plank.

In this dissertation, we investigate baseline data. Some baseline data consists of multivariate observations. The others consist of profiles.

This dissertation focuses on applying data mining technologies (especially clustering analysis) to analyze baseline data in phase I so that the SPC models can be improved. In the past, researchers were mainly interested in monitoring and diagnosing the mean shifts of process or product variables in phase II. The

performance of MSPC models in phase II, however, is determined mostly by whether the models built in phase I capture the nature of the underlying processes expressed through baseline data. So, it is necessary to study baseline data with more attention. Data mining technology should be useful for this purpose since it extracts structures from data as described in Bradley *et al*. (1999).

This dissertation addresses four separate but related questions to improve phase I analysis: (1) Are there multiple operational modes in baseline data? If so, how many? (2) Which periods in historical data collected over a long time period are the baseline? (3) If the baseline consists of profile data, are there outlier profiles? (4) When should the baseline model be updated? Before going into details of their backgrounds in the following sections, here we first introduce the importance of them briefly.

In the first question, the number of operational modes needs to be determined because a manufacturing process may have multiple operational modes, instead of only one as commonly assumed by previous works. An operational mode is a set of settings of process variables such that the product quality is consistently good. Building only one MSPC model for monitoring may have poor performance when multiple operational modes exist. In this dissertation, we present a new method to determine the number of operational modes.

In the second question, the baseline periods have to be extracted from historical data because the baseline data should only consist of observations when the process is manufacturing successfully. However, in practice, the baseline dataset is

rarely given but is selected from a historical dataset, which may consist of many observations in a long time period. In this long time period, the process may have intervals when it is unstable and may experience periods of both successful and unsuccessful production. In this dissertation, we present a new method to extract baseline periods in large historical data.

In the third question, outliers have to be identified before building MSPC models because the existence of outliers in baseline data may bias our MSPC model such that it has poor performance in online monitoring. Outliers are data points that are significantly different from the others. Outlier detection and elimination are two important steps in phase I analysis of baseline data. In this dissertation, we present a new method to detect outlier profiles.

In the fourth question, determining when we should update MSPC models is important because processes usually change from when the baseline data is collected after running for a while, through process improvements or changes of underlying structures. Thus baseline data needs to be updated. Correlation matrix characterizes relationships among variables. In this dissertation, we use a test of hypothesis to determine whether the correlation matrix changes from the baseline. If the changes are identified, the MSPC models may need updated.

1.1 Background of determining the number of operational modes

The first question we are concerned with in this dissertation is determining the number of operational modes in baseline MSPC data. This question was motivated by our experience in a food manufacturing company. We collected baseline data from a

successful food processing system and constructed an MSPC model to monitor the manufacturing process. To our surprise, the MSPC model gave many false alarms soon after it was implemented for on-line monitoring. But the product quality was within specifications.

We found that the alarms were caused by adjustments made by operators to ensure the consistency of product quality after a major switch in raw materials from winter to summer flour. The adjustments constituted a new operational mode. This experience illustrates that it is possible for a manufacturing process to have more than one operational mode, even the same product is manufactured.

There are some other descriptions of multiple operational modes in literature. Chu *et al.* (2004) identify three operational modes and one fault mode in baseline data from a rapid thermal annealing process where the three operational modes correspond to three products manufactured on the same process. Hwang and Han (1999) find eleven operational modes in a blast furnace operation.

There are two ways to handle the baseline dataset which is generated by multiple operational modes. The first is that we build a global MSPC model that encompasses all the modes, and use this global model to monitor the process. The second is to build a separate MSPC model for each operational mode, called a local model, and use this set of MSPC models to monitor the process.

We think the local model method has better performance in signaling an out-of-control process. The comparison of the global and local models is illustrated in Fig. 1, where we have only two variables. The crosses and circles are baseline data,

and the ellipsoids are the in-control areas of Hotelling's T^2 method. Appendix A describes Hotelling's T^2 method in details. Obviously there are two operational modes. If we build a global model with all baseline data, the model will fail to signal points in the space between these two modes, as point P1 in Fig. 1(a) falling inside the in-control area. The local model method avoids this problem, as in Fig. 1(b), where P1 lies outside the in-control areas of these two models, it is signaled by both models.

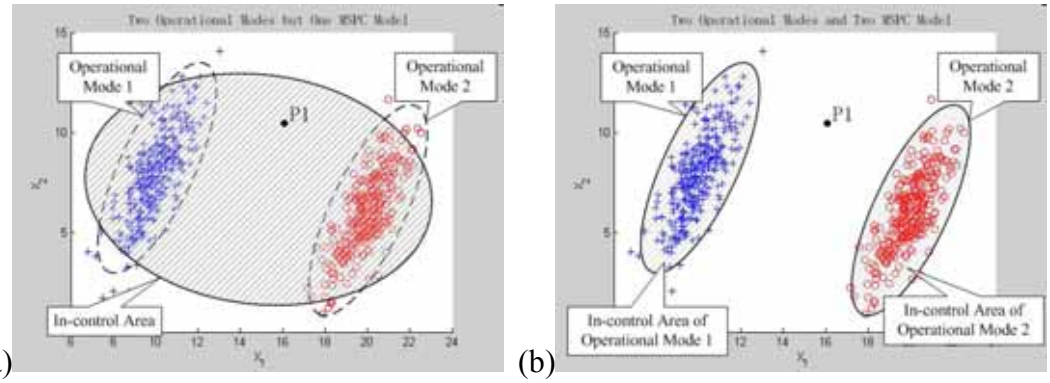


Figure 1. Comparison of (a) global and (b) local models

To build the local model, the number of operational modes in the baseline dataset needs to be determined first. In this dissertation, we assume that each data cluster represents an operational mode. Thus, determining the number of operational modes is equivalent to determining the number of clusters in a baseline MSPC dataset.

Data mining technology has been used widely to identify clusters in many fields including imaging and biology; see Fraley and Raftery (1998), Ertoz *et al.* (2003), Cinque *et al.* (2004) and Su and Liu (2005). Although in data mining, many algorithms exist to determine the number of clusters, each has a serious limitation when applied to identify the number of operational modes in the baseline process control data. So, a new method is proposed to determine the number of operational modes in the baseline MSPC dataset.

To address this question, this dissertation proposes a method that has the capabilities to detect the number of clusters in baseline MSPC data, namely: (1) The method correctly detects exactly one cluster if indeed there is only one. This is, of course, the most common outcome in baseline MSPC data; (2) If there is more than one cluster, it detects the correct number; (3) It does not require us to assume that all clusters are convex in advance; and (4) The number of clusters detected is not sensitive to arbitrarily chosen threshold values. The method we propose has all these properties in contrast to existing methods, each of which has some but not all of the capabilities.

In literature, there are three types of algorithms that determine the number of clusters: model-based, density-based, and scale-based methods. Model-based methods assume that each cluster has its own underlying multinormal distribution and consequently they do not work well on non-convex clusters, as shown in Fraley and Raftery (1998).

Density-based methods define clusters as regions in the data space where the objects are dense, and which are separated from one another by low-density regions; see Daszykowski *et al.* (2001). The number of clusters is the number of dense regions. Shared nearest neighbors, described in Ertoz *et al.* (2003), is one of the density-based methods. The problem with these methods is that the final answer is quite dependent on the threshold values used in algorithms and it is hard to select these values properly.

The scale-based methods overcome the threshold problem by computing the number of clusters over a range of the threshold value. The number of clusters is

selected as the one that persists for the largest range of the threshold value. These methods are very successful; see Kothari and Pitts (1999), Nakamura and Kehtarnavaz (1998), Herbin *et al.* (2001), Costa and Netto (1999) and Wang *et al.* (2004). However, scale-based methods are not capable of determining that there is exactly one cluster, the most common situation in baseline MSPC data.

The method we propose is a scale-based method that has been extended to enable it to identify exactly one cluster when that is indeed the case. The concept for our method, called scale-based with dummy dimension (SBDD), is to create a new augmented dataset which contains the original clusters, plus clones of those clusters, and an additional dimension. Thus, if the baseline dataset has 5 clusters in a three dimensional space, the augmented dataset has 10 clusters in the four-dimension space. Then we safely use a scale-based method to determine the number of clusters in the augmented dataset since the number of clusters is two or greater. To find the number of clusters in the original dataset, we simply divide by two.

To illustrate the proposed SBDD method and the existing methods we apply these to various simulated and actual datasets: a simulated multinormal dataset with one cluster; a dataset with three clusters, one of which is a non-convex cluster; a simulated industrial oven with two zones under engineering control with two operational modes; and a dataset from literature that contains the constituents of three related wine products.

The experimental results show that SBDD method gives the correct number of clusters for all four datasets, while the other methods do not. The model-based method

fails on the non-convex dataset and the wine dataset, which is a real dataset and we do not know whether it has convex or non-convex clusters. The results given by density-based methods depend heavily on the selection of parameters. The scale-based method gives right answers on three of the four datasets. But it gives higher than the actual number when there is only one cluster in the dataset.

In practice, since we do not know whether the clusters are convex or non-convex and whether we have only one or more clusters in advance, the SBDD method proposed here is clearly the safest choice to detect the number of clusters in a baseline dataset.

We do not study clustering errors in this dissertation. The steps in a cluster analysis are (1) find the number of clusters; (2) assign observations to clusters. Clustering error refers to assigning observations to clusters incorrectly. This dissertation focuses on step (1) only. None of the previous methods of determining the number of clusters uses clustering error to evaluate the performance.

1.2 Background of determining baseline periods in historical data

The second topic of this dissertation is determining the baseline periods in a historical dataset with a large number of observations collected in a long time period. We propose a method to automatically extract baseline periods from a large historical dataset of product variables. Baseline periods have a stable distribution of quality, and the most favorable quality. For example, the yield is consistently high or the mean of a particular product variable is close to target with small variation. Also, a baseline period should not include transient periods where the product quality is good for a

short while but the process is set unsuccessfully. For example, in an industrial steel melting tank, the output may still be good for a short while even when the settings of temperatures are poor. Usually engineering expertise can give the minimal length of a baseline period. Any period shorter than this given length is not included among baseline periods.

The motivation to address the problem of identifying baseline periods from a historical dataset collected in a long time period comes from our experience in a continuous process. Observations of process and product variables for approximately one year were collected. We wanted to find baseline periods by analyzing this huge amount of product observations in this long period so that the process and product data in these periods can be used to build SPC models. The yield rate in each batch is the product variable. Our question is: how can we select baseline periods from large amount of historical data of product variables?

In this dissertation, we propose a probability density profile (PDP) clustering method to address this problem. This method is robust to distributions generating the observations. It works in the following way. First, we use a moving window to segment the sequence of product variable into overlapping subsequences, which are in the end transformed into PDPs. Then, the number of PDP clusters is determined and each PDP is assigned to a cluster. We also assign a cluster label to each product observation, which derives clusters of points (product observations). The mean and standard deviation of the product variable in each point cluster are calculated. We select the clusters with the best statistics, such as the highest mean. The periods

spanned by the time stamps of the observations in the selected clusters and longer than a minimal length are baseline periods.

The ideas of segmenting sequence into subsequences, transforming them into PDPs and clustering PDPs are inspired by Guh (2005) and Han and Baker (1995). Guh (2005) uses an overlapping moving window with fixed number of observations to segment a sequence of historical data into subsequences. They are interested in the patterns of subsequences and a neural network was trained to classify these patterns. The trained neural network is applied online to recognize the abnormal patterns in the process. In Han and Baker (1995), in order to cluster protein sequences, they count the frequencies of the occurrence of 20 amino acids in certain positions of each protein sequence. The resulting sequence of frequency distribution is called a profile. Then, the protein sequences are clustered according to the distances in the profiles.

This inspired us that if a sequence contains observations from both successful and unsuccessful productions, the PDPs of subsequences from periods of successful and unsuccessful productions should have different patterns. We can cluster these PDPs to identify periods generated by different distributions.

To our best knowledge, there is no literature addressing how to determine baseline periods from a sequence of quality observations in SPC. As stated in Woodall (2000), “It is doubtlessly disturbing to many practitioners that researchers tend to neglect Phase 1 applications…., (which) cannot be easily placed into a general mathematical framework. Because of this fact, these important practical issues are rarely mentioned in the SPC research literature.” This may explain why such an

important issue in phase I, determining baseline periods, has not been addressed.

In literature, identification of change points in a sequence of observations is the only research work related to this topic. The change point is the last observation before the probability distribution of observations changes. Observations between two consecutive change points follow a single distribution consistently. One possible way of applying methods of change point identification to choose baseline periods is to find all the change points in the sequence of quality observations. Then, statistical tests can be applied to compare different periods and the ones with the most favorable statistics are selected as baseline periods.

Likelihood ratio test (LRT) is a popularly used method for change point identification; see Sullivan and Woodall (1996 and 2000), Hawkins and Zamba (2005), Son and Kim (2005), Herberts and Jensen (2004), Loschi and Cruz (2005), and Ramanayake and Gupta (2002). Sullivan (2002) proposes a method based on hierarchical clustering to identify change points.

The disadvantage of change point identification methods is that they rely heavily on the assumption of observation distributions. When the assumption is violated, which can happen in practice, they may have poor performance. For instance, Sullivan and Woodall (1996 and 2000), Sullivan (2002), Hawkins and Zamba (2005), and Son and Kim (2005) assume normal distributions; Herberts and Jensen (2004) and Loschi and Cruz (2005) assume Poisson distribution; and Ramanayake and Gupta (2002) assume exponential distributions. When the real distribution is different from the assumed ones, they may identify too many change points and segment the

sequence into many short periods. In the end, only a few short periods are selected as the most favorable ones. Consequently, only a small portion of baseline periods can be successfully identified. Chapter 3 shows how an LRT method based on the assumption of normal distribution performs poorly when observations are generated by lognormal or hyper-exponential distributions.

In this dissertation, we assume that there is only one product variable. However, this method can also be extended to cases with multiple product variables.

We apply the proposed PDP clustering method and the LRT method in Sullivan and Woodall (1996) on simulated datasets and a real dataset from a continuous process. The comparison of their performances on these two datasets shows that the PDP clustering method is more robust to distributions and gives more convincing baseline periods from the real dataset.

1.3 Background of detecting outlier profiles

Detecting outlier profiles in baseline data is my third concern in this dissertation. When the quality of processes or products is characterized by profiles, MSPC methods are devised for on-line monitoring in phase II and outlier detection in phase I.

From the definition of profiles (functions of explanatory variables), one may expect to see smooth curves or hyper-planes depicting the functions. However, if we loosen the definition of a function by letting it take any form (even not smooth), we have a more general definition of profiles.

With this more general definition of functions, profiles and high-dimension

vectors are interchangeable. A profile can be treated as a high dimensional vector if we take the index of each value of the explanatory variable as the index of each dimension. The value in that dimension is just the value of the response variable. Fig. 2(a) illustrates how a linear profile with two sample points is transformed into a vector in 2-D space. In Fig. 2(a), the linear profile has two samples when the explanatory variable equals x_1 and x_2, respectively. The response variable takes values y_1 and y_2 accordingly. This linear profile is transformed into a vector in 2-D space by letting y_1 and y_2 be the values in the first and second dimension respectively.

Reversely, a high-dimension vector can be illustrated as a profile, which takes the more general definition described above. To transform a high-dimension vector into a profile, we take the index of dimension as the explanatory variable. The value in each dimension is the value of the response variable. This is illustrated in Fig. 2(b). Albazzaz *et al*. (2005) discuss this in more details, where high dimensional vectors are transformed into profiles for visual interpretation.

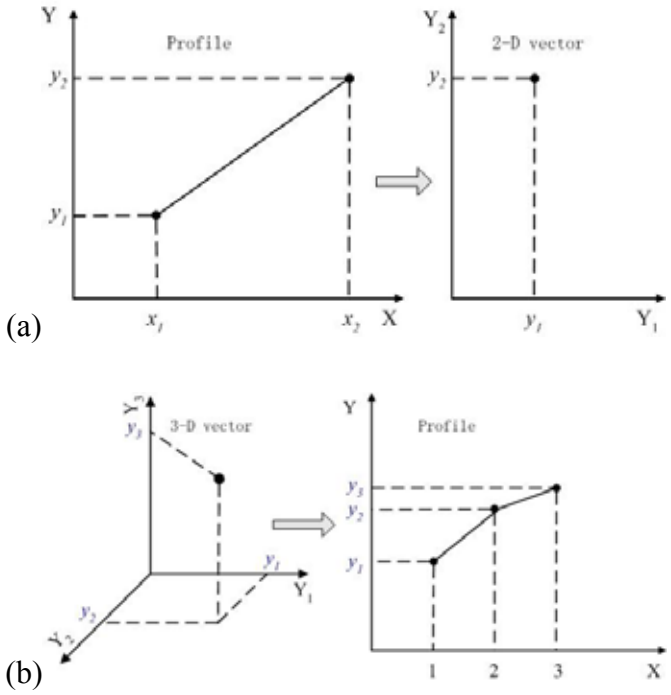


Figure 2. Transformation between profiles and vectors: (a) profiles to vectors; (b) vectors to profiles

We have experience in a real industrial oven process where engineers use profiles and vectors interchangeably. The engineers believed that the quality is determined by the temperature profile a product experiences in the oven. Here, the profile is just the temperature as a function of the locations of the 14 thermocouples. There is no explicit expression for this function. This profile is actually a 14-D vector.

Treating profiles as vectors is especially useful, or sometimes the only option when profiles are too complex. In this case, it is usually hard, if not impossible, to fit a regression model to express the relationship between the response and explanatory variables. In Chapter 4, we use the terms profiles and vectors interchangeably.

We treat profiles as vectors in high-dimension space, so χ^2 control chart can be applied to identify outliers in multivariate datasets. Here we assume that all profiles take fixed values of explanatory variables such that when we treat profiles as

vectors, all vectors are in the same space.

The application of the χ^2 control chart to identify outlier profiles is valuable in statistical process control in two ways: (1) It can be used to identify and remove outliers in the baseline data in phase I enabling the creation of a better model. (2) It can be used for on-line monitoring of processes in phase II by determining whether a newly observed profile is different from the baseline profile, i.e., out-of-control.

The χ^2 control chart method works as follows: Given a set of profiles, we treat it as a set of vectors in high-dimension space. A central vector is derived by finding the median in each dimension. The variance among profiles is estimated by considering the pair-wise differences between profiles. Then each profile is compared to the central vector. A χ^2 statistic is developed to measure their differences. If the χ^2 statistic exceeds a threshold value, it is labeled an outlier.

We assume that there is only one response variable and one explanatory variable. But the χ^2 control chart method can also be applied with one response variable and multiple explanatory variables.

One may think that we can apply methods of outlier detection from data mining area, such as the local outlier factor (LOF) method; see Breunig *et al* (2000). Usually these methods require the number of vectors to be large compared to the number of dimensions. It might not be satisfied in a profile baseline dataset such as the VDP data in Chapter 4 which has only 24 vectors in 314-dimension space.

In Chapter 4, we apply the χ^2 control chart method to simulated and real

data. The simulated profiles are generated from a highly nonlinear complex equation. The χ^2 control chart method is able to identify outliers that are generally too high or too low relative to the preponderance of the profiles. Also, it can identify an outlier that is near "the middle of the pack" but has the wrong shape.

When using simulated profiles, Type I and Type II errors are computed to measure the performance of the proposed χ^2 control chart method. It is compared with the existing methods based on these two errors. The Type I error is the percent of non-outlier profiles that are identified as outliers. The Type II error is the percent of outlier profiles that are identified as non-outliers. In contrast, Mahmoud and Woodall (2004) assess the performance of several methods to detect outliers for linear profiles by considering the probability of identifying at least one outlier, regardless of the number present.

We also apply the χ^2 control chart method to data that gives the density profile of a wood product as a function of the depth into the plank. This data was originally presented in Walker and Wright (2002) and is used in Williams *et al*. (2003) to test an outlier detection method based on non-linear regression. In contrast to the method in Williams *et al*. (2003), the χ^2 control chart method identifies outliers masked by other profiles but with the wrong shape. Also, the χ^2 control chart method does not require qualitative judgment to determine the outliers as in Williams *et al*. (2003).

There is a growing body of research about profiles. Regression-based methods fit an explicit model relating the response and explanatory variables and

focus on the coefficients of the model to determine outliers. Other methods, including the χ^2 control chart method and wavelet transformations, do not create an explicit function and can be used when the profiles are complex and regression would involve too many regression parameters. The power to detect outliers drops significantly when the number of parameters is large, as discussed in Jeong *et al*. (2006).

Focusing on linear profiles, Mahmoud and Woodall (2004) compare their outlier detection method to those proposed by Kang and Albin (2000), Stover and Brill (1998), and Kim *et al*. (2003). Kang and Albin (2000) simultaneously monitor the slope and intercept of a linear profile with a T^2 chart. Kim *et al*. (2003) remove the correlation between the slope and intercept by coding X values such that the mean is zero and separately monitor the slope, intercept and error variance. Mahmoud and Woodall (2004) create two multivariate linear models: one gives the response as a function of the explanatory variable and the other includes an indicator function for each profile as additional explanatory variables. They conclude there are no outliers if the two models are not statistically significantly different. Mahmoud and Woodall (2004) compare these methods for linear profiles on simulated data and conclude that their own method and the method in Kim *et al*. (2003) perform best.

Considering nonlinear profiles, Williams *et al*. (2003) detect outlier profiles by creating a nonlinear regression model and identifying outliers with four T^2 charts. Jin and Shi (2001) and Lada *et al*. (2002) use wavelet transformations for nonlinear profiles. They focus on a subset of coefficients chosen using engineering knowledge.

Jeong *et al*. (2006) also use wavelet transformations but they select the key coefficients with an adaptive procedure. Wavelet methods handle complex profiles well but can be somewhat difficult to interpret. Woodall *et al*. (2004) is a good reference for the research work of applying SPC on linear or nonlinear profiles.

All of the profile methods described, including the proposed χ^2 control chart method, assume that the dataset consists of realizations of one underlying profile plus some outlier profiles. In fact, it is possible that a dataset could contain two or more clusters of profiles as well as some outlier profiles. Data mining methods can be applied in these cases by treating profiles as vectors, such as methods to determine the number of clusters, e.g., the model-based method by Fraley and Raftery (1998), the density-based method in Ertoz *et al*. (2003), and the scale-based method in Zhang and Albin (2007) and Kothari and Pitts (1999). Then the profiles can be clustered.

1.4 Background of detecting the change of correlation matrix

The fourth question addressed in this dissertation is how we can determine whether the MSPC models need update after they are put into on-line monitoring. We answer this question through detecting whether the correlation matrix has changed. If so, the MSPC models may need update.

We describe a test of hypothesis to determine whether the current correlation matrix is significantly different from the baseline. If significant difference is detected, we provide a new method to diagnose which variables may change their mutual correlations. Thus, operators can be guided to check these variables to see whether errors happen among them. The scope of trouble-shooting is greatly narrowed. If no

process error is found, we should consider updating the MSPC baseline data with the latest observations.

In literature, there is no work for detecting correlation changes of multivariate processes. Except for detecting mean shifts, detecting variance changing also absorbs some interests; see Guo and Dooley (1992), Albin *et al.* (1997), Ho and Chang (1999), Acosta-Mejia and Pignatiello (2000), Montgomery (2001), and Yeh and Lin (2002). In literature regarding detecting changes of correlations, only the detection of autocorrelation coefficients in univariate autoregressive process is studied, as shown in Guo and Dooley (1995), Cook at al. (2001), and Hwarng (2004 and 2005).

This dissertation also discusses my future work, the application of MSPC and data mining technologies in a brain neuron system. In that system, we want to find the patterns of the activities of brain neurons when one is planning body movements. Thereafter we can predict from the activities of brain neurons what the brain plans the body parts to do, e.g., move the right arm to reach an object. This is very helpful for disabled people with artificial arms or legs. We can drive the corresponding artificial body parts to fulfill the task as planned by the brain based on the prediction results.

My expertise in MSPC and data mining applies in the following way for my future research. Data mining technologies (especially clustering analysis) can be used to segment neurons into clusters with different activities in brain planning. Different neurons are involved at different levels in planning different body movements. Clustering neurons helps to understand which neurons are active in planning a certain body movement, which in the end facilitates the body movement prediction. Multiple

neurons are involved in planning a certain body movement. Their signals form a multivariate dataset, which can be modeled by MSPC technologies.

The remainder of this dissertation is organized as follows. Chapter 2 describes the method of determining the number of operational modes in MSPC baseline data. The method to determine the baseline periods in a historical dataset is described in chapter 3. A method of detecting outlier profiles in baseline data is provided in chapter 4. The similarity test of correlation matrices and diagnosing methods are given in chapter 5. Chapter 6 concludes the research. Future work of applying MSPC and data mining technologies in the brain neuron system is described in Chapter 7.

2 Determining the number of operational modes in MSPC baseline datasets

In this chapter, a new method to determine the number of operational modes in MSPC baseline datasets is proposed. Section 2.1 describes the methods in data mining context to determine the number of clusters. Section 2.2 describes our proposed method. Section 2.3 compares the performances of the proposed method with the other three existing methods by applying them on four experimental datasets. The experimental results show that our proposed method works better than the existing methods.

2.1 Methods to determine the number of clusters

A difficult and unresolved aspect of clustering is determining the number of clusters, say K^*, in a dataset. There are three categories of methods to find K^*. This section reviews the model-based, density-based and scale-based methods.

Before describing methods to find K^*, we describe an important building block: the method of k-means which, given the number of clusters k, finds cluster centers and assigns points to clusters. It works as follows: Randomly select k points as cluster centers and assign each point in the dataset to the cluster with the nearest center. Calculate new cluster centers that are the averages of the assigned points. Reassign points to the nearest cluster center, calculate new centers, and so on until the centers converge, i.e., they are less than a threshold distance from centers in the previous iteration. The threshold distance can be any arbitrarily small number, such as 0.001 or less. The smaller this value, the longer the algorithm takes to converge. Note the result is not too sensitive to the selection of threshold distance. For different selections of

threshold distance, if they are all small enough, only a few (or even no) points change their memberships. For details of the *k*-means method, refer to Han and Kamber (2001).

2.1.1 Model-based methods

Model-based methods to find K^* assume each cluster is generated by its own multivariate normal distribution and work as follows. First we select K_{max}, an upper bound on possible value of K^* . For k=1, 2,…, K_{max}, use *k*-means (or any other algorithm which takes *k* as input parameter) to partition the data into *k* clusters and estimate the mean vector and the variance-covariance matrix for each cluster. Then, for each *k*, compute an adjusted loglikelihood value $l'(k)=l(k)-f(k)$, where $l(k)$ is a loglikelihood value and $f(k)$ is an overfitting penalty which is an increasing function of *k.* Select K^* to maximize $l'(k)$ as shown in the example in Fig. 3 where K^*=4. For details of model-based methods, please refer to Fraley and Raftery (1998).

The disadvantage of model-based methods is the assumption of mixture of multivariate normal distributions. It tends to incorrectly divide non-convex clusters into several convex clusters. In MSPC applications, in high dimensions, it is hard to know whether normality or even convexity is a safe assumption.

