國立交通大學
工業工程與管理學系

碩士論文

製程能力指標應用於多品質特性
及工具磨耗之製程

PCI Methodology Applied to Multiple Characteristics and Tool Wear Manufacturing Processes

研 究 生：莊雅斐

指導教授：彭文理　教授

中 華 民 國 九 十 五 年 七 月

製程能力指標應用於多品質特性

及工具磨耗之製程

研究生：莊雅斐　　　　　　　　　　　　　　指導教授：彭文理

國立交通大學工業工程與管理學系碩士班

摘要

製程能力指標被廣泛地應用在製造業，做為衡量產品品質的標準。製程能力指標能簡單且直接地表示產品品質，是建立在某些前提假設之下：製程穩定且不受特定因素影響而改變（即管制內）、製程符合常態分配、抽樣觀察值皆獨立。現實中，這些假設並非都能滿足，製程常會因為外在因素而產生變動。

本文中分別針對多品質特性及工具磨耗提出製程能力指標之修正，並分別以兩個實際的產品做案例，以說明如何應用修正之製程能力指標。

第一部分，利用複式抽樣法估計 C_{PU}^{T} 並比較四種方法的估計準確度。結果顯示 BCPB 方法的準確度較高。本文以 BCPB 方法計算，將符合信賴下界及估計準確度的抽樣數列表，提供使用者可容易查出需要的抽樣數，以進行抽樣計畫。

第二部分，考量受外在因素影響的製程，提出監控製程的程序，並依據製程能力需求提供指標臨界值。這個監控製程的程序類似其他管制圖的監控程序，可用來判斷製程是否要停止生產或更換零件。

關鍵字：多品質特性、複式抽樣、信賴下界、工具磨耗、臨界值

PCI Methodology Applied to Multiple Characteristics and Tool Wear Manufacturing Processes

Student: YaFei Chuang Advisor: Dr. W.L. Pearn

Department of Industrial Engineering and Management
National Chiao Tung University

Abstract

Process capability indices have been widely used in the manufacturing industry and provided numerical measures on process performance. However, process capability indices seem to be easy and straightforward to apply. That is because of some assumption must be satisfied: the process under investigation is free from any special or assignable cause, the process characteristic is normal distributed, and the observations of quality characteristics are statistically independent. However, these conditions are not always fulfilled in many manufacturing situations. In real world, process is always influenced by some assignable cause.

In addition, capability measure for processes with single characteristic has been investigated extensively, but capability measure for processes with multiple characteristics is comparatively neglected. In the thesis, the process capability indices for multi-characteristics of one-sided process and the modified index C_{pk} for tool wear process are proposed. Two real-world examples from multiple characteristics process and tool wear process respectively are taken to illustrate the applications of the propose approaches.

For the first example, we used four bootstrap methods to estimating C_{PU}^{T} and compare the estimation precision of these methods. The results indicated that the BCPB method has better performance based on estimation precision than others. The table of the lower confidence bound values and sample sizes required for specified precision of the estimation is provided for the practitioners.

For the second example, a procedure of capability measure with assignable causes is developed, and the critical value for various values capability requirements and sample size are provided. The proposed procedure is similar to those used in monitoring a process with control chart, and used to monitor the process and decide if the process should stop and replace the tool to avoid producing unacceptable products.

Keywords: multiple quality characteristics, bootstrap, lower confidence bound, tool wear, critical value.

誌謝

轉眼間研究所生涯即將結束，這段時間感謝 彭文理老師的諄諄教誨，除了論文的指導，修課時也給予我相當多的提點、幫助我探索更豐富的知識；更不時地在日常生活中講述一些人生的體驗及道理，讓我收穫良多。另外感謝雅甄學姐在論文寫作的過程中，給予我許多幫助，讓我的論文得以順利完成；還有口試時 許錫美老師及 鍾淑馨老師提出的指教，都讓我收穫良多。

除了感謝老師和學姐，我很高興研究所兩年生活中，可以認識一群好朋友。在研究所繁忙、沈重的課業及論文壓力下，看到愛搞笑的大家就會覺得很開心，無論是味覺不好的翊展、還是情境大師柏先，都會讓我笑不停、怕作者太累的電動魔人源翌，你們都是好人同學；517實驗室一姐佩青，論文寫累時和妳聊一聊就會比較輕鬆。當然少不了生管實驗室的幾位同學，這兩年大家可是共同製造了不少的美好回憶，當中要特別感謝宇帆和自誠，在碩一時每天辛苦地載我上課，才能讓我不用每天早起搭公車。

最後多謝家人的支持，尤其是每天跟我聊天的妹妹，是我論文寫作時的重要調劑。

將此論文獻給我親愛的家人與朋友

Content

List of Table

List of Figure

1. Introduction

1.1. Process Capability Indices

There are three basic means which is process yield; process expected loss and process capability indices (PCIs) can be widely applied in measuring product performance. Of the three, process capability indices are easily understood and can be straightforwardly applied to the manufacturing industry, because process capability indices establish the relationship between the actual process performance and the manufacturing specifications. The relationship between the actual process performance and the specification limits or tolerance may be quantified using appropriate process capability indices. The larger process capability index implies the higher process yield. Those capability indices quantifying process potential and process performance are necessary to successful quality improvement activities and quality program implementation. Several capability indices have been widely used in manufacturing industry as follows:

$$C_p = \frac{USL - LSL}{6\sigma},$$

$$C_{PU} = \frac{USL - \mu}{3\sigma},\ C_{PL} = \frac{\mu - LSL}{3\sigma},$$

$$C_{pk} = \min\left\{\frac{USL - \mu}{3\sigma}, \frac{\mu - LSL}{3\sigma}\right\},$$

where USL and LSL are the upper and the lower specification limits, μ is the process mean, σ is the process standard deviation. In the literature, many authors have promoted the use of various process capability indices. Examples include Kushler and Hurley (1992), Vännman and Kotz (1995), Kotz and Lovelace (1998), Pearn and Shu (2003), and reference therein.

In practice process mean μ and process variance σ^2 are unknown. In order to calculate the estimator, however, data must be collected to calculate the index value, and a great degree of uncertainty may be introduced into capability assessments due to sampling errors.

1.2. Literature Review

The assessment of process capability appears to be easy and straightforward to apply. That is because of some assumptions should be satisfied before those capability indices are used. These conditions stipulate that the process under investigation is free from any special or assignable causes (i.e., in-control), the process characteristic is to follow normal distribution, and the observed values of quality characteristics are statistically independent. In addition, capability measure for processes with single characteristic has been investigated extensively, see Kane (1986), Pearn *et al.* (1992, 1998), Chen *et al.* (1998), however, capability measure for processes with multiple characteristics is comparatively neglected.

With the scientific and technological progress, products are various and different from the past. In practice, most of current products are multi-characteristics. For process with multiple characteristics, Bothe(1992) considered a simple measure by taking the minimum measure of each single characteristic. For instance, consider a v- characteristics product with v-yield measures P_1, P_2, …, and P_v. The overall process yield is measured as $P = \min\{P_1, P_2, ..., P_v\}$. It is noted that this approach does not reflect the real situation accurately. Suppose a process has five characteristics ($v = 5$), with equal characteristic yield measures $P_1 = P_2 = P_3 = P_4 = P_5 = 99.85\%$ (or 1500 ppm of non-conformities). Assuming that the five characteristics are mutually independent, then the actual overall process yield should be calculated as $P = P_1 \times P_2 \times ... \times P_5 = 99.2522\%$ (or 7478 ppm of non-conformities), which is significantly less than that calculated by Bothe(1992). Chen and Pearn (2003) modified the process capability index with multi-characteristics, proposed

$$S_{pk}^T = \frac{1}{3}\Phi^{-1}\left\{[\prod_{j=1}^{v}(2\Phi(3S_{pkj}) - 1) + 1]/2\right\}.$$

Then, Wu and Pearn (2004) discussed couplers and wavelength division multiplexers, which are multi-characteristics products with one-sided specification. For the product, they proposed a capability index as

$$C_{PU}^T = \frac{1}{3}\Phi^{-1}\left\{\prod_{j=1}^{v}\Phi(3C_{PUj})\right\}.$$

They estimated confidence bound by bootstrap method. There are four types of bootstrap confidence interval, including the standard bootstrap confidence interval (SB), the percentile bootstrap confidence interval (PB), the biased corrected percentile bootstrap confidence interval (BCPB), and the bootstrap-t (BT) method. However, Wu and Pearn (2004) calculate the confidence bound by PB method.

These conditions (process free from assignable cause, normal distribution, independent observations) are not always fulfilled in many manufacturing situations; process capability analysis with assignable causes has become critical issues. Many researchers realized the fact and thus a number of studies are reported in the literature. Somerville and Montgomery (1996) presented an extensive study to illustrate how poorly the normally based capability indices perform as a predictor of process fallout when the process is non-normally distributed. Several authors have discussed the assessment of the process capability when the process output is non-normally distributed. These methods could be divided into two categories. One is completely new indices, designed to be robust to non-normality (see Johnson *et al.* (1994), Wright (1995), Chen and Pearn(1997)). Alternatively, other authors suggested corrections for existing process capability indices (see Vännman (1995), Castagliola (1996), Chang *et al.* (2002)).

A process capability analysis is valid only when the process under investigation is free from any special or assignable causes. It is also expected that observations are statistically independent; however, processes with uncontrollable but acceptable trend are common in practice. Current practices in assessing process capability in the presence of assignable cause include various techniques. Some approaches attempt to remove the variability associated with systematic cause. Time series modeling to trended data is also suggested by Alwan and Roberts (1988), who recommend using residuals in monitoring the process. Furthermore, others make the general assumption of linear degradation in the tool. Quesenberry (1988) suggested that tool wear can be modeled over an interval of tool life by a regression model and assumes that the tool wear rate is known or a good estimate of it is available.

However, the above approaches assume a static process capability over a cycle. By allowing the process capability to be dynamic within a cycle, as well as from cycle to cycle, circumvents some of the problems encountered. Spiring (1991) proposed an application of assessing process capability index, C_{pm} in the presence of a systematic assignable cause that results in a numerical measure of the actual process capability associated with the process.

1.3. Research Objects

Two real-world examples from multiple characteristics process and tool wear process respectively are taken to illustrate the applications of the propose approaches. For the first example, we used four bootstrap methods to estimating C_{PU}^{T} and compare the estimation precision of these methods. Since the four types of bootstrap confidence interval are different ways to estimate confidence bound the engineers/ practitioners would want to know which one is the best. In this paper, we compare the performance of confidence interval for the one-sided index C_{PU} with multiple characteristics by using these four bootstrap methods. Furthermore, we find that the BCPB method would be the best way to estimate confidence interval when sample size is less than 100. We also provide the tables about the sample sizes required for various designated precision for the engineers/ practitioners to use in their factory applications. For the second example, the estimator of the index C_{pk} for the process where a systematic assignable cause occurs is investigated. Further, a procedure of capability measure with assignable causes is developed, and the critical value for various values capability requirements and sample size are provided. The real-world case taken from IC packaging process is investigated to illustrate the applicability of the proposed procedure. The testing procedure is similar to those used in monitoring a process with control chart and used to monitor the process and decide if the process should stop and replace the tool to avoid producing unacceptable products.

2. Capability Estimation for Process with Multiple Quality Characteristics

2.1. Thin-film Transistor Liquid Crystal Display: Manufacturing Process

TFT-LCD represents thin-film transistor liquid-crystal display. TFT-LCD was first invented in early 1960's, over the years substantial improvements it was developed for commercial production for notebook computers in 1991, hence the TFT-LCD industry began.

The TFT-LCD is a progressive display, which screen picture information by adjusting the amount of light permitted. TFT is a circuit formed with semiconductor films on a thin glass substrate to control liquid crystals. The circuit plays a essential role in controlling each pixel, the basic unit of a picture image. The color filter displays a color image by coating the pixel on a glass substrate. TFT-LCD technology has created a wide range of computer and consumer products. The flat and thin attributes of LCD makes them ideal for mobile or portable applications.

Three key components make the liquid-crystal display module functions properly. Those include the liquid-crystal display, the back lighting, and the peripheral (interface) system. There are three major process groups in TFT-LCD manufacturing process: array process, cell process and module assemble process, as follows:

Array Process: The array process is similar to the semiconductor manufacturing process, except that transistors are fabricated on a glass substrate instead of a silicon wafer.

Cell Process: The cell process joins the arrayed back substrate and the front substrate that is fitted with a color filter. The space between the two substrates is filled with liquid crystal.

Module Assembly Process: The module assembly process involved connecting additional components, such as driver integrated circuits and backlight units, to the fabricated glass panel.