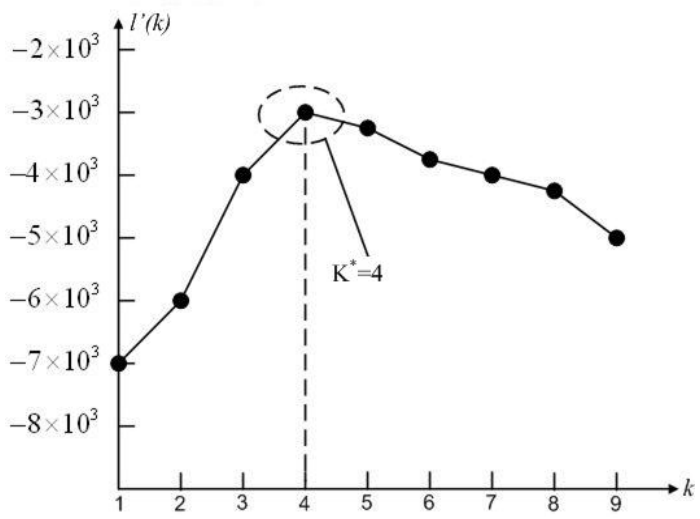


Figure 3. Model-based method for selecting the number of clusters; K*=4

2.1.2 Density-based methods

The second class of methods to find K^* is density-based. Shared nearest neighbors (SNN) is one density-based method; see Daszykowski *et al.* (2001). In SNN, there are three user-specified integers, k_0, k_1 and k_2, and $k_0>k_1$. For each point, k_0 nearest neighboring points are recorded. Then, if two points share at least k_1 nearest neighboring points, we say these two points are connected. If one point has at least k_2 points connected with it, this point is called a core point. Only core points are counted when we determine the number of clusters. The number of clusters is the number of disconnected groups of connected core points. Any two core points in the same group are connected by at least one path, but there is no path connecting any two core points from two different groups.

The major disadvantage of the density-based methods is that k_0, k_1 and k_2 are critical user-specified parameters in the algorithm. The final answer K^* is very sensitive to the values of these parameters. The selection of the parameter values is arbitrary or depends on the user's understanding of the dataset. Therefore, the number

of clusters given by the density-based method may be wrong because of the improper selection of the parameter values.

2.1.3 Scale-based methods

The third type of method is scale-based such as the one proposed by Kothari and Pitts (1999). There is one scale parameter λ_t in the algorithm that gives the minimum allowable distance between cluster centers. As λ_t increases in small increments, the number of clusters detected is monotone decreasing. The algorithm finds K_t, the number of clusters associated with each value of λ_t. A graph of K_t vs. λ_t is constructed and the number of clusters K^* is set equal to the K_t that corresponds to the largest horizontal range of λ_t. In Fig. 4, K_t=3 for λ_t from 0.35 to 0.6; so K^*=3. Herbin *et al.* (2001), Costa and Netto (1999) and Wang *et al.* (2004) provide other versions of scale-based methods.

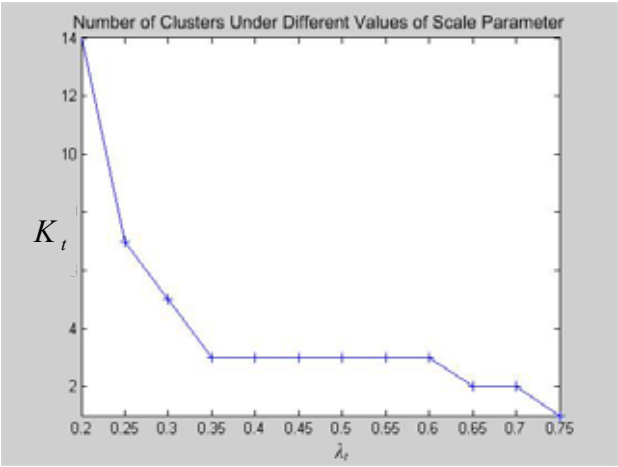


Figure 4. Scale-based method for selecting the number of clusters; K^*=3.

The scale-based method proposed by Kothari and Pitts (1999) can be simplified into the following steps:

Step 0: Standardize the dataset such that each variable has mean 0 and unit

variance, denoted as **X**. Initialize: t=0; K_0= K_{max}, a number greater than the maximal possible number of clusters; λ_0 and the scale increment $\Delta\lambda$.

Step 1: Cluster **X** into K_t clusters using the method of *k*-means (or any other clustering method which segments dataset into K_t clusters). Find K_t cluster centers.

Step 2: If distance between any two cluster centers is less than λ_t, combine into one cluster.

Step 3: Iterate: $t=t+1$; K_t=the number of remaining clusters; $\lambda_t = \lambda_{t-1} + \Delta\lambda$

Step 4: If $K_t = 1$, go to Step 5; otherwise, go to Step 1.

Step 5: Graph K_t vs. λ_t. K^* >1 corresponds to longest horizontal segment.

Scale-based methods are capable of dealing with non-convex clusters in contrast to model-based method. They are not overly sensitive to user selected threshold values in contrast to the density-based methods. However, there is a significant problem: the scale-based methods determine the number of clusters correctly only if there are two or more clusters in the data. The methods are not capable of concluding that the number of clusters is equal to one. This is quite important in MSPC, since one operational mode is the most common situation.

Scale-based methods cannot conclude that there is one cluster because the algorithm stops when K_t=1, as you can see in Step 4 and Fig. 6 above. Therefore the longest horizontal interval can never lead to the result K^*=1. If the algorithm continued, increasing the scale parameter to infinity, the number of clusters detected

remains 1. Therefore the longest horizontal interval would always lead to the result K^*=1. So, the algorithm has to stop when K_t=1. If there is only one cluster in the data set, the scale-based method always concludes that the number is greater than 1. We propose a modification to the scale-based method in Section 2.2 such that it can detect one cluster and can be applied in MSPC.

2.2 Scale-based with dummy dimension (SBDD) method to determine the number of clusters

This section proposes a scale-based method that is able to identify exactly one cluster or more than one cluster. Consider a matrix $\mathbf{X}_{n\times p}$ that contains baseline data where n is the number of observations and p is the number of variables. Also, assume there are K^* operational clusters in $\mathbf{X}$. We construct an augmented dataset $\mathbf{X^D}$ such that it has $2K^*$ clusters. We can safely apply the scale-based method to find the number of clusters in $\mathbf{X^D}$, and then we divide it in half to find the number of clusters in $\mathbf{X}$.

The matrix $\mathbf{X^D}$ consists of the points in $\mathbf{X}$ and a clone of those points and has an extra dummy dimension. Fig. 5 illustrates the concept behind $\mathbf{X^D}$. The stars in Fig. 4 are centers for clusters. Fig. 5(a) shows an original dataset $\mathbf{X}$ in one-dimension with 2 clusters, whose centers are labeled A and B respectively. Fig. 5(b) shows the augmented dataset $\mathbf{X^D}$ in two-dimensions with 4 clusters: the original 2 clusters (A and B) plus 2 clones (A' and B') a distance d =1 away.

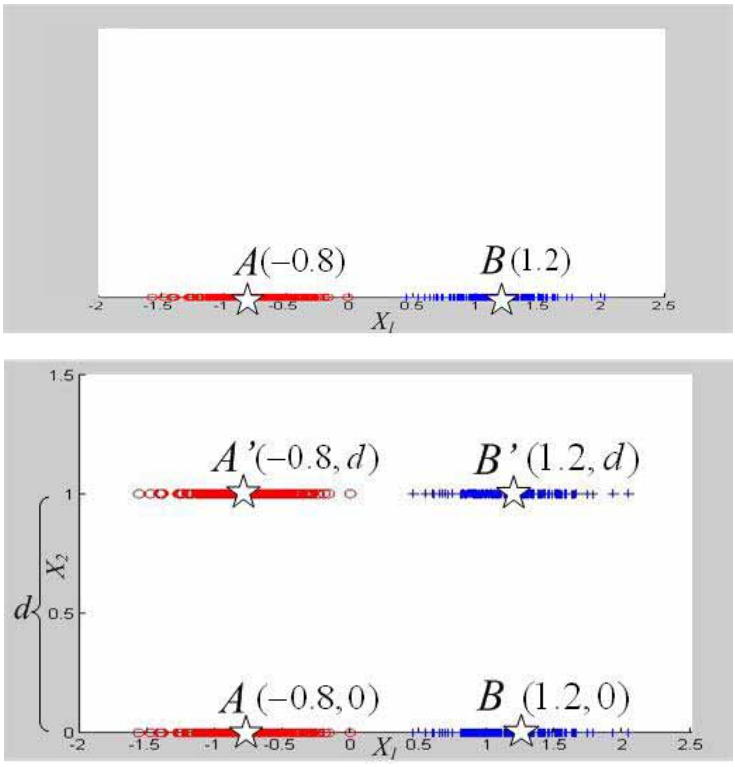


Figure 5. (a) Original data set **X** with 2 clusters in one dimension and (b) dataset $\mathbf{X^D}$ with 4 clusters in two-dimensions

It is important to notice in Fig. 5 that if *d*, the distance between the original and cloned clusters, is properly selected the centers of the clusters in $\mathbf{X^D}$ will follow a certain pattern. Clusters A and B in **X** have centers (-0.8) and (1.2) and clusters A, B, and their clones A' and B' in $\mathbf{X^D}$ have centers (-0.8, 0), (1.2, 0), (-0.8, *d*), (1.2, *d*), where *d*=1. Half the clusters centers, in the dummy dimension, are equal to 0 and half are equal to *d*. We will use this observation about the cluster centers when giving the details of the SBDD method.

2.2.1 Scaled-based with dummy dimension method

The steps of SBDD method are given below:

Step 0. Standardize the data such that each variable has mean zero and variance one to eliminate the effect of scales of different variables. Denote the

standardized matrix by $\mathbf{X}_{n\times p}$.

Step 1. Select a small initial value for *d* and for Δd, which is the increment for *d*. A recommended initial value for *d* is 1 or 2, for Δd is 0.5 or 1.

Step 2. Construct an augmented matrix with dummy dimension as follows:

$$\mathbf{X}^{D}_{(2n)\times(p+1)} = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{X} & \mathbf{D} \end{bmatrix} \tag{1}$$

where $\mathbf{0}$ is an $n\times 1$ vector with each element 0, and $\mathbf{D}$ is an $n\times 1$ vector with each element *d*.

Step 3. Apply the scale-based method described in Section 2.2 to determine $\boldsymbol{K}^{*}_{\boldsymbol{D}}$, the number of clusters in $\mathbf{X^D}$. Compute and record the cluster centers at each λ_t in the scale-based method.

Step 4. Check whether *d* was correctly chosen. The value *d* is correct if half the cluster centers in $\mathbf{X^D}$ have 0 in the dummy dimension and half have *d* in the dummy dimension; i.e., the cluster centers in $\mathbf{X^D}$ have the pattern

$$(\mathbf{X}^{c}_{1},0),(\mathbf{X}^{c}_{2},0),...,(\mathbf{X}^{c}_{M},0),(\mathbf{X}^{c}_{1},d),(\mathbf{X}^{c}_{2},d),...,(\mathbf{X}^{c}_{M},d) \tag{2}$$

where $\mathbf{X}^{c}_{i}, i=1,2,...,M$ are the unknown vectors with dimension $1\times p$ representing the cluster centers in dataset $\mathbf{X}$.

- If *d* is correct, then the number of clusters in $\mathbf{X}$ is $\boldsymbol{K}^{*}_{\boldsymbol{D}}/2$. End.
- If not, then increase *d* to *d*+ $\boldsymbol{\Delta d}$ and go to step 2.

2.2.2 More about selecting *d*

The distance *d* between the original data and the clone must be chosen

correctly. Revisiting Fig. 5, note that if *d* is too small relative to the distance between the cluster centers, the scale-based method will incorrectly detect only two clusters in $\mathbf{X}^{\mathbf{D}}$: one consisting of clusters A and A' with center (-0.8, *d*/2) and the other consisting of clusters B and B' with center (1.2, *d*/2). As you can see from this example, if *d* is too small we can easily detect it since the cluster centers do not follow the pattern in Eqn. (2).

If d is too large relative to the distance between the cluster centers in Fig. 5, the scale-based method will incorrectly detect only two clusters in $\mathbf{X}^{\mathbf{D}}$: one consisting of clusters A and B with center (0, 0) and the other consisting of cluster A' and B' with center (0, *d*). Here, the values of the first dimension of these two centers is 0, because **X** is standardized such that it has zero means and unit variance on each dimension of **X**. However the cluster centers appear to follow the pattern in Eqn. (2)! Since we can easily detect *d* too small but not one that is too large, the algorithm begins with a very small *d* and increases until the desired pattern for cluster centers is observed.

2.3 Experiments

In this section we present four datasets and use the methods described in Sections 2.1 and 2.2 to determine the number of clusters. The experimental results, as shown in Table 1, indicate that SBDD method successfully identifies the correct numbers of clusters in these four datasets. The other three methods only work on some of these four datasets.

		Method to Find Number of Clusters				
Dataset Name	true # clusters	Model Based	Density Based(SNN)		Scale Based	SBDD
			k0=30, k1=20,k2=10	k0=20, k1=10, k2=5		
Multinormal	1	1	2	1	2	1
Non-convex	3	6	9	2	3	3
Oven	2	2	11	2	2	2
Wine	3	2	1	1	3	3

Table 1. Number of clusters identified by four methods on four datasets

In these experiments, for each dataset, we apply the model-based method proposed by Fraley and Raftery (1998), the density-based method by Ertoz *et al.* (2003), the scale-based method by Kothari and Pitts (1999) and the proposed SBDD method on it. For the density-based method, to show the effects of the arbitrarily chosen parameters, we use two different sets of parameters. Then, the results given by these four methods are compared.

2.3.1 Multivariate normal data with one cluster

Consider a dataset generated by a multivariate normal distribution with five variables having the following mean vector and covariance matrix:

$$\vec{\mu} = [10,10,30,25,40]^T, \quad \Sigma = \begin{bmatrix} 4 & 2 & 1 & -2 & 0 \\ 2 & 3 & 2 & 0 & -1 \\ 1 & 2 & 5 & 3 & -2.2 \\ -2 & 0 & 3 & 4 & -2 \\ 0 & -1 & -2.2 & -2 & 3 \end{bmatrix}.$$

We generate 300 observations and since all the data is generated by the same distribution there is only one cluster. The covariance matrix is arbitrarily selected and has the features of a covariance matrix: (1) All diagonal values are positive; (2) Symmetric; (3) Positive definitive.

The first row of Table 1 shows the results when these four methods are applied

on the multinormal dataset. We can see that the SBDD method and the model-based method give the correct answer, and the scale-based method incorrectly selects two clusters. The density-based method gives the correct answer with one set of parameters but gives an incorrect answer with another set.

We generate another two datasets $\mathbf{X}_1$ and $\mathbf{X}_2$ with the same $\boldsymbol{\mu}$ but different $\boldsymbol{\Sigma}$'s, $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, respectively. The SBDD method is applied. These two $\boldsymbol{\Sigma}$'s are:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 6 & 2 & 1 & -2 & 0 \\ 2 & 7 & 2 & 0 & -1 \\ 1 & 2 & 5 & 3 & -2.2 \\ -2 & 0 & 3 & 4 & -2 \\ 0 & -1 & -2.2 & -2 & 3 \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_2 = \begin{bmatrix} 4 & 1 & -1 & -2 & 0 \\ 1 & 3 & 2 & 0 & -1 \\ -1 & 2 & 5 & 3 & -2.2 \\ -2 & 0 & 3 & 4 & -2 \\ 0 & -1 & -2.2 & -2 & 3 \end{bmatrix}.$$

The properly selected value of d for $\mathbf{X}_0$ differs from the one for $\mathbf{X}_1$, but is the same as the one for $\mathbf{X}_2$. However, the detected number of clusters remains the same.

2.3.2 Non-convex cluster

A dataset in two dimensions with three clusters, one of which is a concave, is shown in Fig. 6. This dataset can be accessed at the following address: http://www.stat.rutgers.edu/~jklin/567/hw3data1.txt.

The results of four methods on this dataset are given in the second row of Table 1. The scale-based method and the SBDD method both correctly identify three clusters. The model-based method incorrectly identifies six clusters. The density-based method also fails with either of the two sets of parameters.

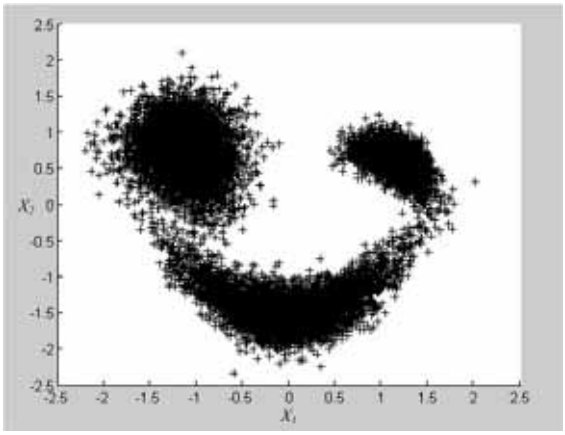


Figure 6. Scatter plot of a two-dimensional dataset with a concave cluster

2.3.3 Two-zone industrial oven simulation

The third dataset we consider comes from a simulation we developed in *Simulink* in *Matlab* of an industrial oven with two zones. As shown in Fig. 7 a conveyor passes through the two zones, each zone having a different target temperature. We simulate this system focusing on the temperatures in each zone.

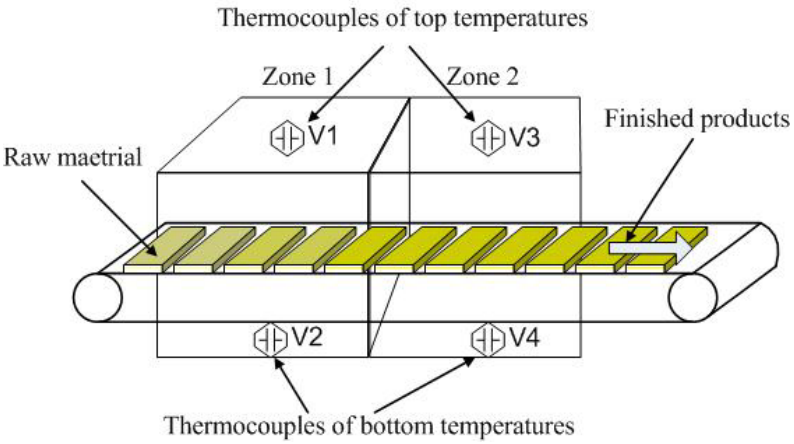


Figure 7. Two-zone industrial oven with PID controllers

We choose this simulated system as an example here because it captures the characteristics of modern manufacturing systems: (1) Controllers are widely used; (2) It is hard to tell whether the clusters formed by the observations of process variables are convex or not. Thus, if one method of determining the number of operational

clusters works in this simulated system, it is promising to work in a real manufacturing system.

The simulated baseline dataset consists of four variables corresponding to the four sensors, i.e., thermocouples, which record the temperatures in the bottom and top of each zone. Data is collected from the simulation after an initial start-up of 1000 seconds. Then data is recorded every 10 seconds from the four sensors for a total of 10,000 seconds. The sampling interval is chosen to avoid autocorrelations in the data.

A typical industrial oven has some important features that are captured in the simulation: (1) The oven has PID controllers in the first zone to insure the target temperatures are maintained. PID controllers are feedback controllers: if the sensor detects that the current temperature is not equal to the target, the level of heat is increased or decreased; (2) The temperature in one part of the oven affects the temperatures in the other parts. The bottom temperature of zone one affects the top temperature of zone one. Both top and bottom temperatures in zone one affect the top and bottom temperatures in zone two; (3) There are several sources of random noise that affect the temperatures in each part of the oven. The noise effect and the controllers are described in detail in the control diagram in Fig. 8.

The simulation is run to create two operational clusters representing two successful operational modes defined by target values for the first zone. For the first 4000 seconds the target values for the top and bottom temperatures at zone one are 300 and 350 degrees, respectively. Then the targets are adjusted to 310 and 360 degrees for the two temperatures. In a plant, such a shift could be caused by the transition from

warmer to colder ambient temperatures. For example, in food processing industry, the operator notes that the product is browning less, the target value is adjusted accordingly to insure that the final product has consistent characteristics.

In manufacturing practice, there may be log documents to record when and how the target values are adjusted. Small adjustments are made to insure the consistency of the product quality. Because of the existence of noise, adjusting target values may or may not lead to multiple clusters in data space. Only those adjustments large and persistent enough may cause multiple clusters. But it is still hard to tell what adjustment can be called “large enough”. So, even with log documents in hand, we still have to analyze the baseline dataset with cluster analysis method to determine the number of clusters.

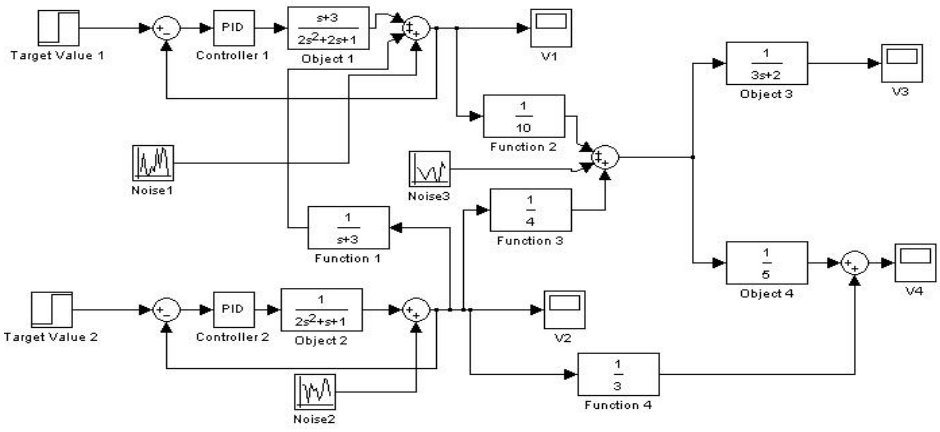


Figure 8. Circuit diagram of two-zone industrial oven with PID control and 4 thermocouples

We can see the results from the third row of Table 1 when we apply the four methods on the industrial oven dataset. Model-based method, scale-based method and SBDD method detect the number of clusters correctly, but density-based method succeeds only with one of the two sets of parameters. However, the convexity of the

data clusters can not be verified in the high-dimensional space. We also have no prior knowledge of the actual number of clusters in the dataset. So, we still recommend SBDD method in MSPC applications.

2.3.4 Wine

The last dataset we analyze has thirteen variables that characterize three wine products. The data is published by UCI Machine Learning Repository and can be accessed as follows: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine/.

Row 4 of Table 1 shows the results of the four methods on this wine data. The scale-based method and SBDD method successfully detect 3 clusters, while the other two do not.

3 Determining baseline periods in historical data

In this chapter, we describe the PDP clustering method to determine baseline periods from a historical dataset of product variables in a long time period. Section 3.1 describes the PDP clustering method in details. The LRT and PDP clustering methods are applied to simulated and real datasets and their performances are compared in Section 3.2.

3.1 PDP clustering method

In this section, we describe the four steps of the proposed PDP clustering method to determine the baseline periods from a sequence of product variable observations.

The sequence of the product variable is denoted as $\mathbf{Y}_N = \{y_1, y_2, ..., y_N\}$, where N is the number of observations in the sequence. The sampling instants of this sequence are denoted as $\mathbf{T}_N = \{t_1, t_2, ..., t_N\}$, where t_i is the sampling time of $y_i, i = 1, 2, ..., N$.

In the remainder of this chapter, without losing generality, we assume that the product with higher value of the product variable has better quality.

3.1.1 Segmenting sequence of product observations into subsequences and transforming subsequences into PDPs

In the proposed PDP clustering method, sequence $\mathbf{Y}_N$ is first segmented into subsequences by a moving window of size w. Instead of defining w as the number of observations in each window as in Guh (2005), more generally, we define w as the

time length covered by the window. In the i^{th} subsequence $\mathbf{S}_i$, y_i is the first observation. Then all the observations whose sampling times are between t_i and $t_i + w$ constitute subsequencie $\mathbf{S}_i$, i.e.,

$$\mathbf{S}_i = \{y_j \mid t_i \le t_j \le t_i + w\} \tag{3}$$

We also denote the observations in subsequence $\mathbf{S}_i$ as $\mathbf{S}_i = \{y_{i1}, y_{i2}, ..., y_{il}, ..., y_{iN_i}\}$, where N_i is the number of observations in subsequence $\mathbf{S}_i$. Fig. 9 illustrates how the subsequences are derived from $\mathbf{Y}_N$.

Two neighboring subsequences are heavily overlapped. For example, let us consider subsequences $\mathbf{S}_i$ and $\mathbf{S}_{i+1}$. From Eqn. (3), $\mathbf{S}_i = \{y_i, y_{i+1}, y_{i+2}..., y_{(i+N_i-1)}\}$, $\mathbf{S}_{i+1} = \{y_{i+1}, y_{i+2}, ..., y_{(i+N_{i+1})}\}$. These two subsequences have overlapped observations $\{y_{i+1}, y_{i+2}..., y_{\min\{i+N_i-1, i+N_{i+1})}\}$. This is illustrated by $\mathbf{S}_1$ and $\mathbf{S}_2$ in Fig. 9.

With this definition of *w*, the numbers of observations in subsequences are varying when the sampling frequency is not fixed, as shown in Fig. 8. Defining *w* as the number of observations in a moving window is just a special case when the sampling frequency is fixed.

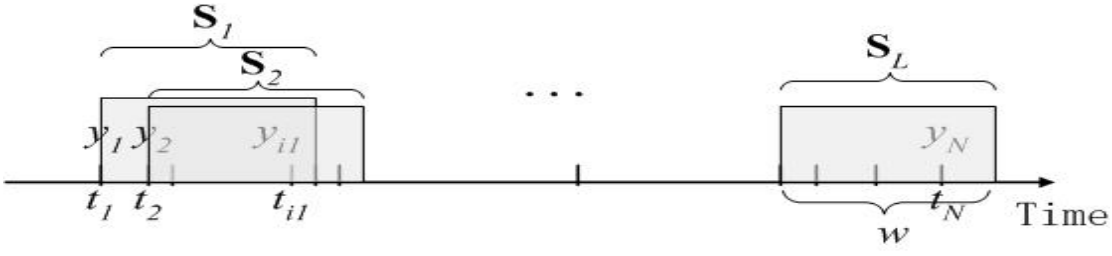


Figure 9. Segmenting sequence into subsequences using overlapping moving window

Since the PDP of a subsequence $\mathbf{S}_i$ is derived from its histogram, we must define the number of categories and their widths for the histograms. We call the

categories bins. To design a set of bins, we need to determine the number of bins and the width of each bin. As stated in Besterfield *et al*. (1999), the number of bins should be between 5 and 20. Broad guidelines are as follows: Use 5 to 9 bins when the number of observations is less than 100; Use 8 to 17 bins when the number of observations is between 100 and 500, and etc. The widths of bins are usually recommended uniform. However, sometimes if we want higher resolution in some certain range of the variable, we can make bins with smaller widths there.

The selection of *w* depends on the sampling frequency of the product variable and the design of bins. We need to choose a large enough *w* such that generally the number of observations in each subsequence is compatible with the number of bins. When the sampling interval of the product variable is not fixed, it may happen that the number of observations in a subsequence is small. PDPs transformed from these subsequences can not characterize the distribution of the observations accurately. Any subsequence whose number of observations is smaller than a certain number, $\underline{N}$, is neglected; otherwise it is reserved in the subsequence set.

A subsequence is then first transformed into a histogram with the set of bins, and the PDP is derived by dividing the histogram with the number of observations in the subsequence.

Suppose that we decide to have a set of *K* bins, whose boundaries consist of a set of *K-1* strictly monotonically increasing values, denoted as $\mathbf{B} = \{b_1, b_2, ..., b_{K-1}\}$. The histogram of subsequence $\mathbf{S}_i$ is just a set of frequencies of observations falling into the j^{th} bin, denoted by n_{ij}, where $n_{ij} = \sum_{l=i}^{i+N_i-1} I(b_{j-1} \le y_l < b_j)$, j=1, 2, …, K. Here,

$I(\cdot)$ is an identity function such that $I(X)=1$ if X is true, and 0 otherwise; $b_0=-\infty$ and $b_K=\infty$.