We focus on the array process of TFT-LCD. The processing steps in array process are similar to semiconductor industry: deposition, photolithography and etching. Following process are included:

PECVD (Plasma Enhanced Chemical Vapor Deposition) Process: Maintaining a vacuum before gas enters the chamber, and heating the glass plate to a specific temperature. The RF voltage is applied from electrodes inside the chamber, which transforms gas into a plasma state when gas flows into the chamber. Precursors are formed and deposited on the glass substrate from this plasma.

Sputtering Process: Sputtering is the process wherein the gas ion, which is the high energy inside the plasma created by RF power or DC power, collides with the target surface, resulting in the deposition of the target material on the plate. Generally, the target materials are mounted on the negative electrode surface. Then, the sputtered target materials are deposited on the plate, which is put on the positive electrode. For sputtering, inactive gases are used, such as helium and argon, so that deposition material chemistry is not affected.

Photolithography Process: Photolithography is the transfer of a pattern from the photo mask onto a substrate. The substrate is coated with an extremely thin liquid film of photosensitive material, called photoresist. Then the light exposes the photoresist, some of which is destroyed when exposed to the light. The unnecessary portion of the material is then cleaned from the surface through another process. Another layer of the photoresist is then deposited to the substrate, exposed, cleaned, until all the layers have been printed or imaged onto the surface.

Dry etch Process: The dry etch process uses reactive species, such as atoms or radicals from the gas plasma, to etch away a portion of the object material. When these species react with the material located on the plate, the open region of material transforms into a volatile state and is removed from the matrix. In this process, the reaction velocity is fast and fine patterns can be formed uniformly.

Characteristics of Thin-Film Transistor Liquid Crystal Display

The photolithography process is a critical step within the LCD manufacturing process because panel quality depends on the entire pattern formation. We focus on some quality characteristics in photolithography process. Another layer is formed (deposited) by sputtering, exposed, and cleaned. Repeating these steps until all layers has been printed. Between one deposited layer and another, an error may be existed, called overlay, see Figure 1. There are three steps in photolithography process: coating, exposure, development. It might result deviation as exposure on panel window, called critical dimension, see Figure 2. In addition, coating photoresist on panel has to be uniform. The specifications of these three key parameters are shown as Table 1.

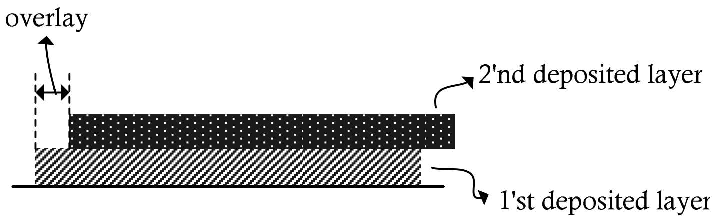

Figure 1. Deposited layers on TFT-LCD.

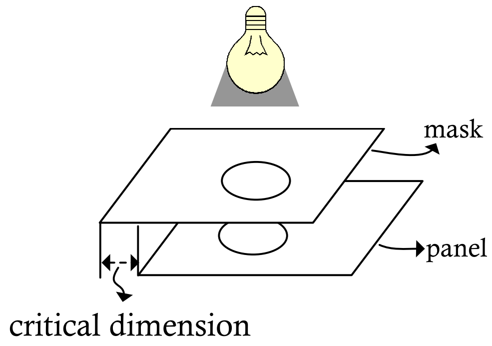

Figure 2. Exposure process on panel window.

Table 1. Specifications for thin-film transistor liquid crystal display.

Parameter	Specifications
Overlay	$\le 0.1\mu m$
Critical Dimension	$\le 0.3\mu m$
Uniformality	≤ 0.03

2.2. Thin-film Transistor Liquid Crystal Display: Capability Measure

One-sided Capability Measurement with Single Characteristic

Process capability index is a function of process parameters and manufacturing specifications, which measures the capability of reproducing products meeting the specifications. For normally distributed processes with one-sided specification limit *USL*, or *LSL*, the process yield is the following, where *Z* follows the standard normal distribution N(0,1).

$$P(X < USL) = P\left(\frac{X-\mu}{\sigma} < \frac{USL-\mu}{\sigma}\right) = \Phi(3C_{PU}),$$

$$P(X < LSL) = P\left(\frac{X-\mu}{\sigma} < \frac{LSL-\mu}{\sigma}\right) = \Phi(3C_{PL}).$$

For convenience of presentation, we let C_I denote either C_{PU} or C_{PL}. Therefore, the corresponding non-conforming units in parts per million (NCPPM) for a well-controlled normal process can be calculated as: NCPPM $=10^6 \times [1-\Phi(3C_I)]$. Consequently, the production yield for usual existing processes should target no more than 88 PPM, noting that NCPPM $\le$ 100 PPM is the common standard used in most microelectronic industries for products with one-sided specification. The production yield for newly set-up processes on safety, strength, or with critical parameters, however, should target no more than 0.8 PPM, a more stringent requirement set for possible mean shift or variation change.

One-sided Capability Measurement with Multiple Characteristics

Capability measure for processes with single characteristic has been investigated extensively. But, capability measure for processes with multiple characteristics is comparatively neglected. For processes with multiple characteristics, a simple measure by taking the minimum of the measure of each single characteristic has been considered. Wu and Pearn (2004) proposed the modified one-sided index C_{PU}^T, and the multi-characteristics process yield index can be rewritten as:

$$C_{PU} = \frac{1}{3}\Phi^{-1}\{\Phi(\frac{USL-\mu}{\sigma})\},$$

where $\Phi(\cdot)$ is the cumulative distribution of the standard normal distribution $N(0,1)$, and Φ^{-1} is the inverse function of $\Phi(\cdot)$. For the process with multiple quality characteristics, the following overall capability index is referred to as C_{PU}^T:

$$C_{PU}^T = \frac{1}{3}\Phi^{-1}\left\{\prod_{j=1}^{v}\Phi(3C_{PUj})\right\},$$

where C_{PUj} denotes the C_{PU} value of the jth characteristic for $j = 1, 2, \ldots, v$, and v is the number of characteristics. The index, C_{PU}^T, can be viewed as a generalization of the single characteristic yield index, C_{PU}. Give $C_{PU}^T = c$, we have

$$\left\{\prod_{j=1}^{v}\Phi(3C_{PUj})\right\} = \Phi(3c).$$

In fact, Wu and Pearn (2004) showed that the relationship between the index C_{PU}^T and the overall process yields P can be established as follows:

$$P = \prod_{j=1}^{v}P_j = \prod_{j=1}^{v}\Phi(3C_{PUj}) = \Phi(3C_{PU}^T).$$

Hence, the new index C_{PU}^T provides an exact measure on the overall process yield. For the example given in Wu and Pearn (2004), if C_{PU}^T = 1.00, then the entire process yield would be exactly 99.865%, and each single characteristic yield is no less than $(0.9986501)^{1/5}$ = 0.9997299 (equivalent to 270 NCPPM).

In order to calculate the estimator, however, sample data must be collected, and a great degree of uncertainty may be introduced into capability assessments due to sampling errors. The approach by simply looking at the calculated values of the estimated indices and then making a conclusion on whether the given process is capable, is highly unreliable as the sampling errors have been ignored. A reliable approach for estimating the true value of process index is to construct the lower confidence bound. The lower confidence bound not only is essential to production yield assurance, but also can be used in capability testing for decision making.

Hence, if the required overall process capability is $C_{PU}^{T} \geq c_0$, let c' be the minimum C_{PU} required for each single characteristic, then

$$\frac{1}{3}\Phi^{-1}\left\{\prod_{j=1}^{v}\Phi(3C_{PUj})\right\} \geq \frac{1}{3}\Phi^{-1}\left\{\prod_{j=1}^{v}\Phi(3c')\right\} \geq c_0 .$$

And then we obtain the lower confidence bound to be

$$c' \geq \frac{1}{3}\Phi^{-1}\left(\sqrt[v]{\Phi(3c_0)}\right).$$

Table 2 displays the lower bound c' of C_{PUj} obtained by Wu and Pearn (2004) for the required overall process capability C_{PU}^{T} are 1.00 and 1.33 for $v = 1(1)5$ characteristics. For example, if a process has capability requirement $C_{PU}^{T} \geq 1.00$ with $v = 5$, i.e., the capability for all the five characteristics is the following $C_{PUj} \geq 1.153$, for j = 1, 2, …, 5.

Table 2. Lower bound of various capability levels for multiple characteristics.

	$C_{PU}^{T} > c_0$	
v	1.00	1.33
1	1.000	1.330
2	1.068	1.383
3	1.107	1.414
4	1.133	1.436
5	1.153	1.452

Hence, given the desired estimation precision $R_{pm}^{(PS)}$, the confidence level γ, a Since given the desired estimation precision R_{PU} and the confidence level γ, ensures that the risk of making incorrect decisions will be no greater than the preset Type I error 1 − R_{PU}. For a given estimation precision $R_{PU} = \hat{C}_{PU} / C_{PU}^{T}$, the sample size determination is important as it directly relates to the cost of the data collection plan. Hence, given the desired estimation precision R_{PU}, the approximate sample size must be obtained. In order to compute the lower confidence bound to determine sample sizes required for specified precision of the estimation on C_{PU}^{T}. We using bootstrap methods to determine the lower confidence bound in the following section.

2.3. Capability Estimation: Bootstrap Approach

2.3.1. The Bootstrap Methodology

For computational tractability, statistical research work generally has depended on the central limit theorem and normal approximations to obtain standard errors and/or confidence interval. Unfortunately, most process data in real world are not normally distributed. Usually the population distribution of data is unknown. Idealized models and assumptions can be replaced with more realistic modeling or by virtually model-free analyses. Efron (1979, 1982) introduced a nonparametric, computational intensive but effective estimation method, called the "Bootstrap", which is a data based simulation technique for statistical inference. One can use the nonparametric bootstrap method to estimate the sampling distribution of a statistic, while assuming only that the sample is a representative of the population from which it is drawn, and that the observations are independent and identically distributed. The merit of the nonparametric bootstrap approach is that it does not rely on any assumptions regarding the underlying distribution. Rather than using distribution frequency tables to compute approximate p probability values, the bootstrap method generates a unique sampling distribution based on the actual sample rather than the analytic methods.

The bootstrap sampling is equivalent to sampling (with replacement) from the empirical probability distribution function. It can be applied whenever the construction of confidence intervals for parameters using the standard statistical techniques becomes intractable. In order to calculate process capability indices sample data must be collected because μ and σ are unknown. Current practices of measuring capability by evaluating point estimate are unreliable because it ignores sampling error. The essence of bootstrapping is that, without any knowledge about a population, the distribution found in a random sample of size n from the population is the best guide to the distribution in the population. By resampling observations from the observed data, the population that consists of the n observed sample values is used to model the unknown real population. The only difference between bootstrapping and randomization is the sampling with replacement.

In the bootstrap, B new samples, each of the same size as the observed data n, are drawn with replacement from the population. Efron and Tibshirani (1986) developed four types of bootstrap confidence interval, including the standard bootstrap confidence interval (SB), the percentile bootstrap confidence interval (PB), the biased corrected percentile bootstrap confidence interval (BCPB), and the bootstrap-t (BT) method. In the following we give an overview of four bootstrap confidence intervals. These are employed to determine the lower confidence bounds of the index.

Standard Bootstrap (SB)

From the B bootstrap estimates $\widehat{C}_{PU}^{T*}$, calculate the sample average and the sample standard deviation

$$\overline{C}_{PU}^{T*} = \frac{1}{B}\sum_{i=1}^{B}\hat{C}_{PU}^{T*}(i),$$

$$S^*_{C^T_{pu}} = \sqrt{\frac{1}{B-1}\sum_{I=1}^{B}[\hat{C}^{T*}_{PU}(i) - \hat{C}^{T*}_{PU}]^2} \ ,$$

where $\hat{C}^{T*}_{PU}(i)$ is the i-th bootstrap estimate. The quantity $S^*_{C^T_{PU}}$ is actually an estimator of the standard deviation of $\hat{C}^T_{PU}$ and if $\hat{C}^T_{PU}$ is approximately normal distribution the (1-2α) 100% SB confidence interval can be obtained as

$$[\hat{C}^T_{PU} - Z_\alpha S^*_{C^T_{PU}}],$$

where Z_α is t he upper α quantile of the standard normal distribution.

The Percentile Bootstrap (PB)

From the ordered collection of $\hat{C}^{T*}_{PU}(i)$, select the α percent and the (1-α) percent points as the end points and the PB confidence interval is

$$[\hat{C}^{T*}_{PU}(\alpha B)].$$

Biased-Corrected Percentile Bootstrap (BCPB)

The bootstrap distribution may be biased while the percentile confidence interval is possible due to sampling errors. In other words, that bootstrap distributions obtained using only a sample of the complete bootstrap distribution may be shifted higher or lower than would expected. Thus, a three steps procedure has been developed to correct for this potential bias (Efron, 1982). First, using the ordered distribution of $\hat{C}^{T*}_{PU}$, calculate the probability of

$$P_0 = P[\hat{C}^{T*}_{PU} \le \hat{c}^T_{PU}],$$

second, calculate

$$Z_o = \Phi^{-1}(P_o),$$

$$P_L = \Phi(2Z_0 - Z_\alpha),$$

$$P_U = \Phi(2Z_0 + Z_\alpha),$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. Finally, the BCPB confidence is obtained as

$$[\hat{C}^{T*}_{PU}(P_L B)].$$

Bootstrap-t (BT)

While the distribution of the statistic is skewed, the percentile bootstrap confidence interval is possible lower. Thus, the bootstrap-t is developed and that the generated distribution will mimic the distribution of T. First, approximate the distribution of a statistic of $T = (\hat{C}^T_{PU} - C^T_{PU}) / S_{C^T_{PU}}$ by using bootstrap. By taking bootstrap samples from the original data values the bootstrap approximation in this case can be obtained, calculate the corresponding estimates $\hat{C}^{T*}_{PU}(i)$ and their standard error, and then finding the T-values $T = (\hat{C}^{T*}_{PU} - \hat{C}^T_{PU}) / S^*_{C^T_{PU}}$. The (1-2α)100% BT confidence interval can be obtained as

$$[\hat{C}^T_{PU} - t^*_\alpha S^*_{C^T_{PU}}],$$

where t_{α}^{*} and $t_{1-\alpha}^{*}$ are the upper α and 1-α quantile of the bootstrap T-distribution respectively.

2.3.2. Performance Comparisons

To compare the estimating performance of these four types of bootstrap interval, we calculate the value $R_{PU} = C_{PU}^{estimating} / C_{PU}^{T}$ (Pearn and Shu, 2003) , which is estimating precision. We also compare the four types of bootstrap interval by ranking. The closer the bootstrap interval met the true value ($C_{PU}^{T} = 1.00 \ or \ 1.33$), the better the rank is. On the contrary, if the bootstrap interval is greater than true value, the rank would be worse.

Calculated the total rank R, which is a weighted value; it shows the rank of the four bootstrap methods (see detail as Appendix A).

$$R = (number\ of\ rank 1\times 1 + number\ of\ rank 2\times 2 + number\ of\ rank 3\times 3 + number\ of\ rank 4\times 4)/500$$

Some random data distributed as normal distribution are generated by MABLAB program (in Appendix B) and these data satisfied the required overall process capability C_{PUj} (see Table 2). For example, if a process has a capability requirement $C_{PU}^{T} \geq 1.00$ with $v = 5$, i.e., the capability for all the five characteristics is the following $C_{PU}^{T} \geq 1.153$, for $j = 1,2,...,5$. In Table 3, the rank of four bootstrap method is illustrated with various sample size n=30(10)100, 125, 150, 200, and v=2(1)5 as $C_{PU}^{T} = 1$ and 1.33. For example, if the sample size is 60, we generate two random observations distributed as normal distribution and $C_{PU}^{T} = 1.33$. From the table, we obtained the four ranks 2.944, 2.028, 1.132, 3.896 respectively.

We found that the method BCPB (Biased-corrected Percentile Bootstrap) is better than other methods in Table 3, (when $C_{pu}^{T} = 1$). The estimating results of four methods are similar as the sample size increasing, shown in Figure 3(a)~(d) and Figure 4(a)~(d). However, the result in Figure 4(c)~(d) (when $C_{pu}^{T} = 1.33$, $v = 4 \sim 5$) is not the same.

These tables show that the method BCPB is distinctly better in small sample size ($n < 100$); however as sample size increase, the difference between four methods' performance is unobvious. In addition, as quality characteristic increase, the rank of BCPB method is large. This indicates that BCPB method is perform worse than the other methods (see Figure 4(d)), actually, the estimation of four methods are similar. However, in small sample size ($n < 100$), the BCPB method is the best one to calculate $\hat{C}_{PU}^{T}$. Hence we use BCPB method to evaluate the $\hat{C}_{PU}^{T}$ in the following section.

Table 3. The total rank of the four bootstrap methods as $C_{PU}^{T}=1,1.33$ and $v=2(1)5$.

$v=2$ / n	$C_{pu}^{T}=1$				$C_{pu}^{T}=1.33$			
	SB	PB	BCPB	PT	SB	PB	BCPB	PT
30	3	2	1.006	3.994	2.996	1.996	1.008	4.000
40	2.996	2.004	1.004	3.996	2.982	2.008	1.034	3.974
50	2.992	2.002	1.024	3.982	2.99	2.012	1.030	3.968
60	2.982	2.014	1.042	3.962	2.984	2.010	1.056	3.950
70	2.994	2.004	1.018	3.984	2.984	2.020	1.030	3.966
80	2.988	2.012	1.026	3.974	2.972	2.028	1.072	3.928
90	2.994	2.016	1.022	3.968	2.970	2.026	1.084	3.920
100	2.980	2.020	1.026	3.974	2.992	2.002	1.072	3.934
125	3.000	2.010	1.018	3.972	2.968	2.040	1.104	3.888
150	2.988	2.012	1.040	3.960	2.974	2.056	1.094	3.876
200	3.004	2.018	1.036	3.942	2.972	2.018	1.148	3.858
$v=3$ / n	$C_{pu}^{T}=1$				$C_{pu}^{T}=1.33$			
	SB	PB	BCPB	PT	SB	PB	BCPB	PT
30	2.966	2.028	1.100	3.906	2.942	2.048	1.112	3.896
40	2.950	2.040	1.106	3.904	2.954	2.046	1.132	3.868
50	2.970	2.034	1.100	3.896	2.946	2.044	1.150	3.860
60	2.942	2.070	1.124	3.864	2.922	2.094	1.216	3.768
70	2.954	2.052	1.120	3.874	2.916	2.112	1.290	3.678
80	2.944	2.062	1.172	3.818	2.904	2.108	1.332	3.656
90	2.940	2.078	1.166	3.816	2.836	2.168	1.428	3.568
100	2.934	2.072	1.162	3.828	2.894	2.136	1.352	3.610
125	2.964	2.060	1.124	3.852	2.832	2.172	1.590	3.406
150	2.970	2.044	1.136	3.848	2.804	2.176	1.620	3.398
200	2.918	2.114	1.200	3.768	2.784	2.230	1.636	3.344
$v=4$ / n	$C_{pu}^{T}=1$				$C_{pu}^{T}=1.33$			
	SB	PB	BCPB	PT	SB	PB	BCPB	PT
30	2.97	2.052	1.116	3.856	2.94	2.076	1.17	3.814
40	2.912	2.084	1.208	3.796	2.882	2.11	1.3	3.708
50	2.922	2.094	1.224	3.76	2.826	2.186	1.508	3.478
60	2.91	2.096	1.222	3.772	2.798	2.188	1.53	3.478
70	2.882	2.126	1.254	3.736	2.822	2.186	1.664	3.328
80	2.916	2.108	1.26	3.716	2.79	2.274	1.666	3.268
90	2.892	2.12	1.318	3.668	2.742	2.272	1.784	3.196
100	2.886	2.124	1.28	3.706	2.73	2.308	1.85	3.11
125	2.922	2.126	1.308	3.644	2.634	2.4	2.054	2.912
150	2.92	2.1	1.328	3.648	2.684	2.412	2.034	2.868
200	2.88	2.152	1.394	3.57	2.632	2.434	2.322	2.606
$v=5$ / n	$C_{pu}^{T}=1$				$C_{pu}^{T}=1.33$			
	SB	PB	BCPB	PT	SB	PB	BCPB	PT
30	2.938	2.084	1.222	3.756	2.86	2.15	1.398	3.59
40	2.878	2.118	1.314	3.688	2.786	2.216	1.652	3.344
50	2.892	2.148	1.326	3.63	2.792	2.24	1.648	3.316
60	2.85	2.16	1.408	3.582	2.758	2.254	1.828	3.156
70	2.846	2.158	1.45	3.542	2.686	2.336	2.026	2.942
80	2.902	2.18	1.382	3.528	2.654	2.4	2.11	2.83
90	2.876	2.172	1.5	3.452	2.572	2.432	2.288	2.702
100	2.83	2.192	1.56	3.406	2.582	2.478	2.384	2.55
125	2.904	2.156	1.514	3.42	2.56	2.482	2.6	2.354
150	2.856	2.228	1.572	3.342	2.466	2.524	2.648	2.346
200	2.834	2.244	1.578	3.34	2.426	2.614	2.878	2.074

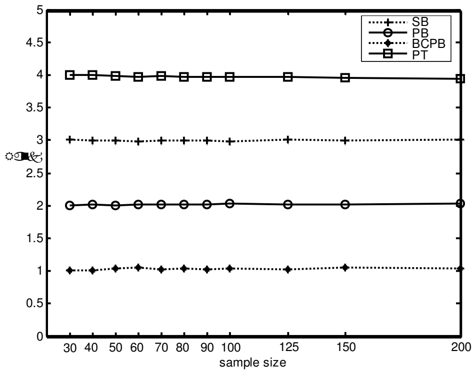

Figure 3(a). The total rank of the four bootstrap methods as $C_{pu}^{T}=1$, $v=2$

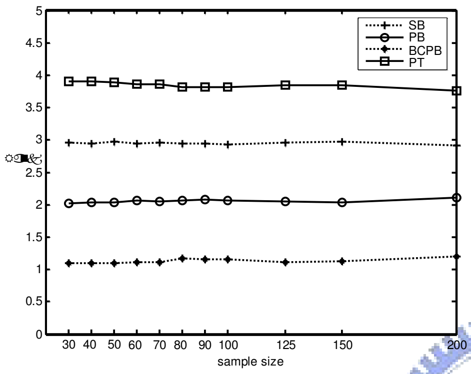

Figure 3(b). The total rank of the four bootstrap methods as $C_{pu}^{T}=1$, $v=3$

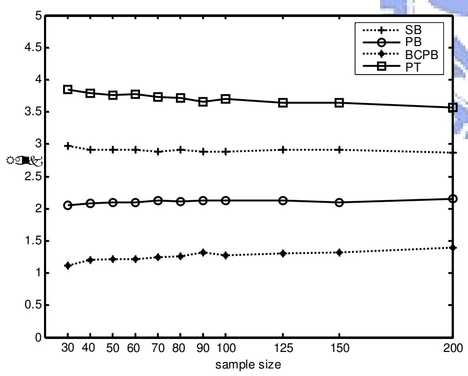

Figure 3(c). The total rank of the four bootstrap methods as $C_{pu}^{T}=1$, $v=4$.

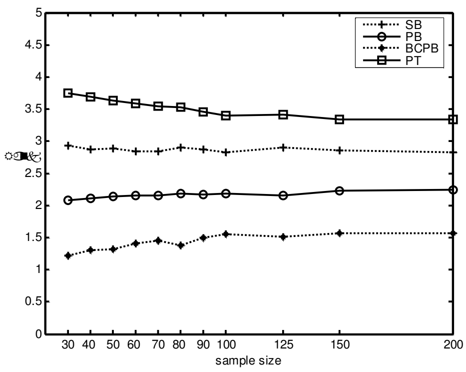

Figure 3(d). The total rank of the four bootstrap methods as $C_{pu}^{T}=1$, $v=5$

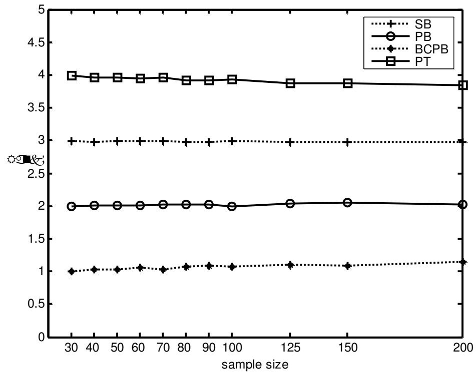

Figure 4(a). The total rank of the four bootstrap methods as $C_{pu}^{T}=1.33$, $v=2$

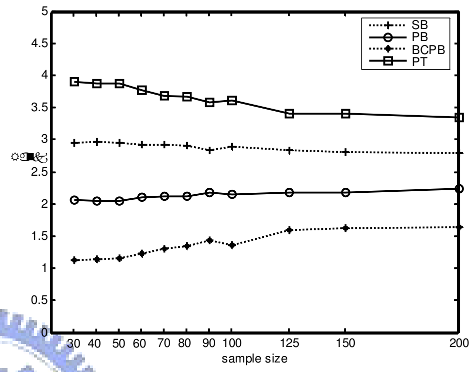

Figure 4(b). The total rank of the four bootstrap methods as $C_{pu}^{T}=1.33$, $v=3$

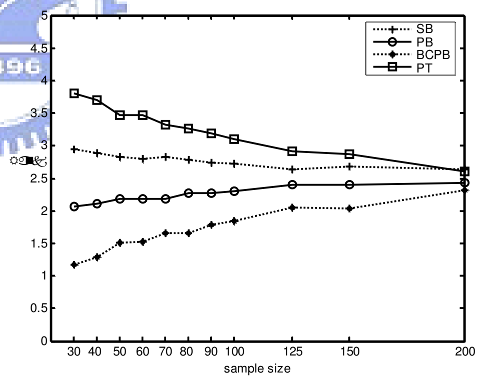

Figure 4(c). The total rank of the four bootstrap methods as $C_{pu}^{T}=1.33$, $v=4$.