The PDP of the i^{th} subsequence $\mathbf{S}_i$ is just its histogram divided by the number of observations in $\mathbf{S}_i$. It can be considered as the normalization of histograms such that histograms of subsequences with different number of observations can be compared directly. We use a set of K variables $\mathbf{f}_i=\{f_{i1}, f_{i2}, ..., f_{iK}\}$ to denote the PDP of $\mathbf{S}_i$, where $f_{ij}=\frac{n_{ij}}{N_i}$, j=1, 2, …. K. Suppose that after the transformation, there are L PDPs reserved, the PDP dataset is:

$$\mathbf{F}=\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \vdots\\ \mathbf{f}_i\\ \vdots\\ \mathbf{f}_L\end{bmatrix}=\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1j} & \cdots & f_{1K}\\ f_{21} & f_{22} & \cdots & f_{2j} & \cdots & f_{2K}\\ \vdots & \vdots & & \vdots & & \vdots\\ f_{i1} & f_{i2} & \cdots & f_{ij} & \cdots & f_{iK}\\ \vdots & \vdots & & \vdots & & \vdots\\ f_{L1} & f_{L2} & \cdots & f_{Lj} & \cdots & f_{LK}\end{bmatrix} \quad (4)$$

If two subsequences are generated by significantly different distributions, their PDPs are two significantly different row vectors in Eqn. (4). It is demonstrated in Fig. 10 where the bars in dark and light colors represent the PDPs of two sequences generated by two different lognormal distributions, respectively.

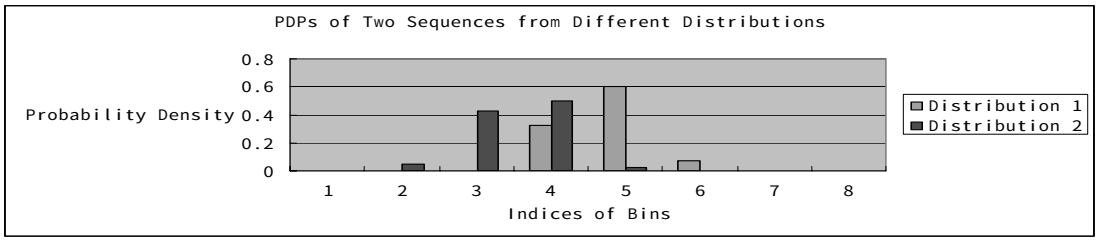


Figure 10. Difference of PDPs of different distributions

3.1.2 Determining the number of PDP clusters and clustering PDPs

Since significantly different distributions have significantly different PDPs, clustering PDPs helps to determine periods of different distributions in a data sequence. Before clustering PDPs, we need to determine the number of PDP clusters. In the proposed PDP clustering method, we take each PDP as a point in a high dimensional space. The PDP dataset **F** in Eqn. (4) is just a dataset of L points in K-dimension space.

There are several types of methods to determine the number of clusters in a dataset, such as the model-based method in Fraley and Raftery (1998), the density based method in Daszykowski *et al.* (2001), and the scale-based method in Zhang and Albin (2007), Kothari and Pitts (1999), Herbin *et al.* (2001), Costa and Netto (1999) and Wang *et al.* (2004). Among these methods, we recommend the scale-based method proposed by Zhang and Albin (2007) since their method can handle non-convex clusters, is not sensitive to user-specified parameters, and can give correct number of clusters even there is only one cluster in the dataset. The other methods do not have all these properties. While in the PDP dataset **F**, it may happen that there is only one PDP cluster since all observations are generated by a single distribution. It is also hard to determine the convexity of the PDP clusters in **F**.

After the determination of the number of PDP clusters, denoted as N^*, in dataset **F**, we need to cluster the PDPs into N^* clusters. In the proposed PDP clustering method, for simplicity, we use K-means to cluster PDPs. There are many other methods of clustering, such as K-median, hierarchical clustering, etc. Please refer to

Han and Kamber (2001) for details. Garcia-Escudero and Gordaliza (2005) claim that they are clustering curves, but essentially they take curves as points in high dimensional space. They use a trimmed *K*-means method to cluster the curves, which is robust to outlying curves. If one wants more robustness to outlying PDPs, the trimmed *K*-means can be applied.

After the PDP clustering by *K*-means, we get the cluster label of each PDP. We denote these cluster membership labels as $\mathbf{M} = \{M_1, M_2, ..., M_L\}$, where M_i=1, 2, …, or N^* .

3.1.3 Point clustering: determining the membership of each single observation

Clustering PDPs only gives the cluster membership of each PDP, but we do not have clear cut-off points segmenting periods with different distributions. For instance, PDPs $\mathbf{f}_1$ to $\mathbf{f}_{100}$ are labeled 1, PDPs $\mathbf{f}_{101}$ to $\mathbf{f}_{200}$ are labeled 2. If PDPs $\mathbf{f}_{100}$ and $\mathbf{f}_{101}$ are derived from subsequence $\mathbf{S}_{100} = \{y_{100}, y_{101}, ..., y_{145}\}$ and $\mathbf{S}_{101} = \{y_{101}, y_{102}, ..., y_{145}, ..., y_{150}\}$, respectively, we can not determine at which product observation the distribution changes. So, we need to determine the cluster membership label of each observation. If y_i and y_{i+1} have different cluster membership label, we say that the distribution till t_i and the distribution after t_{i+1} are different.

One possible way to determine the label of each observation is as follows. For observation y_i, we assign a set of N^* probabilities to it, denoted as $\mathbf{P}_i = \{p_{i1}, p_{i2}, ..., p_{iN^*}\}$, where p_{ij} is the percent of PDPs covering y_i whose membership labels are *j*. We use set $\mathbf{S}_i^C = \left\{\mathbf{S}_{i_1}, \mathbf{S}_{i_2}, ..., \mathbf{S}_{i_{N_i^C}}\right\}$ to denote the set of N_i^C

subsequences covering observation y_i. So, p_{ij} is calculated as:

$$p_{ij} = \frac{\sum_{k=1}^{N_i^C} I(M_{i_k} = j)}{N_i^C}, j = 1, 2, ..., N^* \quad (5).$$

Eqn. (5) is applied on all i=1, 2, …, N. Then, for point y_i, we assign a cluster label, denoted as L_i, to each point by choosing the label corresponding to the maximal probability, i.e., $L_i = \arg\max_j \{p_{ij}, j = 1, 2, ..., N^*\}$.

3.1.4 Calculating statistics of each point cluster and picking up the baseline periods

After the point clustering, the whole time period of the historical dataset is divided into periods with different cluster labels, as illustrated in Fig. 11. In Fig. 11, the whole time period is divided into four periods, two are labeled cluster 1, the remaining two are labeled cluster 2. We need to determine which point clusters have the best quality, e.g., the highest mean. Then, the periods corresponding to the best point clusters are selected as baseline periods.

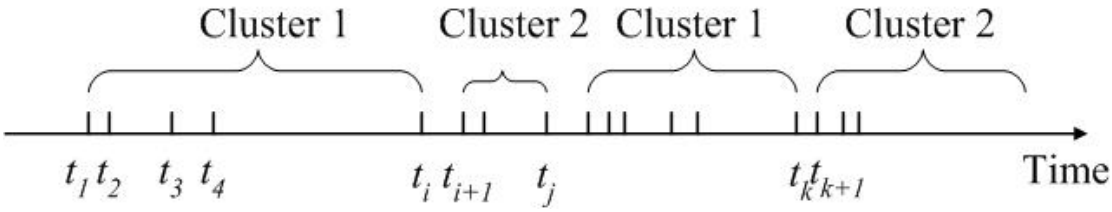


Figure 11. Periods with different cluster labels

Among the selected baseline periods, some may be shorter than the user-specified minimal length. For example, in Fig. 11, suppose cluster 2 has the best quality. The first period of cluster 2, from t_{i+1} to t_j, is too short. These clusters are disregarded, and the remaining selected periods are the baseline.

We determine which periods have the best quality by analyzing commonly used statistics of each point cluster, the mean and standard deviation. We denote the cluster with the highest mean as $C_{(1)}$, the mean of $C_{(1)}$ as $\hat{\mu}_{(1)}$, the standard deviation of $C_{(1)}$ as $\hat{\sigma}_{(1)}$, and the cluster index of $C_{(1)}$ as j^*. We also denote the means and standard deviations of point clusters as $\hat{\boldsymbol{\mu}} = [\hat{\mu}_1, \hat{\mu}_2, ..., \hat{\mu}_{N^*}]$ and $\hat{\boldsymbol{\sigma}} = [\hat{\sigma}_1, \hat{\sigma}_2, ..., \hat{\sigma}_{N^*}]$, respectively. The numbers of observations in these point clusters are $\mathbf{n} = [n_1, n_2, ..., n_{N^*}]$.

Except for only choosing $C_{(1)}$ as the point cluster with the best quality, we also choose any other point clusters whose mean is not significantly smaller and whose standard deviation is not significantly greater than $C_{(1)}$ as clusters with the best quality. So, we conduct tests of hypothesis to compare the other clusters with $C_{(1)}$. The two hypotheses are:

$$\begin{matrix} H_{01}: \mu_i = \mu_{(1)} & H_{02}: \sigma_i \le \sigma_{(1)} \\ H_{11}: \mu_i < \mu_{(1)} & H_{12}: \sigma_i > \sigma_{(1)} \end{matrix} \text{ and } i \ne j^*, i = 1, 2, ..., N^* \qquad (6)$$

If none of H_{01} and H_{02} is rejected, we say that cluster *i* is as good as $C_{(1)}$.

Usually we conduct *t-tests* and *F-tests* to compare means and variances of samples from two distributions, respectively. Readers are referred to Montgomery and Runger (2006) for details of these two tests. Although *t-test* and *F-test* assumes the normal distribution, which may be violated in practice, we still use it here to compare the means and standard deviations of two clusters roughly.

In order to reduce the overall type I error in the *t-tests* and *F-tests* in Eqns. (6), we can choose a small confidence level α, such as α=0.01 when determining the critical values of these two tests. So the overall type I error is still not too large.

3.2 Simulations and real data

In this section, we apply the LRT and PDP clustering methods on simulated and real datasets. Their performances are compared. We use type I and type II errors as the performance measurement. Here, type I error, denoted as β_1, is the percent of the successful production period identified as unsuccessful production. Similarly, type II error, denoted as β_2, is the percent of the unsuccessful production period identified as successful production incorrectly.

The simulation results show that the proposed PDP clustering method has similarly good performance with the LRT methods when the data is generated by normal distributions as they have similarly small type I and II errors. However, PDP clustering method performs better on lognormal and hyper-exponential distributions by having much smaller type I error and similarly small type II error than the LRT method. It shows that PDP clustering method is robust to distributions.

The real dataset is a sequence of a product variable from a continuous process. The proposed PDP clustering method segments the sequence into periods which coincide with the changes in the process variables. The period of successful production in the real dataset is mostly identified correctly as baseline periods by the PDP clustering method. Contrarily, the LRT method only extracts a small portion of successful production period as baseline. We think the PDP clustering method gives more reasonable baseline.

Before describing experimental results, we first briefly introduce the LRT method by Sullivan and Woodall (1996).

3.2.1 LRT method

The procedure of the LRT method is like this. For a given sequence of *N* observations $\mathbf{Y}_N$, the log-likelihood value of the whole sequence by assuming that all observations come from a single normal distribution is calculated, denoted as l_0. By assuming the position of a single change point, m_1, it divides the sequence into two subsequences, $\{y_1, y_2, ..., y_{m_1}\}$ and $\{y_{m_1+1}, y_{m_1+2}, ..., y_N\}$.The log-likelihood of each subsequence is calculated, denoted as l_1 and l_2, respectively, by assuming each subsequence follows a normal distribution. The normalized likelihood ratio is calculated from l_0, l_1 and l_2. The assumed change point m_1 changes from 2 to *N*-2, and the normalized likelihood ratio for each m_1 is calculated. If all normalized likelihood ratios are within a control limit 1, there is no change point; otherwise, the point with the largest normalized likelihood ratio is considered as the most significant change point and the sequence is divided into two subsequences at that point. The same procedure is repeated on each of these two subsequences, until no change point is identified.

For a given m_1, the log-likelihood values l_0, l_1 and l_2 are calculated as:

$l_0 = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\hat{\sigma}^2) - \frac{N}{2}$, $l_1 = -\frac{m_1}{2}\log(2\pi) - \frac{m_1}{2}\log(\hat{\sigma}_1^2) - \frac{m_1}{2}$, and

$l_2 = -\frac{N-m_1}{2}\log(2\pi) - \frac{N-m_1}{2}\log(\hat{\sigma}_2^2) - \frac{N-m_1}{2}$.

Here, $\hat{\sigma}^2$, $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ are maximum likelihood estimates of the variance of the whole sequence $\mathbf{Y}_N$, $\{y_1, y_2, ..., y_{m_1}\}$ and $\{y_{m_1+1}, y_{m_1+2}, ..., y_N\}$, respectively.

The normalized likelihood ratio, denoted as *Nlrt*(m_1 , *N*- m_1), is calculated as:

$Nlrt(m_1, N-m_1) = \frac{lrt(m_1, N-m_1)}{UCL \times E[lrt(m_1, N-m_1)]}$, where

$lrt(m_1, N-m_1) = -2(l_0 - (l_1 + l_2))$. The two terms in the denominator can be approximated by

$UCL = \frac{1}{1.7}\chi^2_{(1-p)^{1/k^*},k^*}$, and $E[lrt(m_1, N-m_1)] = 2\left[\frac{N-2}{(m_1-1)(N-m_1-1)}\right]$, where $\chi^2_{(1-p)^{1/k^*},k^*}$ is the $(1-p)^{1/k^*}$ percentile of the χ^2 distribution with degrees of freedom $k^* = -4.76 + 3.18\log(N)$, and p=0.05.

3.2.2 Simulations with different distributions

Each simulated dataset consists of 1000 observations of a single variable and 200 replicates. So, each dataset is a 1000×200 matrix. In each replicate, the first 400 observations are generated with distribution $f(\boldsymbol{\theta}_1)$, the remaining 600 observations are generated with distribution $f(\boldsymbol{\theta}_2)$. Fig. 12 illustrates the dataset.

$$\mathbf{D} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1,200} \\ y_{21} & y_{22} & \cdots & y_{2,200} \\ \vdots & \vdots & & \vdots \\ y_{400,1} & y_{400,2} & \cdots & y_{400,200} \\ y_{401,1} & y_{401,2} & \cdots & y_{401,200} \\ \vdots & \vdots & & \vdots \\ y_{1000,1} & y_{1000,2} & \cdots & y_{1000,200} \end{bmatrix} \begin{matrix} \} f(\boldsymbol{\theta}_1) \\ \\ \\ \\ \} f(\boldsymbol{\theta}_2) \\ \\ \\ \end{matrix}$$

Figure 12. Simulated dataset

When generating each dataset, we let $f(\boldsymbol{\theta}_1)$ has higher mean. If two periods have the same mean, then the period with the smaller variance has better quality.

For each dataset, we apply the LRT and the proposed PDP clustering methods on each replicate, β_1 and β_2 are calculated. The averages, $\overline{\beta}_1$ and $\overline{\beta}_2$ in all 200 replicates are recorded.

When applying PDP clustering method, we assume that the sampling interval is a constant. So defining window size *w* as the time length covered by the window is

equivalent to defining it as the number of observations in a moving window. We let w=40 observations. For each dataset, we plot the observations of the first replicate to have a view of the sequence and design the bins accordingly. In the simulation, we let the number of bins K=8. When transforming sequence into PDPs, we let $\underline{N}$ =20, i.e., each PDP is built from a subsequence with at least 20 observations.

3.2.2.1 Normal distribution

To simulate the dataset generated by normal distributions, we let

$$\begin{aligned} f(\boldsymbol{\theta}_1) &= N(\mu_1, \sigma_1^2) = N(0,1) \\ f(\boldsymbol{\theta}_2) &= N(\mu_2, \sigma_2^2) \end{aligned} \tag{7}$$

where μ_2 <0 and $\sigma_2 \geq 1$. We take factor pair [μ_2 , σ_2] to generate simulated 15 datasets, where $\mu_2 = [-1, -2, -3, -4, -5]$ and $\sigma_2 = [1, 2, 3]$.

Tables 2 and 3 give $\overline{\beta}_1$ and $\overline{\beta}_2$ of LRT and PDP clustering methods. Table 4 gives the boundaries of bins $\mathbf{B} = \{b_1, b_2, ..., b_7\}$ for PDP clustering method. By comparing Tables 2 and 3, we can see that LRT and PDP clustering method perform similarly well. Type I and II errors β_1 and β_2 in Tables 2 and 3 are both very close to 0. It means almost the whole baseline period is identified correctly, and almost no period of unsuccessful production is identified incorrectly as baseline. It is not a surprise for LRT methods since it is based on the assumption of normal distribution.

		μ_2				
		-1	-2	-3	-4	-5
σ_2	1	2.1 (0.2)	1.4 (0.1)	0.8 (0)	1.1 (0)	0.7 (0)
	2	1.8 (0.3)	1.0 (0.1)	0.9 (0.1)	1.2 (0)	0.8 (0)
	3	0.7 (0.3)	0.8 (0.1)	0.4 (0.1)	0.6 (0.1)	0.2 (0)

Table 2. $\overline{\beta}_1$ ($\overline{\beta}_2$) of LRT method on simulated datasets by normal distributions

		μ_2				
		-1	-2	-3	-4	-5
σ_2	1	0.9 (0.5)	0.4 (0.3)	0.2 (0.2)	0.2 (0.2)	0.2 (0.1)
	2	0.7 (0.7)	0.5 (0.3)	0.4 (0.2)	0.3 (0.2)	0.2 (0.2)
	3	0.5 (0.4)	0.5 (0.3)	0.3 (0.4)	0.3 (0.3)	0.3 (0.2)

Table 3. $\overline{\beta}_1$ ($\overline{\beta}_2$) of PDP clustering method on simulated datasets by normal distributions

$[\mu_2, \sigma_2]$	$\mathbf{B} = \{b_1, b_2, ..., b_7\}$
[1,1], [1,2], [1,3]	{-2, -1, -0.5, 0, 0.5, 1, 2}
[2,1], [2,2], [2,3]	{-4, -3, -2, -1, 0, 1, 2}
[3,1], [3,2], [3,3]	{-4, -3, -2, -1, -0.5, 0.5, 1}
[4,1], [4,2], [4,3]	{-5, -4, -3, -1, -0.5, 0.5, 1}
[5,1], [5,2], [5,3]	{-6, -5, -4, -1, -0.5, 0.5, 1}

Table 4. Boundaries of bins for normal distribution

3.2.2.2 Lognormal distributions

The lognormal distribution is a long-tail distribution with probability density function $f(y) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln y - \mu)}{2\sigma^2}\right]$, $0 < y < \infty$. The mean and variance of Y are $E(Y) = \mu_Y = e^{\mu+\sigma^2/2}$ and $Var(Y) = \sigma_Y^2 = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1)$.

In the simulations with lognormal distributions, we let $\boldsymbol{\theta}_1 = [\mu_{Y1}, \sigma_{Y1}] = [0.9, 0.3]$ and $\boldsymbol{\theta}_2 = [\mu_{Y2}, \sigma_{Y2}]$, where $\mu_{Y2} = [0.8, 0.7, 0.6, 0.5, 0.4]$, and $\sigma_{Y2} = [0.3, 0.5, 0.7]$. We take the 15 different factor pairs $[\mu_{Y2}, \sigma_{Y2}]$ to generate simulated datasets.

Fig. 13 shows the plot of 100 observations generated by a lognormal distribution, where the first 40 observations are generated with parameter $\boldsymbol{\theta}_1 = [0.9, 0.03]$, the remaining 60 observations are generated with parameter $\boldsymbol{\theta}_2 = [0.8, 0.03]$. We can see that visually it is almost impossible to find a proper

segmentation point to divide this sequence into two parts.

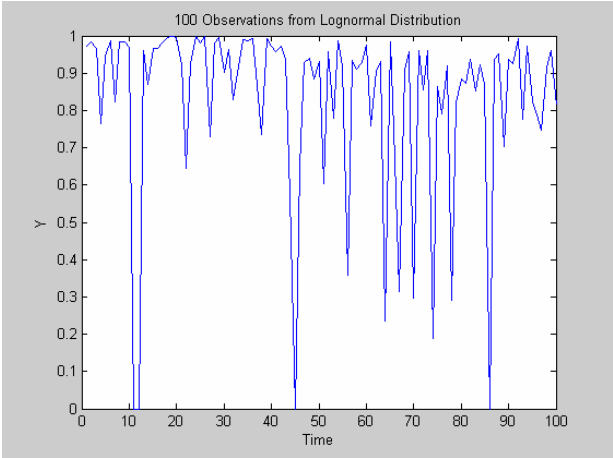


Figure 13. Plot of 100 observations from lognormal distribution

Tables 5 and 6 show $\overline{\beta}_1$ and $\overline{\beta}_2$ after we apply LRT and PDP clustering method on these 15 simulated datasets. PDP clustering method performs much better than LRT method by having much smaller $\overline{\beta}_1$. Tables 5 and 6 also show that the PDP clustering method has higher $\overline{\beta}_2$ than LRT method. However, since PDP clustering method has far smaller $\overline{\beta}_1$ than LRT method, we still conclude that generally PDP clustering method has much better performance than LRT method on lognormal distributions. Table 7 gives the boundaries of bins used in the PDP clustering method on these datasets.

		μ_{Y2}				
		0.8	0.7	0.6	0.5	0.4
σ_{Y2}	0.3	88.7(0.2)	87.7(0)	85.0(0)	89.5(0)	82.3(0)
	0.5	88.3(0.6)	89.2(0.1)	86.8(0)	84.8(0)	87.4(0)
	0.7	90.2(1.1)	89.0(0.1)	87.8(0)	85.1(0)	87.6(0)

Table 5. $\overline{\beta}_1$ ($\overline{\beta}_2$) of LRT on simulated datasets by lognormal distributions

		μ_{Y2}				
		0.8	0.7	0.6	0.5	0.4
σ_{Y2}	0.3	1.0(1.1)	0.5(0.3)	0.5(0.3)	0.4(0.2)	0.4(0.2)
	0.5	5.7(10.7)	0.7(0.5)	0.5(0.3)	0.5(0.3)	0.4(0.3)
	0.7	9.8(24.5)	1.1(1.4)	0.5(0.4)	0.5(0.3)	0.5(0.3)

Table 6. $\overline{\beta}_1$ ($\overline{\beta}_2$) of PDP clustering method on simulated datasets by lognormal distributions

$[\mu_{Y2}, \sigma_{Y2}]$	$\mathbf{B} = \{b_1, b_2, ..., b_7\}$
[0.8,0.3], [0.8,0.5], [0.8,0.7]	{0.75, 0.78, 0.8, 0.82, 0.89, 0.91, 0.93}
[0.7,0.3], [0.7,0.5], [0.7,0.7]	{0.67,0.70,0.73,0.82,0.89,0.91,0.94}
[0.6,0.3], [0.6,0.5], [0.6,0.7]	{0.57,0.60,0.63,0.82,0.89,0.91,0.94}
[0.5,0.3], [0.5,0.5], [0.5,0.7]	{0.47,0.50,0.53,0.72,0.89,0.91,0.94}
[0.4,0.3], [0.4,0.5], [0.4,0.7]	{0.37,0.40,0.43,0.55,0.87,0.90,0.93}

Table 7. Boundaries of bins for lognormal distribution

3.2.2.3 Hyper-exponential distributions

In our simulation with hyper-exponential distribution, we use the mixture of two exponential distributions to generate datasets, i.e., $f_X(x) = pf_1(x) + (1-p)f_2(x)$. Function $f_i(x) = \lambda_i e^{\lambda_i x}$, *i*=1, 2. Parameter λ_1=1 is fixed in all simulated datasets. To simulate datasets, we have two factors, λ_2 and p. They take values {2, 3, 4, 5, 6} and {0.9, 0.8, 0.7}, respectively. In each replicate of every dataset, the first 400 observations are generated with parameter $\boldsymbol{\theta}_1 = [\lambda_2, p]$, the remaining 600 observations are generated with $\boldsymbol{\theta}_2 = [\lambda_2, 1\text{-}p]$, i.e., their pdfs are $f_{\boldsymbol{\theta}_1}(x) = pf_1(x) + (1-p)f_2(x)$ and $f_{\boldsymbol{\theta}_2}(x) = (1-p)f_1(x) + pf_2(x)$, respectively.

Fig. 14 shows the plot of 100 observations generated by a hyper-exponential distribution, where [λ_2 , p]=[3, 0.8]. The first 40 observations are generated by $f_{\boldsymbol{\theta}_1}(x)$, the remaining 60 observations are by $f_{\boldsymbol{\theta}_2}(x)$. Like the lognormal distribution in Fig. 13, it is difficult to tell which periods are the baseline.

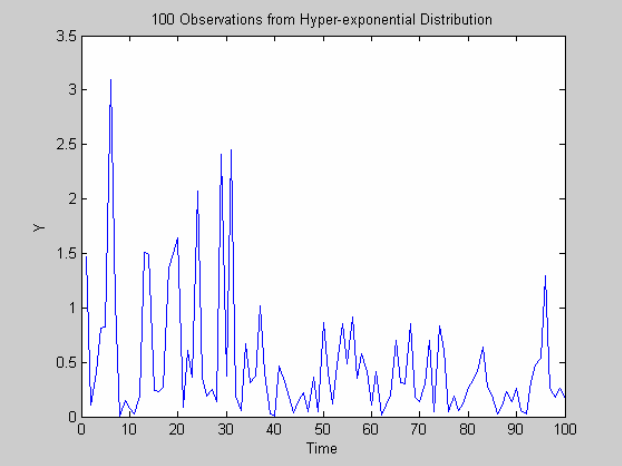


Figure 14. Plot of 100 observations from hyper-exponential distribution

Tables 8 and 9 give $\overline{\beta}_1$ and $\overline{\beta}_2$ when we apply the LRT and PDP clustering methods on these 15 datasets. PDP clustering method has much smaller $\overline{\beta}_1$ than and similar $\overline{\beta}_2$ as the LRT method in all circumstances. For a given value of λ_2, with the increase of p, both $\overline{\beta}_1$ and $\overline{\beta}_2$ increase because the difference between the means of $f_{\theta_1}(x)$ and $f_{\theta_2}(x)$ decreases. For instance, when [λ_2 , p]=[3, 0.9], $\mu_{\theta_1} = 0.93$, $\mu_{\theta_2} = 0.4$, $\mu_{\theta_1} - \mu_{\theta_2} = 0.53$; when [λ_2 , p]=[3, 0.8], $\mu_{\theta_1} = 0.87$, $\mu_{\theta_2} = 0.47$, $\mu_{\theta_1} - \mu_{\theta_2} = 0.4$. The boundaries of bins used by the PDP clustering method are $\mathbf{B} = \{b_1, b_2, ..., b_7\} = \{0.2,0.4,0.6,0.8,1,1.2,1.4\}$ in all 15 simulated datasets.

		λ_2				
		2	3	4	5	6
p	0.9	21.8(11.3)	29.0(7.1)	22.2(6.8)	27.6(6.4)	23.1(5.8)
	0.8	27.1(17.8)	25.4(17.0)	29.8(13.7)	33.9(13.9)	24.4(14.7)
	0.7	34.5(24.9)	37.3(29.7)	35.7(18.7)	40.6(22.9)	33.3(14.9)

Table 8. $\overline{\beta}_1$ ($\overline{\beta}_2$) of LRT on simulated datasets by hyper-exponential distributions

		λ_2				
		2	3	4	5	6
p	0.9	9.6(7.5)	5.1(8.6)	4.4(5.7)	3.1(7.0)	3.5(5.1)
	0.8	17.9(19.1)	12.1(13.7)	13.0(15.4)	11.8(13.1)	11.8(14.5)
	0.7	23.8(29.6)	28.6(29.9)	20.7(20.1)	27.8(26.3)	19.5(17.4)

Table 9. $\overline{\beta}_1$ ($\overline{\beta}_2$) of PDP clustering method on simulated datasets by hyper-exponential distributions

3.2.3 Real dataset

The real dataset comes from a continuous manufacturing process. The yield of each batch is recorded as the product variable. It is ranged between 0 and 1. Fig. 15 shows the real dataset with 483 batches. The sampling interval is 4 to 5 hours and averagely we have around 5 samples each day. From Fig. 15 we can see that the process experiences a period of unsuccessful production from t_1 to t_2. Some of the batches before t_1 have high percentages of defected products, so do batches after t_2.

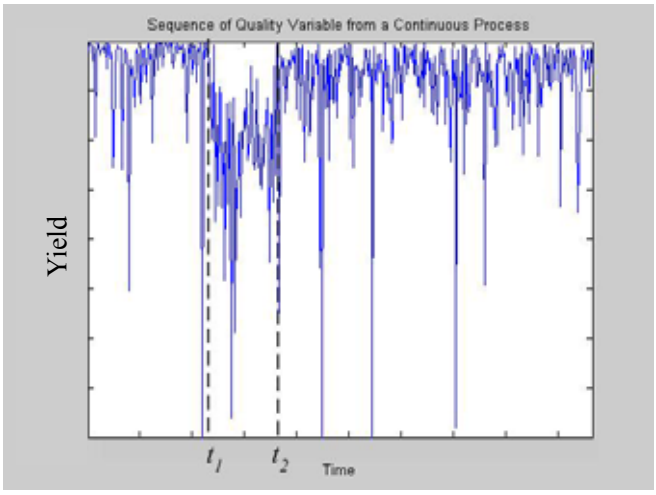


Figure 15. Plot of data from a real continuous process

We apply the PDP clustering method and the LRT method on this real dataset. When applying the PDP clustering method, we let the moving window size w=9 days. So, each moving window roughly has 40 to 50 observations. Any subsequence whose

number of observations is less than 20 is neglected, i.e., $\underline{N}$ =20. The boundaries of bins are **B**={0.79, 0.82, 0.85, 0.88, 0.91, 0.94, 0.97}. After the segmentation by PDP clustering or the LRT method, any period with less than 10 observations is disregarded.

The PDP clustering method segment the sequence into three parts, the start and end indices, the means and the standard deviations of these three segments are given in Table 10. The period between observations 1 and 118 is selected as the period of successful production because it has significantly smaller mean than others.

Period	Start Index	End Index	Mean	Standard Deviation
1	1	115	0.92	0.13
2	116	185	0.81	0.13
3	186	483	0.89	0.13

Table 10. Periods segmented by PDP clustering method and statistics

The selection of baseline periods is supported by the events happening in the process variables. A big change happened in the process variables after the 122nd batch. Thereafter, the process engineers adjusted the process variables by trials and the adjustment procedure stopped after the 178th batch. The process settings after the 178th batch are very stable, so are the settings before the 122nd batch. However, they are different.

So, it should be reasonable to segment the sequence into three periods: batches 1 to 122, batches 123 to 178 and batches 179 to 483. The average yield in the first period is the highest and should be considered as the baseline. Comparing with the period identified by the PDP clustering method, we can see that β_1 =1-115/122=5.7%, and β_2 =0.

Contrarily, the LRT method picks up periods of batches 13 to 41 and 233 to 245 as the ones representing the successful production. Thus, β_1=1-(41-13+1)/122 =76.2%, and β_2=(245-233+1)/(483-122)=3.5%. The majority of information about the successful production is lost. If we use the observations of the process variables in these two periods identified by LRT method to build MSPC models, they can not characterize the successful working conditions from batches 1 to 122 and are very likely to have poor performance in online monitoring.

3.2.4 Sensitivity of PDP clustering method to designations of bins

PDP clustering method needs a set of bins to transform subsequences into PDPs, as described in section 3.1.1. Now we show that the PDP clustering method is not sensitive to the designation of bins, as long as the bins are reasonable to the data.

We demonstrate this by applying the PDP clustering method on the datasets in section 3.2.2 with newly designed bins, whose boundaries are **B**={0.51, 0.58, 0.65, 0.72, 0.79, 0.86, 0.93} constantly. This set of bins is reasonable since the product variable is valued from 0 to 1. We want to focus more on the higher end of the value range, so we design bins with width 0.07 each from 0.51 to 1.0, and let [0, 0.51) to constitute a single bin.

Table 11 lists the $\overline{\beta}_1$ and $\overline{\beta}_2$ of the PDP clustering method. It shows that with these newly designed bins, PDP clustering method has similarly good performance (small $\overline{\beta}_1$ and $\overline{\beta}_2$) as when the boundaries of bins are designed as in Table 7.

		μ_{Y2}				
		0.8	0.7	0.6	0.5	0.4
σ_{Y2}	0.3	1.0(1.3)	0.4(0.3)	0.4(0.2)	0.3(0.2)	0.4(0.2)
	0.5	4.3(11.0)	0.7(0.6)	0.4(0.3)	0.4(0.3)	0.4(0.2)
	0.7	10.5(23.6)	1.0(1.3)	0.4(0.4)	0.5(0.3)	0.4(0.3)

Table 11. $\overline{\beta}_1$ ($\overline{\beta}_2$) of PDP clustering method on simulated datasets by lognormal distributions with newly-designed bins

4 Detecting outlier profiles

In this chapter, we apply χ^2 control chart to detect the outliers in profile baseline data. Section 4.1 describes the proposed χ^2 control chart method for detecting outliers in a profile dataset. The method is described in the context of statistical process control assuming that the dataset is baseline process control data. Section 4.2 shows simulated and real examples of the application of the χ^2 control chart method, and compares it with the nonlinear regression method by Williams *et al*. (2003). Section 4.3 discusses the robustness of the variance estimator and applies the method to on-line monitoring of profiles.

4.1 χ^2 control chart method to detect outlier profiles in baseline

The baseline profile dataset consists of *N* profiles. Denote the response variable by *Y* and the single explanatory variable by *X*. The explanatory variable takes a set of *M* fixed values $\{x_1, x_2, ..., x_M\}$. The i^{th} profile, i=1, 2,…, *N,* is a 1-by-*M* vector $\{y_{i1}, y_{i2}, ..., y_{iM}\}$ where y_{ij} is the response *Y* for the i^{th} profile when $X = x_j$.

Among the *N* profiles, there are *P* outlier profiles and *N*-*P* non-outlier profiles. Denote the set of outlier profiles by S_1 and non-outlier profiles by S_0. It is reasonable to assume that $P<N/2$.

Model the profiles in S_0 as follows:

$$y_{ij} = f_s(x_j) + \varepsilon_{ij},\ j = 1, 2, ..., M;\ i \in S_0 \tag{8}$$

As commonly assumed in literature, such as Kang and Albin (2000), Kim *et al*. (2003) and Williams *et al*. (2003), the noise terms ε_{ij}'s in Eqn. (8) are independent identically-distributed (*iid*) normal random variables with mean 0 and variance σ_s^2

for all *i* and *j*. (Section 4.1.4 extends the results to the case where the variances of ε_{ij} differ for different x_j's.) Function $f_s(\cdot)$ can be of any form, linear or nonlinear. Since the possible values of x_j are fixed for profiles in S_0, write $f_s(x_j) = y_j$.

Model the profiles in set S_1 as follows:

$$y_{kj} = f_k(x_j) + \varepsilon_{kj},\ j = 1, 2, ..., M;\ k \in S_1. \tag{9}$$

The model in Eqn. (9) for S_1 may differ from the model in Eqn. (8) for S_0 in two ways: (1) $f_k(\cdot) \neq f_s(\cdot)$ and/or (2) $\varepsilon_{kj} \sim N(0, \sigma_k^2), \sigma_k^2 > \sigma_s^2$. Any two different profiles *k* and *l* in set S_1 can be generated by the same or different underlying models, i.e., either $f_k(\cdot) \neq f_l(\cdot)$ or $f_k(\cdot) = f_l(\cdot)$ and either $\sigma_k^2 \neq \sigma_l^2$ or $\sigma_k^2 = \sigma_l^2$.

Among the *N* profiles, there are *NM* noise terms in Eqns. (8) and (9) that are assumed to be independent.

When $f_s(\cdot)$ and $f_k(\cdot)$ in Eqns. (8) and (9) are complex, it is difficult to fit explicit expressions for them. We take the *N* profiles as points in *M*-dimension space, among which *P* profiles are outliers. From Eqn. (8), profiles in S_0 can be considered *N-P* normally distributed points with mean vector $\boldsymbol{\mu}_s = [y_1, y_2, ..., y_M]^T$, and variance-covariance matrix $\boldsymbol{\Sigma}_s = \sigma_s^2 \mathbf{I}$, where $\mathbf{I}$ is an *M*-by-*M* identity matrix. Similarly, from Eqn. (9), profile *k* in S_1 can be considered a normally distributed point with mean vector $\boldsymbol{\mu}_k \neq \boldsymbol{\mu}_s$ and/or variance-covariance $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$, where $\sigma_k^2 > \sigma_s^2$.

In the remainder of this section, we first describe the χ^2 control chart to identify outliers. Then, we propose robust estimators of the mean vector $\boldsymbol{\mu}_s$ and the variance σ_s^2. The test statistic plotted on the χ^2 control chart is revised based on the estimators of $\boldsymbol{\mu}_s$ and σ_s^2, and its approximate distribution is derived. In the end of

this section, the situation when the variance differs at different x_j's is discussed.

4.1.1 χ^2 control chart

The χ^2 control chart works as follows. Suppose we know the mean vector $\boldsymbol{\mu}_s$ and the variance-covariance matrix $\boldsymbol{\Sigma}_s$ of the *M*-variate baseline normal distribution. In the case of identifying outlier profiles, $\boldsymbol{\mu}_s = [y_1, y_2, ..., y_M]^T$ and $\boldsymbol{\Sigma}_s = \sigma_s^2 \mathbf{I}$. Given profile *i* in baseline $\mathbf{y}_i = [y_{i1}, y_{i2}, ..., y_{iM}]^T$, we construct a statistic:

$$\Delta_i = [\mathbf{y}_i - \boldsymbol{\mu}_s]^T \boldsymbol{\Sigma}_s^{-1} [\mathbf{y}_i - \boldsymbol{\mu}_s] = \sum_{j=1}^{M} \frac{(y_{ij} - y_j)^2}{\sigma_s^2} \tag{10}$$

If profile *i* is non-outlier, Δ_i is a sample from a χ^2 distribution with *M* degrees of freedom. The upper control limit for the χ^2 control chart is $UCL = \chi^2_{\alpha,M}$, the upper 100α percentile of the χ^2 distribution with *M* degrees of freedom. The χ^2 control chart plots Δ_i against *i* for *i*=1, 2, …, *N*. Profile *i* with $\Delta_i > UCL$ is considered an outlier.

Mean vector $\boldsymbol{\mu}_s$ and σ_s^2 in Eqn. (10) are usually unknown and can only be estimated from the baseline data. The following subsection gives their robust estimators.

4.1.2 Estimating $\boldsymbol{\mu}_s$ and σ_s^2

Since profile baseline data may contain outliers, we need to derive estimators of $\boldsymbol{\mu}_s$ and σ_s^2 which are robust to the presence of outliers.

The estimator of the mean vector $\boldsymbol{\mu}_s$ is the median of the points in profile baseline data, denoted as $\hat{\boldsymbol{\mu}}_s$, i.e.,

$$\begin{aligned} &\hat{\boldsymbol{\mu}}_s = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_M] \\ &\hat{y}_j = \tilde{y}_{\cdot j} = median(y_{ij}, \quad i = 1, 2, ..., m); \; j = 1, ..., M \end{aligned} . \tag{11}$$

To estimate σ_s^2, we calculate the *N(N-1)/2* pair-wise differences between profiles *i* and *k* at each x_j:

$$\delta_{(i,k)j} = y_{ij} - y_{kj}, \quad i,k = 1,2,...,N, \quad i \neq k; \quad j = 1,2,...,M . \tag{12}$$

For each pair *i* and *k*, average the squared differences over the values of x_j to obtain *N(N-1)/2* estimates of σ_s^2, called pair-wise estimator:

$$\hat{\sigma}^2_{(i,k)s} = \frac{1}{2M}\sum_{j=1}^{M}\delta^2_{(i,k)j} \quad i,k = 1,2,...,N, \quad i \neq k . \tag{13}$$

(There is a 2 in the denominator since $V(\delta^2_{(i,k)j}) = 2\sigma_s^2$ in Eqn. (12).) Then estimate σ_s^2 with the median of the *N(N-1)/2* estimates in Eqn. (13) as follows:

$$\hat{\sigma}_s^2 = median(\hat{\sigma}^2_{(i,k)s}, i,k = 1,2,...,N; i < k) . \tag{14}$$

We approximate the statistical property of the estimator $\hat{\sigma}_s^2$ in Eqn. (14) by studying the statistical property of $\overline{\hat{\sigma}}_s^2 = \frac{1}{N(N-1)/2}\sum_{i<k}^{N}\hat{\sigma}^2_{(i,k)s}$ since mean has much better known statistical property than median. In appendix, we prove that $\overline{\hat{\sigma}}_s^2$ is an unbiased and asymptotically effective estimator.

The reason of using estimator $\hat{\sigma}_s^2$ in Eqn. (14) other than the regular sample variance $\hat{\sigma}^2_{Sample,s} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N-1}\sum_{i=1}^{N}(y_{ij} - \bar{y}_{\cdot j})^2$ is that $\hat{\sigma}_s^2$ is more robust to outliers than $\hat{\sigma}^2_{Sample,s}$, which is equivalent to $\overline{\hat{\sigma}}_s^2$. It is demonstrated by simulations in section 4.2.

4.1.3 Revising test statistic Δ_i

After estimating $\hat{\boldsymbol{\mu}}_s$ and $\hat{\sigma}_s^2$, we need to revise the test statistic Δ_i in Eqn. (8) since it assumes that $\boldsymbol{\mu}_s$ and σ_s^2 are known. When we use the estimators in Eqns. (8) and (14), the test statistic should be:

$$\Delta_i^{'} = [\mathbf{y}_i - \hat{\boldsymbol{\mu}}_s]^T \hat{\boldsymbol{\Sigma}}_s^{-1} [\mathbf{y}_i - \hat{\boldsymbol{\mu}}_s] = \sum_{j=1}^{M} \frac{(y_{ij} - \hat{y}_j)^2}{\frac{N-1}{N}\hat{\sigma}_s^2} \tag{15}$$

Profile *i* is considered an outlier if $\Delta_i^{'} > UCL$, where $UCL = \chi^2_{\alpha,M}$.

Now, we prove that statistic $\Delta_i^{'}$ is a sample from a random variable of an approximate χ^2 distribution with *M* degrees of freedom if profile *i* is non-outlier.

To simplify the proof, we first assume that there is no outlier in baseline and the variance σ_s^2 is known. As when we approximate the statistical property of $\hat{\sigma}_s^2$ in Eqn. (14), we use the average, instead of median in Eqn. (11) to estimate the mean vector and study the approximate statistical property of $\Delta_i^{'}$ in Eqn. (15).

The average vector of the *N* baseline profiles is:

$$\hat{\boldsymbol{\mu}}_s = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_M], \text{ where } \hat{y}_j = \bar{y}_{.j} \tag{16}$$

The difference between profile *i* and $\hat{\boldsymbol{\mu}}_s$ is $[y_{i1} - \hat{y}_1, ..., y_{ij} - \hat{y}_j, ..., y_{iM} - \hat{y}_M]$. From the definition of non-outlier profiles in Eqn. (8), we can prove that $y_{ij} - \bar{y}_{.j}$ is a sample from a normal distribution with mean 0 and variance $\frac{N-1}{N}\sigma_s^2$. So, $\sum_{j=1}^{M} \frac{(y_{ij} - \hat{y}_j)^2}{\frac{N-1}{N}\sigma_s^2}$ is a sample from a χ^2 distribution with *M* degrees of freedom. We release the assumptions and substitute the estimators of the mean vector and variance with their robust estimators in section 4.1.2. Then $\Delta_i^{'}$ is a sample from an approximate χ^2 distribution.

Readers may think that it is more proper to use Hotelling's T^2 control chart than to use the χ^2 control chart when the mean vector and the variance-covariance matrix are estimated from the sample and the sample size is small. In Hotelling's T^2

control chart, the *UCL* of the test statistic is $\frac{(N-1)^2}{N}\beta_{\alpha,M/2,(N-M-1)/2}$, where $\beta_{\alpha,M/2,(N-M-1)/2}$ is the upper 100α percentile of a beta distribution with parameters *M/2* and *N-M-1*; see Montgomery (2001) for details. The *UCL* of Hotelling's T^2 control chart requires the number of points (*N*) is larger than their dimension (*M*). This condition is sometimes not satisfied when we treat profiles as points. Profiles may have a huge number of fixed values of the explanatory variable but we only have a few profiles. Therefore we recommend using χ^2 control chart instead.

4.1.4 When variance of noise differs at different X_j's

In the previous subsections, we assume $\varepsilon_{ij} \sim iid\ N(0, \sigma_s^2)$. Now we consider the case where the variance of ε_{ij} differs at different *j*=1, 2,…, *M* and $\varepsilon_{ij} \sim iid\ N(0, \sigma_{sj}^2)$.

We revise Eqn. (15) as

$$\Delta_i' = \sum_{j=1}^{M} \frac{(y_{ij} - \hat{y}_j)^2}{\frac{N-1}{N}\hat{\sigma}_{sj}^2} \tag{17}$$

where $\hat{\sigma}_{sj}^2$ can be the regular sample variance, i.e., $\hat{\sigma}_{sj}^2 = \frac{\sum_{i=1}^{N}(y_{ij} - \bar{y}_{\cdot j})^2}{N-1}$. If one wants higher robustness to outliers, mean absolute deviation (MAD) estimator can be used to estimate σ_{sj}, which is $\hat{\sigma}_{MAD,j} = 1.25 \times \frac{1}{N}\sum_{i=1}^{N}\left|y_{ij} - \bar{y}_{\cdot j}\right|$ when $X = x_j$. Readers are referred to Montgomery *et al.* (1990) for more details of MAD estimators.

4.2 Examples

In this section, we study the performance of the χ^2 control chart method in

detecting outlier profiles in phase I by applying it to two nonlinear profile datasets: one simulated and one real. Its performance on these two datasets is compared with the nonlinear regression method by Williams *et al*. (2003). To determine the *UCL*s of the test statistics in these two methods, we choose $\alpha = 0.05$. Simulation results show that the χ^2 control chart method has better performance than the nonlinear regression method.

4.2.1 Nonlinear regression method

In this subsection, we briefly introduce the nonlinear regression method by Williams *et al.* (2003). For each profile, four T^2 statistics of the regression coefficients and the mean squared error (*MSE*) of the regression model are calculated. If any of these five statistics exceeds its control limit, that profile is identified as an outlier.

Suppose for profile *i*, the least square estimates of regression coefficients are $\hat{\boldsymbol{\beta}}_i = [\hat{\beta}_{i0}, \hat{\beta}_{i1}, ..., \hat{\beta}_{ik}]^T$, $MSE_i = \sum_{j=1}^{M} (y_{ij} - \hat{y}_{ij})^2 / (M - p)$, where $L=k+1$; $i=1, 2, \ldots, N$. Here, $\hat{y}_{ij} = f(x_j; \hat{\boldsymbol{\beta}}_i)$, where $f(\cdot)$ is the function we use to fit the profile, e.g., $f(x; \boldsymbol{\beta}) = \sum_{l=0}^{k} \beta_l x^l$.

The first Hotelling's T^2 statistic is calculated by

$$T_{1,i}^2 = (\hat{\boldsymbol{\beta}}_i - \bar{\hat{\boldsymbol{\beta}}})^T \mathbf{S}_1^{-1} (\hat{\boldsymbol{\beta}}_i - \bar{\hat{\boldsymbol{\beta}}}) \tag{18}$$

where $\bar{\hat{\boldsymbol{\beta}}} = \frac{1}{N} \sum_{i=1}^{N} \hat{\boldsymbol{\beta}}_i$ and $\mathbf{S}_1 = \frac{1}{N-1} \sum_{i=1}^{N} (\hat{\boldsymbol{\beta}}_i - \bar{\hat{\boldsymbol{\beta}}})(\hat{\boldsymbol{\beta}}_i - \bar{\hat{\boldsymbol{\beta}}})^T$. Similarly, the second T^2 statistics $T_{2,i}^2$ is calculated by substituting $\mathbf{S}_1$ in Eqn. (18) with $\mathbf{S}_2 = \frac{\hat{\mathbf{V}}\hat{\mathbf{V}}'}{2(N-1)}$, where $\hat{\mathbf{V}} = [\hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3, ..., \hat{\mathbf{v}}_N]$ and $\hat{\mathbf{v}}_{i+1} = \hat{\boldsymbol{\beta}}_{i+1} - \hat{\boldsymbol{\beta}}_i$, $i=1, 2, \ldots, N-1$. Substituting $\mathbf{S}_1$ in

Eqn. (20) with $\mathbf{S}_3 = \frac{1}{N}\sum_{i=1}^{N} MSE_i \times (\hat{\mathbf{D}}_i'\hat{\mathbf{D}}_i)^{-1}$, where

$$\mathbf{D}_i = \frac{\partial f(\mathbf{X}_i, \boldsymbol{\beta}_i)}{\partial \boldsymbol{\beta}_i} = \begin{bmatrix} \frac{\partial f(x_{i1}, \boldsymbol{\beta}_i)}{\partial \beta_{i0}} & \frac{\partial f(x_{i1}, \boldsymbol{\beta}_i)}{\partial \beta_{i1}} & \cdots & \frac{\partial f(x_{i1}, \boldsymbol{\beta}_i)}{\partial \beta_{ik}} \\ \frac{\partial f(x_{i2}, \boldsymbol{\beta}_i)}{\partial \beta_{i0}} & \frac{\partial f(x_{i2}, \boldsymbol{\beta}_i)}{\partial \beta_{i1}} & \cdots & \frac{\partial f(x_{i2}, \boldsymbol{\beta}_i)}{\partial \beta_{ik}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(x_{iM}, \boldsymbol{\beta}_i)}{\partial \beta_{i0}} & \frac{\partial f(x_{iM}, \boldsymbol{\beta}_i)}{\partial \beta_{i1}} & \cdots & \frac{\partial f(x_{iM}, \boldsymbol{\beta}_i)}{\partial \beta_{ik}} \end{bmatrix},$$

we get $T^2_{3,i}$

The control limit for $T^2_{1,i}$ is $UCL_1 = \frac{(N-1)^2}{N} B_{\alpha, L/2, (N-L-1)/2}$, where $B_{\alpha, L/2, (N-L-1)/2}$ is the $100\,\alpha$ upper percentile of a beta distribution with shape parameters *L*/2 and (*N*-*L*-1)/2. Control limit $UCL_2 = \frac{(f-1)^2}{f} B_{\alpha, L/2, (f-L-1)/2}$, where $f = \frac{2(N-1)^2}{3N-4}$, and $UCL_3 = \chi^2_{\alpha, L}$. The upper and lower control limits of *MSE* are $\overline{S^2} \pm h_{\alpha, N, \infty}\hat{\sigma}$, where *h* is a critical value given in Nelson (1983), $\overline{S^2} = \frac{1}{N}\sum_{i=1}^{N} MSE_i$, and $\hat{\sigma} = \overline{S^2}\sqrt{2(N-1)/(N(M-L))}$.

The exact distribution of the fourth T^2 statistic is unknown, so we skip it when we apply the nonlinear regression method on simulated data.

4.2.2 Nonlinear profiles: simulated datasets

In the simulated dataset, we use Type I and Type II detection errors to assess the proposed χ^2 control chart method and compare it with the nonlinear regression method. Suppose among the *N* profiles in a dataset, there are *P* outliers. If, among the *N*-*P* non-outlier profiles, P_1 profiles are incorrectly identified as outlier profiles, and among the *P* outlier profiles, P_2 profiles are correctly identified, then

Type I Error $= \frac{100P_1}{N-P}$ and Type II Error $= \frac{100(P-P_2)}{P}$.

We generate profile datasets, each with N=200 profiles, that consist of 200-P non-outlier and P outlier profiles where P takes values 20, 40, 60 or 80. For each profile there are M=100 values of X; i.e. X=0.08, 0.16, …, 8.

The profiles are generated as follows:

$$y_{ij} = f_a(x) + \varepsilon_{ij},$$
$$f_a(x) = 10 - 20ae^{-ax_j}\sin(\sqrt{4-a^2}\,x_j)/\sqrt{4-a^2} + 10e^{-ax_j}\cos(\sqrt{4-a^2}\,x_j) \quad (19)$$
$$\text{and } \varepsilon_{ij} \sim N(0,\sigma^2)$$

The non-outlier profiles have a=0.5 and $\sigma = 1$. Fig. 16 shows $f_{0.5}(x)$ and $f_{1.1}(x)$. Regression models could not easily model these profiles.

The nonlinear regression model we choose to fit the profiles is a multinomial function $f(x;\boldsymbol{\beta}) = \sum_{l=0}^{5} \beta_l x^l$. Here, order 5 is chosen because in Fig. 14 there are four obvious points where $f_a'(x) = 0$. So, $f_{0.5}'(x) = \prod_{i=1}^{4}(x - x_{i0}) = b_1 + b_2x + b_3x^2 + b_4x^3 + b_5x^4$. We also conduct likelihood ratio test to compare models of order 5 and order 6; see Rawlings *et al.* (1998). It shows that the model of order 6 is not significantly better than order 5.

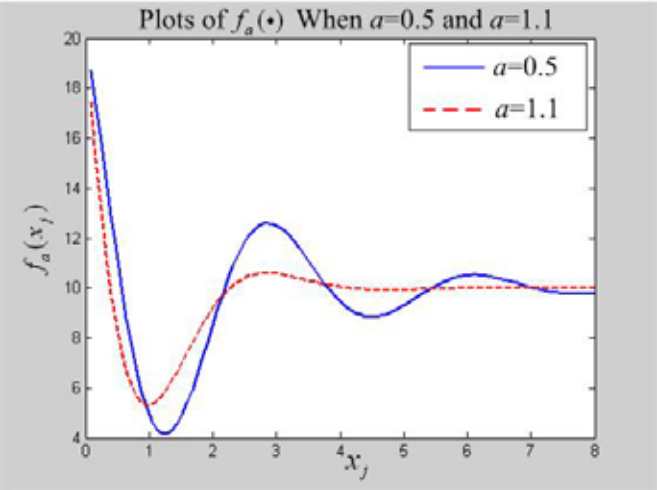


Figure 16. Plots of $f_{0.5}(x)$ and $f_{1.1}(x)$

We perform two simulation experiments. In the first experiment, the factors are P, the number of outliers, and a, the function parameter, with levels a=0.5, 0.7, …, 1.9. In the second simulation experiment, the factors are P and σ, the standard deviation of the noise term, with levels 1.2, 1.4, …, 3.0. For each (P, a) or (P, σ) combination, we generate 300 profile datasets, apply the χ^2 control chart method and the nonlinear regression method.

The simulation results show that the χ^2 control chart method generally outperforms the nonlinear regression method. The average Type I and II errors of applying these two methods on simulated data are listed in the following tables, where the standard deviations of the Type I and II errors in 300 replications are the numbers in parentheses.