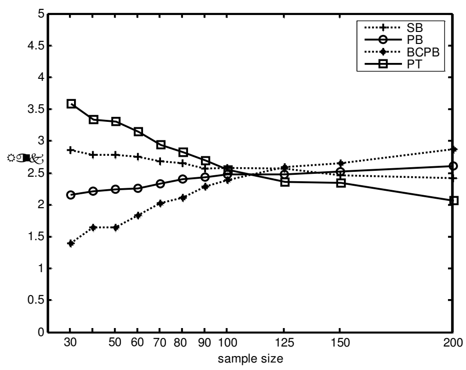

Figure 4(d). The total rank of the four bootstrap methods as $C_{pu}^{T}=1.33$, $v=5$

2.3.3. Recommended Bootstrap with Sample Size Information

These simulation results indicate that the estimating precision R_{pu} approaches to 1 as the sample size increases in all cases that have been investigated in Table 4-5.

Table 4. The relationship of the sample size and estimating precision with $C_{PU}^T = 1$.

	$v=2$	$v=3$	$v=4$	$v=5$
n	R_{pu}	R_{pu}	R_{pu}	R_{pu}
30	0.8458	0.8406	0.8387	0.8384
40	0.8598	0.8571	0.8554	0.8563
50	0.8712	0.8696	0.8694	0.8686
60	0.8775	0.8780	0.8794	0.8804
70	0.8873	0.8874	0.8885	0.8886
80	0.8931	0.8936	0.8952	0.8957
90	0.8973	0.8995	0.9012	0.9020
100	0.9022	0.9039	0.9061	0.9070
125	0.9115	0.9150	0.9169	0.9180
150	0.9182	0.9222	0.9243	0.9260
200	0.9289	0.9327	0.9351	0.9371

Table 5. The relationship of sample size and estimating precision with $C_{PU}^T = 1.33$.

	$v=2$	$v=3$	$v=4$	$v=5$
n	R_{pu}	R_{pu}	R_{pu}	R_{pu}
30	0.8520	0.8436	0.8423	0.8397
40	0.8622	0.8589	0.8577	0.8540
50	0.8731	0.8696	0.8670	0.8670
60	0.8803	0.8793	0.8774	0.8774
70	0.8875	0.8865	0.8856	0.8845
80	0.8929	0.8926	0.8924	0.8925
90	0.8980	0.8979	0.8980	0.8985
100	0.9030	0.9034	0.9027	0.9027
125	0.9124	0.9125	0.9140	0.9142
150	0.9190	0.9204	0.9218	0.9229
200	0.9294	0.9316	0.9328	0.9341

Chou and Owen (1989) showed that under normality assumption the estimator $\hat{C}_{PU}$ and $\hat{C}_{PL}$ are distributed as $(3\sqrt{n})^{-1} t_{n-1}(\delta)$, where $t_{n-1}(\delta)$ is distributed as the noncentral t distribution with $n-1$ degrees of freedom and noncentrality parameter $\delta = 3\sqrt{n} C_{PU}$ and $\delta = 3\sqrt{n} C_{PL}$, respectively. A $100(1-\alpha)\%$ lower confidence bound L_C for C_{PU} satisfies. It can be written as:

$$\Pr(\frac{USL-\mu}{3\sigma} \geq L_C) = \Pr(t_{n-1}(\delta_1) \leq t_1) = 1-\alpha,$$

where $t_1 = 3\sqrt{n}\hat{C}_{PU}$ and $\delta_1 = 3\sqrt{n} L_C$. Therefore, we can calculate the lower confidence bound (LCB) by solving the cumulative distribution function (CDF) of noncentral t distribution with $n-1$ degrees of freedom and noncentrality parameter $\delta_1 = 3\sqrt{n}L_C$.

To compute the sample size required n, we develop a MATLAB program (available on request). The simulation data is the same one in Section2.3.2 (random data generated from normal distribution). Let the desired estimation precision be R_{PU} and the confidence level be γ, and then the minimum sample size n (always rounding up if n is not an integer) can be calculated. Table 6 displays the sample size n required for $R_\gamma \geq R_{PU}$ i th R_{PU} = 0.75(0.01)0.95 and $\gamma = 0.9, 0.95, 0.975, 0.99$. We also provide the actual estimation precision R_γ in the Table 6. For example, if R_{PU} is set to 0.89, then with γ = 0.95 the sample size needed is n =76. We conclude that a minimum sample size of n =76 is required to be 95% certain that the true C_{PU} is no less than R_γ = 89.12% of the sample estimate $\hat{C}_{PU}$. Thus, if the sample estimate $\hat{C}_{PU}$ =1.2, then the true value of C_{PU} is no less than 1.2 × 89.12% = 1.069, with 95% confidence. The MATLAB program computing the minimum sample sizes required for specified precision of the estimation for the C_{PU}^T is shown in Appendix B MATLAB Program.

Table 6. Sample size n required for $R_\gamma \geq R_{PU}$, with $R_{PU} = 0.75(0.01)0.95$, $\gamma = 0.9, 0.95, 0.975, 0.99$, three quality characteristics and C_{PU}^T =1

R_{PU}	$\gamma = 0.90$		$\gamma = 0.95$		$\gamma = 0.975$		$\gamma = 0.99$	
	n	R_γ	n	R_γ	n	R_γ	n	R_γ
0.75	-	-	-	-	-	-	16	0.7518
0.76	-	-	-	-	6	-	21	0.7609
0.77	-	-	-	-	7	0.7776	24	0.7731
0.78	-	-	-	-	14	0.7832	28	0.7807
0.79	-	-	-	-	18	0.7904	31	0.7901
0.80	-	-	6	-	22	0.8005	36	0.8012
0.81	-	-	12	0.8119	26	0.8107	40	0.8103
0.82	-	-	17	0.8222	32	0.8226	49	0.8201
0.83	-	-	23	0.8316	38	0.8304	56	0.8305
0.84	-	-	28	0.8414	44	0.8403	65	0.8414
0.85	6	-	35	0.8536	52	0.8502	75	0.8502
0.86	18	0.8608	41	0.8600	63	0.8613	88	0.8609
0.87	26	0.8708	51	0.8710	73	0.8700	105	0.8710
0.88	33	0.8805	60	0.8802	88	0.8800	124	0.8812
0.89	44	0.8909	76	0.8912	105	0.8908	146	0.8610
0.90	54	0.9005	93	0.9004	128	0.9000	176	0.9005
0.91	71	0.9107	115	0.9102	158	0.9103	213	0.9101
0.92	92	0.9205	146	0.9201	197	0.9204	268	0.9202
0.93	121	0.9306	188	0.9303	253	0.9300	339	0.9302
0.94	164	0.9400	251	0.9402	337	0.9400	451	0.9400
0.95	231	0.9500	350	0.9502	473	0.9505	634	0.9500

In Table 6, we can found that as the sample size increases, the R_{PU} and γ increase. However, some values of sample size can not be obtained when the values of precision R_{PU} and confidence level γ are small. This is due to the problem of the bootstrap resampling procedure.

We consider the following case taken from a manufacturing factory located on the Science-Based Industrial Park in Taiwan, making the liquid crystal display. With focus on these key parameters: overlay, critical dimension and uniformality. To obtain the sample size required n under the desired estimation precision R_{pu}, we can look it up in Table 6. Table 6 displays the sample size n required for $R_{\gamma} \geq R_{pu}$ with $R_{pu} = 0.75(0.01)0.95$ and γ= 0.9, 0.95, 0.975, and 0.99. If the practitioners set R_{pu} to be 0.92 and γ = 0.95 then the sample size needed is n = 146. We conclude that a minimum sample size of n = 150 is required to be 95% certain that the true C_{PU}^{T} is no less than R_{γ} = 92.01% of the sample estimate $\hat{C}_{PU}^{T}$. Thus, if the sample estimate $\hat{C}_{PU}^{T}$ = 1.3, then the true value of C_{PU}^{T} is no less than 1.3 × 92.01% =1.20, with 95% confidence. Hence sample data collected from 150 LCD are displayed in Table 7. The upper specification limit, the calculated sample mean, sample standard deviation, the estimated $\hat{C}_{PU_j}$ for overlay, critical dimension and uniformality are summarized in Table 8.

Table 7. The 150 sample observations for three quality characteristics.

Overlay (μm):									
0.0779	0.0697	0.0764	0.0763	0.0834	0.0860	0.0778	0.0849	0.0846	0.0649
0.0853	0.0801	0.0711	0.0847	0.0817	0.0747	0.0886	0.0777	0.0889	0.0716
0.0802	0.0776	0.0800	0.0811	0.0873	0.0804	0.0810	0.0729	0.0782	0.0794
0.0711	0.0712	0.0724	0.0839	0.0831	0.0846	0.0803	0.0851	0.0701	0.0741
0.0706	0.0826	0.0665	0.0843	0.0862	0.0824	0.0810	0.0804	0.0838	0.0693
0.0757	0.0842	0.0765	0.0742	0.0838	0.0832	0.0837	0.0745	0.0820	0.0911
0.0786	0.0751	0.0738	0.0801	0.0853	0.0667	0.0778	0.0888	0.0890	0.0638
0.0796	0.0859	0.0718	0.0799	0.0637	0.0789	0.0878	0.0926	0.0674	0.0745
0.0859	0.0913	0.0863	0.0695	0.0878	0.0753	0.0790	0.0798	0.0801	0.0736
0.0746	0.0885	0.0788	0.0746	0.0862	0.0787	0.0753	0.0793	0.0776	0.0945
0.0833	0.0709	0.0804	0.0780	0.0888	0.0842	0.0794	0.0793	0.0771	0.0835
0.0691	0.0806	0.0805	0.0735	0.0843	0.0837	0.0727	0.0834	0.0752	0.0877
0.0771	0.0850	0.0755	0.0826	0.0776	0.0833	0.0669	0.0740	0.0839	0.0743
0.0781	0.0754	0.0840	0.0840	0.0962	0.0780	0.0801	0.0742	0.0781	0.0908
0.0911	0.0849	0.0764	0.0932	0.0783	0.0732	0.0722	0.0775	0.0787	0.0715

Critical Dimension (μm):									
0.2559	0.2627	0.2717	0.2656	0.2756	0.2747	0.2645	0.2671	0.2588	0.2703
0.2689	0.2633	0.2694	0.2573	0.2691	0.2776	0.2550	0.2632	0.2624	0.2605
0.2783	0.2623	0.2691	0.2571	0.2616	0.2759	0.2670	0.2688	0.2598	0.2620
0.2788	0.2507	0.2661	0.2726	0.2807	0.2735	0.2673	0.2478	0.2831	0.2653
0.2691	0.2792	0.2718	0.2791	0.2770	0.2581	0.2731	0.2660	0.2612	0.2718
0.2657	0.2711	0.2579	0.2649	0.2760	0.2707	0.2769	0.2605	0.2648	0.2723
0.2657	0.2650	0.2764	0.2827	0.2734	0.2676	0.2757	0.2662	0.2758	0.2753
0.2514	0.2654	0.2754	0.2842	0.2524	0.2734	0.2687	0.2743	0.2631	0.2719
0.2726	0.2828	0.2750	0.2721	0.2633	0.2608	0.2877	0.2628	0.2894	0.2638
0.2700	0.2654	0.2819	0.2728	0.2713	0.2670	0.2580	0.2730	0.2652	0.2794
0.2656	0.2850	0.2735	0.2774	0.2730	0.2757	0.2640	0.2707	0.2564	0.2634
0.2638	0.2727	0.2681	0.2647	0.2720	0.2687	0.2627	0.2828	0.2838	0.2700
0.2638	0.2640	0.2797	0.2708	0.2704	0.2475	0.2713	0.2710	0.2870	0.2610
0.2651	0.2729	0.2698	0.2702	0.2694	0.2586	0.2619	0.2790	0.2723	0.2833
0.2709	0.2592	0.2740	0.2598	0.2557	0.2790	0.2714	0.2874	0.2656	0.2789