# outliers	Parameter a for Outliers (a=0.5 for non-outliers)							
(P)	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
20	7(1)	9(1)	7(1)	7(1)	8(1)	8(1)	8(1)	8(1)
40	7(1)	6(1)	3(1)	3(1)	4(1)	4(0)	4(0)	4(0)
60	5(1)	3(1)	1(0)	1(0)	2(1)	2(0)	2(0)	3(2)
80	7(1)	8(2)	3(1)	2(1)	3(1)	5(1)	5(1)	6(1)

Table 12. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by χ^2 control chart method when a shifts, Type I error

# outliers	Parameter a for Outliers (a=0.5 for non-outliers)							
(P)	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
20	N/A	52(10)	1(2)	0(0)	0(0)	0(0)	0(0)	0(0)
40	N/A	70(6)	10(4)	0(0)	0(0)	0(0)	0(0)	0(0)
60	N/A	82(4)	39(6)	3(2)	0(0)	0(0)	0(0)	0(0)
80	N/A	91(3)	75(0)	41(5)	8(3)	1(1)	0(0)	0(0)

Table 13. Average percent (and standard deviation) of outlier profiles incorrectly identified as non-outliers by χ^2 control chart method when a shifts, Type II error

Tables 12 and 13 list the simulation results with factor pair (P, a) of the χ^2 control chart method. In Table 12, when a equals 0.5, there are no outlier profiles and we see that 5 to 7 percent of profiles are incorrectly identified as outliers, though the Type I error in the test of hypothesis was set at 5 percent. The realized Type I error is higher because the statistic Δ_i' in Eqn. (15) is a sample from an approximately χ^2 distribution.

Table 13 shows that at each P, the Type II error drops quickly with the increase of a. It also shows that at the same value of a, as the number of outliers increases, the Type II error increases. Note that in Table 13, when a=0.5, the calculation is not applicable (N/A) because there are no outlier profiles.

Fig. 17 illustrates that depending on visual recognition of an outlier profile is not realistic. The figure shows 200 non-outlier profiles in gray with a=0.5 and one

outlier profile in black with a=1.1. It would be difficult to visually pick out this outlier. The proposed method though would find it almost certainly, according to Table 13.

Table 14 and 15 show the Type I and II errors of the χ^2 control chart method with factor pair (P, σ). Table 14 shows that the χ^2 method retains most of the non-outlier profiles and Table 15 shows that when σ increases sufficiently, most of the outlier profiles are detected.

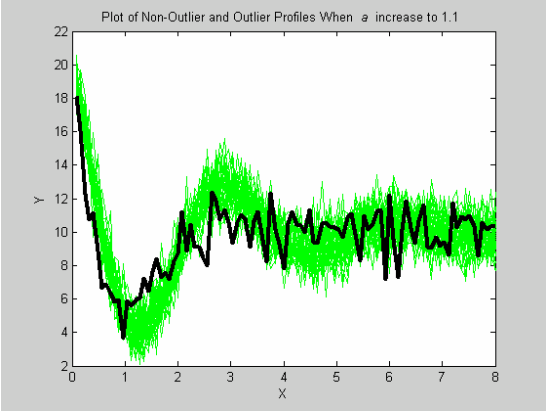


Figure 17. 200 Non-outlier profiles in gray and one outlier profile in bold

# outliers	Parameter σ for outliers (σ=1 for non-outliers)									
(P)	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
20	10(1)	10(0)	10(0)	10(0)	10(0)	10(0)	10(0)	10(0)	10(0)	10(0)
40	4(0)	2(0)	2(1)	2(1)	2(1)	2(1)	2(1)	2(1)	2(1)	2(1)
60	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
80	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)

Table 14. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by χ^2 control chart method when σ increases, Type I error

# outliers	Parameter σ for outliers (σ=1 for non-outliers)									
(P)	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
20	16(8)	0(1)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
40	26(6)	1(1)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
60	40(5)	4(3)	0(1)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)	0(0)
80	48(4)	16(3)	6(2)	3(2)	1(1)	1(1)	0(1)	0(1)	0(0)	0(0)

Table 15. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by χ^2 control chart method when σ increases, Type II error

# outliers	Parameter *a* for Outliers (*a*=0.5 for non-outliers)							
(*P*)	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
20	61(46)	64(45)	61(46)	67(44)	60(45)	64(44)	63(44)	60(45)
40	64(45)	63(45)	67(42)	69(40)	67(39)	68(37)	67(36)	72(32)
60	55(47)	61(45)	70(39)	73(31)	79(24)	86(19)	89(13)	93(9)
80	66(45)	66(41)	75(30)	89(15)	93(8)	96(5)	97(3)	99(1)

Table 16. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by nonlinear regression method when *a* shifts, Type I error

	Parameter *a* for Outliers (*a*=0.5 for non-outliers)							
# outliers	0.5	0.7	0.9	1.1	1.3	1.5	1.7	1.9
20	N/A	16(21)	1(2)	0(0)	0(0)	0(0)	0(0)	0(0)
40	N/A	19(23)	1(3)	0(0)	0(0)	0(0)	0(0)	0(0)
60	N/A	27(30)	4(6)	0(0)	0(0)	0(0)	0(0)	0(0)
80	N/A	29(34)	10(12)	1(2)	0(0)	0(0)	0(0)	0(0)

Table 17. Average percent (and standard deviation) of outlier profiles incorrectly identified as non-outliers by nonlinear regression method when *a* shifts, Type II error

	Parameter σ for outliers (σ=1 for non-outliers)									
# outliers	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
20	56(48)	57(48)	52(49)	59(49)	57(49)	54(49)	63(48)	59(49)	52(49)	58(48)
40	63(47)	60(48)	62(48)	63(48)	55(49)	61(48)	60(48)	59(46)	63(41)	70(34)
60	63(48)	57(49)	58(49)	60(49)	55(49)	63(45)	62(42)	73(31)	80(21)	92(10)
80	60(49)	56(50)	64(48)	60(47)	57(44)	65(39)	80(24)	91(10)	97(3)	99(1)

Table 18. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by nonlinear regression method when σ increases, Type I error

	Parameter σ for outliers (σ=1 for non-outliers)									
# outliers	1.2	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0
20	34(37)	18(21)	7(8)	1(2)	0(1)	0(1)	0(0)	0(0)	0(0)	0(0)
40	31(39)	23(28)	11(14)	4(6)	1(2)	0(1)	0(0)	0(0)	0(0)	0(0)
60	34(41)	29(34)	19(22)	9(11)	5(6)	2(3)	1(2)	0(1)	0(1)	0(0)
80	35(42)	32(37)	21(28)	16(20)	13(13)	7(8)	4(5)	3(4)	2(2)	1(2)

Table 19. Average percent (and standard deviation) of non-outlier profiles incorrectly identified as outliers by nonlinear regression method when σ increases, Type II error

The results of applying the nonlinear regression method on the same simulated datasets are shown in Tables 16 to 19. We now compare the performance of the χ^2 control chart method and the nonlinear regression method. The χ^2 control chart method has much lower Type I error than the nonlinear regression

method in all simulated datasets, no matter whether *a* shifts or σ increases; see Tables 12 vs. 16 and Tables 14 vs. 18. High Type I error of the nonlinear regression method is caused by the high correlations among nonlinear regression coefficients, which causes the near-singularity of the variance-covariance matrices $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$ in section 3.1. So, even a small shift in the regression coefficient vector leads to high testing statistics $T_{1,i}^2$, $T_{2,i}^2$ or $T_{3,i}^2$.

The χ^2 control chart method has higher Type II error than the nonlinear regression method when the shift of parameter *a* is small; see Tables 13 vs. 17. This is not a surprise since χ^2 control chart is known to be insensitive to small and moderate mean shifts. To have a higher sensitivity to small and moderate shift in the mean vector, one can choose multivariate EWMA control chart to identify outlier profiles; see Montgomery (2001) for details.

When σ increases, the χ^2 control chart method generally has smaller Type II error than the nonlinear regression method; see Tables 15 vs. 19. It shows that the χ^2 control chart method is more effective in detecting variance increases than the nonlinear regression method.

4.2.3 Vertical density profile data

In this subsection, we apply the χ^2 control chart method to the Vertical Density Profile (VDP) dataset which can be accessed at http://bus.utk.edu/stat/walker/VDP/ Allstack.txt. Each of the 24 profiles consists of the density of a board measured at fixed depths across the thickness of the board with 314 measurements taken 0.002 inches apart. One VDP profile is shown in Fig.

18.

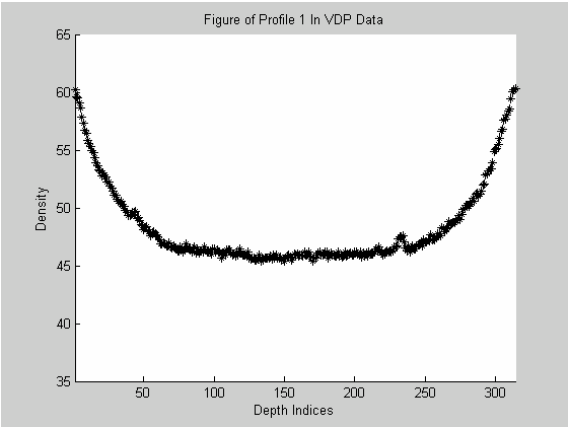


Figure 18. Example of a VDP profile

To apply the χ^2 control chart method, we first check whether the noise term ε_{ij} has the same variance at different values of $X=x_j$. Fig. 19 shows that the variances are obviously different at different X's and thus we use the regular sample variance to estimate σ_{sj}^2 as in Section 4.1.4.

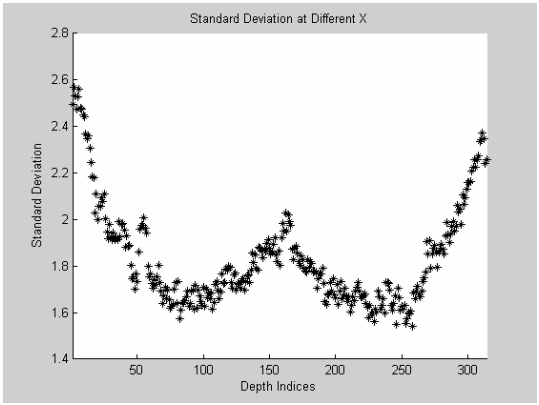


Figure 19. Standard deviation vs. X In VDP data

Profiles 3, 6, 9, 10 and 14 are identified as outliers as shown in Fig. 20 which gives the dissimilarity measure $\Delta_i^{'}$ from Eqn. (17) for each of the 24 profiles compared to the threshold value.

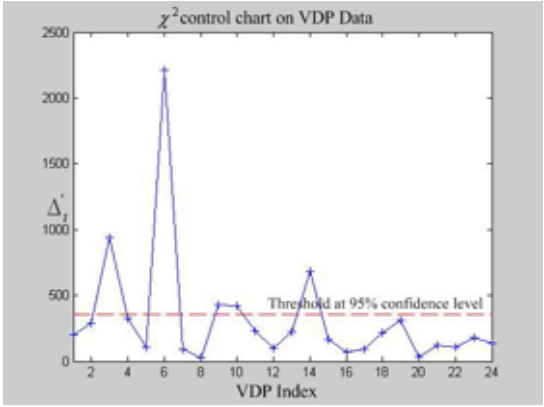


Figure 20. χ^2 control chart on VDP data

The outlier profiles are highlighted in Fig. 21 with black curves and the non-outlier profiles are shown in gray. Profile 3 is an outlier because the density is too high across all x; profiles 6, 9 and 14 are outliers because the density is too low across all x. Profile 10 is an outlier because its shape is not consistent with the other profiles; the density decreases too quickly at low depths and increases too quickly at high depths.

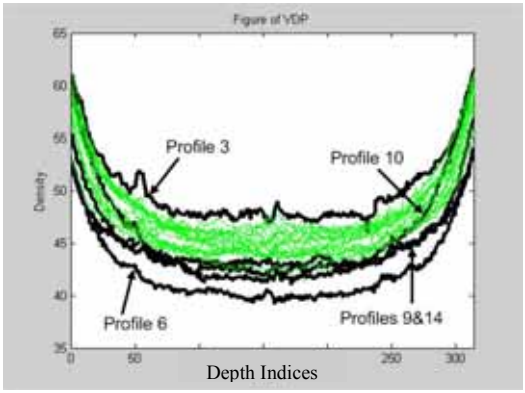
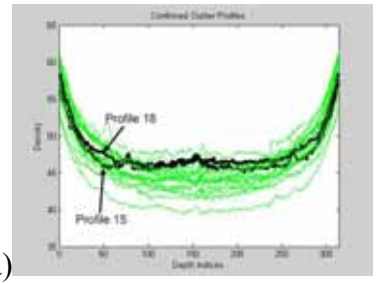


Figure 21. VDP outlier profiles identified by χ^2 control chart method

In Williams *et al*. (2003) each profile is modeled with a non-linear regression and four T^2 charts identify profiles 4, 9, 15, 18, and 24 as outliers. Upon further examination, especially of the regression coefficients, they conclude that profiles 15

and 18, shown in Fig. 22(a), are outliers while profiles 4, 9, and 24, shown in Fig. 22(b), require further investigation. Visually, it is not obvious why these are outliers and why some other profiles are not labeled as outliers.

(a) 

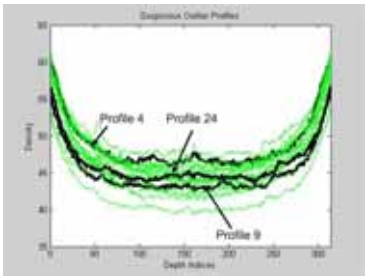
(b) 

Figure 22. VDP outlier profiles identified by Williams *et al*. (2003) (a) outliers; (b) possible outliers

4.3 Discussions

4.3.1 Robustness of estimator $\hat{\sigma}_s^2$ in Eqn. (14)

In Eqn. (16), we use the median of *N(N-1)*/2 pair-wise estimators $\hat{\sigma}_{(i,k)s}^2$ to estimate σ_s^2. Now, using simulated data, we illustrate that estimator $\hat{\sigma}_s^2$ in Eqn. (14) is more robust to outliers than MAD estimator $\hat{\sigma}_{MAD,s} = 1.25\frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N}\left|y_{ij} - \bar{y}_{\cdot j}\right|$ and regular sample variance $\hat{\sigma}_{Sample,s}$. In each dataset, 200 profiles are generated where 160 nonoutlier profiles are generated by Eqn. (19) with *a*=0.5 and $\sigma = 1$. The remaining *P*=40 outliers are generated by increasing *a* from 0.7 to 1.9 and keeping σ unchanged. Table 20 lists the means (and standard deviations in brackets) of these three estimators in 300 replicates. It shows that $\hat{\sigma}_s$ is closer to 1, which is the true value of σ, than the other two estimators. The conclusion holds even when *P*=80.

Estimators	*a* (*a*=0.5 for nonoutliers)						
	0.7	0.9	1.1	1.3	1.5	1.7	1.9
$\hat{\sigma}_s$	1.021 (0.002)	1.04 (0.001)	1.042 (0.001)	1.042 (0.001)	1.042 (0.001)	1.042 (0.001)	1.042 (0.001)
$\hat{\sigma}_{MAD,s}$	1.021 (0.003)	1.060 (0.003)	1.096 (0.003)	1.125 (0.003)	1.149 (0.003)	1.169 (0.003)	1.186 (0.003)
$\hat{\sigma}_{Sample,s}$	1.025 (0.003)	1.065 (0.003)	1.102 (0.003)	1.133 (0.003)	1.158 (0.004)	1.179 (0.003)	1.197 (0.004)

Table 20. Three estimates of σ_s when *P*=40 and *a* increases

4.3.2 Application of the χ^2 control chart method to on-line profile monitoring

After removing the outlier profiles from the baseline data, we can derive a central point as in Eqn. (11) with the remaining profiles. We will consider a newly observed profile, $y_{new,j}$, j=1, …, M, out-of-control if the dissimilarity measure

$$\Delta_{new}^{'} = \sum_{j=1}^{M} \frac{(y_{new,j} - \hat{y}_j)^2}{\frac{N+1}{N}\hat{\sigma}_s^2} \tag{20}$$

exceeds the 95th percentile of the χ^2 distribution with M degrees of freedom. Note that the dissimilarity measure is slightly different than in Eqn. (15).

Eqn. (20) is obtained by observing that $y_{new,j}$ is independent from the y_{ij}'s in the baseline profiles. It follows that $y_{new,j} - \hat{y}_j \sim iid\ N(0, \frac{N+1}{N}\sigma_s^2)$ and $\frac{y_{new,j} - \hat{y}_j}{\sqrt{\frac{N+1}{N}\sigma_s^2}} \sim iid\ N(0,1)$ and $\Delta_{new}^{'}$ in Eqn. (20) has an approximate χ^2 distribution with M degrees of freedom.

5 Detect and diagnose changes of correlation matrix

In this chapter, we first describe the problems that regular MSPC methods may have when correlation matrix changes. Then, in section 5.2, we describe in details a test of hypothesis on the similarity of two correlation matrices. A diagnose method is proposed in this section when the similarity is denied. Section 5.3 gives simulation results of the testing and diagnosing.

5.1 Problems when correlation matrix changes

When correlation matrices change, the performance of MSPC methods in detecting mean shifts may be jeopardized. In this section, we first show that regular MSPC methods have poor performance in detecting correlation matrix changes in some specific situations. Then, we demonstrate through a simulation example that correlation matrix changes may deteriorate the performance of MSPC methods in detecting process mean shifts.

Let us consider the simplest bi-variate situation where the baseline correlation matrix is $\boldsymbol{\Sigma}_0 = \begin{bmatrix} 1 & \theta \\ \theta & 1 \end{bmatrix}$, $\theta > 0$. We use a Hotelling's T^2 control chart to monitor these two variables simultaneously. The stars in Fig. 23 represent baseline observations. Two new orthogonal axis, *z-1* and *z-2*, represent the first and the second main variation directions, respectively. The in-control area of the T^2 model, if plotted on the two dimensional space, is the ellipsoid in Fig. 23.

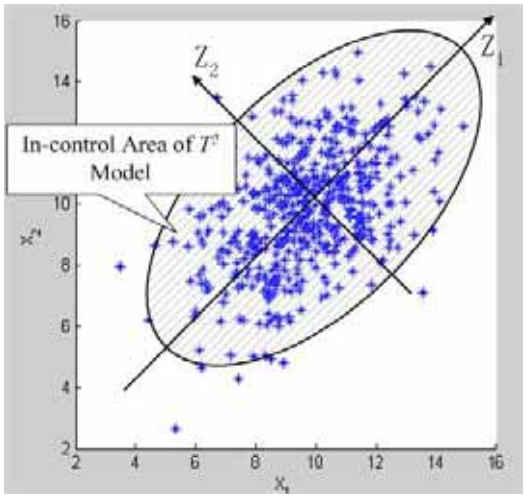


Figure 23. Scatter plot of observations with correlation matrix $\mathbf{\Sigma}_0$.

Now we change $\mathbf{\Sigma}_0$ to $\mathbf{\Sigma}_1 = \begin{bmatrix} 1 & \theta + \Delta_{12} \\ \theta + \Delta_{21} & 1 \end{bmatrix}, \Delta_{12} = \Delta_{21} = \delta > 0$. The principal axes will not change, and the points generated with the new correlation matrix will be more compressed along axis *z-1*, shown as circles in Fig. 24. We can see that these new observations are less likely to exceed the in-control area. It is difficult for T^2 chart to detect this correlation change.

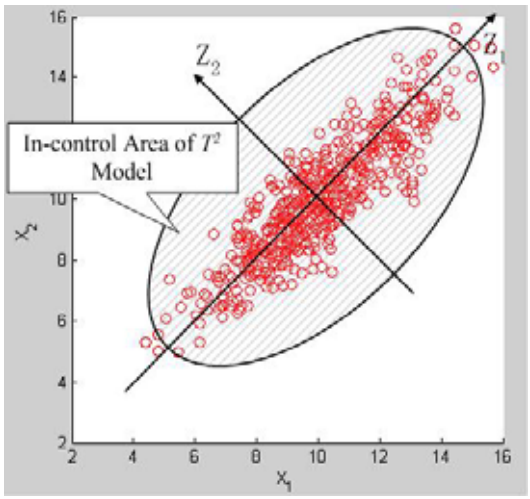


Figure 24. Scatter plot of observations with correlation matrix $\mathbf{\Sigma}_1$ where $\delta > 0$.

Now, let us see what will happen when $\delta < 0$. If $\delta < 0$ and $\theta + \delta > 0$, the new observations generated with $\mathbf{\Sigma}_1$ have more variation along axis *z-2* and the T^2

chart signals easily. Operators may be able to find the cause of the correlation change during inspections. Furthermore, if $\theta+\delta<0$, the main variation direction will be in the direction of *z-2* , so points appear more frequently out of the in-control area; see Fig. 25.

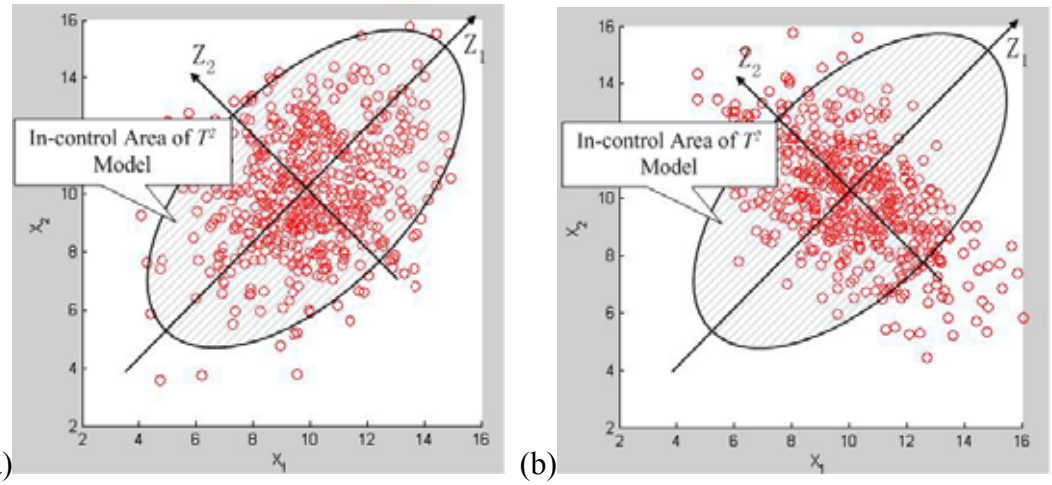


Figure 25. Scatter plot of observations with $\mathbf{\Sigma}_1$ where (a) $\delta<0$ and $\theta+\delta>0$ (b) $\delta<0$ and $\theta+\delta<0$.

This shows that the T^2 chart is sensitive to directions of correlation coefficient changes. It is only sensitive to the correlation changes in some specific directions. The same situation happens when $\theta<0$ or more than 2 variables are under consideration. If $\theta>0$ and $\delta>0$, or $\theta<0$ and $\delta<0$, we call it the change in the sign direction, otherwise, we call it the reverse direction.

The intuitive explanation for this phenomenon is that when the correlation changes in the sign direction, the directions of the major variations change a little (or do not change in bi-variate situation), while the effects of the more compression along these directions are far beyond the change of main variation directions. So, it will be hard for a T^2 chart to signal. On the other hand, for the change in reverse directions, T^2

chart can signal easily because either of the more sparse distribution of data on some directions or of the great direction changes of the major variations.

The direction sensitiveness of the T^2 chart is very dangerous in practice, because the correlation changing in the sign direction may mask the shifting of mean values, which makes it even less capable of signaling the mean shifts.

Table 21 shows quantitatively how correlation changes may affect the capability of a T^2 chart to detect process mean shifts. Table 21 is generated as follows: 200 bi-variable normal distributed baseline observations are generated with mean [10, 10], variance [3, 2] and correlation coefficient 0.5. We denote these two variables as X_1 and X_2. A T^2 chart is used to monitor this process and the control limit T^2_{UCL} is calculated such that a point whose T^2 statistic exceeds T^2_{UCL} is considered out-of-control at 95% confidence level. So, when there is no process error, the ARL should be 1/0.05=20.

Then we apply the T^2 chart to monitor the process on-line. The monitored observations are generated with two factors: the shifting amount of the mean of X_1 and the change of correlation coefficient, denoted by $\Delta\mu_1$ and δ respectively. Factor $\Delta\mu_1$ takes values [0, 1, 2, 3, 4] and δ takes values [-0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4]. For each [$\Delta\mu_1$, δ] combination, 200 replications are run. In each replication, N=1000 normally distributed observations are generated with mean [10+$\Delta\mu_1$, 10], variance [3, 2] and correlation coefficient 0.5+δ. The index of the first observation whose T^2 statistic exceeds the T^2_{UCL} is recorded as the run length (RL). Then average RL (ARL) is calculated by averaging these RLs in 200 replications.

In Table 21, when $\Delta\mu_1 \le 3$ and δ>0, the ARLs are larger than when δ=0, i.e., no correlation change. This shows that the correlation changes may delay the detection of small mean shifts by T^2 charts.

		δ						
		Reverse Direction		No Shift	Sign Direction			
		-0.2	-0.1	0	0.1	0.2	0.3	0.4
$\Delta\mu_1$	0	15.8	17.4	21.2	28.0	33.7	45.3	30.8
	1	10.9	13.0	15.3	20.2	21.2	25.7	25.7
	2	5.3	5.7	5.6	6.9	7.7	10.6	12.4
	3	2.9	2.7	2.9	2.5	3.2	3.4	4.3
	4	1.8	1.9	1.8	1.7	1.7	1.7	1.7

Table 21. ARL of T^2 charts in detecting mean shifts when correlation changes

5.2 Test of correlation matrix similarity and correlation change diagnosis method

In this section, we first describe a test procedure for the similarity of two correlation matrices. Then we propose a diagnose method when the test shows significant difference between them to give the possible responsible variables.

5.2.1 Testing similarity between two correlation matrices

Rencher (2002) derives a statistic *u* to measure the difference between the desired and the actual correlation matrix. When *u* exceeds the threshold value, we say that the current correlation matrix is significantly different from the baseline one. The test procedure in Rencher (2002) is as follows:

$$H_0 : \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_0$$
$$H_1 : \boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_0$$

$$u = v[\ln|\boldsymbol{\Sigma}_0| - \ln|\mathbf{S}| + tr(\mathbf{S}\boldsymbol{\Sigma}_0^{-1}) - p] \quad (21)$$

where **S** is the estimate of the current correlation matrix $\boldsymbol{\Sigma}_1$, $\boldsymbol{\Sigma}_0$ is the desired one,

p is the number of process variables to be monitored, v=n-1, n is the sample size when estimating $\mathbf{\Sigma}_1$, and $tr(\mathbf{X})$ is the trace of square matrix $\mathbf{X}$. If statistic $u > C$, reject H_0, where C is the threshold value.

If the n observations come from a distribution with the same correlation matrix as $\mathbf{\Sigma}_0$, when v is large, u in Eqn. (21) is approximately χ^2 distributed with degrees of freedom $df=\frac{1}{2}p(p+1)$. The value of C in the test procedure is usually set as the α percentile of the χ^2 distribution, denoted as $\chi^2_\alpha(\frac{1}{2}p(p+1))$. Regularly we choose α=0.99 or 0.95. If v is small, the following statistic u' is a better approximation to the χ^2 distribution.

$$u' = [1 - \frac{1}{6v-1}(2p+1-\frac{2}{p+1})]u \tag{22}$$

Eqn. (21) can be expressed in terms of eigenvalues $\lambda_1, \lambda_2, ..., \lambda_p$ of matrix $\mathbf{S}\mathbf{\Sigma}_0^{-1}$ by noting that $tr(\mathbf{S}\mathbf{\Sigma}_0^{-1}) = \sum_{i=1}^{p}\lambda_i$ and $\ln|\mathbf{\Sigma}_0| - \ln|\mathbf{S}| = -\ln|\mathbf{\Sigma}_0^{-1}| - \ln|\mathbf{S}| = -\ln|\mathbf{S}\mathbf{\Sigma}_0^{-1}|$ $= -\sum_{i=1}^{p}\ln\lambda_i$. So, Eqn. (21) is equivalent to:

$$u = v[\sum_{i=1}^{p}(\lambda_i - \ln\lambda_i) - p] \tag{23}$$

Note that if $\mathbf{S} = \mathbf{\Sigma}_0$, $\mathbf{S}\mathbf{\Sigma}_0^{-1} = \mathbf{I}$, where $\mathbf{I}$ is a $p \times p$ identity matrix. Then $\lambda_i = 1, i = 1, 2, ..., p$. So, in Eqn. (21), $\sum_{i=1}^{p}(\lambda_i - \ln\lambda_i) = p$, and u=0. Otherwise, if $\mathbf{S} \neq \mathbf{\Sigma}_0$, there exist some eigenvalues not equal to 1. Fig. 26 shows the curve of $\lambda - \ln\lambda$ vs. λ, from which it is clear that $\lambda - \ln\lambda$ can get its minimal value 1 at $\lambda = 1$, otherwise $\lambda - \ln\lambda > 1$. So, when $\mathbf{S} \neq \mathbf{\Sigma}_0$, u>0.

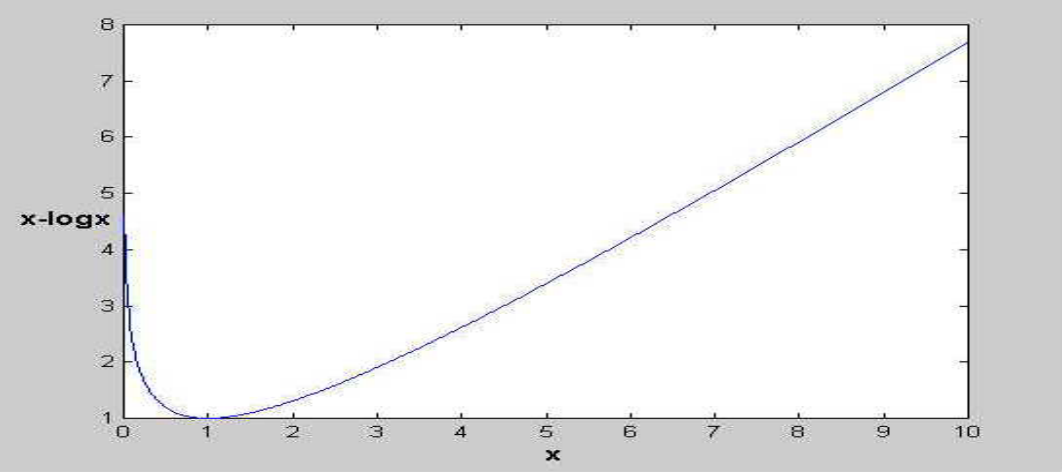


Figure 26. $(x-\ln x)\sim x$ plot

5.2.2 Correlation change diagnosis (CCD) method when H_0 is rejected

Eqn. (23) also tells us how each $\lambda_i = 1, i = 1, 2, ..., p$ contributes to statistic u. Now, our question is that whether we can get any information from those $\lambda's$ and their corresponding eigenvectors on the possible causes when H_0 is rejected.

When H_0 is rejected, with Eqn. (23), we can rank the value of $\lambda_i - \ln \lambda_i$, i=1, 2, …, p, in decreasing order, to see the contribution of each λ_i to the value of u. We name $\{\lambda_i : \lambda_i - \ln \lambda_i > 1\}$ as the contributing eigenvalues and their corresponding eigenvectors as contributing eigenvectors. In the following proposition, we propose that the contributing eigenvectors can give us information of possible variables responsible for the rejection of H_0. We use these information to diagnose the causes of rejecting H_0. We call this method as correlation change diagnosis (CCD) method. Its proof is provided in Appendix C.

Proposition 5.1: Suppose there are p variables with correlation matrix $\mathbf{\Sigma}_0$. We denote the set of p variables as S. We also denote a variable set consisting of all the variables whose mutual correlation coefficients change as S_1, the number of

variables in S_1 as k, and the subset consisting of the remaining variables as S_2. We reorganize the p variables such that the first k variables are in S_1, and the last p-k variables are in S_2. We denote the new correlation matrix as $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_0 + \mathbf{\Delta}_\Sigma$, where $\mathbf{\Delta}_\Sigma = \begin{bmatrix} \mathbf{\Delta}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $\mathbf{\Delta}_{11}$ is a k-by-k symmetric matrix. Then, for any of the contributing eigenvalues of matrix $\mathbf{D} = \mathbf{\Sigma}_1 \mathbf{\Sigma}_0^{-1}$, the contributing eigenvectors will have all zero elements on positions corresponding to variables in S_2;

We illustrated the above proposition in the following example.

Example:

$$\mathbf{\Sigma}_0 = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & -0.1 \\ 0.3 & -0.1 & 1 \end{bmatrix}$$

$$\mathbf{\Sigma}_1 = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & -0.2 \\ 0.3 & -0.2 & 1 \end{bmatrix} = \mathbf{\Sigma}_0 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -0.1 \\ 0 & -0.1 & 0 \end{bmatrix}$$

Then, the eigenvalues of $\mathbf{D} = \mathbf{\Sigma}_1 \mathbf{\Sigma}_0^{-1}$ and their corresponding eigenvectors are:

$\lambda_1 = 1, \lambda_2 = 0.83, \lambda_3 = 1.10$

$\mathbf{t}_1 = [-0.86, -0.43, -0.26]^T$

$\mathbf{t}_2 = [0, -0.67, -0.74]^T$

$\mathbf{t}_3 = [0, 0.67, -0.74]^T$

Since only variable 2 and 3 change their correlation, in the eigenvectors corresponding to the contributing eigenvalues, only the 2nd and 3rd elements have non-zero values.

However, in practice, because of the existence of noise and the limit of the number of observations to estimate the new correlation matrix $\mathbf{\Sigma}_1$, the estimate of the new correlation matrix may not be identical to the actual one. For the eigenvectors

corresponding to the contributing eigenvalues of **D** matrix, in the positions of variables in S_2 , non-zero values may appear, but with small absolute values. So, in practice, we can plot a bar chart, with the horizontal axes being the variables. The vertical axes of the bar chart is the absolute values of the eigenvector elements corresponding to the λ which has the maximum $\lambda - \ln\lambda$ value. We denote this eigenvector as *diagnosis eigenvector*.

For instance, after the similarity of correlation matrices is denied, we get a *diagnosis eigenvector* as:

[0.11, 0.04, 0.06, -0.12, -0.10, 0.55, -0.11, -0.04, 0.02, 0.80], then the contribution plot can be plotted as Fig. 27. Actually, the correlation change occurs between variables 6 and 10. Fig. 27 shows that variables 6 and 10 have the highest bar. Thus operators can be guided to narrow their inspection scope onto variables 6 and 10.

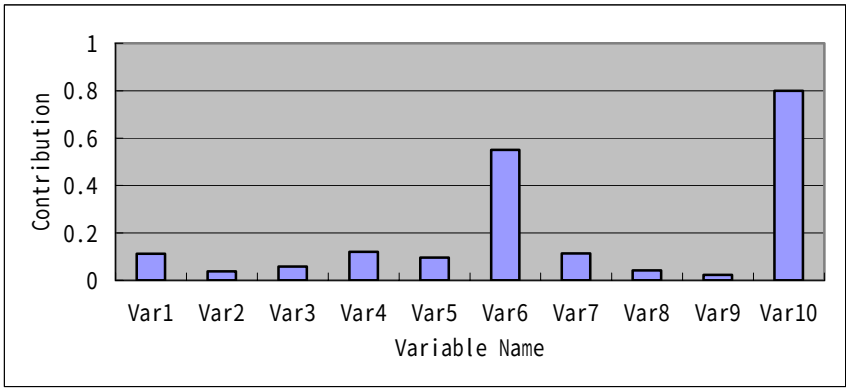


Figure 27. Contribution bar chart of CCD method

Accuracy of the CCD method is one thing we are concerned with. In practice, the operators do not know in advance how many variables change their correlations, so an applicable way is to define a set with a fixed size, for instance, set the size as 3. We say the CCD method is a success if the actual fault-causing variables are

included in the set after the diagnosis.

For example, in Fig. 27, we set the diagnosis set size as 3. Variables 4, 6 and 10 have the highest bars, so this set equals {4, 6, 10}. We know the actual variables which change their mutual correlations are 6 and 10, so $\{6, 10\} \subset \{4, 6, 10\}$ and this diagnosis is a successful one. If more than 3 process variables change correlations, the operators can first inspect these 3 variables and correct any error among them, and then put the process into running. After the similarity of the latest and the baseline correlation matrix is denied again, use CCD method to find out another 3 possible contributing variables.

After we put MSPC methods in on-line monitoring, we need to test the similarity of the latest and the baseline correlation matrix regularly. We can use a non-overlapping window of size w. The estimate of the current correlation matrix is derived from the latest w observations. For instance, if w=100, we estimate the correlation matrix at t=100 with the first 100 observations, and at t=200 with the second 100 observations, etc. The value of w can be determined pragmatically from the number of variables and how fast we want to be able to detect correlation matrix changes. With the larger w, we can have the more accurate estimate of correlation matrix, but the detection of changes of correlation matrices becomes slower.

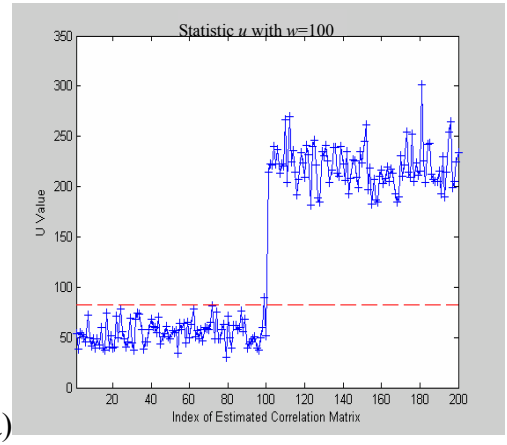
5.3 Simulation for similarity testing and diagnosing

In this section, several simulations are conducted to demonstrate the performance of similarity test in detecting correlation changes and the CCD method in diagnosing possible responsible variables. The simulation results show that the

similarity test can pick up the correlation changing quickly and give the diagnosis with high accuracy.

These simulations are conducted under these conditions: 1000 observations of 10 process variables from a real manufacturing process running under successful conditions are gathered to estimate the correlation matrix $\boldsymbol{\Sigma}_0$. The correlation between a randomly chosen variable pair, variables 5 and 7, is changed by an amount $\Delta_{5,7} = \Delta_{7,5} = \delta$, thus leading to a new correlation matrix $\boldsymbol{\Sigma}_1$. The correlation coefficient between variables 5 and 7 in $\boldsymbol{\Sigma}_0$ is 0.37. Then 10000 normally distributed observations are generated with mean vector $\boldsymbol{\mu} = \mathbf{0}$, correlation matrix= $\boldsymbol{\Sigma}_0$ and variance vector $\boldsymbol{\sigma} = \mathbf{1}$. Another 10000 observations are generated with N($\mathbf{0}$, $\boldsymbol{\Sigma}_1$), which follow the first 10000 observations to form a dataset with 20000 observations. Then the similarity test is implemented on the 20000 observations, with different *w*. Fig. 28 shows the charts of *u* statistics for different *w*'s. It shows that the similarity test can detect the significant difference between two correlation matrices correctly.

(a)



(b)

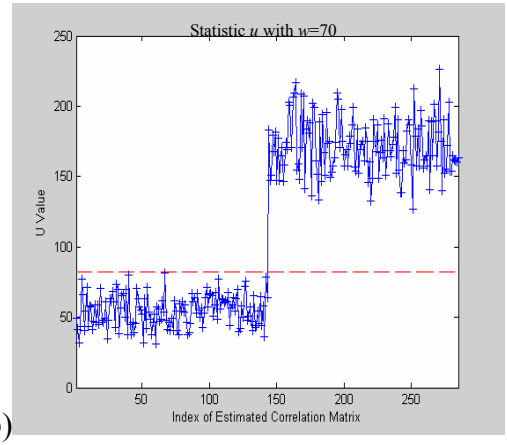


Figure 28. Chart of *u* statistics with (a) *w*=100 (b) *w*=70

In Fig. 28(a), the *u* statistic of the 101st moving window exceeds the threshold. It is the first estimate of the correlation matrix after the correlation matrix changes to

$\boldsymbol{\Sigma}_1$. In Fig. 28(b), *w*=70, the *u* statistic starts to go beyond the threshold value after the 144th estimate of correlation matrix. The *u* statistic of the 143rd moving window, which consists of observations 9941~10010, does not exceed the threshold value since in these 70 observations, 60 observations are generated with correlation matrix $\boldsymbol{\Sigma}_0$, the remaining 10 come from the new correlation matrix $\boldsymbol{\Sigma}_1$.

To study how accurate the CCD method can diagnose the possible variables responsible to the correlation matrix change, for each combination of *w* and δ , 200 replications are run. Factor *w* takes values 70, 90 and 130, δ has three levels, 0.3, -0.3 and -0.45, respectively. Each time when the similarity test denies H_0, the CCD method is applied. The percentages of successful diagnosis in the 200 replications under different *w* and δ are recorded in Table 22. We can see that CCD method always gives high accuracy rate under different *w* and δ values, and the CCD method gives more accurate diagnosis if the window size is higher, i.e., we use more observations to estimate the current correlation matrix.

		w		
		70	90	130
δ	0.3	97.5	99.5	98.5
	-0.3	92.5	92.5	96.5
	-0.45	96.5	99	99

Table 22. Percentage of successful diagnosis

6 Conclusion

This dissertation covers four topics to improve the SPC model (models) constructing in phase I: determining the number of operational modes in MSPC baseline data; determining baseline periods in historical data collected in a long time period; detecting outlier profiles in complex profile baseline data; and determining whether the MSPC model needs to be updated.

A new SBDD method is first proposed to determine the number of operational modes in a baseline MSPC dataset. The proposed method has the following specific features: (1) It detects the correct number of clusters whether the dataset has one or more clusters; (2) It detects the correct number of clusters whether the clusters are convex or non-convex; (3) It is not sensitive to user-specified parameters.

To demonstrate the performance of the proposed method, we apply it, along with the three existing data mining methods for clustering, on each of four datasets and compare their performances. Three of these four datasets are simulated and the remaining one is a real dataset of the ingredients of three wine products. The numbers of clusters in these datasets are previously known. The results show that the proposed method gives the correct numbers of clusters on all four datasets, while the others do not.

We propose a PDP clustering method to determine baseline from a sequence of historical product observations collected in a long time period. It uses overlapping moving windows to segment the sequence into subsequences. These subsequences are transformed into PDPs. Clustering methods are applied to group these PDPs, and to

cluster each single observation. Basic statistics of each point cluster are calculated and the clusters with the most satisfying statistics are selected. The periods corresponding to the points in the selected clusters are considered as baseline periods.

We apply the proposed PDP clustering method on simulated and real datasets and its performance is compared with the LRT method. The results on simulated datasets show that the proposed PDP clustering method is robust to distributions which generate the data, but the LRT method is not. The PDP clustering method is insensitive to the reasonable designation of bins, which are used to transform subsequences into PDPs. On the real dataset, the selection of period of successful production by the PDP clustering method is supported by the changes of process variables. However, the LRT method only picks up a small portion of period of successful production. We think the PDP clustering method gives more convincing result.

The limitation of the proposed PDP clustering method is that it is difficult to be applied when the number of product variable is large. The proposed PDP clustering method assumes that there is only one product variable. If we want to extend it to cases with multiple product variables, we need to define a set of grids in high dimension space. A sequence of observations can thus be transformed into PDPs and the PDP clustering method can be applied. However, with the increase of dimension, the number of grids increases exponentially. So, the number of observations in a subsequence has to be very large to be compatible with the number of grids, which might not be feasible.

There are two possible ways to handle these cases. One is that we select only a few (one or two) product variables which dominate the product quality. Alternatively, we use data reduction methods, such as principal component analysis, to reduce the dimensionality by choosing only a few latent variables to represent the majority of variance in the original variables. Each latent variable is a linear or nonlinear combination of the original product variables. Then, the PDP clustering method can still be applied on the selected dominating product variables or latent variables.

We apply the χ^2 control chart method to detect outlier profiles that does not require fitting regression models and can be applied to profiles of any complexity. This is accomplished by treating profiles as vectors in high-dimension space. This method is useful in process control in removing outliers from baseline data and also in monitoring new profiles. It may sometimes be the only option when the profiles are so complex that all other methods do not apply.

This method uses the median of the baseline profiles to estimate the mean vector of the nonoutlier profiles (vectors). The difference between a profile and the center vector is measured by a statistic that is approximately χ^2 distributed. A profile is identified as an outlier if this statistic exceeds a threshold value.

This method can be applied to profiles that are too complex to model with linear or non-linear regression, as illustrated in the simulation experiments we conducted. The χ^2 control chart method successfully identifies most of the outliers while retaining most of the non-outlier profiles. When applied to actual data where the profiles describe the density of a wood product along the depth, the χ^2 control chart

method yields convincing results.

The fourth method proposed in this dissertation is to test the similarity of correlation matrices in MSPC applications and diagnose when the similarity is denied. A test statistic is computed to measure the difference between the current and the baseline correlation matrices. If there is significant difference between them, a new method is devised to diagnose the responsible process variables to see whose mutual correlations have changed. The operators can be guided to inspect whether there are process errors in the diagnosed process variables. If no error is found, we should consider building a new MSPC model.

We apply the correlation matrix testing and diagnosis methods on simulated datasets. The results show that our diagnosis method can find the responsible variables which cause the change of correlation matrix with high accuracy.

7 Future work

In previous chapters, we have shown applications of MSPC and data mining technologies in various manufacturing processes, such as the industrial oven process in Chapter 2 and the continuous process in Chapter 3. In fact, MSPC and data mining technologies can be widely applied in almost all data-rich systems, such as biomedical, manufacturing, health care, and other service systems such as insurance and financial systems.

Among these data-rich systems, my future research will focus on a biomedical system, the brain neuron system. This research opportunity is brought to me by the post doctoral position in Arizona State University.

The biomedical system shares the following four features with the other data-rich systems, which make my expertise in MSPC and data mining useful: (1) It has many variables; (2) Most variables are highly correlated; (3) It is so complicated that it is almost impossible to build explicit physical models to describe it; (4) The patterns of data are usually unknown in such a high dimensional space. Such a system generates a huge amount of data every day, but only a small fraction is utilized. Data mining technologies can extract the patterns in the huge amount of multivariate data and MSPC methods can build statistical models to these patterns.

My future research topic will be finding the patterns of brain neuron activities when one is planning body movements. The brain neurons control our body movements. Signals (commands) are sent from the brain to our body parts such as arms and legs, and movements of these parts are accomplished accordingly.

Discovering these patterns and connecting it with the planned body movements help to predict body movements based on detected neuron signals.

This research is conducted by studying the signals of brain neurons of monkeys. Huge amount of data, e.g., the intra-cortical signals, of the brain neurons of monkeys can be measured by microelectrodes inserted into certain areas of the cortex, as the square area shown in Fig. 29. The areas where the electrodes are inserted depend on the functions of those areas and the tasks we want the monkeys to conduct.

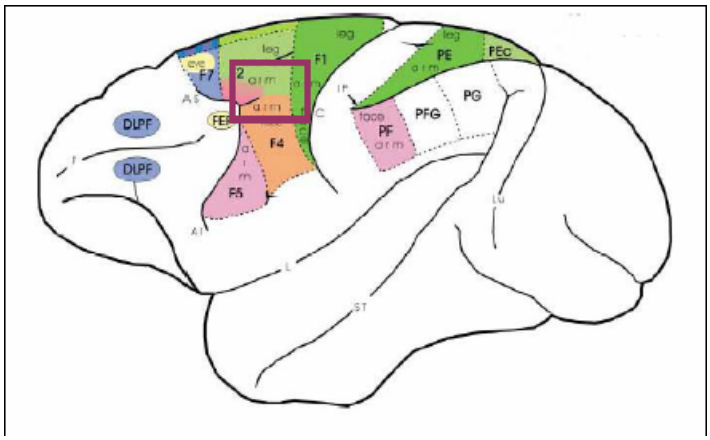


Figure 29. Areas of brain with different functions

The purpose of this research is to investigate the set-related neurons according to their different firing patterns when a monkey is preparing to fulfill a simple task, such as reaching a target with its right hand. Set-related neurons are those in a certain area of the brain that are believed to be involved in preparing for a certain task.

The firing of neurons is defined as follows. There is always a difference in electrostatic potential between the inside and outside of a neural cell: the cell is polarized. We call this electrostatic potential the membrane voltage. The membrane voltage of an inactive cell remains close to a resting potential, which is approximately -70 mV (the negative sign signifies excess negative charge inside the cell relative to

the outside). When the membrane of an excitable cell becomes depolarized beyond a threshold, the cell undergoes an action potential, we say the cell “fires”, often called a “spike”. This threshold generally is about 15 mV more positive than the cell’s resting potential.

Fig. 30 shows the process of the firing of a neuron. In the beginning, the neuron is inactive. The membrane voltage is around -75 mV, which is its resting potential. Then the cell is activated and the membrane voltage increases. When the membrane voltage increases beyond the threshold, the cell “fires”.

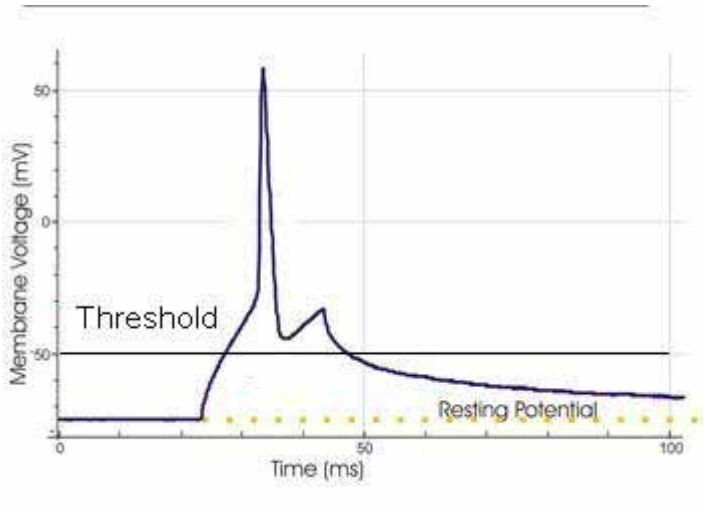


Figure 30. Firing of a neuron

The main hypotheses that we test in this study are as follows. Hypothesis I is to test whether the firing patterns are inhibited by a localized set of neurons, i.e., only a few neurons in that certain area are involved in the task preparing. Hypothesis II is to test whether neurons fire signals to communicate with one another when the animal is preparing the task. The questions addressed in this research are: (1) Which neurons are involved in preparing the task under different conditions? (2) If multiple neurons are involved in preparing for the task, do they work independently or do they

communicate with each other? (3) Do these neurons show different activity pattern when preparing to complete the task under different conditions?

In the remainder of this chapter, we first describe the experiments we conduct to collect the data. Then we propose methods to analyze the data to test the hypothesis described before.

7.1 Empirical Studies

The experiments are conducted in a 3-D virtual reality environment (VRE), which is presented to the monkey through a mirror in front of the monkey's eyes. In the VRE, there are only five objects that can show up: a stationary starting position (green solid sphere), a true target (a green flashing solid sphere), a false target (a green non-flashing semi-transparent sphere), an obstacle (cylinder) and a mobile cursor (red sphere). The position of the cursor in the 3-D VRE is determined by a sensor taped to the wrist of the monkey. When the monkey's hand moves, the cursor in the 3-D VRE also moves.

The monkey is trained to move the cursor from the starting position to hit the true target and hold the cursor there for at least 0.1 second, with the presence or absence of obstacle in the middle of the straight path from the starting position to the target. There are four types of failures when the monkey conducts the task: (1) Fail by curvature, i.e., the monkey takes a detoured trajectory to reach the target when the obstacle is absent; (2) Fail by hit obstacle, i.e., the monkey hits the obstacle because of failing to take the detoured trajectory when the obstacle is present; (3) Fail by wrong target, i.e., the monkey hits the false target; (4) Fail by target hold time, i.e.,

the monkey holds the target less than 0.1 second.

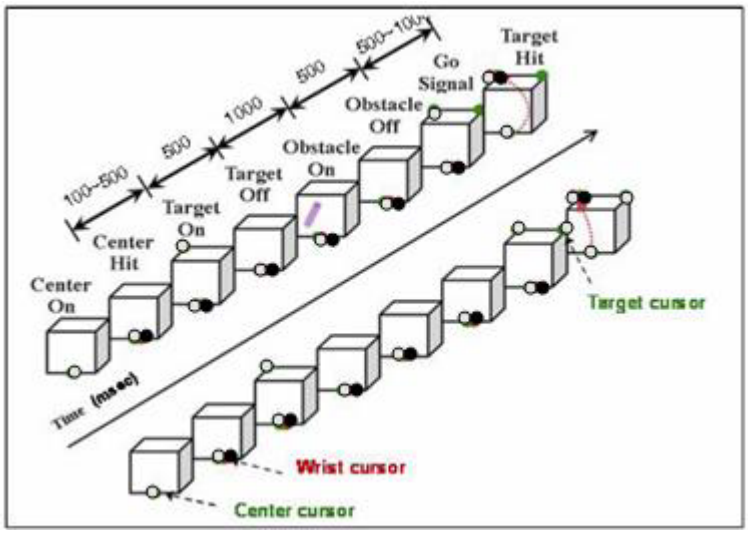


Figure 31. Signals given to monkeys in each trial

There are two factors in the design of experiments: the position of the true target and the existence of the obstacle. The position of the true target has two levels: left-top and right-top corner. The obstacle also has two levels: presence and absence. There are totally 4 different experimental conditions. Several replicates are run under each of the experimental conditions. In each replicate, signals of the same 19 neurons are recorded.

In each trial, signals of eight events are given to the monkey in a certain order, as shown in Fig. 31. In Fig. 31, the cube represents the VRE presented to the monkey. The light gray dot in the bottom represents the stationary starting position. The light gray dots in the up-right or up-left corner represent the true and false targets. The black dot represents the cursor.

The first signal is “Center On”, where the light gray sphere is shown at the bottom of the VRE. The monkey moves its right hand to hit it with the cursor. The time when it is hit is recorded as “Center Hit”. Then, 100 to 500 milliseconds later,

the true target is shown, denoted as “Target On”. The monkey is given 500 milliseconds to memorize the position of the target before it is turned off, as the “Target Off” in Fig. 31. The time 1000 milliseconds later than the “Target Off” is when obstacle appears (“Obstacle On”), if there is an obstacle in the trial. The obstacle is shown as a gray line in the cube of the VRE in Fig. 31. The obstacle cylinder (if any) disappears 500 milliseconds later, denoted as “Obstacle Off”. The “Go” tone is given to the monkey 100 to 500 milliseconds later, and the true and false targets, and the obstacle (if any) are also shown in the VRE. This signal allows the monkey to move the cursor to hit the target. The time when the cursor hit the target is recorded as “Target Hit”.

The preparing time period is the time between “Obstacle On” and “Go”. The monkey is trained to know that if there is no obstacle showing up at the time point of “Obstacle On”, there will be no obstacle in the trial. So, by then the monkey has all the information of the true target and the presence/absence of the obstacle. He starts preparing the task.