Uniformality:									
0.0272	0.0264	0.0255	0.0267	0.0248	0.0272	0.0270	0.0267	0.0257	0.0265
0.0264	0.0265	0.0252	0.0278	0.0263	0.0272	0.0252	0.0264	0.0264	0.0247
0.0271	0.0276	0.0268	0.0293	0.0283	0.0265	0.0269	0.0275	0.0277	0.0257
0.0255	0.0269	0.0259	0.0271	0.0273	0.0256	0.0278	0.0283	0.0267	0.0277
0.0254	0.0265	0.0280	0.0283	0.0262	0.0269	0.0267	0.0266	0.0263	0.0261
0.0269	0.0270	0.0262	0.0279	0.0252	0.0255	0.0277	0.0254	0.0262	0.0279
0.0265	0.0271	0.0286	0.0252	0.0261	0.0266	0.0278	0.0270	0.0255	0.0274
0.0244	0.0272	0.0279	0.0259	0.0266	0.0265	0.0256	0.0274	0.0266	0.0282
0.0268	0.0260	0.0256	0.0253	0.0268	0.0287	0.0270	0.0294	0.0265	0.0258
0.0275	0.0265	0.0282	0.0270	0.0266	0.0267	0.0254	0.0270	0.0277	0.0257
0.0278	0.0255	0.0274	0.0260	0.0273	0.0269	0.0256	0.0293	0.0256	0.0274
0.0249	0.0265	0.0269	0.0269	0.0268	0.0267	0.0262	0.0266	0.0271	0.0269
0.0252	0.0257	0.0286	0.0267	0.0265	0.0270	0.0270	0.0261	0.0264	0.0263
0.0280	0.0271	0.0267	0.0274	0.0266	0.0277	0.0252	0.0268	0.0267	0.0257
0.0264	0.0273	0.0244	0.0263	0.0264	0.0258	0.0268	0.0260	0.0276	0.0256

Table 8. Calculations for process capability of overlay, critical dimension, and uniformality.

Characteristics	USL	$\bar{x}$	$\hat{\sigma}$	$\hat{C}_{PU_j}$	$\hat{C}_{PU}^{T}$
Overlay	$0.1\mu m$	0.0795	0.0065	1.0499	
Critical dimension	$0.3\mu m$	0.2693	0.0083	1.2298	1.0087
uniformality	0.03	0.0267	0.00097	1.1423	

3. Capability Measure for IC Manufacturing with Tool wear

In this chapter, a case taken from a integrated-circuit (IC) manufacturing factory will be proposed to illustrate the corrective process capability indices for tool wear. First, we will introduce the manufacturing process of IC manufacturing in section 3.1. Then, describing the tool wear problem in section 3.2. Finally, a case will be proposed to illustrate how to apply the corrective process capability indices.

3.1. Integrated-Circuit: Manufacturing Process

The integrated circuit was first conceived in 1952, and the first integrated circuits were manufactured in 1959. Programmable integrated circuits were developed in the 1980s. The devices can be programmed by the user, rather than being fixed by the integrated circuit manufacturer.

An integrated circuit (also knows as IC or chip) is a miniaturized electronic circuit which has been manufactured in the surface of a thin substrate of semiconductor material.

This process of IC manufacturing can be summarized in four major process steps: wafer fabrication, wafer probe and sort, IC assembly, and burn-in and final test.

Step 1: Wafer Fabrication

Various layers of substances are formed within the wafer, or deposited on the surface of it in wafer fabrication process. These layers are typically formed in the following way: A thin film of oxide is formed or deposited on the surface of the wafer in a process called oxidation. Then, a photoengraving process called photolithography (also known as "masking" or "imaging") is used to transfer a desired pattern onto the surface of a silicon wafer. Portions of the oxide surface under the pattern are then dissolved away in a process called etching. Finally, in a process called doping, impurities are introduced into the exposed surface to form device elements such as the source and drain of a transistor. Thin films may also be deposited on the wafer to form elements such as the polysilicon gate of a transistor.

Step 2: Wafer Probe and Sorting

In the second step of wafer manufacturing, each die on a fabricated wafer is functionality tested. The dice that fail are marked with an ink spot. The wafer is then sectioned into individual die by scribing lines between the dice and breaking the wafer along these lines. The defective dice are discarded, and the remaining dice are usually sent from the fabrication facility to a die bank inventory. Die lots will be withdrawn from the inventory and assembled when they are scheduled for release.

Step 3: IC Assembly (Package)

In the third Step of wafer manufacturing, die that have been fabricated and tested are assembled for product release. Individual integrated circuit die can be mounted in a wide variety of packages. A chip generally assembled by placing it on a frame, attaching electrical leads to it at contact points (for connections to the outside world), and sealing the assembly in a protective housing.

Step 4: Final Test and Burn-in

In this final Step, packaged chips are subjected to an extensive series of electrical tests and burn-in operations to ensure that the circuit functions correctly and will continue to do so reliably. (For example, they may be operated for several hours in a high- temperature environment).

IC packaging

Integrated circuit packaging (called simply package or assembly) is the final stage of IC manufacturing per se, followed by IC testing. Once wafer probe and sorting, the wafer is scored and then broken into individual dice. Only the good, undyed chips will be packaged. Packaging involves mounting the die, connecting the die pads to the pins on the package, and sealing the die. The packaged chips are retested to ensure that they were not damaged during packaging and that the die-to-pin interconnects operation was performed correctly. The operations at this stage as follows:

Die attachment: the step during the integrated circuit packaging phase of semiconductor device fabrication during which a die is mounted and fixed to the package or support structure.

IC bonding: bonding is a method of making interconnections between a microchip and the outside would as part of semiconductor device fabrication.

IC encapsulation: refers to the design and manufacturing of protective packages for integrated circuits.

Wafer back grinding

Wafer back grinding must be done after wafer fabrication and wafer probe, in order to thin wafer thickness. Subsequently, IC packaging will be done. Semiconductor wafers are routinely thinned prior to cube to aid the sawing operation and to allow the final assembled package thickness to be minimized. For semiconductor devices required to operate at high power levels, wafer thinning improves the ability to dissipate heat by lowering the thermal resistance of die. As final thickness is decreased, the wafer progressively becomes less able to support its own weight and to resist the stresses generated by post back grinding processes. Thus, it is important to reduce the damage caused by back grinding and improve its quality.

Technical products of users' requirement are more and more handy and small; the size of IC has to conform the customers' requests. However, the circuits on the wafer are possibly damaged if the thickness of wafer is too thin. Hence, wafer back grinding must be done to thin the wafer without damaging the circuits on wafer.

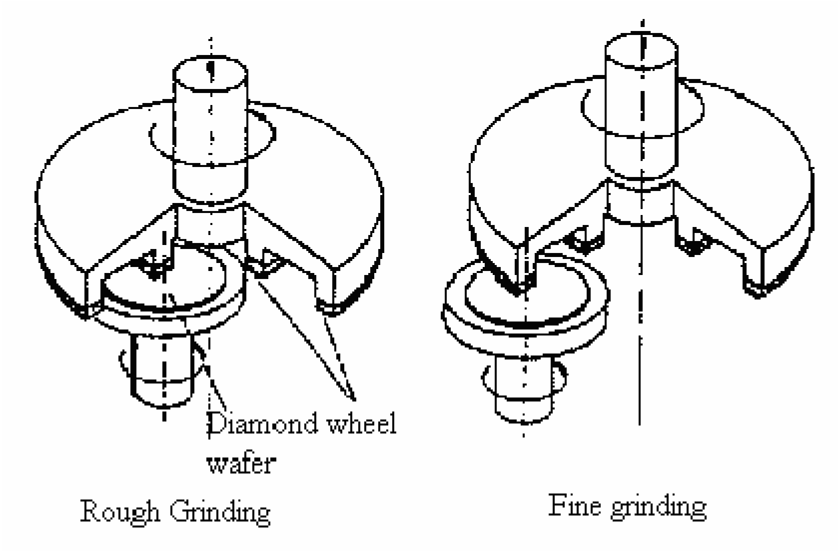

Figure 5. Wafer back grinding.

3.2. Integrated-Circuit: The Tool Wear Problem

Traditionally it is assumed that process capability is to be assessed only when the values are statistically independent. The issue of correlation among the samples and its effect on control chart limits has been studied by many authors (see Vasilopoulos and Stamboulis (1978)). However, the effect of correlation in estimating process capability has rarely been considered. There are some situations when assignable causes are systematic, such as tool wear, so that their effects can be decomposed before capability is evaluated.

When systematic assignable causes are present and tolerated, the overall variation on the process (σ^2) is composed of variation due to random causes (σ_r^2) and variation due to assignable causes (σ_a^2), i.e. $\sigma^2 = \sigma_r^2 + \sigma_a^2$. The traditional PCI measures neglects that portions of the overall variation, in the presence of tool wear, will be due to assignable causes. Therefore, any estimates of process capability will confound the true capability with these two causes. In order to get a true measure of process capability, any variation due to assignable cause must be removed from the measure of process capability. Spiring (1989, 1991) viewed this as a dynamic process that is constantly changing as the process, tools, age, etc. In the dynamic model, the capability of the process will vary, possibly in a predictable trend. Spiring has devised a modification of C_{pm} index for this dynamic process under the effect of systematic assignable causes. In the state, the goal is to maintain some minimum requirement of capability at all times. As a result, the capability will be cyclical in nature, its period defined by the frequency of process adjustments. Even when assignable cause variation is not systematic, as is the case with tool wear, it needs to be able to deal with random fluctuations of the process mean over time. Typically, deviations from the target value are due to easily determined assignable causes, such as shift-to-shift changes, differences in raw material batches, environmental factors, etc.

The most general case discussed will assume only a reasonable predictable recurring pattern with known upper and lower specification limits, target value and the existence of a tool wear problem. Figure 6 illustrates a general relationship that may occur when a tool wear problem exists, includes the process specifications (i.e. USL, LSL and T), the starting, stopping and the process output. The tool wear is pictured in a non-linear, increasing trend but could be nay reasonably consistent recurring pattern. The process illustrated in Figure 6 depicts a systematic tool wear problem with non-linear fashion. Similar to measuring variation in any process all sources of variation must be examined when considering tool wear. In a process exhibiting a tool wear problem, the traditional measure of process capability index C_{pk} is influenced by tool wear slope, as Figure7. Hence, such measure is since if fails to acknowledge that portions of the overall variation will be due to assignable causes.

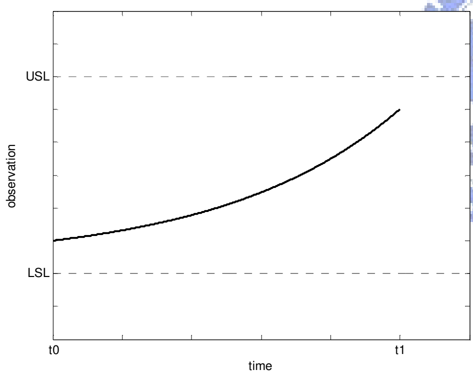

Figure 6. An example of tool wear problem.

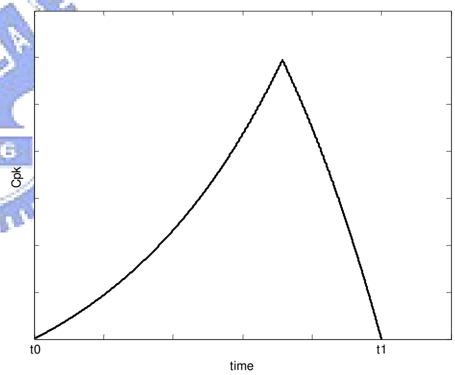

Figure 7. Plot of the changing capability of a process with tool wear.

3.3. Capability Measure for Wafer Back Grinding

In this section, a modified C_{pk} index for dynamic process under the affect of systematic assignable causes will be introduced (Pearn *et al.*, 2006). Subsequently, a example of wafer grinding will be propose to illustrate how the modified C_{pk} applied to.

3.3.1. Estimation of the C_{pk}

Allowing the process capability to be dynamic, the objective will be to maintain some minimum level of capability. Using a process capability index, the changing ability of the process can be monitored. Pearn *et al.* (2006) proposed a modified C_{pk} index for dynamic processes under the affect of systematic assignable cause as:

$$C_{pk} = \frac{\min\{USL - \mu_t, \mu_t - LSL\}}{3\sigma_{rt}},$$

where USL and LSL denote the upper and lower specification limits respectively, μ_t represents the mean and σ_{rt} the variation (due to random causes only) of the process at time period t.