Two types of signals are recorded from each neuron: spike train data and waveform data. Spike train data is the time stamps of all neuronal spikes. Fig. 32 shows the spike train data of a neuron in around twenty successful trials, where the *X*-axis is the time axis, each row represents a successful trial, and each dot represent a neuron spike. Four behavioral events are illustrated in Fig. 31 by different symbols: “Obstacle On”, “Obstacle Off”, “Go Signal”, and “Target Hit”. Time zero is aligned to “Obstacle On”. The trials are sorted by “Go Signal” in an ascending order.

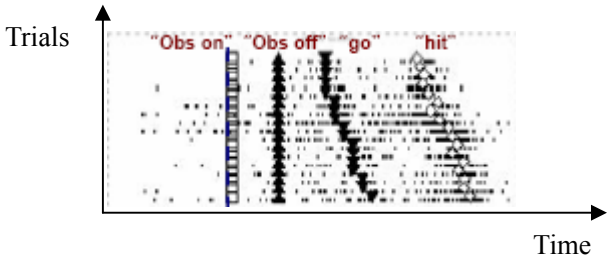


Figure 32. Spike train data of a neuron

Waveform data records the waveform shape of each spike. Fig. 30 is an example of the waveform data of a neural spike.

7.2 Methods

In this section, we describe how to analyze spike train and waveform data, respectively. We model the spike train data of each neuron. The model parameters are used to cluster neurons into clusters of involved and uninvolved neurons when preparing for a task.

The purpose of analyzing waveform data is to build connections between different wave shapes and different body movements. Waveform data records the shape of each spike. Based on the assumption that the commands sent by the brain neurons are coded by different wave shapes, mapping spike shapes to the body movements may decode the spike shapes to commands.

7.2.1 Spike train data

The purpose of this analysis is to find neuron clusters that are involved in preparing the task under each experimental condition and the dependency among these neurons.

For the spike train data of one experimental condition, there are two ways to model it: (1) model the spike train data in each trial, or (2) pool the data of a single

neuron in all successful trials together, transform it into a histogram, and model the histogram. Then we find the cluster patterns in these model parameters.

7.2.1.1 Analyzing spike train data in each trial

We can analyze the spike train data of each trial separately. From Fig. 29, we can see that the preparing process of a neuron can be segmented into several stages, each of which has different arrival rate of spikes. It is equivalent to say that the inter-spike time in different stages is distributed differently. If we plot the inter-spike time of a neuron in a successful trial, we have the plot in Fig. 33.

Change point identification method can be applied to segment the spike train data into several periods, each of which is assumed to be generated by a stable probability distribution. It is reasonable to assume that the inter-arrival time between two consecutive spikes is exponentially distributed, but with different parameters at different stages. Methods of change point identification in exponential distribution can be found in literature, e.g., Ramanayake and Gupta (2002). We can directly use it here. After segmenting spike train data into periods by identified change points, the distribution parameter in each period can be estimated. Thus, the spike train data of a successful trial can be modeled by several change points and the distribution parameter in each period.

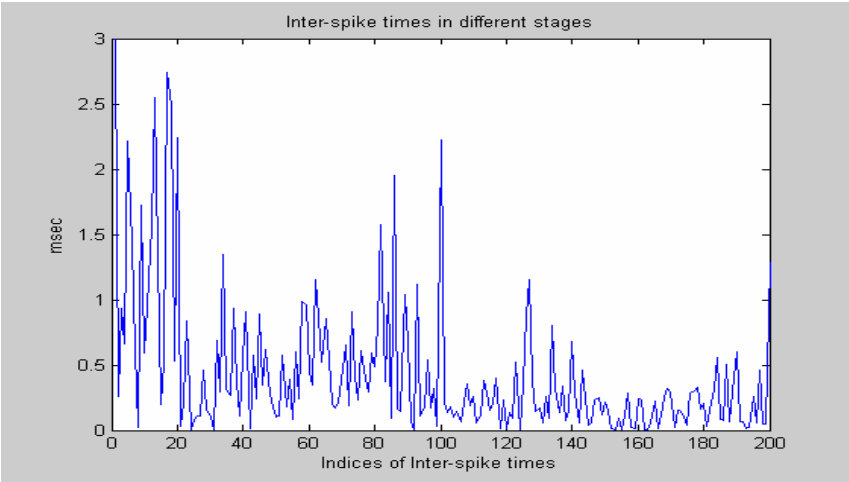


Figure 33. Inter-spike time of a neuron in a successful trial

We can choose only a few model parameters of interests, such as the starting time and the distribution parameter of the most active stage, to represent the spike train data of a neuron in a successful trial. For the same neuron, these selected parameters should have similar values in all of the successful trials. Under a certain experimental condition, the involved neurons should have similar values of these selected parameters. These parameters of these neurons should fall into a cluster (or multiple clusters), and the uninvolved neurons should fall into other clusters.

7.2.1.2 Clustering neurons by their histograms

To transform spike train data into histograms, the time period of task preparing is segmented into a set of contiguous and equal-size bins. The histogram of a neuron is just a set of integer numbers, each of which is the count of spikes in a bin in all the successful trials. So, totally we can have 19 histograms, one for each neuron. Fig. 34 demonstrates the spike train data of four neurons and their histograms.

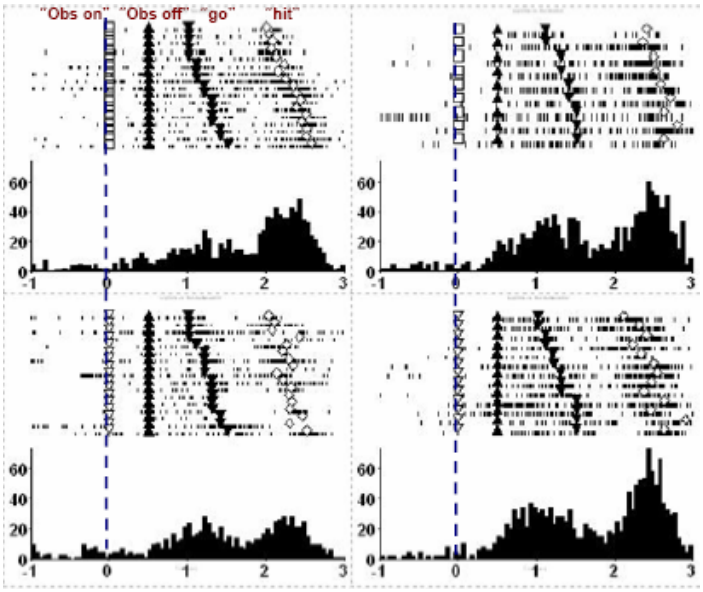


Figure 34. Spike train data of four neurons and histograms

There are several possible methods that can be applied on these 19 histograms to find their different patterns under the k^{th} experimental condition. These methods include: (1)Regression based method, and (2)Fourier or wavelet-transformation based method.

(1) Regression based method uses nonlinear regression to fit each histogram on the same nonlinear model. Each histogram can be represented by a set of regression parameters. In this way, a high-dimension histogram is projected to a lower dimensional coefficient space. Clustering analysis can be applied to study the cluster pattern in the coefficient space. The histograms of those neurons which are involved in the preparing of the action under the experimental condition should have significantly different histogram with those uninvolved neurons. So, the coefficient vectors of those involved neurons should form a cluster, and the other coefficient vectors either appear as outliers, or form another cluster.

(2) Fourier-transformation or wavelet-transformation based method applies

Fourier or wavelet transformation to individual histogram. Filters can be applied on the transformation coefficients to select coefficients of our interest. The reserved coefficients can be clustered. The reserved coefficients of the neurons involved in preparing the action under the experimental condition should form a cluster, and the others should form another cluster or are just outliers.

Principal component analysis (PCA) can be applied on the regression or selected Fourier or wavelet transformation coefficients since these coefficients should be correlated. If we only use 2 or 3 principal components to represent these coefficients, we can plot a 2-D or 3-D scatter plot of the projections of these coefficients. Thus, the cluster pattern in these coefficients can be visually accessed.

7.2.1.3 Studying the dependency among neurons

After we cluster the neurons, the relationships among the neurons involved in preparing the action under a certain experimental condition can be studied. The purpose of studying the relationship among them is to find how the neurons are cooperating and communicating with each other in the task preparing.

The simplest relationship, linear relationship can be studied by the correlations of the histograms of the involved neurons. However, correlations only capture the linear relationship between two histograms. Nonlinear relationships, which are more likely to be the fact among neurons in the brain, can not be captured.

Dependency describes the relationship between two variables more generally, which includes linear and nonlinear relationships. It can be captured by mutual information. The mutual information of two random variables is large if they are

dependent, and small otherwise. A bottom-up hierarchical clustering method can be applied on the histograms of neurons to cluster them. In this hierarchical clustering, the mutual information is used as the measure of similarity. The bottom-up hierarchical clustering works in the following way. In the beginning, we take each neuron as a single cluster. Two neurons with the largest mutual information are combined into one cluster. Then, any two clusters with the largest mutual information are combined into one. The algorithm stops when some stopping criterion is satisfied. In the end, the neurons in the same cluster are considered dependent on each other. Readers are referred to Kojadinovic (2004) for more details about this hierarchical clustering based on mutual information.

7.2.2 Waveform data

Waveform data is also called profiles. We can either model the profiles and analyze the model's coefficients, or treat profiles as points in a high-dimension space, where the dimension is the number of observations we have in a profile. The methods of profile modeling include nonlinear regression, Fourier transformation or wavelet transformation, etc.

Then, profiles can be clustered by clustering their model coefficients or by clustering profiles as points directly. In all successful trials when preparing the same task, spike profiles of a single neuron may show similar cluster patterns, e.g., at a certain stage of the task preparing, the spike profiles have similar shapes, but different shapes at another shape. This may imply that different commands are sent at different stages. The spike profile shapes may change in the same stage but when

preparing different tasks. Finding the difference in the spike profile shapes and mapping it to the difference in the body movement may uncover which spike profile shape corresponds to what command of body movement.

Appendix A. MSPC methods

Commonly used MSPC methods include Hotelling's T^2 in Tracy *et al.* (1992), Mason *et al.* (1995 and 1997) and Mason and Young (1999 and 2000), Multivariate EWMA (MEWMA), as discussed in Lowry *et al.* (1992), Testik *et al.* (2002) and Montgomery (2001), Multivariate CUSUM (MCUSUM) proposed by Crosier (1988), Principal Component Analysis (PCA) in Jackson (1991), Kourti and MacGregor (1996) and Kano *et al.* (2004), and Partial Least Squares (PLS); see Geladi and Kowalski (1986), Kresta *et al.* (1991), Wurl *et al.* (2001) and Xu and Albin (2002).

The above MSPC methods can be classified into two categories: subspace and full-space methods. PCA and PLS are subspace methods because points in the full dimensional space are projected onto a subspace with lower dimension. The remaining methods, Hotelling's T^2, MEWMA and MCUSUM are categorized as full-space methods.

Hotelling's T^2 is a popular MSPC method to monitor a multivariate process. For a p-variable process, the T^2 statistic is defined as:

$$T^2 = (\mathbf{x} - \bar{\mathbf{x}})' \mathbf{S}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \tag{A1}$$

where $\bar{\mathbf{x}}$ and $\mathbf{S}$ are the estimates of the mean vector and variance-covariance matrix of these p variables obtained from baseline data respectively.

Assuming the multivariate normality of process variables while the process is in control, Hotelling's T^2 statistic is proportional to an F distribution. The upper control limit of the T^2 statistic is calculated as

$$T^2_{UCL} = \frac{(N-1)(N+1)p}{N(N-p)} F_\alpha(p, N-p) \tag{A2}$$

where N is the number of observations in baseline data to build the MSPC model, $F_\alpha(p, N-p)$ is the upper 100α percentile of F distribution with degrees of freedom p and N-p.

A Hotelling's T^2 control chart can be built by plotting statistic T_i^2 of vector $\mathbf{x}_i$ vs. the time tag of $\mathbf{x}_i$ or i when we apply it for on-line process monitoring. For a new observation $\mathbf{x}_i$, if its T_i^2 exceeds T_{UCL}^2, we conclude that the current mean of the underlying p-variable process is significantly different from the baseline mean. When we are not confident with the assumption of multinormality of $\mathbf{x}$, we can let T_{UCL}^2 be the 99th or 95th percentile of the T^2's of baseline observations.

Hotelling's T^2 statistic can be decomposed into the sum of p elements, as shown in the following paragraphs. We also show that the in-control space $T^2 \le T_{UCL}^2$ is just a hyper-ellipsoid in the p-dimension space.

Since $\mathbf{S}$ is a positive definitive and symmetric matrix, it can be decomposed as $\mathbf{S} = \mathbf{P}\Lambda\mathbf{P}'$, where $\mathbf{P} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_p]$ is the eigenvector matrix and $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_p)$ is the diagonal eigenvalue matrix of $\mathbf{S}$ respectively. Matrix $\mathbf{S}^{-1}$ has the same eigenvector matrix $\mathbf{P}$ as $\mathbf{S}$, and its eigenvalue matrix is just the inverse of the eigenvalue matrix of $\mathbf{S}$, i.e., $\mathbf{\Lambda}^{-1} = diag(\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_p^{-1})$. So, $\mathbf{S}^{-1} = \mathbf{P}\Lambda^{-1}\mathbf{P}'$

Thus, the T^2 statistic in Eqn. (A1) can be rewritten as:

$$T^2 = (\mathbf{x} - \bar{\mathbf{x}})'\mathbf{P}\Lambda^{-1}\mathbf{P}'(\mathbf{x} - \bar{\mathbf{x}}) \tag{A3}$$

If we define $\mathbf{z} = \mathbf{P}'(\mathbf{x} - \bar{\mathbf{x}})$, which is a new *p-by-1* vector, Eqn. (A3) is equivalent to:

$$T^2 = \mathbf{z}'\Lambda^{-1}\mathbf{z} = \sum_{i=1}^{p} \frac{z_i^2}{\lambda_i} \tag{A4}$$

Since $\mathbf{PP}' = \mathbf{P}'\mathbf{P} = \mathbf{I}$, where $\mathbf{I}$ is a *p-by-p* identity matrix, $\mathbf{z} = \mathbf{P}'(\mathbf{x} - \bar{\mathbf{x}})$ is just projecting the point $\mathbf{x}$ onto a new coordinate system, whose axes are $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_p$ with origin $\bar{\mathbf{x}}$. So, the in-control area defined by $T^2 = \sum_{i=1}^{p} \frac{z_i^2}{\lambda_i} \le T_{UCL}^2$ is just a hyper-ellipsoid in this new coordinate system.

Fig. 35 illustrates the Hotelling's T^2 more clearly with a bi-variable example, i.e., p=2. In Fig. 35, the circles are baseline observations and the dashed ellipse is the edge of the in control area. This ellipse is centered at the baseline average $\bar{\mathbf{x}}$. The directions of the two new axes are $\mathbf{t}_1$ and $\mathbf{t}_2$, which are the two eigenvectors of the **S** matrix derived from the baseline data.

In Fig. 35, note two points A and B. Point A is inside the ellipse, $T_A^2 < T_{UCL}^2$ and is considered an in-control point. Point B is outside of the ellipse, $T_B^2 > T_{UCL}^2$ and is considered an out-of-control point.

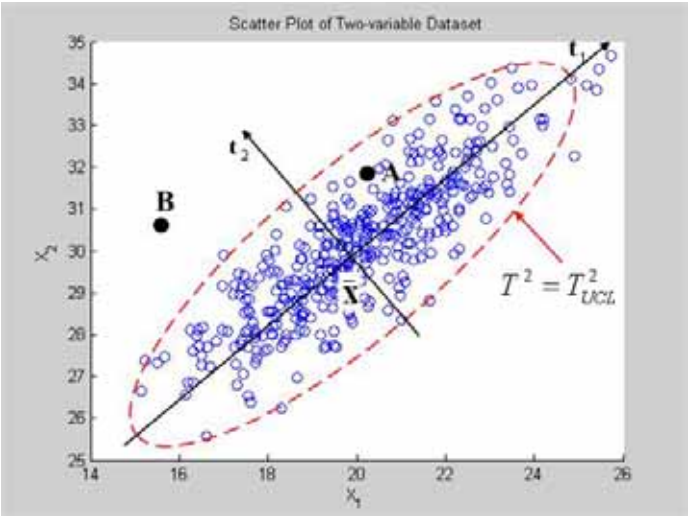


Figure 35 Hotelling's T^2 in a bi-variable example

Before introducing MEWMA, let's first introduce its univariate version, exponentially weighted moving average (EWMA) control chart. EWMA is devised to

detect subtle shifts of the mean of a single variable from the target μ_0. The EWMA statistic is defined as: $z_i = \lambda x_i + (1-\lambda) z_{i-1}$, $i = 1, 2, 3, ...; z_0 = \mu_0; 0 < \lambda < 1$. Sometimes the average of baseline data is used as the starting value of the EWMA, i.e., $z_0 = \overline{x}$.

When we apply the EWMA control chart on on-line monitoring, we plot z_i versus the sample number *i* (or time). The center line and the control limits for the z_i are as follows, $UCL = \mu_0 + L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]}$; $Center\ line = \mu_0$;

$LCL = \mu_0 - L\sigma\sqrt{\frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]}$, where σ is the standard deviation of the variable. For details of EWMA control chart and how to select the proper values of *L* and λ, please refer to Montgomery (2001).

Multivariate EWMA (MEWMA) monitors the small and moderate shifts of the mean vector of *p* variables. Similar with EWMA, a new vector $\mathbf{z}_i$ is defined as $\mathbf{z}_i = \lambda(\mathbf{x}_i - \boldsymbol{\mu}_0) + (1-\lambda)\mathbf{z}_{i-1}, i = 1, 2, 3, ...; \mathbf{z}_0 = \mathbf{0}, 0 < \lambda \leq 1$. Then, for vector $\mathbf{z}_i$, a Hotelling's T^2 statistic can be calculated as $T_i^2 = \mathbf{z}_i' \boldsymbol{\Sigma}_{\mathbf{z}_i}^{-1} \mathbf{z}_i$, where

$$\boldsymbol{\Sigma}_{\mathbf{z}_i} = \frac{\lambda}{2-\lambda}[1-(1-\lambda)^{2i}]\boldsymbol{\Sigma} \qquad \text{(A5)}$$

Parameters $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}$ are the mean vector and variance-covariance matrix of variable vector $\mathbf{x}$ when the process is in control. When these two parameters are unknown, they can be substituted by the estimates from baseline data. The MEWMA control chart is constructed by plotting T_i^2 versus *i*. The control limits for the MEWMA control chart can be retrieved from tables given in Montgomery (2001).

Cumulative Sum (CUSUM) control chart is another method to detect small

shifts in the mean value of a single variable. Multivariate CUSUM is the application of CUSUM in multivariate cases. In Montgomery (2001), to monitor the process by individual observations, CUSUM statistics are defined as:

$$\begin{aligned} C_i^+ &= \max[0, x_i - (\mu_0 + K) + C_{i-1}^+] \\ C_i^- &= \max[0, (\mu_0 + K) - x_i + C_{i-1}^-] \end{aligned} \tag{A6}$$

where $C_0^+ = C_0^- = 0$. In Eqn. (A6), $K = \frac{|\mu_1 - \mu_0|}{2}$, where μ_0 is the target process mean and μ_1 is the shifted mean that we want to detect it as fast as possible. If C_i^+ or C_i^- of observation x_i exceeds decision interval *H*, the process is considered out-of-control. A reasonable value for *H* is five times the process standard deviation.

Crosier (1988) proposes two schemes of multivariate CUSUM (MCUSUM): CUSUM of *T* statistic (COT) and regular MCUSUM. Method COT just applies the univariate CUSUM on Hotelling's *T* statistic. COT statistic is calculated as:

$$S_i = \max(0, S_{i-1} + T_i - k), \; i = 1, 2, 3, \ldots \tag{A7}$$

where $S_0 = 0$ and T_i is the positive square root of T_i^2 as calculated in Eqn. (A1) for observation $\mathbf{x}_i$. If $S_i > h$, the process is considered out of control. Parameter *k* is selected such that the mean vector shift of our interest can be detected as fast as possible. For instance, if we want to detect mean vector shift in the amount of one standard deviation, $k = \sqrt{p}$, where *p* is number of process variables. Parameter *h* is selected such that the COT chart has the desired false alarm rate.

The regular MCUSUM scheme is expressed as follows: Let

$C_i = [(\mathbf{s}_{i-1} + \mathbf{x}_i - \boldsymbol{\mu}_0)' \boldsymbol{\Sigma}^{-1} (\mathbf{s}_{i-1} + \mathbf{x}_i - \boldsymbol{\mu}_0)]^{1/2}$, then

$$\mathbf{s}_i = \mathbf{0} \qquad \text{if } C_i \le k$$
$$\mathbf{s}_i = (\mathbf{s}_{i-1} + \mathbf{x}_i - \boldsymbol{\mu}_0)(1 - k / C_i) \quad \text{if } C_i > k$$

where $\mathbf{s}_0 = \mathbf{0}$, k>0, $\boldsymbol{\mu}_0$ is the target mean vector, and $\boldsymbol{\Sigma}$ is the variance-covariance matrix of process variables. Let $Y_i = [\mathbf{s}_i' \boldsymbol{\Sigma}^{-1} \mathbf{s}_i]^{1/2}$. The regular MCUSUM control chart signals when $Y_i > h$. Selections of parameters k and h are similar with the COT scheme such that the control chart has the desired performance to detect the interested mean shift.

The Hotelling's T^2, MEWMA and MCUSUM methods may have serious problems when the process variables are highly correlated. Many highly correlated variables lead to the $\mathbf{S}$ matrix in Eqn. (A1) and the $\boldsymbol{\Sigma}$ matrix in Eqn. (A5) being near singular. This implies that some λ_i's in Eqn. (A4) are very small or near zero. It makes the Hotelling's T^2 very sensitive to deviations of z_i's in Eqn. A(4) corresponding to these small λ_i's, i.e., even a very small deviation in these z_i's can make the statistic of Hotelling's T^2, MEWMA or MCUSUM exceed the control limit. So, the false alarm rate will be very high when these methods are applied to monitor highly correlated multivariate process.

Subspace methods, such as principal component analysis (PCA) and partial least squares (PLS), solve this problem by using only a few orthogonal latent variables. Latent variables are linear combinations of original variables and are independent to each other. When variables are highly correlated, only a few latent variables can account for most of the variance of variables in the original full space. The space spanned by the latent variables is called a subspace. The variance not accounted by the few latent variables is considered noise.

PCA only concerns **X** variables and constructs a few latent variables to capture the major variance in **X** variables. PLS captures the information contained in **X** matrix (usually the process variables) that accounts for the major variation in the **Y** matrix (product variables) with only a few latent variables.

The baseline dataset of PLS consists of both process observations **X** *(n-by-p)* and corresponding product observations **Y**(*n-by-m)*. Each row of **X** represents an observation of *p* process variables, and each row of **Y** represents an observation of *m* product variables.

Before building PCA and PLS models, to eliminate the effects of different scales of different variables, each variable in the baseline dataset is standardized with mean 0 and variance 1.

In PCA, we denote the standardized vector of process variables by $\mathbf{x}$. We project vector $\mathbf{x}$ in a *p*-dimension space into an *A*-dimension subspace (A<<*p*) by:

$$\mathbf{z} = \mathbf{P}_A^{'}\mathbf{x} \tag{A8}$$

where $\mathbf{P}_A = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_A]$, $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_A$ are A eigenvectors of variance-covariance matrix $\mathbf{S}$ corresponding to its A largest eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_A \geq \lambda_{A+1} \geq ... \geq \lambda_p$. Eqn. (A8) is equivalent to project a point $\mathbf{x}$ onto a subspace spanned by $\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_A$. So, $\mathbf{z}$ is an *A*-by-1 vector. Usually *A* is much smaller than *p*, which implies that we project a vector in high dimensional full space into a subspace with much smaller dimensions. The projections of $\mathbf{x}$ on the remaining dimensions defined by $\mathbf{t}_{A+1}, \mathbf{t}_{A+2}, ..., \mathbf{t}_p$ are usually considered noise.

We can transform the projected point $\mathbf{z}$ in the *A*-dimension subspace back to

the full space by:

$$\hat{\mathbf{x}} = \mathbf{P}_A \mathbf{z} \tag{A9}$$

The squared prediction error (SPE) gives a measure of how close the observation $\mathbf{x}$ is to its projection $\hat{\mathbf{x}}$ in the full space:

$$SPE_{\mathbf{x}} = \sum_{i=1}^{p} (x_i - \hat{x}_i)^2 \tag{A10}$$

Another measure, T^2, is just the Hotelling's T^2 of the projected point in the subspace. We can prove that in the A-dimension space, the variance-covariance matrix of the projections of the baseline data points is $\mathbf{S}_A = diag(\lambda_1, \lambda_2, ..., \lambda_A)$. Thus the T^2 for the projected point of $\mathbf{x}$ is just:

$$T^2 = \sum_{i=1}^{A} \frac{z_i^2}{\lambda_i} \tag{A11}$$

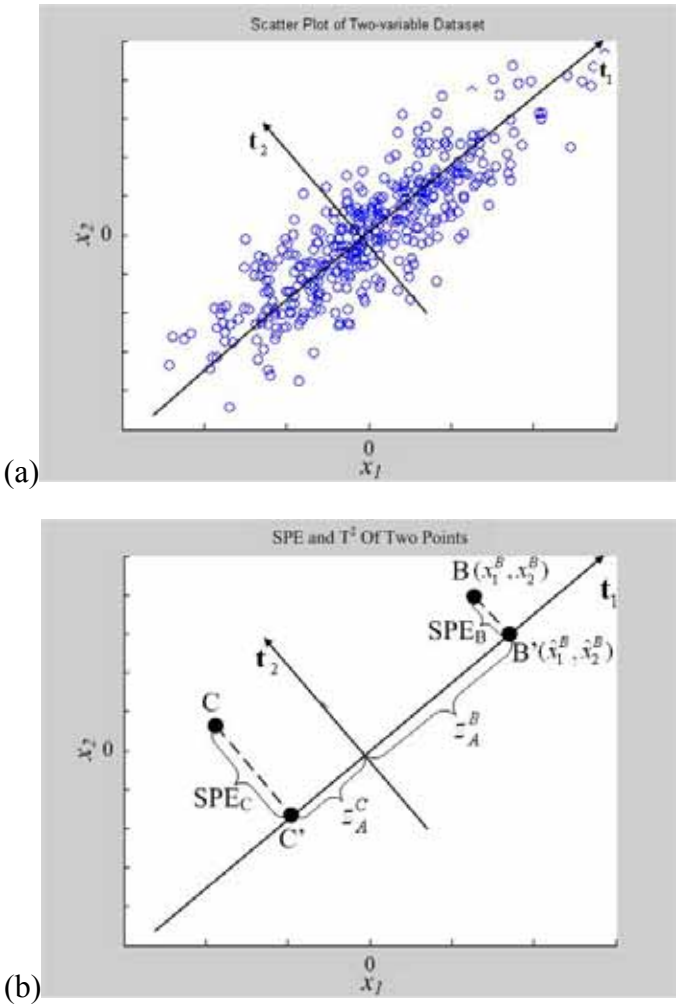


Figure 36. Illustration of PCA, SPE and T^2

Fig. 36 illustrates the PCA, SPE and T^2 in a more straightforward way. In Fig. 36(a), the circles are baseline data points. We can see that these points are mainly varying along the direction defined by $\mathbf{t}_1$. If we only choose one latent variable, A=1. The variance along direction $\mathbf{t}_2$ can be considered noise. Point B in Fig. 36(b) is projected onto the one-dimension subspace whose coordinate is defined by $\mathbf{t}_1$. Point B' is the projected point. The location of B' in the one-dimension space is z_A^B. If we transform the coordinate of B' back to the original 2-dimension space, its coordinate is $[\hat{x}_1^B, \hat{x}_2^B]$, just as shown in Fig. 36(b). The SPE of point B equals

$(x_1^B - \hat{x}_1^B)^2 + (x_2^B - \hat{x}_2^B)^2$ and the T^2 equals $\frac{(z_A^B)^2}{\lambda_1}$.