We have to finding the value of C_{pk} or a suitable estimate at carious times t over each cycle in the lifetime of the tool to monitor a process capability. Assume the effect of the tool deterioration to be linear over the sampling window only, estimates of C_{pk} are possibly that will in fact be free from any contribution of the assignable cause. Thus, the proposed estimator of process capability can be obtained by replacing μ_t and σ_{rt} by the estimators $\bar{X}_t$ and $[(n-2)MSE_t/(n-1)]^{1/2}$, respectively. Then obtained

$$\hat{C}_{pk} = \frac{\min\{USL - \bar{X}_t, \bar{X}_t - LSL\}}{3\hat{\sigma}_{rt}} = \frac{d - \left|\bar{X}_t - M\right|}{3\sqrt{\frac{(n-2)MSE_t}{n-1}}}.$$

The variation $\hat{\sigma}_{rt}$ is removed by considering of the sequentially selected points (i.e., t_{a1}, t_{a2}, …, t_{an}) instead of the sample variance. The MES_t is the mean square error associated with the regression equation $\hat{X}_{ai} = \hat{\alpha}_a + \hat{\beta}_{ai}$, where t_{ai} is the sequence number of the sampling unit and $\hat{\beta}_a$ will denote the linear change in the tool wear given a unit change in time.

$$MSE_t = \frac{\sum_{i=1}^{n}(X_{t_{ai}} - \hat{X}_{t_{ai}})^2}{n-2}.$$

3.3.2. Sampling Distribution and Critical Value for Dynamic Process

Pearn *et al.* (2006) derived the cumulative distribution function of $\hat{C}_{pk}$, as follow:

$$F_{\hat{C}_{pk}}(x) = 1 - \int_0^{b\sqrt{n}} G\left(\frac{(n-2)(b\sqrt{n}-t)^2}{9nx^2}\right)[\phi(t+\xi\sqrt{n}) + \phi(t-\xi\sqrt{n})]dt, \text{ for } x > 0.$$

where $b = d/\sigma$.

Using ordinary least square (OLS) estimates of α_a, β_a and assuming the sampling scheme to be sequential, the computational formula for $\hat{C}_{pk}$ can be expressed alternatively as

$$\hat{C}_{pk} = \frac{d - \left|\bar{X}_{ta} - M\right|}{3\left[\dfrac{\sum_{i=1}^{n} X_{t_{ai}}^{2}}{n-1} - \dfrac{2n(2n+1)}{(n-1)^{2}}\bar{X}_{t_a}^{2} - \dfrac{12\left(\sum_{i=1}^{n}(iX_{ai})\right)^{2}}{n(n^{2}-1)(n-1)} + \dfrac{12\bar{X}_{t_a}\sum_{i=1}^{n}(iX_{t_{ai}})}{(n-1)^{2}}\right]^{1/2}}.$$

where n denotes the subgroup sample size, and $X_{t_{ai}}$ represents the ith value of the quality characteristic in the sampling period t_a. The proposed sampling plot is similar to those plots used in monitoring a process of control charting procedures. The general form will be to gather k subgroups of size n from each cycle over the lifetime of the tool. The value of k will be unique to each process and in fact may change from cycle to cycle within a process. On the other hand, sample size of less than five are advised against, while larger samples ($n > 30$) may also pose a problem. The optimal sample size for measuring process capability in the presence of systematic assignable cause will vary fro each process considered (Spiring, 1991).

Based on the CDF of $\hat{C}_{pk}$, given values of capability requirement C, the parameter ξ ($\xi = (\mu - M)/\sigma$), sample size n, and risk α, the critical value c_α can be obtained by solving $P(\hat{C}_{pk} \geq c_\alpha | C_{pk} = C) = \alpha$ using available numerical integration methods. That is,

$$\int_0^{(3C+|\xi|)\sqrt{n}} G\left(\frac{(n-2)\left((3C+|\xi|)\sqrt{n}-t\right)^2}{9nc_\alpha^2}\right)[\phi(t+\xi\sqrt{n})+\phi(t-\xi\sqrt{n})]dt = \alpha .$$

The above equation is an even function of ξ. Hence, for either $\xi = \xi_0$ or $\xi = -\xi_0$ we will get the same critical value c_α. Pearn *et al.* (2006) execute extensive calculation to obtain the critical values c_α for ξ =0(0.05)3.00, n =5(5)50, C_{pk} =0.5(0.5)2.0 with risk α =0.05. The results show the maximum c_α value at ξ=1.00.

In addition, Pearn *et al.* (2006) provide practitioners a table to apply the proposed procedure. The critical values of $\hat{C}_{pk}$ for α =0.01 and 0.05 with n=5(5)30 are listed in Table 9. For example, if C=1.33 is the minimum capability requirement, for α=0.05 with sample size n=10, we can find c_α=2.305. That is, as the estimated process capability drops below the critical value of $\hat{C}_{pk}$, the practitioner should stop the process and reset the tool because there is an evidence to think the process is nearing the end of its ability to produce qualified product. On the other hand, if the value of $\hat{C}_{pk}$ is grater that the critical value, then the process is thought capable.

Table 9. The critical value c_α for dynamic process with various parameters.

n	C_{pk} =1.00		C_{pk} =1.33		C_{pk} =1.67		C_{pk} =2.00	
	α =0.01	α =0.05	α =0.01	α =0.05	α =0.01	α =0.05	α =0.01	α =0.05
5	5.206	2.967	6.867	3.918	8.591	4.903	10.269	5.862
10	2.266	1.750	2.980	2.305	3.720	2.881	4.441	3.442
15	1.826	1.517	2.404	2.000	3.002	2.500	3.584	2.987
20	1.644	1.412	2.163	1.863	2.701	2.329	3.226	2.783
25	1.539	1.350	2.026	1.782	2.532	2.229	3.023	2.664
30	1.471	1.309	1.937	1.728	2.420	2.162	2.891	2.584

We consider the following case take from an IC package and testing factory to explain how the proposed procedure be established and applied to diamond wheels of wafer back grinder are investigated. The capability analysis focused on the key characteristic, wafer thickness. The upper and lower manufacturing specification limits are $USL = 330.2\mu m$ and $LSL = 279.4\mu m$, respectively. If the characteristic data is out of specification limits, the diamond wheel of wafer back grinder is considered to make a replacement.

The thickness of wafer is measured and recorded when the product come out of the process. The collect data showing tool wear consist of 100 observations in ten subgroups of size ten each, which exhibited in Table 10. Figure 8 plots the individual values in the series data. It can be seen that the observations starting from a lower value (close to lower specification limit) gradually increase to the upper specification limit due to diamond wheel deterioration. The trend seems to linear in increasing. Also, the values of the thickness of wafer are influenced by tool wear, which is likely to be dependent on the condition of the tool as previous component was processed. Now, the goal is to maintain minimum level of capability at all times and to monitor the processes under the affect of systematic assignable cause. The process should be stopped and the tool should be replaced when the measure of process capability is lower than the minimum acceptable level. Suppose the capability requirement for the wafer back grinding process is defined as "Capable" if $C_{pk} > 1.00$. Therefore, applying the capability measure for dynamic which is proposed by Pearn *et al.* (2006), the practitioners can monitor the process by calculating the measure of C_{pk}. The proposed testing procedure for tool wear process is similar to those used in monitoring a process with control chart.

Table 10. The collected 10 subgroups of size ten (Unit: μm).

i	1	2	3	4	5	6	7	8	9	10
t_1	280.05	281.48	282.25	282.25	283.68	284.07	284.40	284.78	284.78	285.50
t_2	285.88	286.60	286.98	288.80	286.98	288.80	288.41	288.80	286.98	288.08
t_3	286.98	286.98	288.80	288.80	288.80	288.80	286.98	290.61	290.61	292.42
t_4	293.53	293.86	296.06	294.24	296.77	295.34	294.24	297.87	296.44	297.87
t_5	297.87	299.36	298.97	297.87	299.69	297.87	301.50	300.78	300.45	301.50
t_6	301.50	303.37	303.37	305.18	302.98	301.50	305.18	305.18	305.18	312.45
t_7	310.63	308.10	308.82	307.00	306.62	308.82	308.82	312.45	314.26	316.07
t_8	316.07	319.38	318.99	319.71	319.71	316.07	319.38	320.48	320.81	320.81
t_9	320.09	320.81	320.81	321.52	319.71	321.19	319.71	319.71	323.34	325.21
t_{10}	323.34	325.21	323.34	327.02	327.02	325.21	327.02	329.55	327.35	327.02

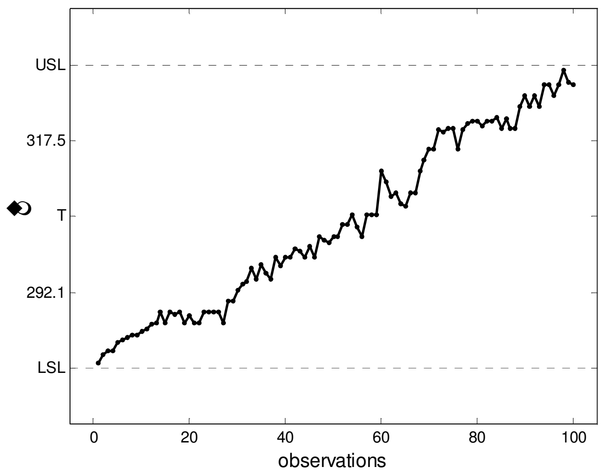

Figure 8. Plot of the 100 observations.

Assume the risk α =10, sample size n =10, and minimum capability requirement C =1.00, in this case we can obtain the critical value of $\hat{C}_{pk}$ is 1.75 by checking Table 9. While the estimated process capability drops below the critical value of $\hat{C}_{pk}$, the engineers should stop the process and replacement the tool because there is an evidence to consider that the process is close the end of its ability to produce acceptable product. If the values of $\hat{C}_{pk}$ are greater than 1.75 the process is considered capable and is allowed to continue producing. Based on the observation listed in Table 10, the calculated $\hat{C}_{pk}$ for dynamic process at each time period are shown in Table 11. Figure 9 illustrate the measure of process capability $\hat{C}_{pk}$ for dynamic process at each time period over a single cycle of the process. It is observed that the estimated $\hat{C}_{pk}$ comes to maximum at time period t_5 and then drops below the minimum C_{pk} line (C_{pk} =1.75) at time period t_{10}. Hence, based on these results, we would suggest that the process should be stopped and the tool should be replaced to avoid produce unacceptable products.

Table 11. The estimated C_{pk} for dynamic process at each time period.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
$\hat{C}_{pk}$	2.9316	3.0805	2.9058	4.8999	6.9571	3.7553	2.9135	2.6374	2.01	1.0158

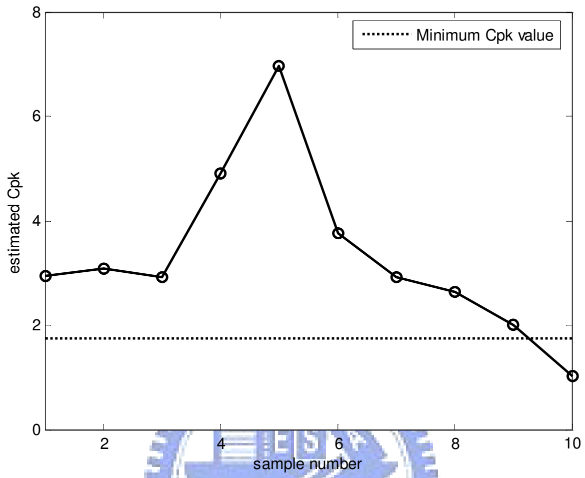

Figure 9. Capability plot for dynamic process at each time period.

4. Conclusion

Process capability indices have been widely used to measure manufacturing process capability in industry. However, these indices for multiple characteristics and tool wear problem are comparatively neglected. In the thesis, two real-world cases of multiple characteristics and tool wear respectively, in manufacturing industry are considered in order to illustrate how the modified indices apply to real-world problems.