Similarly, we can calculate the SPE and T^2 values of point C. We can see that point C has a higher SPE but a smaller T^2 than B. SPE measures the similarity of the relationships among variables to the relationships among variables in baseline data. In Fig. 36(a), the relationship between these two variables in baseline data is captured by the first principal component $\mathbf{t}_1$, which means all the points should be very close to this line. In Fig. 36(b), point B has a smaller SPE than point C, because point B is closer to line $\mathbf{t}_1$ than point C. So, the relationship between these two variables of point B is more similar to the baseline data than point C. Statistic T^2 captures how far the projected point to the origin of the *A*-dimension space.

When we apply PCA for MSPC, we first determine the definition of the subspace, i.e., the value of *A*, $(\lambda_1, \lambda_2, ..., \lambda_A)$ and $\mathbf{P}_A = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_A]$. Then, we can have SPE and T^2 charts for on-line monitoring. The process is considered out-of-control if either SPE or T^2 statistic of an observation exceeds its corresponding control limit.

For details of how to determine the value of *A* and how to calculate the upper control limit of the SPE chart, please refer to Jackson (1991), where he denotes the SPE statistic by Q-statistic. The upper control limit of T^2 chart can be calculated similarly as Eqn. (A2), just substituting *p* with *A*, i.e., $T_{UCL}^2 = \frac{(N-1)(N+1)A}{N(N-A)} F_\alpha(A, N-A)$; see Jiji *et al.* (2003).

PLS is another subspace method. For any process observation

$\mathbf{x} = (x_1, x_2, ..., x_p)'$, PLS constructs A components, $(t_1, t_2, ..., t_A)'$, where t_i is a linear combination of x_j's, j=1,2,…, p. The number of PLS components, A, is usually much smaller than the number of process variables, p.

The PLS components are calculated sequentially from baseline data. The first PLS component $t_1 = w_{11}x_1 + w_{12}x_2 + \cdots + w_{1p}x_p = \mathbf{w}_1'\mathbf{x}$, where $\mathbf{w}_1 = (w_{11}, w_{12}, ..., w_{1p})'$, is calculated by maximizing the covariance between the linear combination of x_j's and the linear combination of y_j's. Using the baseline ($\mathbf{X}$,$\mathbf{Y}$), the covariance maximization problem can be written as:

$$\begin{aligned} &Max\ Cov^2(\mathbf{Xw}, \mathbf{Yc}) \\ &s.t.\ \|\mathbf{w}\| = \sqrt{\mathbf{w}'\mathbf{w}} = 1,\ \|\mathbf{c}\| = \sqrt{\mathbf{c}'\mathbf{c}} = 1 \end{aligned}$$

The optimal solution $\mathbf{w}_1$ is used for the weighting vector for the first PLS component.

Let $\mathbf{t}_1$ be a vector containing the first PLS component for all process observations in the baseline $\mathbf{X}$, or $\mathbf{t}_1 = \mathbf{Xw}_1$. Then $\mathbf{t}_1$ can be used to predict $\mathbf{X}$ and $\mathbf{Y}$, or $\hat{\mathbf{X}}_1 = \mathbf{t}_1\mathbf{p}_1'$ and $\hat{\mathbf{Y}}_1 = \mathbf{t}_1\mathbf{q}_1'$, where $\mathbf{p}_1 = (p_{11}, p_{12}, ..., p_{1p})'$ contains the regression coefficients of the columns of $\mathbf{X}$ regressed on $\mathbf{t}_1$, and $\mathbf{q}_1 = (q_{11}, q_{12}, ..., q_{1m})'$ contains the regression coefficients of the columns of Y regressed on $\mathbf{t}_1$. Let $\mathbf{X}_2 = \mathbf{X} - \mathbf{t}_1\mathbf{p}_1'$ and $\mathbf{Y}_2 = \mathbf{Y} - \mathbf{t}_1\mathbf{q}_1'$. The residual matrix $\mathbf{X}_2$ and $\mathbf{Y}_2$ comprise the information contained in $\mathbf{X}$ and $\mathbf{Y}$ unrelated to the first component $\mathbf{t}_1$.

Similarly the weighting vector $\mathbf{w}_2$ for the second PLS component can be computed by maximizing the covariance between $\mathbf{X}_2\mathbf{w}$ and $\mathbf{Y}_2\mathbf{c}$. Let $\mathbf{t}_2 = \mathbf{X}_2\mathbf{w}_2$, then $\mathbf{p}_2$ and $\mathbf{q}_2$ can be obtained by regressing the columns of $\mathbf{X}_2$ and $\mathbf{Y}_2$ on $\mathbf{t}_2$.

The residual matrices in this iteration are $\mathbf{X}_3 = \mathbf{X}_2 - \mathbf{t}_2\mathbf{p}_2'$ and $\mathbf{Y}_3 = \mathbf{Y}_2 - \mathbf{t}_2\mathbf{q}_2'$. The matrices $\mathbf{X}_3$ and $\mathbf{Y}_3$ comprise the information contained in *X* and *Y* but unrelated to both $\mathbf{t}_1$ and $\mathbf{t}_2$.

Proceeding in this way, the vectors $\mathbf{w}_i$, $\mathbf{p}_i$ and $\mathbf{q}_i$ are obtained from the baseline, the PLS components for any new process observation $\mathbf{x} = (x_1, x_2, ..., x_p)'$ can be sequentially constructed in the following manner: $t_1 = \mathbf{w}_1'\mathbf{x}$, $\mathbf{x}_2 = \mathbf{x} - t_1\mathbf{p}_1$, $t_2 = \mathbf{w}_2'\mathbf{x}_2$, $\mathbf{x}_3 = \mathbf{x}_2 - t_2\mathbf{p}_2$, $t_3 = \mathbf{w}_3'\mathbf{x}_3, ..., t_A = \mathbf{w}_A'\mathbf{x}_A$. The prediction of $\mathbf{x}$ by the PLS model can be written as $\hat{\mathbf{x}} = t_1\mathbf{p}_1 + t_2\mathbf{p}_2 + ... + t_A\mathbf{p}_A$. Hence $\mathbf{x} - \hat{\mathbf{x}}$ represents the information contained in $\mathbf{x}$ but not captured by t_i's.

Similar with PCA, we use SPE to capture the similarity of variables' relationships of new observation $\mathbf{x}_i$ to the variable relationships of baseline data. The SPE is calculated as SPE($\mathbf{x}$)= $\left\|\mathbf{x} - \hat{\mathbf{x}}\right\|^2 = \sum_{j=1}^{p}(x_j - \hat{x}_j)^2$. In PLS, if the SPE of an observation is out of the control limit, we can infer that the product quality will be bad with high probability.

For details of PLS, please refer to Geladi and Kowalski (1986), Kresta *et al.* (1991), Wurl *et al.* (2001) and Xu and Albin (2002).

Appendix B. Approximate the expectation and variance of estimator $\hat{\sigma}_s^2$

We prove that if there are no outlier profiles,

$$\bar{\hat{\sigma}}_s^2 = \frac{1}{N(N-1)/2}\sum_{i<k}^{N}\hat{\sigma}_{(i,k)s}^2 \qquad \text{(B1)}$$

is an unbiased and asymptotically effective estimator of σ_s^2.

We first prove that $\bar{\hat{\sigma}}_s^2$ is unbiased. From Eqn. (12), the difference between

two profiles *i* and *k* is $\delta_{(i,k)j} = \varepsilon_{ij} - \varepsilon_{kj}$, where ε_{ij} and $\varepsilon_{kj} \sim iid\ N(0,\ \sigma_s^2)$. Therefore $\delta_{(i,k)j} \sim N(0, 2\sigma_s^2)$ and the expectation of $\hat{\sigma}^2_{(i,k)s}$ in Eqn. (13) is σ_s^2. From Eqn. (B1), $E(\hat{\sigma}_s^2)=\sigma_s^2$, i.e., the estimator in Eqn. (B1) is unbiased.

Estimator $\overline{\hat{\sigma}}_s^2$ is called an effective estimator of σ_s^2 if it is unbiased and its variance satisfies the following equation:

$$Var(\overline{\hat{\sigma}}_s^2) = \frac{1}{W \cdot I(\sigma_s^2)} \tag{B2}$$

where W is the sample size and $I(\sigma_s^2)$ is the Fisher information. The Fisher information about parameter σ_s^2 of a normal distribution with mean 0 and variance σ_s^2 is $I(\sigma_s^2) = \frac{1}{2\sigma_s^4}$; see DeGroot (1986), pp.420-424. In profile baseline data, $W=NM$.

For $\overline{\hat{\sigma}}_s^2$ to be an asymptotically effective estimator, we need to prove:

$$\lim_{N\to\infty} Var(\overline{\hat{\sigma}}_s^2) = \frac{2}{NM}\sigma_s^4 \tag{B3}$$

Since $\overline{\hat{\sigma}}_s^2 = \frac{1}{N(N-1)/2}\sum_{i<k}^{N}\hat{\sigma}^2_{(i,k)s} = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{N-1}\sum_{i=1}^{N}(y_{ij} - \overline{y}_{\cdot j})^2$ and y_{ij}'s are independent at different *j*, the variance of random variable $\overline{\hat{\sigma}}_s^2$ is:

$$Var(\overline{\hat{\sigma}}_s^2) = \frac{1}{(M(N-1))^2}\sum_{j=1}^{M} Var(\sum_{i=1}^{N}(y_{ij} - \overline{y}_{\cdot j})^2) \tag{B4}$$

We know that $\frac{\sum_{i=1}^{N}(y_{ij} - \overline{y}_{\cdot j})^2}{\sigma^2}$ follows χ^2 distribution with *N*-1 degrees of freedom.

So $Var(\sum_{i=1}^{N}(y_{ij} - \overline{y}_{\cdot j})^2) = 2(N-1)\sigma^4$. Consequently, Eqn. (B4) is equivalent to:

$$Var(\overline{\hat{\sigma}}_s^2) = \frac{2(N-1)}{(M(N-1))^2}\sum_{j=1}^{M}\sigma^4 = \frac{2\sigma^4}{M(N-1)} \tag{B5}$$

Take limits of both sides to obtain:

$$\lim_{N\to\infty} Var(\bar{\hat{\sigma}}_s^2) = \frac{2}{NM}\sigma_s^4 \tag{B6}$$

Thus, we have proved that the estimator in Eqn. (B1) is an asymptotically effective estimator of σ_s^2. □

Appendix C. Proof of proposition 5.1

Proof:

Suppose λ^* is a contributing eigenvalue. Eigenvector $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2]^T$ is the eigenvector corresponding to λ^*, where $\mathbf{V}_1$ is a *k-by-1* vector and $\mathbf{V}_2$ is a *(p-k)-by-1* vector.

Since $\mathbf{V}$ is the eigenvector corresponding to λ^* of matrix $\mathbf{D} = \boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1}$, we have $\mathbf{DV} = \lambda^*\mathbf{V}$. It is equivalent to $(\boldsymbol{\Sigma}_0 + \boldsymbol{\Delta}_\Sigma)^* \boldsymbol{\Sigma}_0^{-1}\mathbf{V} = \lambda^*\mathbf{V}$. So

$$(\mathbf{I} + \boldsymbol{\Delta}_\Sigma\boldsymbol{\Sigma}_0^{-1})\mathbf{V} = \lambda^*\mathbf{V} \tag{C1}$$

where **I** is a p-by-p identity matrix.

We can also rewrite matrix $\boldsymbol{\Sigma}_0^{-1}$ as blocks $\boldsymbol{\Sigma}_0^{-1} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$, where $\mathbf{A}_{11}$, $\mathbf{A}_{12}$, $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ are *k-by-k*, *k-by-(p-k)*, *(p-k)-by-k* and *(p-k)-by-(p-k)* matrices, respectively. Then

$$\boldsymbol{\Delta}_\Sigma\boldsymbol{\Sigma}_0^{-1} = \begin{bmatrix} \boldsymbol{\Delta}_{11} & \mathbf{0}_{12} \\ \mathbf{0}_{21} & \mathbf{0}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}_{11}\mathbf{A}_{11} & \boldsymbol{\Delta}_{11}\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{C2}$$

From Eqns (C1) and (C2), we get

$$\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Delta}_{11}\mathbf{A}_{11}\mathbf{V}_1 + \boldsymbol{\Delta}_{11}\mathbf{A}_{12}\mathbf{V}_2 \\ \mathbf{0} \end{bmatrix} = \lambda^*\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \tag{C3}$$

Thus, we have $\begin{bmatrix} \boldsymbol{\Delta}_{11}\mathbf{A}_{11}\mathbf{V}_1 + \boldsymbol{\Delta}_{11}\mathbf{A}_{12}\mathbf{V}_2 \\ \mathbf{0} \end{bmatrix} = (\lambda^* - 1)\begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix}$. Since $\lambda^* \neq 1$, in order for $(\lambda^* - 1)\mathbf{V}_2 = \mathbf{0}$, $\mathbf{V}_2 = \mathbf{0}$. □

References

1. Acosta, C. A. and Pignatiello, J. J. (2000), "Monitoring Process Dispersion Without Subgrouping", *Journal of Quality Technology*, Vol. 32, No. 2, pp89-102.
2. Albazzaz, H., Wang, X.Z. and Marhoon, F. (2005), "Multidimensional visualization for process historical data analysis: a comparative study with multivariate statistical process control". *Journal of Process Control*, Vol. 15, pp285-294.
3. Albin, S.L., Kang, L. and Shea, Gerald (1997), "An X and EWMA Chart for Individual Observations", *Journal of Quality Technology*, Vol. 29 No. 1.
4. Besterfield, D.H., Besterfield-Michna, C., Besterfield, G.H. and Besterfield-Sacre, Mary (1999), "Total Quality Management", Prentice-Hall, Inc.
5. Bradley, P.S., Fayyad, U.M. and Mangasarian, O.L. (1999), "Mathematical Programming for Data Mining: Formulations and Challenges", *INFORMS Journal on Computing*, Vol. 11, No. 3.
6. Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. (2000), "LOF: Identifying Density-Based Local Outliers", MOD 2000, Dallas, TX USA, pp93-104.
7. Chu, Y.-H., Qin, S. J. and Han, C. (2004), "Fault Detection and Operation Mode Identification Based on Pattern Classification with Variable Selection", *Industrial & Engineering Chemistry Research* 43, pp 1701-1710.
8. Cook, D. F., Zobel, C. W. and Nottingham, Q. J. (2001), "Utilization of neural networks for the recognition of variance shifts in correlated manufacturing process parameters", *International Journal of Production Research*, Vol. 39 No. 17, pp3881-3887.
9. Cinque, L., Foresti, G. and Lombardi, L. (2004), "A clustering fuzzy approach for image segmentation", *Pattern Recognition*, Vol. 37, pp 1797-1807.
10. Costa, J. A. F. and Netto, M. L. D. A. (1999), "Estimating the number of clusters in multivariate data by self-organizing maps", *International Journal of Neural Systems*, Vol. 9, No. 3, pp 195-202.
11. Crosier, R. B. (1988), "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes", *Technometrics*, Vol. 30, No. 3.
12. Daszykowski, M., Walczak, B. and Massart, D.L. (2001), "Looking for natural patterns in data Part 1. Density-based approach", *Chemometrics and Intelligent Laboratory System*, Vol.56, pp83-92.
13. DeGroot, M. H. (1986), "Probability and Statistics", Addison-Wesley Pub. Co..
14. Ertoz, L., Steinbach, M. and Kumar, V. (2003), "Finding clusters of different sizes, shapes and densities in noisy high dimensional data", *SIAM International Conference on Data Mining, San Francisco, California, USA*, May 2003.
15. Fraley, C. and Raftery, A. (1998), "How many clusters? Which clustering method? Answers via model-based cluster analysis", *The Computer Journal*, Vol. 42, No. 8.
16. Garcia-Escudero, L.A. and Gordaliza, A. (2005), "A Proposal for Robust Curve Clustering", Journal of Classification, Vol. 22, No. 2.

17. Geladi, P. and Kowalski, B.R. (1986), "PLS tutorial", *Anal. Chim. Acta*, Vol. 185, pp1-17.
18. Guh, R. S. (2005). "Real-time pattern recognition in statistical process control: a hybrid neural network/decision tree-based approach", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 219, pp283-298.
19. Guo, Y. and Dooley, K. J. (1992), "Identification of change structure in statistical process control", *International Journal of Production Research*, Vol. 30 No. 7, pp1655-1669
20. Guo, Y. and Dooley, K. J. (1995), "Distinguishing between mean, variance and autocorrelation changes in statistical quality control", *International Journal of Production Research*. Vol. 33 No. 2, pp497-510
21. Han, J. and Kamber, M. (2001), *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, CA
22. Han, K.F. and Baker, David (1995), "Recurring Local Sequence Motifs in Proteins", *Journal of Molecular Biology*. Vol. 251, pp176-187
23. Hawkins, D. M. and Zamba, K. D. (2005). "A Change-Point Model for a Shift in Variance", *Journal of Quality Technology*, Vol. 37, No. 1, pp21-31.
24. Herberts, T. and Jensen, U. (2004). "Optimal Detection of a Change Point in a Poisson Process for Different Observation Schemes", *Scandinavian Journal of Statistics*, Vol. 31, pp347-366.
25. Herbin, M., Bonnet, N. and Vautrot, P. (2001), "Estimation of the number of clusters and influence zones", *Pattern Recognition Letters*, Vol. 22, pp 1557-1568.
26. Ho, E. S. and Chang, S. I. (1999), "An integrated neural network approach for simultaneous monitoring of process mean and variance shifts-a comparative study", *International Journal of Production Research*, Vo. 37, No. 8, pp1881-1901.
27. Hwang, D.-H. and Han, C. (1999), "Real-time monitoring for a process with multiple operating modes", *Control Engineering Practice* 7, pp 891-902.
28. Hwarng, H. B. (2004), "Detecting process mean shift in the presence of autocorrelation: a neural-network based monitoring scheme", *International Journal of Production Research*, Vol. 42 No. 3, pp573-595
29. Hwarng, H. B. (2005), "Simultaneous identification of mean shift and correlation change in AR(1) processes", *International Journal of Production Research*, Vol. 43 No. 9, pp1761-1783
30. Jackson, J.E. (1991), "A User's Guide to Principal Components", J. Wiley and Sons, New York.
31. Jeong, M.K., Lu, J.-C and Wang, N. (2006), "Wavelet-based SPC procedure for complicated functional data". *International Journal of Production Research*, Vol. 44, No. 4, pp729-744.
32. Jiji, R.D., Hammond, M.H., Williams, F.W. and Rose-Pehrsson, S.L. (2003), "Multivariate statistical process control for continuous monitoring of networked

early warning fire detection (EWFD) systems". *Sensors and Actuators B*, Vol. 93, pp107-116.
33. Jin, J. and Shi, J. (2001), "Automatic feature extraction of waveform signals for in-process diagnostic performance improvement", *Journal of Intelligent Manufacturing 12*, pp 257-268.
34. Kang, L. and Albin, S. L. (2000), "On-line Monitoring When the Process Yields a Linear Profile", *Journal of Quality Technology*, Vol. 32, No.4, pp418-426.
35. Kano, M., Hasebe, S., Hashimoto, I. and Ohno, H. (2004), "Evolution of multivariate statistical process control: application of independent component analysis and external analysis", *Computers and Chemical engineering* 28, pp 1157-1166.
36. Kim, K., Mahmoud, M. A. and Woodall, W. H. (2003), "On The Monitoring of Linear Profiles", *Journal of Quality Technology*, Vol. 35, No. 3, pp317-328.
37. Kojadinovic, I. (2004), "Agglomerative hierarchical clustering of continuous variables based on mutual information", *Computational Statistics & Data Analysis*, Vol. 46, No. 2, pp 269-294
38. Kothari, R. and Pitts, D. (1999), "On finding the number of clusters", *Pattern Recognition Letters*, Vol.20, pp 405-416.
39. Kourti, T. and Macgregor, J.F. (1996), "Multivariate SPC Methods for Process and Product Monitoring", *Journal of Quality Technology*, Vol. 28, No. 4.
40. Kresta, J.V., MacGregor, J.F. and Marlin, T.E. (1991), "Multivariate Statistical Monitoring of Process Operating Performance", *Canadian Journal of Chemical Engineering*, Vol. 69, pp35-47
41. Lada, E.K., Lu J.C. and Wilson J.R. (2002), "A Wavelet-Based Procedure for Process Fault Detection", *IEEE Transations On Semiconductor Manufacturing*, Vol. 15, No. 1, 2002.
42. Loschi, R. H. and Cruz, F. R. B. (2005). "Bayesian Identification of Multiple Change Points in Poisson Data", *Advances in Complex Systems*, Vol. 8, No. 4, pp465-482.
43. Lowry, C. A., Woodall, W. H., Champ, C. W. and Rigdon, S.E. (1992), "A Multivariate Exponentially Weighted Moving Average", *Technometrics*, Vol. 34, No. 1.
44. Mahmoud, M. A. and Woodall, W. H. (2004), "Phase I Analysis of Linear Profiles With Calibration Applications", *Technometrics*, Vol. 46, No. 4, pp380-391.
45. Mason, R.L., Tracy, N.D. and Young, J.C. (1995), "Decomposition of T^2 multivariate control chart interpretation", *Journal of Quality Technology*, Vol. 27, No. 2.
46. Mason, R.L., Tracy, N.D. and Young, J.C. (1997), "A Practical Approach for Interpreting Multivariate Control Chart Signals", *Journal of Quality Technology*, Vol.29, No. 4.
47. Mason, R.L. and Young, J.C. (1999), "Improving the Sensitivity of the T^2 Statistic in Multivariate Process Control", *Journal of Quality Technology*, Vol. 31, No. 2.

48. Mason, R. L. and Young, J. C. (2000), "Interpretive Features of a T^2 chart in Multivariate SPC", *Quality Progress*, Vol. 33, No. 4.
49. Montgomery, D. C. (2001), "Introduction to Statistical Quality Control (Fourth Edition)", John Wiley & Sons, Inc., New York
50. Montgomery, D. C. and Runger, G. (2006). "Applied Statistics and Probability for Engineers", John Wiley & Sons Inc.
51. Nakamura, E. and Kehtarnavaz, N. (1998), "Determining number of clusters and prototype locations via multi-scale clustering", *Pattern Recognition Letters*, Vol. 19, No. 14, pp 1265-1283.
52. Ramanayake, A. and Gupta, A. K. (2002). "Change Points with Linear Trend Followed by Abrupt Change for the Exponential Distribution", *Journal of Statistical Computation and Simulation*, Vol. 74, No. 4, pp263-278.
53. Rencher, A. C. (2002), "Methods of Multivariate Analysis, 2nd edition", J. Wiley, New York.
54. Son, Y. S. and Kim, S. W. (2005). "Bayesian single change point detection in a sequence of multivariate normal observations", *Statistics*, Vol. 39, No. 5, pp373-387.
55. Stover, F.S. and Brill, R.V. (1998), "Statistical Quality Control Applied to Ion Chromatography Calibrations", *Journal of Chromatography A,* Vol. 804, pp37-43.
56. Su, M. and Liu, Y. (2005), "A new approach to clustering data with arbitrary shapes", *Pattern Recognition*, accepted 22 April 2005.
57. Sullivan, J. H. (2002). "Detection of Multiple Change Points from Clustering Individual Observations", *Journal of Quality Technology*, Vol. 34, No. 4, pp371-383.
58. Sullivan, J. H. and Woodall, W. H. (1996). "A Control Chart for Preliminary Analysis of Individual Observations", *Journal of Quality Technology*, Vol. 28, No. 3, pp265-278.
59. Sullivan, J. H. and Woodall, W. H. (2000). "Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations", *IIE Transactions*, Vol. 32, pp537-549
60. Testik, M. C., Runger, G. C. and Borror, C. M. (2002), "Robustness Properties of Multivariate EWMA Control Charts", *Quality and Reliability Engineering International*, 2003, 19.
61. Tracy, N.D., Young, J.C. and Mason, R.L. (1992), "Multivariate control charts for individual observations", *Journal of Quality Technology*, Vol. 24, No. 2, pp88-95.
62. Walker, E. and Wright, S. P. (2002), "Comparing Curves Using Additive Models", *Journal of Quality Technology*, Vol. 34, No. 1, pp118-129.
63. Wang, W.-J., Tan, Y.-X., Jiang, J.-H., Lu, J.-Z., Shen, G.-L. and Yu, R.-Q. (2004), "Clustering based on kernel density estimation: nearest local maximum searching algorithm", *Chemometrics and intelligent laboratory systems*, Vol. 72, pp 1-8.
64. Williams, J. D., Woodall, W. H. and Birch, J. B. (2003), "Phase I Monitoring of

Nonlinear Profiles", *2003 Quality & Productivity Research Conference*, Yorktown Heights, NY.

65. Woodall, W. H. (2000). "Controversies and Contradictions in Statistical Process Control", *Journal of Quality Technology*, Vol. 32, No. 4, pp341-350.
66. Woodall, W. H., Spitzner, D. J., Montgomery, D. C. and Gupta, S. (2004), "Using Control Charts to Monitor Process and Product Quality Profiles", *Journal of Quality Technology*, Vol. 36, No. 3, pp309-320.
67. Wurl, R. C., Albin, S. L. and Shiffer, I. J. (2001), "Multivariate Monitoring Of Batch Process Startup", *Quality and Reliability Engineering International*, 2001, 17.
68. Xu, D. and Albin, S. L. (2002), "Manufacturing Start-up Problem Solved By Mixed-integer Quadratic Programming and Multivariate Statistical Modeling", *International Journal of Production Research*, Vol. 40, No. 3.
69. Yeh, A. B and Lin D. K. J. (2002), "A New Variables Control Chart for Simultaneously Monitoring Multivariate Process Mean and Variability", *International Journal of Reliability, Quality and Safety Engineering*, Vol. 9, No. 1, pp41-59.
70. Zhang, H. and Albin, S.L., (2007), "Detecting the Number of Operational Modes in Baseline Multivariate SPC Data", Accepted by *IIE Transactions*.

Curriculum Vita

Hang Zhang

9/1994-7/1999	Tsinghua University, Automation, Bachelor of Science
9/1999-7/2002	Tsinghua University, Automation, Master of Science
9/2002-5/2005	Rutgers University, Industrial and Systems Engineering, Master of Science
9/2002-5/2007	Rutgers University, Statistics, Master of Science
9/2002-8/2007	Rutgers University, Industrial and Systems Engineering, Doctor of Philosophy

Fellowship, Teaching Assistant and Graduate Assistant

Publications

H. Zhang and X.H. Yang, "Web-based Statistical Analysis and Utilization of Process Data", Automation Panorama, Vol. 17, 2002, pp 41-43.

H. Zhang and X.H. Yang, "Fieldbus Remote Control System Based on Web", Proceedings of 2002 International Fieldbus/Control and Management Integration Conference and Exhibition, Shanghai, China, 2002, pp42-46.

H. Zhang and S. L. Albin, "Detecting the Number of Operational Modes in Baseline MSPC Data", accepted by *IIE Transactions*.

H. Zhang and S. L. Albin, "Detecting Outliers in Complex Profiles Using χ^2 control chart method", under revision for *IIE Transactions*.

H. Zhang and S. L. Albin, "Determining Statistical Process Control Baseline Periods in Historical Data Collected in a Long Time Period", paper ready to submit.