The first part, we considered the problem of finding the lower confidence bound and sample sizes required for specified precision of the estimation for the C_{PU}^{T}. The lower confidence bounds present a measure on the minimum capability of the process based on the sample data. The sample size determination is directly related to the cost of data collection plan. We used the bootstrap method to calculate the estimator C_{PU}^{T} and compare the precision performance of four bootstrap methods. The results indicated that the BCPB method has good performance when the sample size is small than 100. We also investigated the behavior of the lower confidence bound values and sample sizes required for specified precision of the estimation by using BCPB method. The proposed approach ensures that the risk of making incorrect decisions will be no greater than the preset Type I error $1-\gamma$. We also provided tables for the engineers/ practitioners to use for their in-plant applications. A real-world example taken from TFT-LCD manufacturing process is investigated to illustrate the applicability of our approach.

The second part, the estimator of the index C_{pk} for the process where a systematic assignable cause occurs is investigated. Further, a procedure of capability measure with assignable causes is developed, and the critical value for various values capability requirements and sample size are provided. The proposed procedure is similar to those used in monitoring a process with control chart, and used to monitor the process and decide if the process should stop and replace the tool to avoid producing unacceptable products. A real-world case taken from IC packaging process is investigated to illustrate the applicability of the proposed procedure.

Appendix A The rank of the four bootstrap methods

The rank of the four bootstrap methods as $C_{pu} = 1$ and $v = 2$

	Rank	1	2	3	4	total
n = 30	SB	6	21	473	0	2.934
	PB	5	474	21	0	2.032
	BCPB	473	0	1	26	1.160
	PT	16	5	5	474	3.874
n = 40	SB	6	20	472	2	2.940
	PB	8	472	20	0	2.024
	BCPB	472	2	0	26	1.160
	PT	14	6	8	472	3.876
n = 50	SB	2	27	469	2	2.942
	PB	2	470	27	1	2.054
	BCPB	469	1	2	28	1.178
	PT	27	2	2	469	3.826
n = 60	SB	2	30	465	3	2.938
	PB	9	466	24	1	2.034
	BCPB	468	0	1	31	1.190
	PT	22	3	10	465	3.836
n = 70	SB	4	19	475	2	2.950
	PB	4	478	18	0	2.028
	BCPB	477	0	0	23	1.138
	PT	15	3	7	475	3.884
n = 80	SB	5	26	467	2	2.932
	PB	2	468	30	0	2.056
	BCPB	470	3	0	27	1.168
	PT	23	4	2	471	3.842
n = 90	SB	4	26	470	0	2.932
	PB	4	469	24	3	2.052
	BCPB	472	1	2	25	1.160
	PT	20	4	4	472	3.856
n = 100	SB	6	27	466	1	2.924
	PB	8	467	24	1	2.036
	BCPB	466	1	2	31	1.196
	PT	20	5	8	467	3.844
n = 125	SB	4	19	475	2	2.950
	PB	5	476	19	0	2.028
	BCPB	477	1	0	22	1.134
	PT	14	4	6	476	3.888
n = 150	SB	2	22	470	6	2.960
	PB	4	471	19	6	2.054
	BCPB	477	0	4	19	1.130
	PT	17	7	8	468	3.854
n = 200	SB	2	26	461	11	2.962
	PB	7	467	24	2	2.042
	BCPB	470	4	2	24	1.160
	PT	21	3	13	463	0.216

The rank of four bootstrap methods as $C_{pu} = 1$ and $\nu = 3$

		1	2	3	4	total
n = 30	SB	4	17	477	2	2.954
	PB	0	480	17	3	2.046
	BCPB	481	0	4	15	1.106
	PT	15	3	2	480	3.894
n = 40	SB	1	13	483	3	2.976
	PB	1	486	12	1	2.026
	BCPB	486	1	1	12	1.078
	PT	12	0	4	484	3.920
n = 50	SB	6	23	466	5	2.940
	PB	4	469	24	3	2.052
	BCPB	469	3	4	24	1.166
	PT	21	5	6	468	3.842
n = 60	SB	1	26	466	7	2.958
	PB	2	471	26	1	2.052
	BCPB	474	2	1	23	1.146
	PT	23	1	7	469	3.844
n = 70	SB	5	27	463	5	2.936
	PB	5	464	30	1	2.054
	BCPB	469	5	3	23	1.160
	PT	21	4	5	470	3.848
n = 80	SB	7	25	464	4	2.930
	PB	11	463	23	3	2.036
	BCPB	465	2	10	23	1.182
	PT	17	10	3	470	3.852
n = 90	SB	10	21	457	12	2.942
	PB	3	466	29	2	2.060
	BCPB	466	4	4	26	1.180
	PT	21	9	10	460	3.818
n = 100	SB	7	20	464	9	2.950
	PB	6	467	25	2	2.046
	BCPB	468	6	3	23	1.162
	PT	19	7	8	466	3.842
n = 125	SB	7	39	444	10	2.914
	PB	10	447	33	10	2.086
	BCPB	461	7	9	23	1.188
	PT	22	7	14	457	3.812
n = 150	SB	7	41	435	17	2.924
	PB	13	438	39	10	2.092
	BCPB	446	9	9	36	1.270
	PT	37	9	19	435	3.704
n = 200	SB	13	36	426	25	2.926
	PB	15	434	42	9	2.090
	BCPB	446	14	7	33	1.254
	PT	27	15	25	433	3.728

The rank of four bootstrap methods as $C_{pu} = 1$ and $v = 4$

		1	2	3	4	total
n = 30	SB	7	24	466	3	2.930
	PB	3	467	26	4	2.062
	BCPB	469	3	4	24	1.166
	PT	21	6	4	469	3.842
n = 40	SB	5	30	459	6	2.932
	PB	4	460	31	5	2.074
	BCPB	467	5	1	27	1.176
	PT	24	5	9	462	3.818
n = 50	SB	9	31	450	10	2.922
	PB	3	457	36	4	2.082
	BCPB	460	3	3	34	1.222
	PT	28	9	11	452	3.774
n = 60	SB	9	33	445	13	2.924
	PB	5	453	38	4	2.082
	BCPB	455	6	5	34	1.236
	PT	31	8	12	449	3.758
n = 70	SB	9	44	434	13	2.902
	PB	11	434	45	10	2.108
	BCPB	450	11	7	32	1.242
	PT	31	10	14	445	3.746
n = 80	SB	6	47	434	13	2.908
	PB	16	428	47	9	2.098
	BCPB	439	13	3	45	1.308
	PT	41	10	17	432	3.680
n = 90	SB	12	41	425	22	2.914
	PB	14	431	43	12	2.106
	BCPB	442	10	9	39	1.290
	PT	33	17	24	426	3.686
n = 100	SB	8	48	422	22	2.916
	PB	9	433	48	10	2.118
	BCPB	441	11	12	36	1.286
	PT	43	7	20	430	3.674
n = 125	SB	15	63	392	30	2.874
	PB	16	404	64	16	2.160
	BCPB	425	15	15	45	1.360
	PT	46	16	29	409	3.602
n = 150	SB	17	55	394	34	2.890
	PB	17	409	52	22	2.158
	BCPB	416	21	20	43	1.380
	PT	51	15	33	401	3.568
n = 200	SB	20	53	386	41	2.896
	PB	22	401	63	14	2.138
	BCPB	412	24	10	54	1.412
	PT	46	23	40	391	3.552

The rank of four bootstrap methods as $C_{pu} = 1$ and $\nu = 5$

		1	2	3	4	total
n = 30	SB	10	28	455	7	2.918
	PB	5	458	34	3	2.070
	BCPB	465	5	3	27	1.184
	PT	20	9	8	463	3.828
n = 40	SB	7	47	432	14	2.906
	PB	9	435	50	6	2.106
	BCPB	437	9	7	47	1.328
	PT	48	8	11	433	3.658
n = 50	SB	9	61	418	12	2.866
	PB	12	416	61	11	2.142
	BCPB	420	13	12	55	1.404
	PT	61	8	10	421	3.582
n = 60	SB	14	63	399	24	2.866
	PB	13	410	62	15	2.158
	BCPB	420	11	12	57	1.412
	PT	54	15	28	403	3.560
n = 70	SB	7	67	395	31	2.900
	PB	17	402	71	10	2.148
	BCPB	406	20	10	64	1.464
	PT	70	12	23	395	3.486
n = 80	SB	15	76	384	25	2.838
	PB	10	391	78	21	2.220
	BCPB	406	15	10	69	1.484
	PT	69	18	28	385	3.458
n = 90	SB	20	73	381	26	2.826
	PB	19	385	74	22	2.198
	BCPB	390	23	15	72	1.538
	PT	72	18	30	380	3.436
n = 100	SB	16	78	374	32	2.844
	PB	20	388	78	14	2.172
	BCPB	393	21	15	71	1.528
	PT	74	11	34	381	3.444
n = 125	SB	25	82	348	45	2.826
	PB	34	352	90	24	2.208
	BCPB	366	44	17	73	1.594
	PT	76	21	45	358	3.370
n = 150	SB	28	103	325	44	2.770
	PB	37	330	102	31	2.254
	BCPB	355	40	25	80	1.660
	PT	80	27	50	343	3.312
n = 200	SB	22	86	340	52	2.844
	PB	50	340	95	15	2.150
	BCPB	365	45	19	71	1.592
	PT	70	22	46	362	3.400

The rank of four bootstrap methods as $C_{pu} = 1.33$ and $\nu = 2$

		1	2	3	4	total
n = 30	SB	3	13	481	3	2.968
	PB	3	483	14	0	2.022
	BCPB	483	2	0	15	1.094
	PT	11	2	5	482	3.916
n = 40	SB	8	27	464	1	2.916
	PB	4	465	29	2	2.058
	BCPB	465	0	2	33	1.206
	PT	23	8	5	464	3.820
n = 50	SB	3	18	477	2	2.956
	PB	4	480	15	1	2.026
	BCPB	482	2	1	15	1.098
	PT	11	1	6	482	3.918
n = 60	SB	6	17	476	1	2.944
	PB	5	477	17	1	2.028
	BCPB	477	1	1	21	1.132
	PT	12	5	6	477	3.896
n = 70	SB	6	21	470	3	2.940
	PB	3	474	22	1	2.042
	BCPB	472	2	3	23	1.154
	PT	19	3	5	473	3.864
n = 80	SB	6	33	455	6	2.922
	PB	5	459	32	4	2.070
	BCPB	462	2	4	32	1.212
	PT	27	6	9	458	3.796
n = 90	SB	3	26	468	3	2.942
	PB	4	467	26	3	2.056
	BCPB	474	2	0	24	1.148
	PT	20	4	6	470	3.852
n = 100	SB	3	38	453	6	2.924
	PB	7	456	34	3	2.066
	BCPB	458	4	3	35	1.230
	PT	32	2	10	456	3.780
n = 125	SB	4	29	455	12	2.950
	PB	3	459	33	5	2.080
	BCPB	470	5	1	24	1.158
	PT	23	7	11	459	3.812
n = 150	SB	10	30	450	10	2.920
	PB	9	459	27	5	2.056
	BCPB	459	4	5	32	1.220
	PT	22	7	18	453	3.804
n = 200	SB	6	34	445	15	2.938
	PB	13	448	33	6	2.064
	BCPB	448	12	8	32	1.248
	PT	33	6	14	447	3.750

The rank of four bootstrap methods as $C_{pu} = 1.33$ and $\nu = 3$

		1	2	3	4	total
n = 30	SB	5	21	473	1	2.940
	PB	4	472	20	4	2.048
	BCPB	474	2	3	21	1.142
	PT	17	5	5	473	3.868
n = 40	SB	4	30	459	7	2.938
	PB	4	461	32	3	2.068
	BCPB	462	4	3	31	1.206
	PT	30	5	6	459	3.788
n = 50	SB	3	32	460	5	2.934
	PB	4	460	33	3	2.070
	BCPB	463	4	2	31	1.202
	PT	30	4	5	461	3.794
n = 60	SB	9	37	448	6	2.902
	PB	2	453	40	5	2.096
	BCPB	456	2	3	39	1.250
	PT	33	8	9	450	3.752
n = 70	SB	11	52	428	9	2.870
	PB	3	427	61	9	2.152
	BCPB	440	10	4	46	1.312
	PT	46	11	8	435	3.664
n = 80	SB	11	58	421	10	2.860
	PB	4	420	60	16	2.176
	BCPB	427	7	7	59	1.396
	PT	58	15	12	415	3.568
n = 90	SB	15	51	406	28	2.894
	PB	14	420	51	15	2.134
	BCPB	417	19	14	50	1.394
	PT	54	10	29	407	3.578
n = 100	SB	8	65	405	22	2.882
	PB	8	414	66	12	2.164
	BCPB	426	8	11	55	1.390
	PT	58	13	18	411	3.564
n = 125	SB	17	79	372	32	2.838
	PB	24	382	86	8	2.156
	BCPB	387	18	10	85	1.586
	PT	75	18	35	372	3.408
n = 150	SB	21	102	340	37	2.786
	PB	20	359	101	20	2.242
	BCPB	368	27	21	84	1.642
	PT	91	14	37	358	3.324
n = 200	SB	27	120	317	36	2.724
	PB	39	304	121	36	2.308
	BCPB	329	38	27	106	1.820
	PT	106	38	35	321	3.142

The rank of four bootstrap methods as $C_{pu} = 1.33$ and $\nu = 4$

		1	2	3	4	total
n = 30	SB	5	31	457	7	2.932
	PB	2	463	28	7	2.080
	BCPB	467	1	6	26	1.182
	PT	26	5	9	460	3.806
n = 40	SB	8	49	437	6	2.882
	PB	7	438	48	7	2.110
	BCPB	442	7	10	41	1.300
	PT	43	6	5	446	3.708
n = 50	SB	18	73	387	22	2.826
	PB	13	395	78	14	2.186
	BCPB	400	16	14	70	1.508
	PT	69	16	22	393	3.478
n = 60	SB	17	81	389	13	2.796
	PB	15	390	80	15	2.190
	BCPB	398	13	12	77	1.536
	PT	71	16	19	394	3.472
n = 70	SB	16	94	356	34	2.816
	PB	22	368	104	6	2.188
	BCPB	369	20	12	99	1.682
	PT	93	18	28	361	3.314
n = 80	SB	12	113	344	31	2.788
	PB	18	350	108	24	2.276
	BCPB	369	19	19	93	1.672
	PT	102	17	29	352	3.262
n = 90	SB	23	114	332	31	2.742
	PB	26	330	126	18	2.272
	BCPB	343	32	15	110	1.784
	PT	110	23	26	341	3.196
n = 100	SB	26	123	315	36	2.722
	PB	23	316	132	26	2.310
	BCPB	329	27	24	120	1.870
	PT	119	35	28	318	3.090
n = 125	SB	34	153	278	37	2.644
	PB	28	280	155	373	5.090
	BCPB	291	31	35	143	2.060
	PT	147	36	34	283	2.906
n = 150	SB	30	157	256	57	2.680
	PB	29	274	161	36	2.408
	BCPB	294	32	33	141	2.042
	PT	147	37	51	265	2.868
n = 200	SB	34	186	206	74	2.640
	PB	45	234	182	39	2.430
	BCPB	242	37	45	176	2.310
	PT	181	41	68	210	2.614

The rank of four bootstrap methods as $C_{pu} = 1.33$ and $\nu = 5$

		1	2	3	4	total
n = 30	SB	10	45	436	9	2.888
	PB	12	429	47	12	2.118
	BCPB	440	8	7	45	1.314
	PT	39	17	11	433	3.676
n = 40	SB	11	80	392	17	2.830
	PB	9	399	80	12	2.190
	BCPB	400	13	10	77	1.528
	PT	80	9	17	394	3.450
n = 50	SB	24	115	335	26	2.726
	PB	7	359	121	13	2.280
	BCPB	358	9	18	115	1.780
	PT	111	17	26	346	3.214
n = 60	SB	20	126	327	27	2.722
	PB	25	319	133	23	2.308
	BCPB	338	31	17	114	1.814
	PT	119	22	24	335	3.150
n = 70	SB	24	157	285	34	2.658
	PB	27	284	160	29	2.382
	BCPB	293	25	27	155	2.088
	PT	158	32	31	279	2.862
n = 80	SB	26	177	262	35	2.612
	PB	25	268	173	34	2.432
	BCPB	279	29	26	166	2.158
	PT	171	27	37	265	2.792
n = 90	SB	24	181	257	38	2.618
	PB	38	253	175	34	2.410
	BCPB	260	37	30	173	2.232
	PT	182	26	37	255	2.730
n = 100	SB	23	192	245	40	2.604
	PB	34	245	189	32	2.438
	BCPB	250	35	24	191	2.312
	PT	193	29	42	236	2.642
n = 125	SB	42	212	197	49	2.506
	PB	23	220	226	31	2.530
	BCPB	217	27	35	221	2.520
	PT	218	41	43	198	2.442
n = 150	SB	36	223	191	50	2.510
	PB	30	202	241	27	2.530
	BCPB	197	36	31	236	2.612
	PT	239	38	38	185	2.338
n = 200	SB	43	267	152	38	2.370
	PB	37	168	262	33	2.582
	BCPB	146	31	38	285	2.924
	PT	275	34	47	144	2.120

Appendix B MATLAB Program

MATLAB Program for Sample Size

```
%-----------------------------------------------------------------
% Input basic data Rpu, Cpu_true, upbound, lowbound
%-----------------------------------------------------------------
Rpu = 0.75;
Cpu_true = 1;
upbound = 2000;
lowbound = 0;

%-----------------------------------------------------------------
% Find sample size with binary search
%-----------------------------------------------------------------
counter = 0;

while (upbound - lowbound) > 1;

    counter = counter + 1;
    for i = 1:500;
        CI_BCPB(i) = bootstrap3_Cpu1(floor((upbound+lowbound)/2));
    end;
    size(counter) = floor((upbound+lowbound)/2);
    CI_BCPB_mean = mean(CI_BCPB);

    if CI_BCPB_mean/Cpu_true >= Rpu;
        upbound = floor((upbound+lowbound)/2);
    else
        lowbound = floor((upbound+lowbound)/2);
    end;

    R_gamma1(counter) =CI_BCPB_mean/Cpu_true;

end;
```

```
%-----------------------------------------------------------------------------------------------
% Function files included — bootstrap3_Cpu1.m
%-----------------------------------------------------------------------------------------------
function [CI_BCPB] = bootstrap3_Cpu1(size);
flag = -1;
while flag == -1
%-------------------------------------------------------------------
% Input data n, B, USL, alpha
%-------------------------------------------------------------------
    n = size;
    B = 10000;
    USL = 3.4;
    alpha = 0.05;
%-------------------------------------------------------------------
% Randomize sample data distributed normal distribution and calculate parameter of
data
%-------------------------------------------------------------------
    data1 = normrnd(3.2273, 0.052, n, 1);
    data2 = normrnd(3.2273, 0.052, n, 1);
    data3 = normrnd(3.2273, 0.052, n, 1);

    test1 = 0;
    counter1 = 0;
    while test1 == 0;
        sample_data1 = bootstrp(B, 'sort', data1);
        sample_mu1 = mean(sample_data1, 2);
        sample_sigma1 = std(sample_data1,0,2);
        test1_data = sort(sample_sigma1);
        if test1_data(1)> 0
            test1 =1;
            cpu_hat1 = (USL - sample_mu1)./(3*sample_sigma1);
        else
            counter1 = counter1 +1;
        end;
        if counter1 == 500;
            stop;
        end;
    end;

    test2 = 0;
    counter2 = 0;
    while test2 == 0;
```

```
            sample_data2 = bootstrp(B, 'sort', data2);
            sample_mu2 = mean(sample_data2, 2);
            sample_sigma2 = std(sample_data2,0,2);
            test2_data = sort(sample_sigma2);
            if test2_data(1)> 0
                  test2 =1;
                  cpu_hat2 = (USL - sample_mu2)./(3*sample_sigma2);
            else
                  counter2 = counter2 +1;
            end;
            if counter2 == 500;
                  stop;
            end;
      end;

      test3 = 0;
      counter3 = 0;
      while test3 == 0;
            sample_data3 = bootstrp(B, 'sort', data3);
            sample_mu3 = mean(sample_data3, 2);
            sample_sigma3 = std(sample_data3,0,2);
            test3_data = sort(sample_sigma3);
            if test3_data(1)> 0
                  test3 =1;
                  cpu_hat3 = (USL - sample_mu3)./(3*sample_sigma3);
            else
                  counter3 = counter3 +1;
            end;
            if counter3 == 500;
                  stop;
            end;
      end;
%-------------------------------------------------------------------
% Calculate $\hat{C}_{pu}^{T}$
%-------------------------------------------------------------------
      cpu_total_hat =
(1/3)*norminv(normcdf(3*cpu_hat1).*normcdf(3*cpu_hat2).*normcdf(3*cpu_hat3)
);
      if cpu_total_hat ~= Inf
            flag = 1;
      end;
end;
```

```
%-------------------------------------------------------------------
% Estimate $\hat{C}_{pu}^{T}$ with BCPB method
%-------------------------------------------------------------------
cpu_total_hat_mu = mean (cpu_total_hat);
cpu_total_hat_sort = sort(cpu_total_hat);
p0 = length(find(cpu_total_hat_sort <=
cpu_total_hat_mu))/length(cpu_total_hat_sort);
z0 = norminv(p0);
pl = normcdf(2*z0-norminv(1-alpha));
CI_BCPB = cpu_total_hat_sort(round(pl*B));
```

Reference

1. Alwan, L. C. and Roberts, H. V. (1988). Time series modeling for statistical process control. *Journal of Business and Economic Statistics*, 6(1), 87-95.
2. Castagliola, P. (1996). Evaluation of non-normal process capability indices using Burr's distributions. *Quality Engineering,* 8, 587-593.
3. Chen K. S., Huang M. L., Li R. K. (2001). Process capability analysis for an entire product, *International Journal of Production Research*, 39(17), 4077-4087.
4. Chen K. S., Pearn W. L. (1997). An application of non-normal process capability indices. *Quality and Reliability Engineering International*, 13, 355-360.
5. Chen K. S., Pearn W. L., Lin P. C. (2003). Capability measures for process with multiple characteristics, *Quality and Reliability Engineering International*, 19, 101-110.
6. Chang, Y. S., Choi, I. S. and Bai, D. S. (2002) Process capability indices for skewed populations. *Quality and Reliability Engineering International*, 18, 383-393.
7. Chou Y. M., Owen D. B. (1989). On the distributions of the estimated process capability indices. *Commun Stat Theory Methods,* 18(2), 4549-4560.
8. Efron B. (1979). Bootstrap methods: another look at the jackknife. *Ann Stat*, 7, 1–26.
9. Efron B. (1981). Nonparametric standard errors and confidence intervals. *Can J Stat*, 9, 139–172.
10. Efron B. (1982). The jackknife, the bootstrap and other resampling plans. *Society for Industrial and Applied Mathematics*, 38, Philadelphia, PA.
11. Efron B., Tibshirani R. J. (1986). Bootstrap methods for standard errors, confidence interval, and other measures of statistical accuracy. *Stat. Sci.*, 1, 54-77.
12. Johnson N. L., Kotz, S., Pearn W. L. (1994). Flexible process capability indices. *Pakistan Journal of Statistics,* 10(1), 23-31.
13. Kotz, S. and Johnson N. L. (1993). *Process Capability Indices*. Chapman & Hall, London, U.K.
14. Kotz, S., and Lovelace, C. (1998). *Process Capability Indices in Theory and Practice*. Arnold, London, U.K.
15. Kushler, R. and Hurley, P. (1992). Confidence bounds for capability indices. *Journal of Quality Technology,* 24, 188-195.
16. Quesenberry, C. P. (1988). An SPC approach to compensating a tool wear process. *Journal of Quality Technology,* 20(4), 220-229.
17. Pearn, W. L. and Shu, M. H. (2003). Lower confidence bounds with sample size information for C_{pm} with application to production yield assurance. *International Journal of Production Research,* 41(15), 3581-3599.
18. Pearn, W. L., Chang Y. C. and Wu C. W. (2005). Bootstrap approach for

estimating process quality yield with application to light emitting diodes, *International Journal of Advanced Manufacturing Technology,* 25, 560-570.

19. Pearn, W. L., Hsu Y. C. and Wu C. W. (2006). Estimating and testing process capability in the presence of systematic assignable cause, *International Journal of Production Research*. (In press)

20. Somerbille, S. E., Montgomery, D. C. (1996-97). Process capability indices and non-normal distributions. *Quality Engineering*, 9, 305-316.

21. Spring, F. A. (1989). An application of C_{pm} to the tool-wear problem. *ASQC Quality Congress Transactions, Toronto*, 123-128.

22. Spiring, F. A. (1991). Assessing process capability in the presence of systematic assignable cause. *Journal of Quality Technology*, 23(2), 125-134.

23. Vännman, K. (1995). A unified approach to capability indices. *Statistica Sinica*, 5, 805-820.

24. Vännman, K. and Kotz, S. (1995). A superstructure of capability indices distributional properties and implications. *Scandinavian Journal of Statistics*, 22, 477-491.

25. Vasilopoulos, A. V. and Stamboulis, A. P. (1978). Modification of control chart limits in the presence of data correlation. *Journal of Quality Technology*, 10(1), 20-30.

26. Wright P. A. (1995). A process capability index sensitive to skewness. *Journal of Statistical Computation and Simulation*, 52, 195-203.